Kernel Smoothing Toolbox [*] for MATLAB

Jan Koláček and Jiří Zelinka

Contents

[*] Jan Koláček and Jiří Zelinka, Department of Mathematics and Statistics, Masaryk University, Kotlářská 2, Brno, Czech Republic, available on *http://www.math.muni.cz/english/science-and-research/developed-software/232-matlab-toolbox.html*

1 Kernels in MATLAB toolbox

In the toolbox, kernel functions are created by the command `K_def`.

Syntax

```
K = K_def(type)
K = K_def('gauss',s)
K = K_def(method,par1,par2,par3)
```

Description

`K_def` creates a kernel function which satisfies conditions of the definition in Section 1.1. in [3]

`K = K_def(type)` creates a kernel as a predefined `type`, where `type` is a string variable.

Predefined types are:

`'epan'`	Epanechnikov kernel
`'quart'`	quartic kernel
`'rect'`	uniform (rectangular) kernel
`'trian'`	triangular kernel
`'gauss'`	Gaussian kernel

`K = K_def('gauss',s)` creates the Gaussian kernel with variance $\mathtt{s}^2$.

`K = K_def(method,par1,par2,par3)` creates a kernel by a specified `method` (string) with parameters `par1`, `par2`, `par3` (string or double). All possible combinations of `method` and parameters are listed in Table 1.

Output

The output variable `K` is a structure array with fields described in Table 1.

method	parameters	purpose
'opt'	double values ν, k, μ	optimal kernel from $S_{\nu,k}$ of smoothness μ
'str'	`par1` a string formula (in variable 'x'), `par2`, `par3` double values	kernel defined by the formula `par1` normalized on the support [`par2`, `par3`]
'pol'	`par1` double vector, `par2`, `par3` double values	polynomial kernel defined by the vector of coefficients `par1` normalized on the support [`par2`, `par3`]
'fun'	`par1` a string, `par2`, `par3` double values	kernel defined by the external function `par1` normalized on the support [`par2`, `par3`]

Table 1: Combinations of parameters for `K_def`

Example (Gaussian kernel)
`K = K_def('gauss')` gives

```
K =     type:  'gau'
        name:  ' '
        coef:  1
     support:  [-Inf, Inf]
          nu:  0
           k:  2
          mu:  Inf
         var:  0.2821
        beta:  1
```

Example (the Epanechnikov kernel)
In this case, we have two possibilities how to create the kernel. `K = K_def('opt',0,2,0)` or `K = K_def('epan')` gives

field	**description**
`type`	type of the kernel `'opt'` optimal kernel (default) `'str'` string (variable denoted as 'x') `'pol'` polynomial kernel `'fun'` external function `'tri'` triangular kernel `'gau'` Gaussian kernel
`name`	string with kernel expression or function name, ignored for optimal and polynomial kernel
`coef`	coefficients of the optimal or polynomial kernel
`support`	support of the kernel, default $[-1, 1]$
`nu`, `k`, `mu`	order and smoothness of the kernel
`var`	variance of the kernel $V(K)$
`beta`	k-th moment of the kernel β_k

Table 2: Structure of output

```
K =      type:  'opt'
         name:  ' '
         coef:  [-3/4 0 3/4]
      support:  [-1, 1]
           nu:  0
            k:  2
           mu:  0
          var:  3/5
         beta:  1/5
```

For evaluation of the kernel `K` in a vector `x` use the function `K_val` with syntax `value = K_val(K,x)`.

2 Univariate kernel density estimation

2.1 Running the program

Toolbox for kernel density estimates can be launched by command `ksdens`. Launching without parameters will cause the start to the situation when only data input (button ①) or terminating the program (button ②) is possible (see Figure 1). In the data input subroutine (Figure 2) you can choose reading data from the workspace (button ③), from the external file (button ④) or to create simulated data (button ⑤). After choosing the source of data select the name of the variable containing the random variable (button ⑥). If you know the true density (*e.g.*, for simulated data), you can put it to the text field ⑦ with 'x' as variable. It can be used to compare with the final estimate of the density.

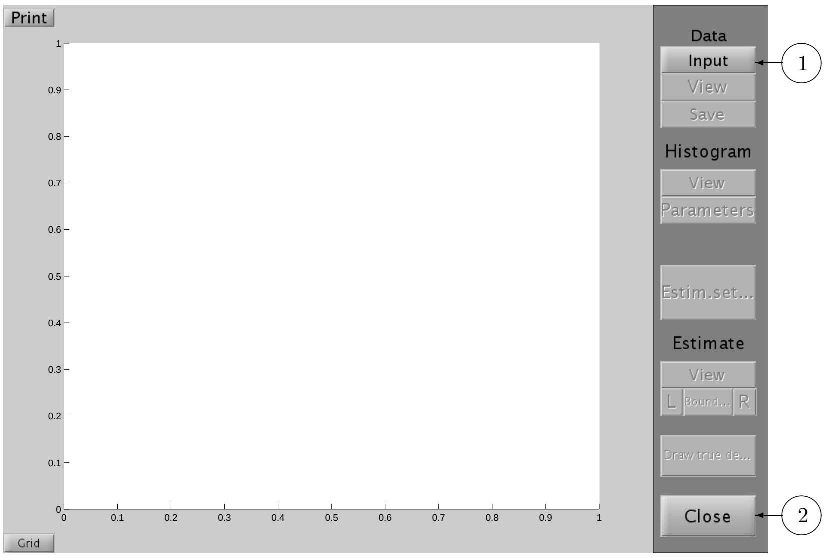


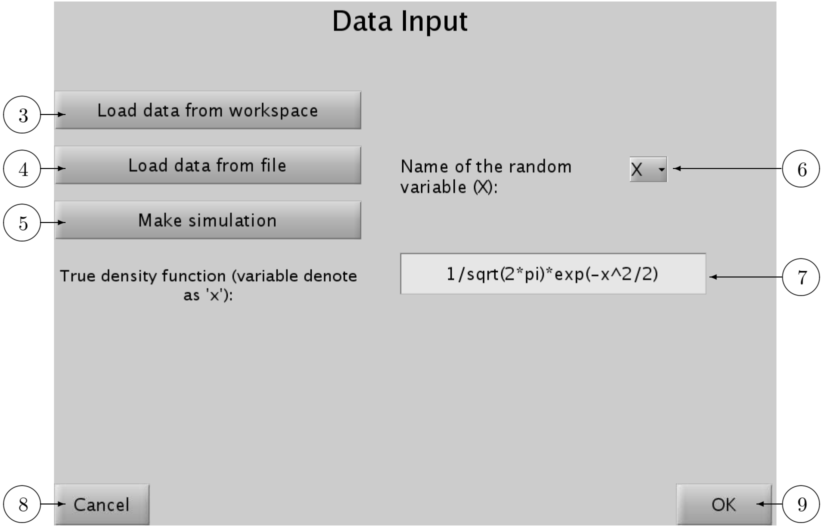
Figure 1: Start of the program.

At the end you can cancel the subroutine (button ⑧) or confirm data (button ⑨).

2.2 Main figure

After data input you will see data and you obtain another possibilities (Figure 3) in the main figure. The same situation is invoked if the main program is called with a parameter, *i.e.*, by command `ksdens(X)`. The variable `X` contains a sample of random variable which density we want to estimate.

Pressing button ⑩ calls the same figure as Figure 3. It is also possible to save selected variables (button ⑪ , see Figure 4) into a file (button ⑮) or into the workspace (button ⑯). It should be point out that the most of variables is empty at the beginning of the program.



Name of the random variable (X): X

True density function (variable denote as 'x'): 1/sqrt(2*pi)*exp(-x^2/2)

Figure 2: Data input.

Another two possibilities after data input concern the histogram: you can display it by button ⑫ (see Figure 5) and set the number of bins for the histogram by button ⑬ .

2.3 Setting the parameters

Button ⑭ invokes the setting the parameters for smoothing (Figure 6). The upper part of the window contains the brief instructions for entering the parameters. In the left part of the window you can choose a predefined kernel (button ⑰) or the optimal kernel (buttons ⑱). Confirm the choice of the kernel by pressing button OK. By button ⑲ you get the picture with the shape of the kernel.

If the kernel is selected, the bandwidth can be chosen (field ⑳). Basic methods of bandwidth selection are implemented (see Section 2.4 in [3] for details). Button ㉒ calls the automatic procedure for setting of all parameters described in Section 2.6 in [3]. In the right part of the window there are boxes where points for drawing the estimate of the density can be set up (㉓). Finally, you can confirm the parameters (button ㉔) or close the window without change of the parameters (button ㉕).

2.4 Eye-control method

The bandwidth can be chosen by so-called *"Eye-control" method* (button ㉑). This button invokes other window (see Figure 7). In boxes ㉖ and ㉗ the bandwidth and the step for its increasing or decreasing can be set. By pressing button ㉘ (the middle one) the estimate for actual bandwidth is displayed, the arrows at the left and right side cause increasing or decreasing the bandwidth by a step and redrawing the figure. By buttons ㉙ you can run and stop the gradual increasing the bandwidth and drawing

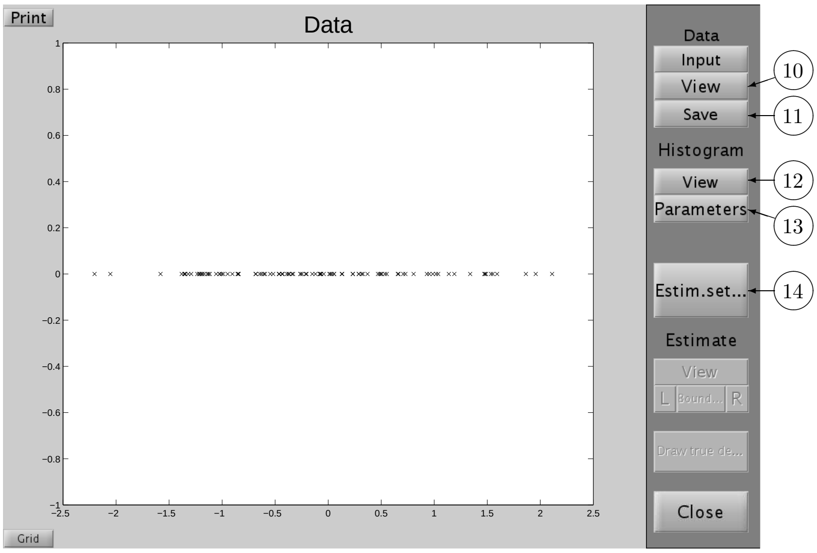


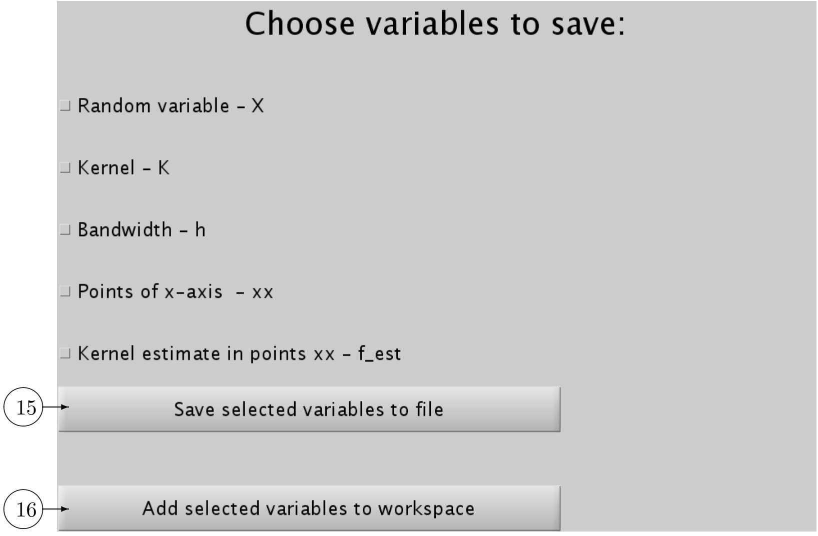
Figure 3: Data view.

the corresponding estimate. Finally it is possible to accept selected bandwidth (button ㉚) or cancel the procedure (button ㉛) .

2.5 The final estimation

If all smoothing parameters are set, the estimate of the density function can be displayed (Figure 8, button ㉜). For correction of boundary effects buttons ㉝ can be applied. In the separate window you can set the left and the right boundaries for the removal the boundary effects. Then pressing buttons “L” or “R” the boundary effects correction is applied at the corresponding part of the estimate.

Button ㉞ is intended to display the true density function if it was specified (see Figure 9). Button ㉟ can be used for displaying the grid in the figure that may help to identify the values of the graph. By the last button ㊱ you obtain the separate window with the density estimate without the buttons for other manipulation with the graph (exporting into some graphical format *etc.*)



Choose variables to save:

□ Random variable - X

□ Kernel - K

□ Bandwidth - h

□ Points of x-axis - xx

□ Kernel estimate in points xx - f_est

15
Save selected variables to file
16
Add selected variables to workspace

Figure 4: Data saving.

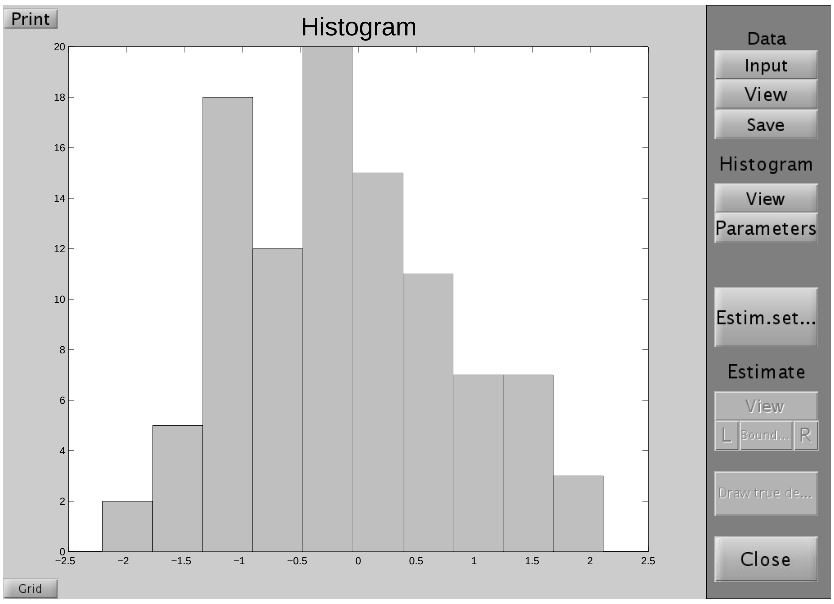
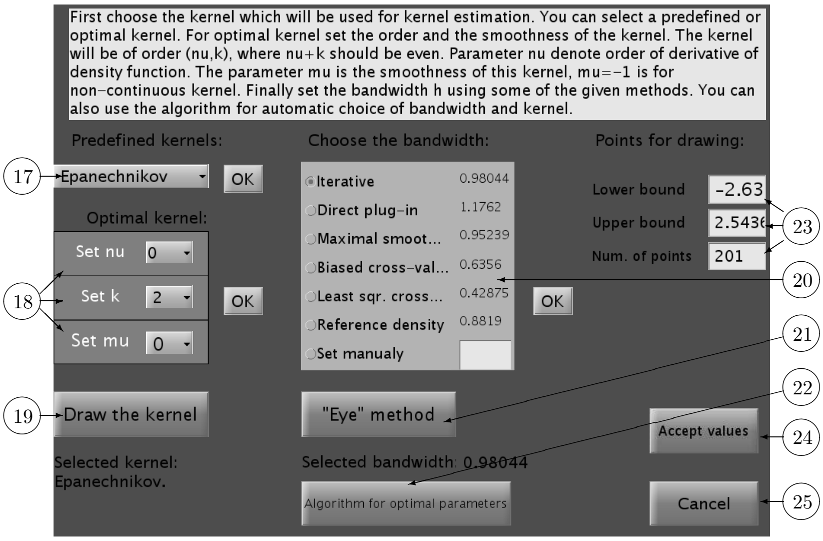


Figure 5: Histogram.



First choose the kernel which will be used for kernel estimation. You can select a predefined or optimal kernel. For optimal kernel set the order and the smoothness of the kernel. The kernel will be of order (nu,k), where nu+k should be even. Parameter nu denote order of derivative of density function. The parameter mu is the smoothness of this kernel, mu=−1 is for non−continuous kernel. Finally set the bandwidth h using some of the given methods. You can also use the algorithm for automatic choice of bandwidth and kernel.

Predefined kernels: Epanechnikov

Optimal kernel:

Set nu 0

Set k 2

Set mu 0

Choose the bandwidth:

● Iterative	0.98044
○ Direct plug−in	1.1762
○ Maximal smoot...	0.95239
○ Biased cross−val...	0.6356
○ Least sqr. cross...	0.42875
○ Reference density	0.8819
○ Set manualy	

Points for drawing:

Lower bound −2.63

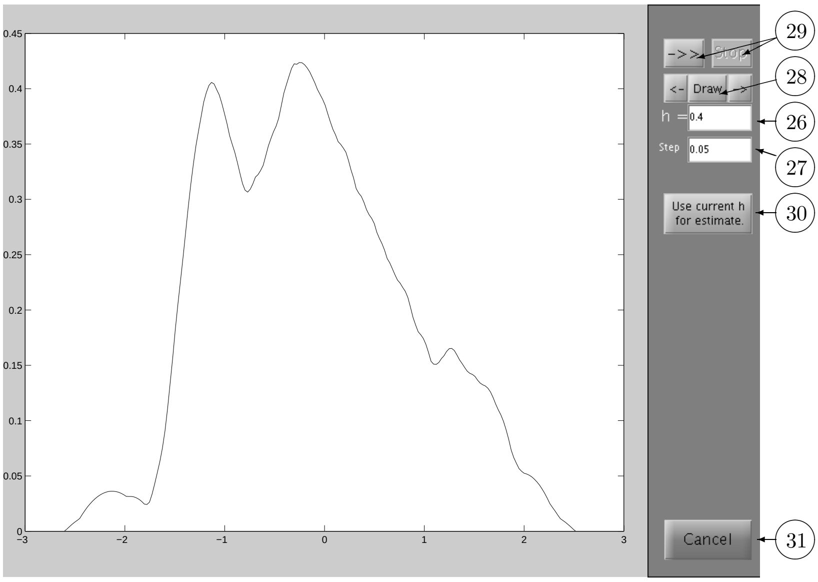
Upper bound 2.5436

Num. of points 201

Selected kernel: Epanechnikov.

Selected bandwidth: 0.98044

Figure 6: Setting parameters.



h = 0.4

Step 0.05

Figure 7: "Eye-control" method.

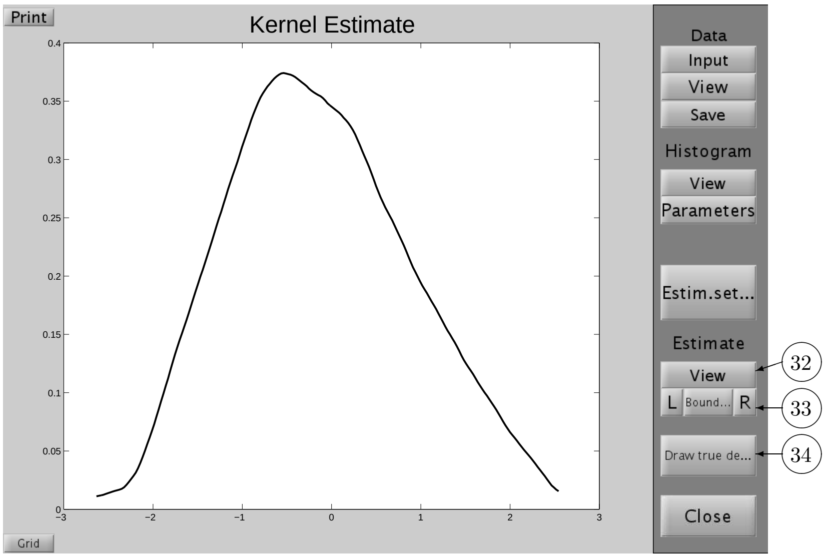


Figure 8: Kernel estimate.

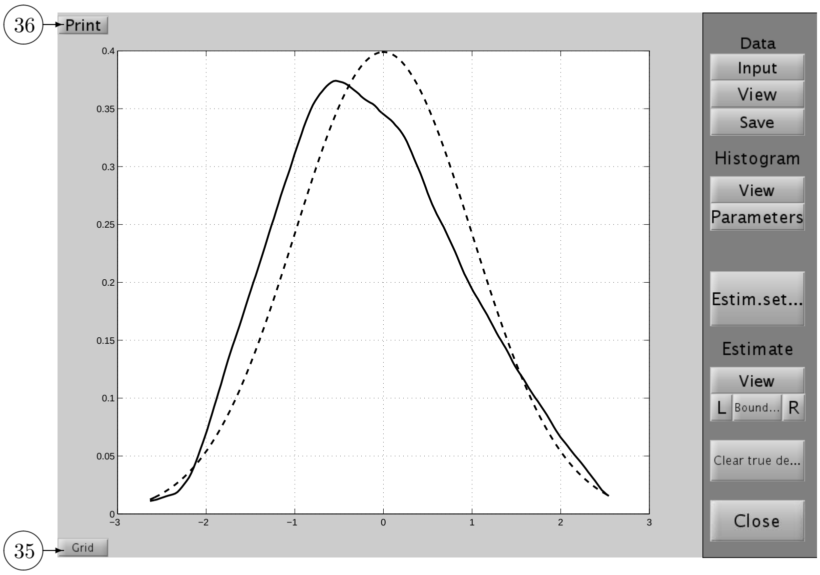


Figure 9: Estimate with the true density.

3 Kernel estimation of a distribution function

3.1 Running the program

The toolbox is launched by command `kscdf`. Launching without parameters will cause the start to the situation when only data input (button ①) or terminating the program (button ②) is possible (see Figure 10). In the subroutine for data input (Figure 11) you can choose the source of the data – you can choose reading data from the workspace (buttons ③), from the file (button ④) or create the simulated data (button ⑤). Then select the name of the variable in the list ⑥ . If the true distribution function is known (for simulated data, for instance), put it to the text field ⑦ . You can use it to compare with the final estimate of the distribution function.

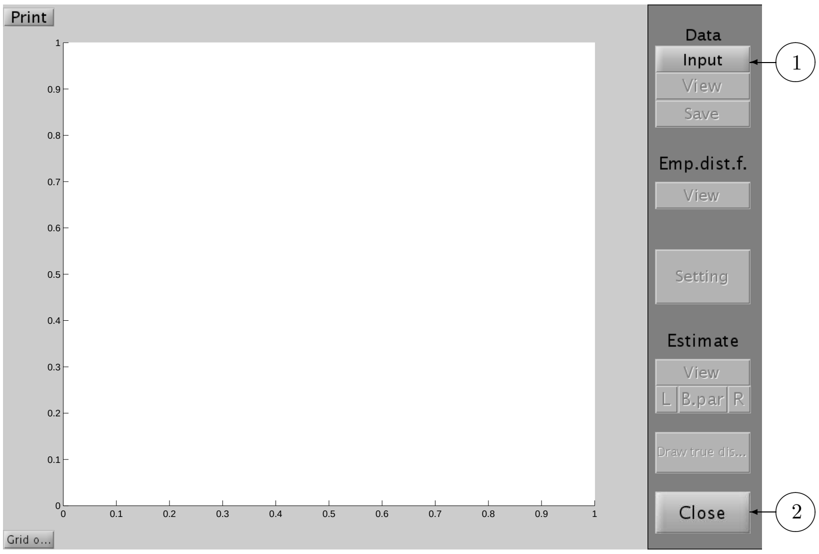


Figure 10: Start of the program.

The data input can be cancelled by button ⑧ or confirmed by button ⑨ .

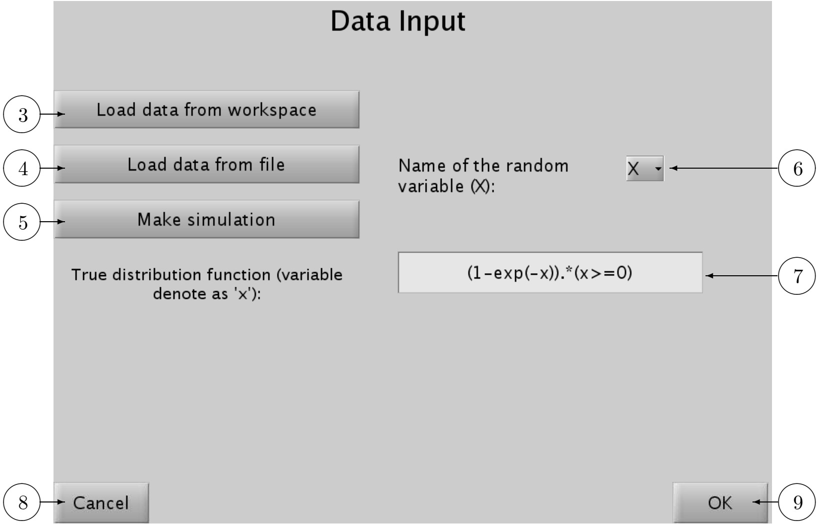
3.2 Main figure

After data input you can view data (button ⑩ , Figure 12) and save chosen values (button ⑪).

Button ⑫ displays the empirical distribution function (Figure 13). Button ⑬ invokes the setting of the parameters for smoothing (Figure 14).

3.3 Setting the parameters

In this subroutine you can choose a predefined kernel (button ⑯), an optimal kernel (button ⑰) and draw the kernel (button ⑱), but only kernels of order $\nu = 0, k = 2$ are available. The bandwidth can be selected in box ⑲ if the kernel is selected. Four



Name of the random variable (X): X

True distribution function (variable denote as 'x'): (1-exp(-x)).*(x>=0)

Figure 11: Data input.

methods for bandwidth choice are implemented (see Section 3.3 in [3]). The points for drawing the estimate of the density can be set up (㉑) in the right part of the window. Finally, you can confirm the setting (button ㉒) or close the windows without change of the parameters (button ㉓).

3.4 Eye-control method

You can use the *"Eye-control" method* (see Figure 15) for bandwidth choice by button ⑳ . First set the initial bandwidth and the step for its increasing or decreasing in boxes ㉔ and ㉕ . Pressing the middle button from ㉖ displays the estimate for actual bandwidth, the arrows at the left and the right side cause increasing or decreasing the bandwidth by a step and redrawing the figure. By buttons ㉗ you can run and stop the gradual increasing the bandwidth and drawing the corresponding estimate. Finally it is possible to accept selected bandwidth (button ㉘) or cancel the procedure (button ㉙) .

3.5 The final estimation

If the kernel and the bandwidth are set the estimate is displayed (button ㉚ , Figure 16). Now, it is possible to set the correction of the boundary effects (the middle button in ㉛) and display the result by pressing "L" or "R" for the left or the right boundary (see Figure 17). We can also display the true distribution function (button ㉜ , Figure 17), if it was entered.

The grid and the separate window with the estimate is also at disposal (buttons ㉝ and ㉞).

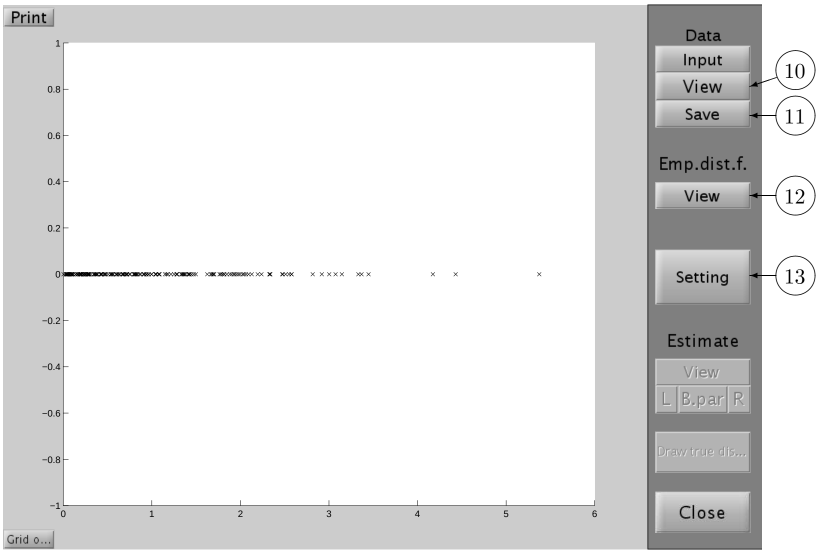


Figure 12: Data view.

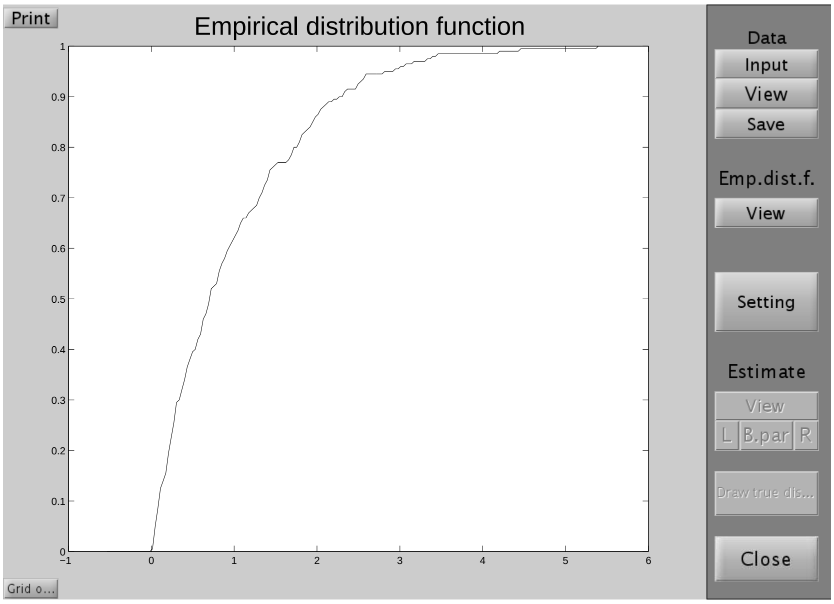
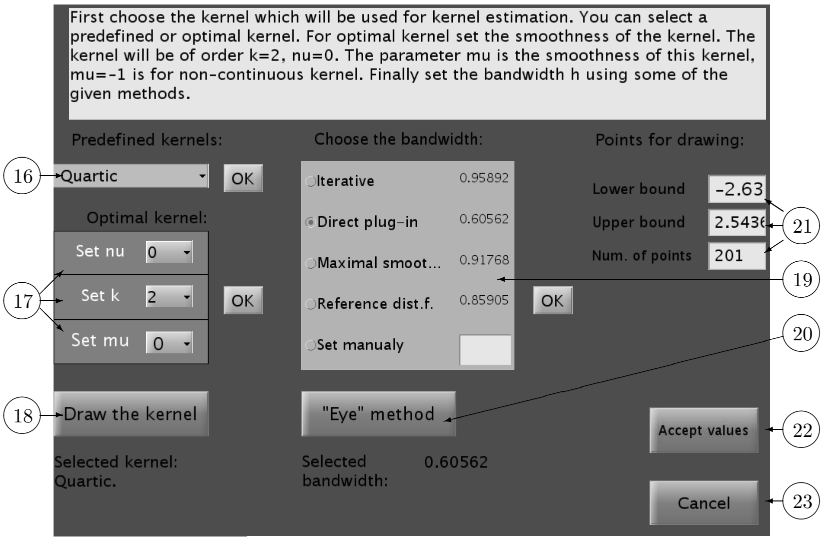


Figure 13: Empirical distribution function.


First choose the kernel which will be used for kernel estimation. You can select a predefined or optimal kernel. For optimal kernel set the smoothness of the kernel. The kernel will be of order k=2, nu=0. The parameter mu is the smoothness of this kernel, mu=-1 is for non-continuous kernel. Finally set the bandwidth h using some of the given methods.

Predefined kernels: Quartic

Optimal kernel:
Set nu 0
Set k 2
Set mu 0

Choose the bandwidth:

○ Iterative	0.95892
● Direct plug–in	0.60562
○ Maximal smoot...	0.91768
○ Reference dist.f.	0.85905
○ Set manualy	

Points for drawing:
Lower bound -2.63
Upper bound 2.543(
Num. of points 201

Selected kernel: Quartic.

Selected bandwidth: 0.60562

Figure 14: Setting the parameters.

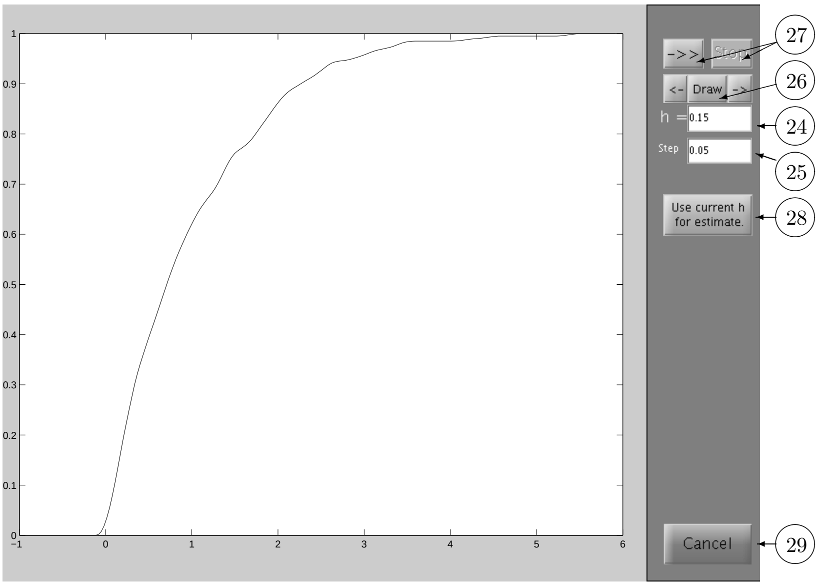

h = 0.15
Step 0.05

Figure 15: “Eye-control” method.

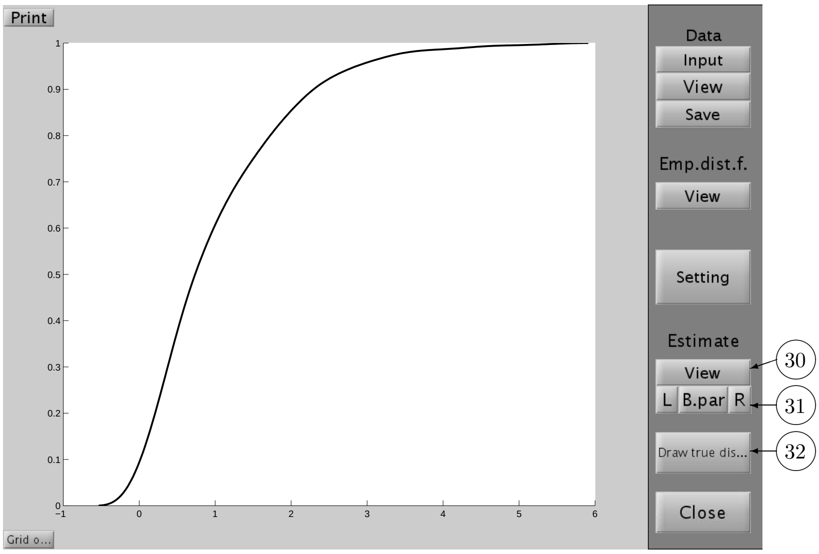


Figure 16: Kernel estimate.

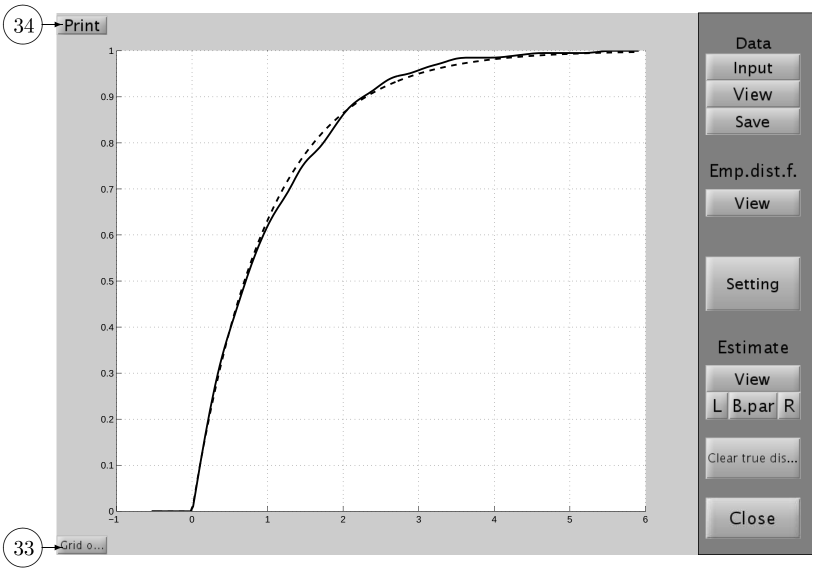
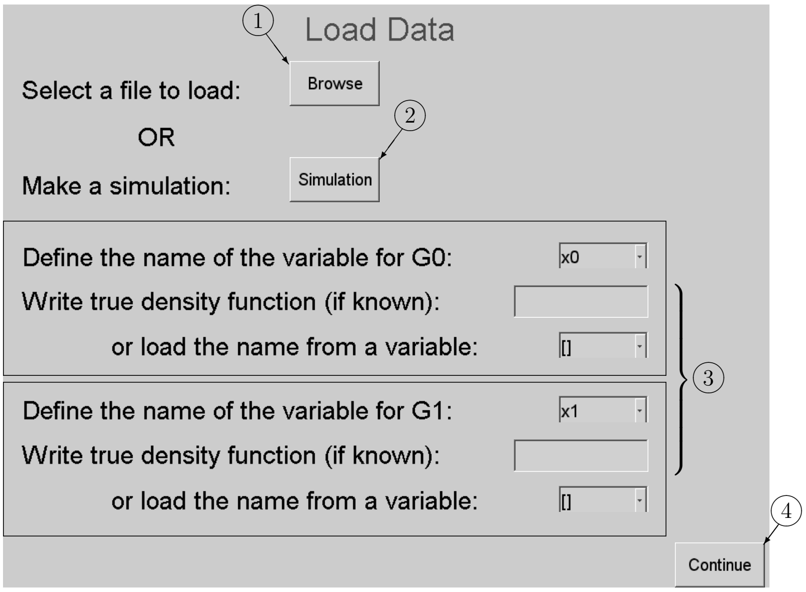


Figure 17: Estimate with boundary correction and the true distribution function.

4 Kernel estimation and reliability assessment

4.1 Running the program

The *Start menu* (Figure 18) for kernel estimation of quality indices is called up by the command `ksquality`.



Select a file to load:

OR

Make a simulation:

Define the name of the variable for G0: x0

Write true density function (if known):

or load the name from a variable: []

Define the name of the variable for G1: x1

Write true density function (if known):

or load the name from a variable: []

Figure 18: Start menu.

You can skip this menu by typing input data as an argument `ksquality(x0,x1)`, where the vectors `x0` and `x1` are score results for two groups $\mathcal{G}_0$ and $\mathcal{G}_1$. If we know also their densities f_0 and f_1 (for example for simulated data), we can set them as the next arguments. For more see `help ksquality`. After the execution of this command the window in Figure 22 is called up directly.

4.2 Start menu

In the *Start menu*, you have several possibilities how to define input data. You can load it from a file (button ①) or simulate data (button ②). In the fields ③ you can list your variables in the current workspace to define input data. If your workspace is empty, these fields are nonactive. If you know the true densities for both groups, you can write them to text fields or load them from selected variables. If you need to simulate values for a model, press button ② . Then the menu for simulation (Figure 19) is called up.

4.3 Simulation menu

In the *Simulation menu*, first it is necessary to generate random samples for both groups by buttons ⑤ . Either of these buttons calls up the *Data generating menu* (Figure 20).

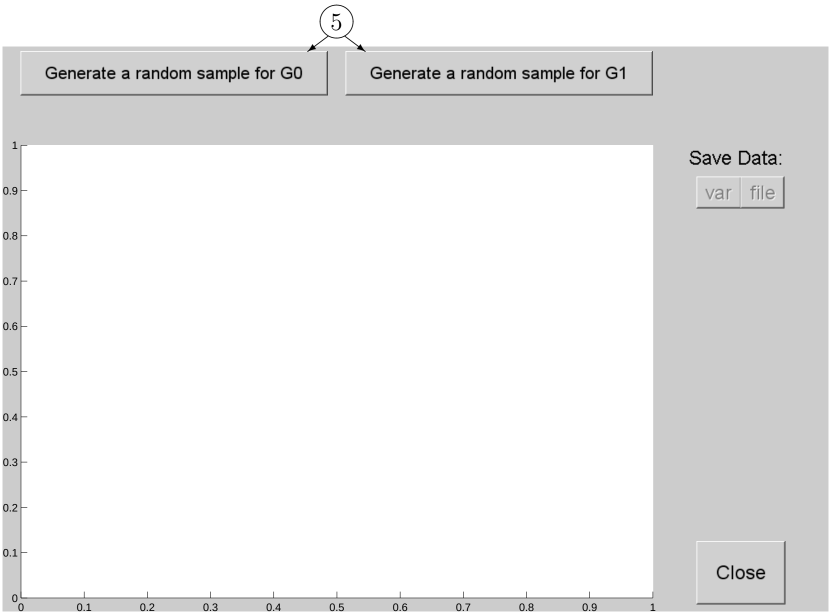
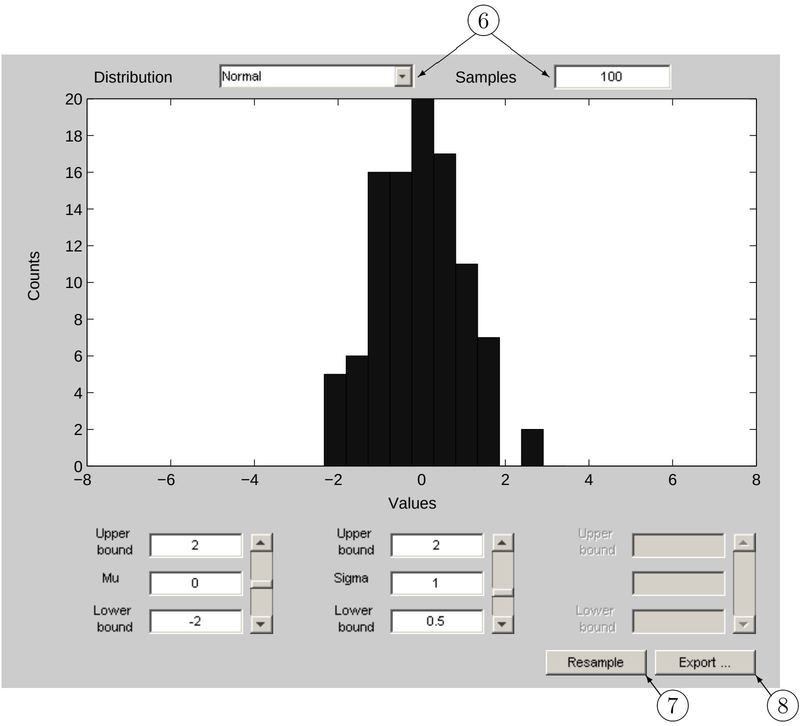


Figure 19: Simulation menu – start.

In this menu, you can set the type of distribution and the size of a sample (fields ⑥). In the figure, the histogram for the generated sample is illustrated. In the bottom part of the menu, you can change parameters of a distribution and make a new sample by ⑦ . If you have done the sample generating, you need to export your data. The button ⑧ calls up the menu where you specify the name of variable and confirm by pressing "OK". Then the *Data generating menu* will be closed and you will be returned to the *Simulation menu*. After data generating for both groups it looks like Figure 21. In the figure, the histograms of generated samples for both groups are illustrated. The cyan color represents data for $\mathcal{G}_0$ and the red color is for $\mathcal{G}_1$. In this stage you can save data to variables or as a file by using buttons ⑨ . If you have done the simulation, press button ⑩ . The *Simulation menu* will be closed and you will be returned to the *Start menu* (Figure 18). In this menu, you can redefine the input data. If you want to continue, press button ④ . The menu will be closed and the *Basic menu* (see the next paragraph) will be called up.

4.4 The final estimation

This menu (Figure 22) was called up from the *Start menu* or directly from the command line (see `help ksquality`). At the start of this application, you can see some color symbols in the figure. The blue crosses represent the score values $X_{01}, \ldots, X_{0n_0}$ for the first group $\mathcal{G}_0$, the red circles illustrate the score values $X_{11}, \ldots, X_{1n_1}$ for the second group $\mathcal{G}_1$.



Distribution: Normal

Samples: 100

Upper bound: 2

Mu: 0

Lower bound: -2

Upper bound: 2

Sigma: 1

Lower bound: 0.5

Figure 20: Data generating menu.

In the menu, there are six fields for computation of indices for reliability assessment. Each field contains two buttons for estimation. The "empiric" button represents the empirical estimate of the related index and the "kernel" button stands for the estimate obtained by kernel smoothing. For all cases the Epanechnikov kernel is used and the optimal smoothing parameter is estimated by the method of maximal smoothing principle. To avoid the boundary effects in estimation the reflection method is applied. For more details see Chapter 2 and 3 in [3].

Let us describe all fields more properly:

- "ROC" – in the first field you obtain the ROC curve for the actual model. At the right hand side of the used button the value of AUC (defined by (4.8) in [3]) is written. The actual curve is plotted in the figure, see Figure 23.
- "Gini" – this field stands for the estimation of the Gini index defined in (4.11) in [3].
- "MIS" – estimates the Kolmogorov – Smirnov statistics denoted as MIS (Maximum Improvement of Sensitivity over chance diagonal, see §4.3.2 in [3]).
- "Inf. Value" – the estimate of the Information Value (see (4.16) in [3]) is computed and the related function f_{IV} is plotted.

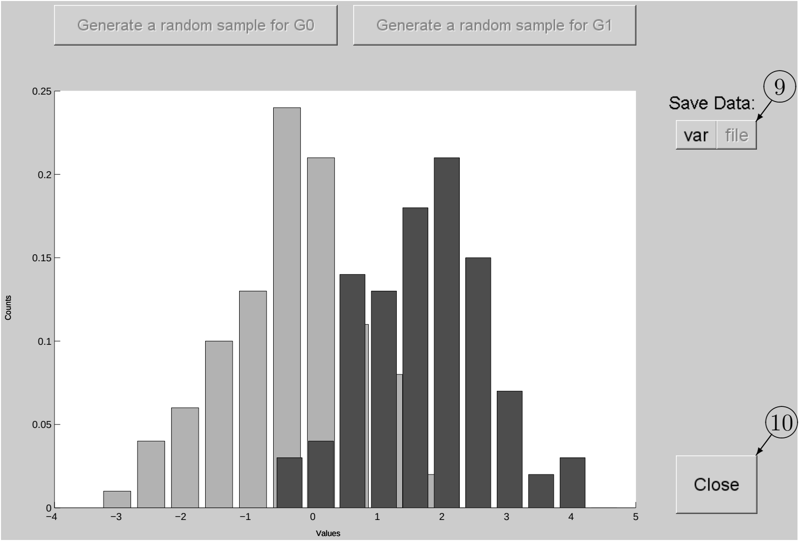


Figure 21: Simulation menu.

- "KR" – this field is for estimating the KR index (see (4.17) in [3]) and plotting the function χ^2.
- "Lift" – set the value q (from interval $(0,1)$) first and then estimate the cumulative Lift function (see (4.12) in [3]) at this point, the estimate is plotted in the current figure. The terms "LR" and "IRL" stand for the estimation of the Lift Ratio (see 4.14 in [3]) and the Integrated Relative Lift (see 4.15 in [3]), respectively.

If you want to show only the actual curve, use button ⑪ . You can also save data to variables and then as a file by using buttons ⑫ . Button ⑬ ends the application.

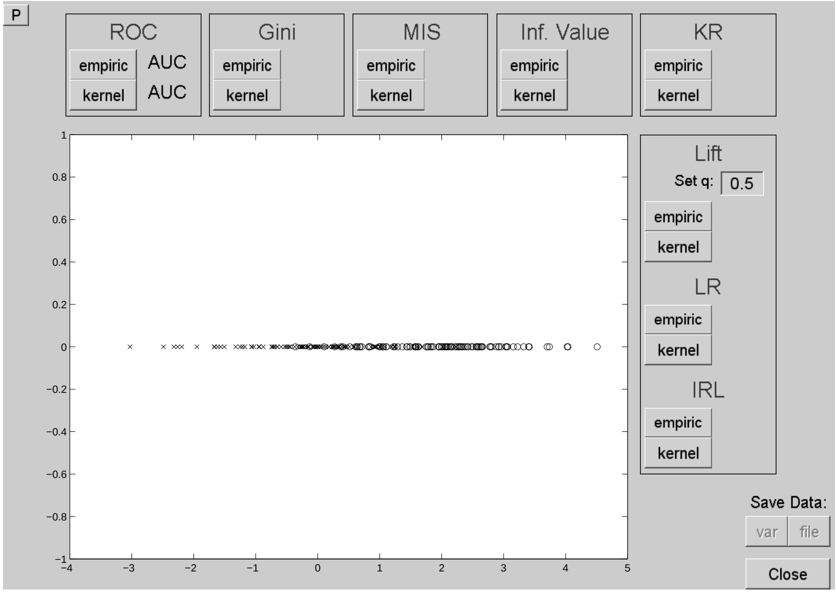


Figure 22: Basic menu.

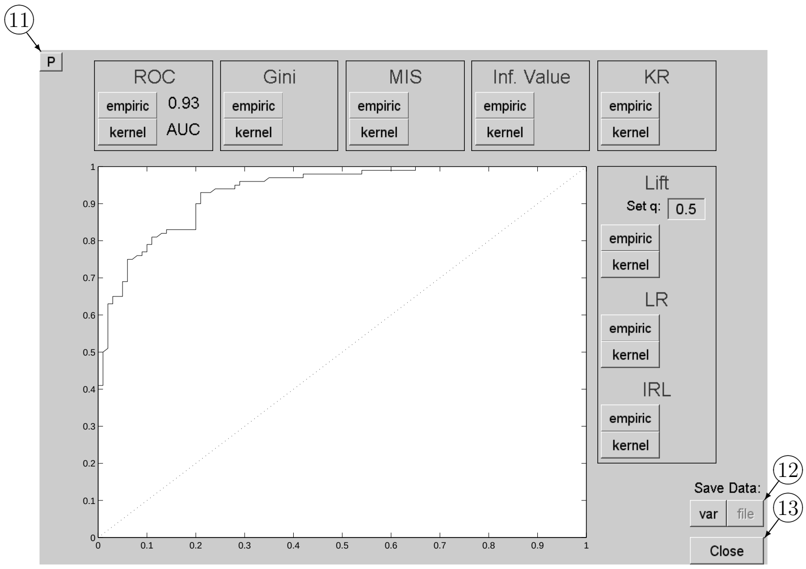


Figure 23: Basic menu.

5 Kernel estimation of a hazard function

5.1 Running the program

The program can be launched by command `kshazard`. If we launch it without parameters it is possible only input data (button ①) or terminate the program (button ②) Figure 24).

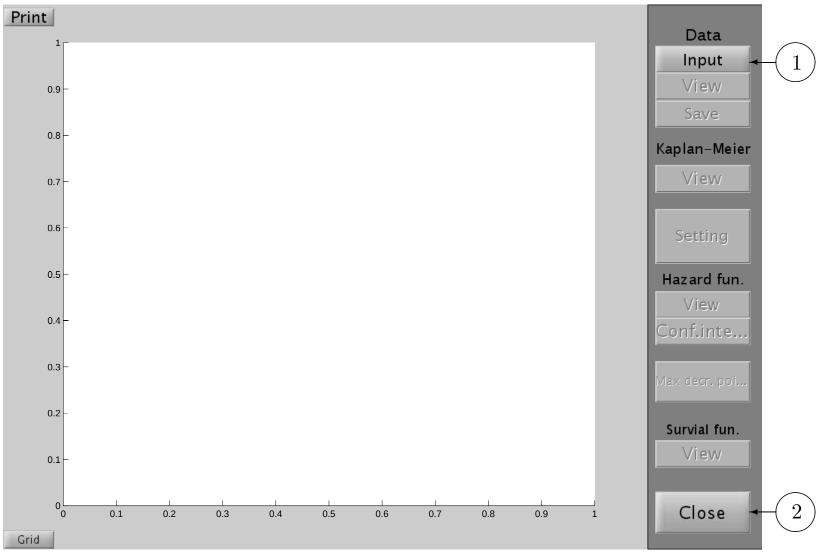


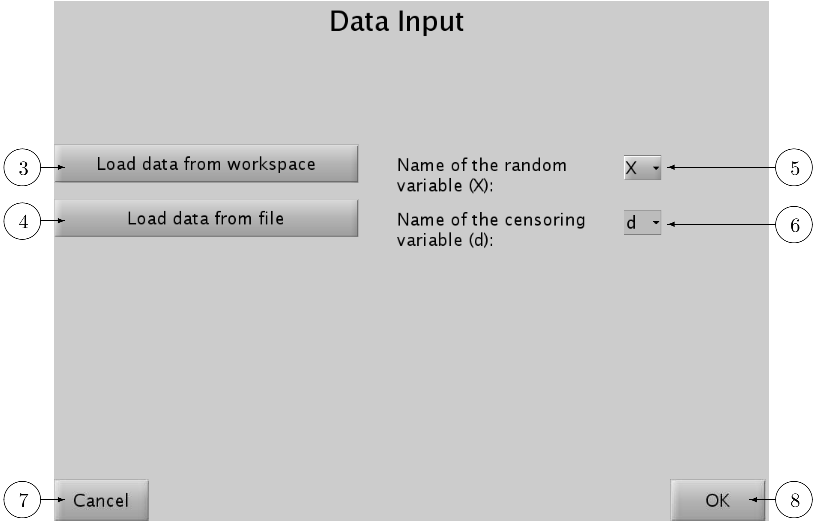
Figure 24: Start of the program.

There is no simulation in the data input window (Figure 25) as the creating of simulated censored data for given hazard function is rather complicated (see Section 5.7 in [3]). We can select the source of the data (the workspace by button ③ or the file by button ④). Then we choose the names of used variables containing the lifetimes and censoring indicator (buttons ⑤ and ⑥). Finally confirm the values or cancel the subroutine (buttons ⑦ and ⑧).

5.2 Main figure

Data input causes the same situation as if the toolbox is called with parameters (*e.g.*, `kshazard(X,d)`). A window with data are displayed, uncensored data is represented by crosses, censored by circles (see Figure 26, button ⑨). Button ⑩ will open the window for data saving.

It is also possible to display Kaplan–Meier estimate of the survival function by button ⑪ (Figure 27). Button ⑫ opens the window, where the parameters for kernel estimate of hazard function can be set (see Figure 28).



Data Input

Name of the random variable (X): X

Name of the censoring variable (d): d

Figure 25: Data input.

5.3 Setting the parameters

In this subroutine you can choose a predefined kernel (button ⑮), an optimal kernel (button ⑯) and draw the kernel (button ⑰). The optimal kernel can be obtained only for $\nu = 0$. Only three methods for bandwidth selection (button ⑱) are available (see Section 5.3 in [3]).

In boxes ⑳ in the setting window the points for drawing the estimates can be set. Finally we can confirm the selected values by button ㉑ or cancel the subroutine by button ㉒ .

5.4 Eye-control method

Procedure with the *"Eye-control" method* for bandwidth selection can be invoked by button ⑲ (see Figure 29). In boxes ㉓ and ㉔ we can set the bandwidth and the step for its increasing or decreasing. Pressing the middle button in ㉕ displays the estimate for the actual bandwidth. The two arrows at the left and the right side cause increasing or decreasing the bandwidth by a step and redrawing the figure. By buttons ㉖ we can run and stop the gradual increasing the bandwidth and drawing the corresponding estimate. Finally it is possible to accept actual bandwidth (button ㉗) or cancel the procedure (button ㉘) .

5.5 The final estimation

After setting the parameters we obtain a window from Figure 30 with the kernel estimate of hazard function (also button ㉙). In addition, we have other options: to display the confidence intervals for the kernel estimate of the hazard function (button ㉚ , Figure 31), to show the points of the most rapid change (decreasing) of the estimate

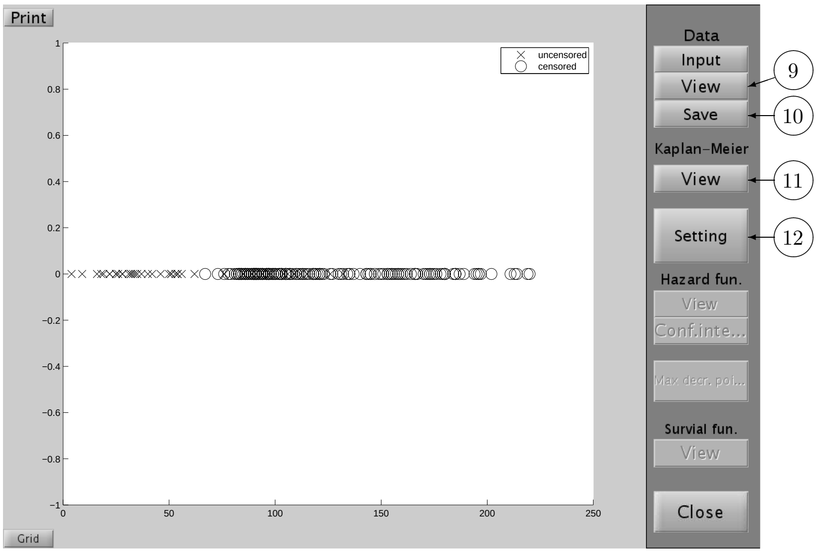


Figure 26: Data view.

of the hazard function (button ㉛ , Figure 32) and to present the kernel estimate of the survival function (button ㉜ , Figure 33).

The grid and the separate window with the estimate can be also invoked (buttons ㉝ and ㉞).

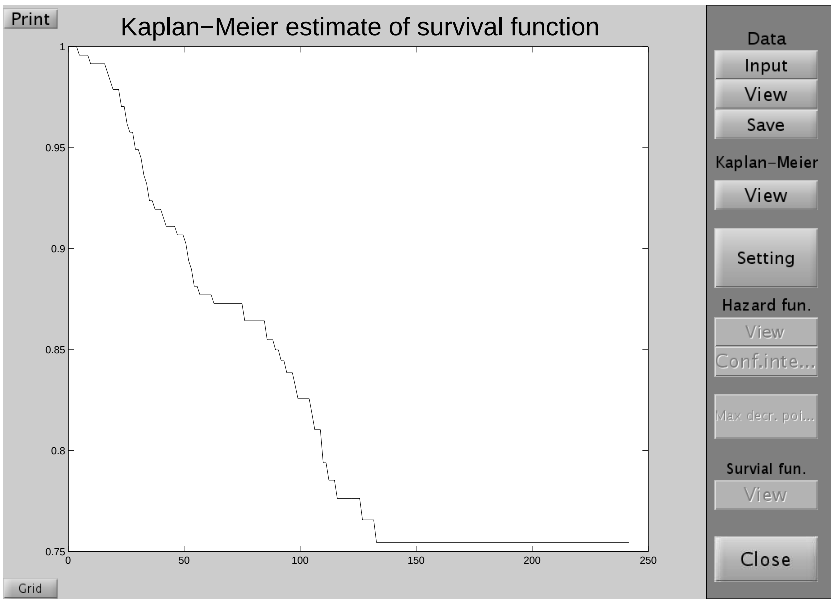
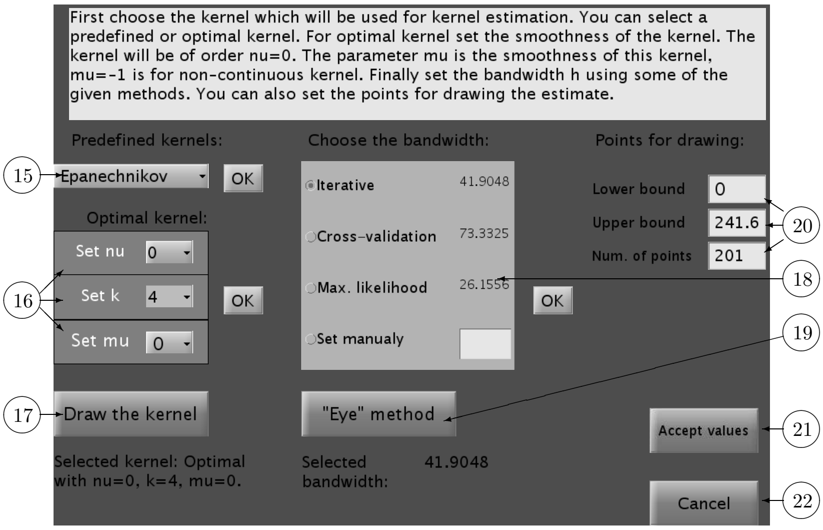


Figure 27: Kaplan–Meier estimate of survival function.



First choose the kernel which will be used for kernel estimation. You can select a predefined or optimal kernel. For optimal kernel set the smoothness of the kernel. The kernel will be of order nu=0. The parameter mu is the smoothness of this kernel, mu=-1 is for non-continuous kernel. Finally set the bandwidth h using some of the given methods. You can also set the points for drawing the estimate.

Predefined kernels: Epanechnikov

Optimal kernel:

Set nu 0

Set k 4

Set mu 0

Choose the bandwidth:

● Iterative 41.9048

○ Cross−validation 73.3325

○ Max. likelihood 26.1556

○ Set manualy

Points for drawing:

Lower bound 0

Upper bound 241.6

Num. of points 201

Selected kernel: Optimal with nu=0, k=4, mu=0.

Selected bandwidth: 41.9048

Figure 28: Setting the parameters.

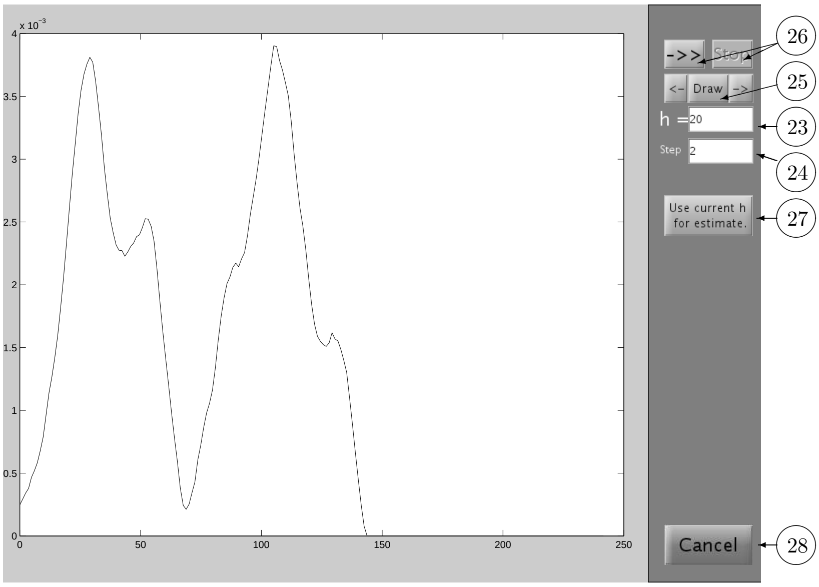


Figure 29: “Eye-control” method.

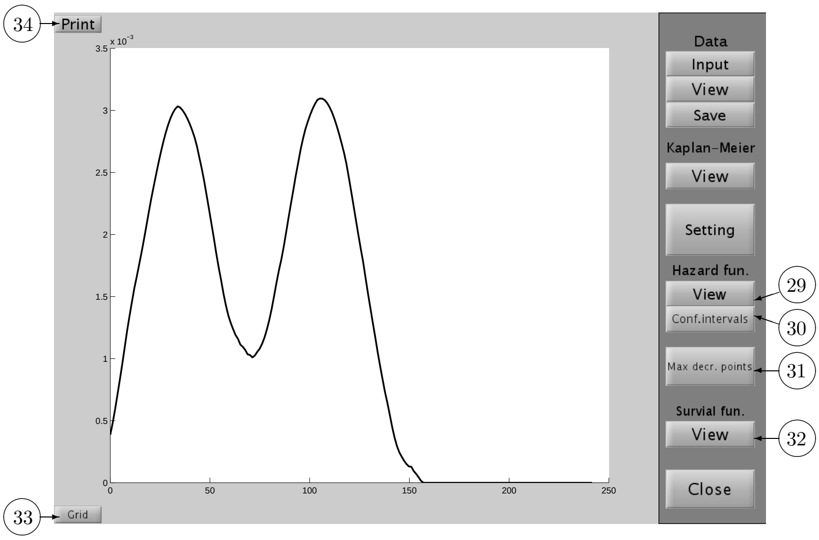


Figure 30: Estimate of the hazard function.

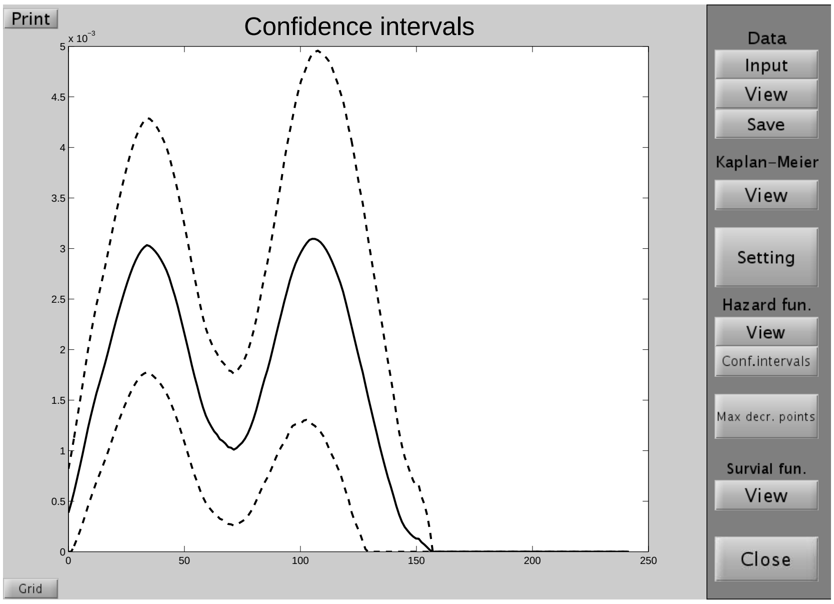


Figure 31: Confidence intervals.

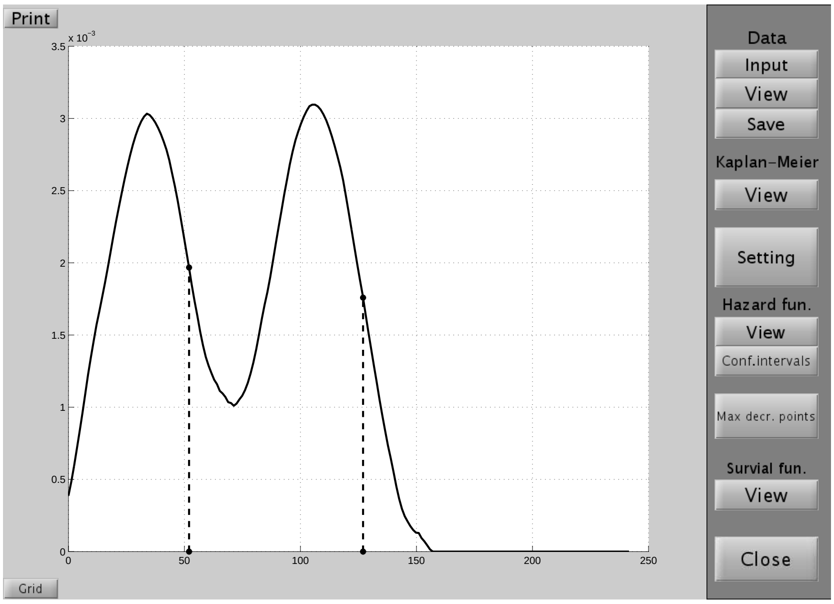


Figure 32: Points of the most rapid decreasing.

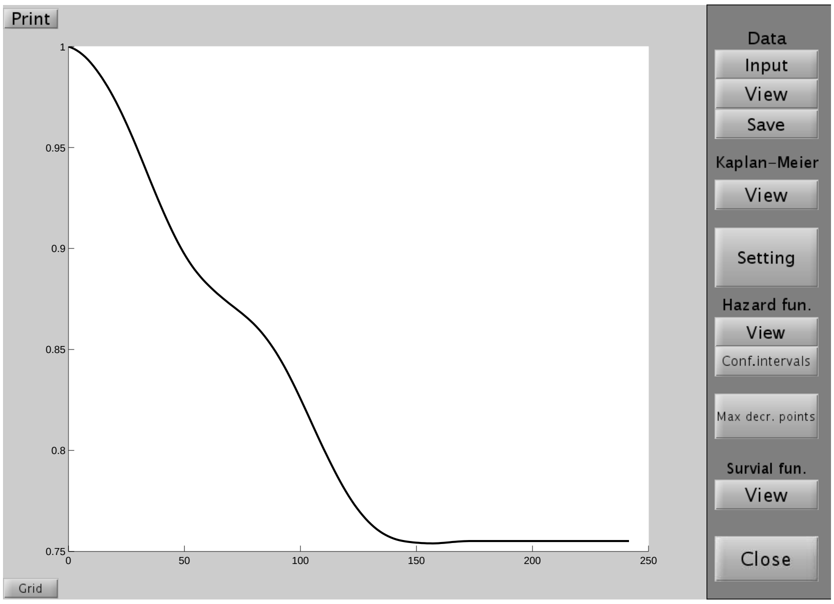


Figure 33: Kernel estimate of the survival function.

6 Kernel estimation of a regression function

6.1 Running the program

The *Start menu* (Figure 34) for kernel regression is called up by the command `ksregress`.

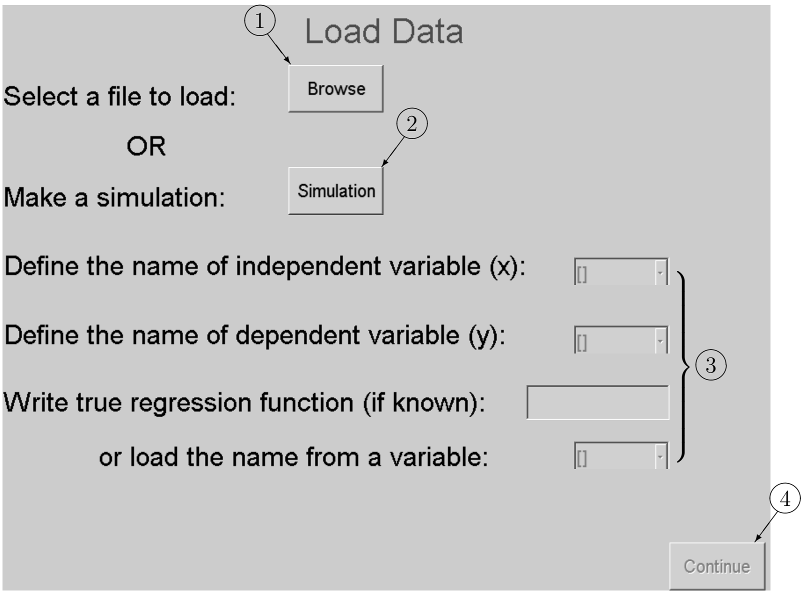
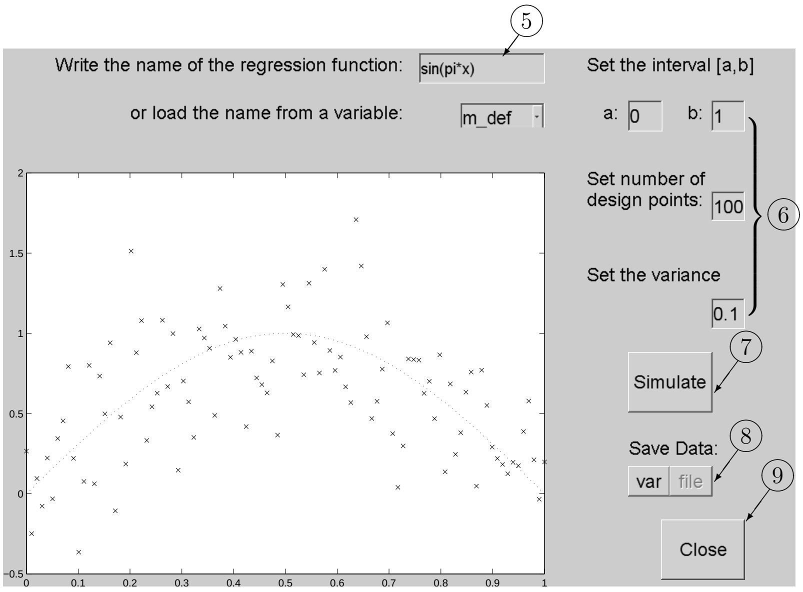

1
Load Data
Select a file to load:
Browse
2
OR
Make a simulation:
Simulation
Define the name of independent variable (x):
[]
Define the name of dependent variable (y):
[]
3
Write true regression function (if known):
or load the name from a variable:
[]
4
Continue

Figure 34: Start menu.

You can skip this menu by typing input data as an argument `ksregress(x,y)`, where the vectors x and y should be of the same length n and they mark x and y axes of measurements. If we know also the true regression function $m(x)$ (for example for simulated data), we can set it as the next argument. For more see `help ksregress`. After executing this command directly the window in Figure 36 is called up.

In the *Start menu*, you have several possibilities how to define input data. You can load it from a file (button ①) or simulate data (button ②). In the fields ③ you can list your variables in the current workspace to define input data. If your workspace is empty, these fields are nonactive. If you know the true regression function of the model, you can write it to the text field or load it from a variable. If you need to simulate a regression model, press button ② . Then the menu for simulation (Figure 35) is called up.

In the *Simulation menu*, first, set the regression function. You can write it to the text field ⑤ (after doing it press ENTER or click anywhere outside the field) or load it from a variable. In the fields ⑥ specify the interval, the number of design points and the variance. You can simulate a regression model by pressing button ⑦ . Then you can save data to variables and then as a file by using buttons ⑧ . If you have done the simulation, press button ⑨ . The *Simulation menu* will be closed and you will be returned to the *Start menu*. In this menu, you can redefine the input data. If you want



Write the name of the regression function: sin(pi*x)

or load the name from a variable: m_def

Set the interval [a,b]

a: 0 b: 1

Set number of design points: 100

Set the variance 0.1

Figure 35: Simulation menu.

to continue, press button ④ . The menu will be closed and the *Basic menu* (see Figure 36) will be called up.

6.2 Main figure

This menu (Figure 36) was called up from the *Start menu* or directly from the command line (see `help ksregress`). The values of independent variable x are automatically transformed to the interval $[0, 1]$. Symbols × mark the measurements after this transformation. If you want to show the original data, use button ⑪ . Button ⑫ ends the application. Press button ⑩ to continue. Other buttons are nonactive.

6.3 Setting the parameters

Button ⑩ calls up the menu for setting of the parameters which will be used for kernel regression, see Figure 37.

In the array ⑬ , we can set kernel regression parameters. First, set the order of the kernel (ν, k), where $\nu + k$ should be even, for the regression estimation $\nu = 0$ is used. The parameter μ is the smoothness of this kernel. If you want to draw the kernel, use button ⑭ . Finally set the bandwidth h. The value should be between 0 and 1. To confirm the setting use ⑮ , to close the window, use ⑯ . The other buttons are useful for choosing the optimal bandwidth.

Button ⑰ calls up the *"Eye-control" menu* (see Figure 39), where we can change the value of h and observe the effect upon the final estimate. Button ⑱ starts the al-

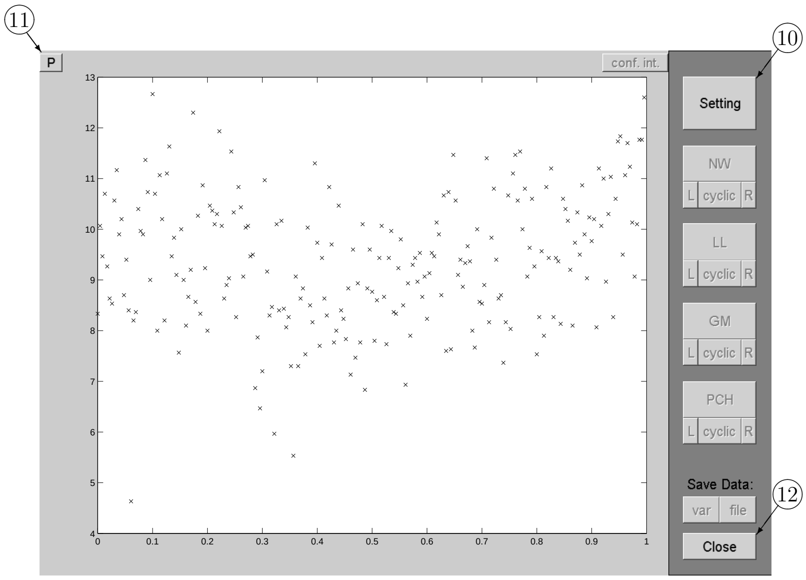
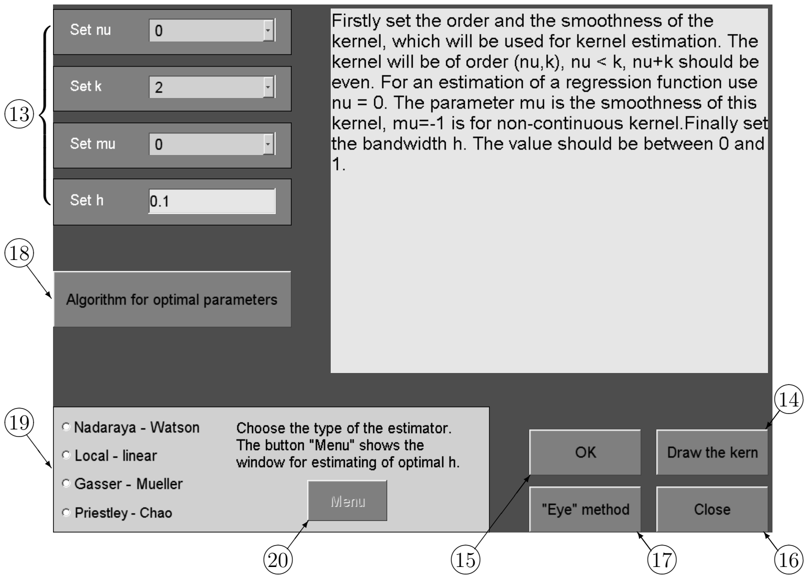


Figure 36: Basic menu.

gorithm for estimation of optimal kernel order and optimal bandwidth (see §6.4). This algorithm automatically sets the values of optimal parameters in the array ⑬ . By selecting one type of kernel estimators in ⑲ you make active ⑳ . This button calls up the menu for using and comparing various methods for choosing the optimal smoothing parameter h (see §6.6).



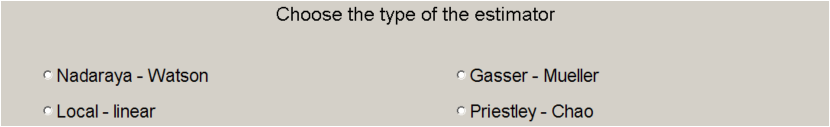
Firstly set the order and the smoothness of the kernel, which will be used for kernel estimation. The kernel will be of order (nu,k), nu < k, nu+k should be even. For an estimation of a regression function use nu = 0. The parameter mu is the smoothness of this kernel, mu=-1 is for non-continuous kernel.Finally set the bandwidth h. The value should be between 0 and 1.

Choose the type of the estimator. The button "Menu" shows the window for estimating of optimal h.

Figure 37: Setting of parameters.

6.4 Estimation of optimal parameters

Button ⑱ calls up an algorithm for the estimation of optimal order of kernel and optimal bandwidth. At first, it is necessary to set the type of kernel estimator:


Choose the type of the estimator

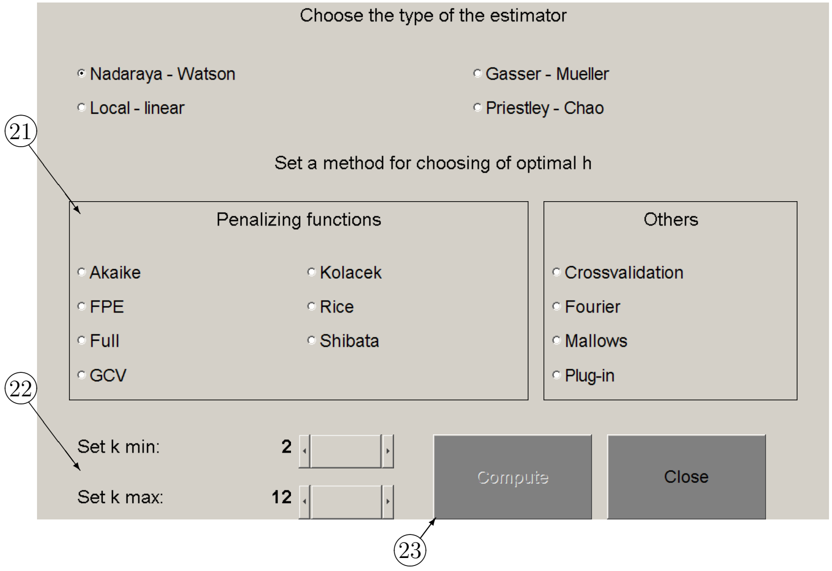
Next, the menu for estimating the optimal bandwidth is called up (see Figure 38).

By choosing one of the methods in the array ㉑ we make active button ㉓ which starts the computation of optimal parameters k and h (we suppose $K \in S^0_{0,k}$). In the array ㉒ , we can set limits for the parameter k, the default values are $k_{min} = 2$, $k_{max} = 12$. The results of the computation are automatically set in the array ⑬ .

6.5 Eye-control method

Button ⑰ calls up a window, where we can change the value of parameter h and observe the effect of these changes upon the final estimate (see Figure 39).

In the arrays ㉔ , set the starting value of parameter h and the step (it can be positive or negative number) for the size of changes of h. The left button ㉕ starts a sequence of pictures representing the quality of estimation dependence on h. The right button stops the sequence. You can change the value of h only one step more or less by buttons ㉖ .



Choose the type of the estimator

● Nadaraya - Watson ○ Gasser - Mueller

○ Local - linear ○ Priestley - Chao

Set a method for choosing of optimal h

Penalizing functions

○ Akaike ○ Kolacek

○ FPE ○ Rice

○ Full ○ Shibata

○ GCV

Others

○ Crossvalidation

○ Fourier

○ Mallows

○ Plug-in

Set k min: 2

Set k max: 12

Figure 38: Estimation of optimal parameters.

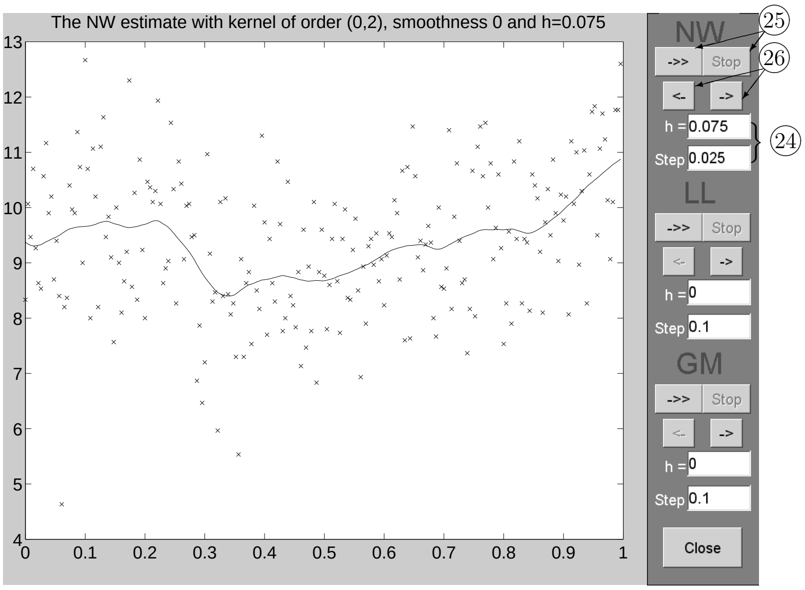
6.6 Comparing of methods for bandwidth selection

Button ⑳ calls up the window for using and comparing of various methods for the optimal bandwidth selection (see Figure 40).

In this window, all mentioned bandwidth estimators are summarized. In the array ㉗ , there are presented:

- method of penalizing functions – see §6.3.3 in [3]
- the cross-validation method (denoted as "Classic CV") – see §6.3.2 in [3].

By clicking on the button with the method's name, the optimal bandwidth is computed by the actual method. The button "draw" calls up a graph of the minimized error function. To draw all penalizing functions applied up to this time use ㉘ . Button ㉙ represents Mallows' method (see §6.3.1 in [3]), button ㉚ denotes the method of Fourier transformation described in the first part of §6.3.4 in [3] and ㉛ marks the plug-in method, *i.e.*, the bandwidth estimate $\hat{h}_{\mathrm{PI}}$ given by (6.35) in [3]. If the original regression function is known (for example for simulated data), we can compute the value of optimal bandwidth as a minimum of AMISE$\{\widehat{m}(\cdot, h)\}$ (see the formula (6.12) in [3]) in the array ㉜ . To do this computation, it is necessary to have the *Symbolic toolbox* installed on your computer. If this toolbox is not installed or the regression function is not defined on input, the array is not active. For the graph of all error functions and their minimal values use ㉝ . Button ㉞ closes the application.



NW: h = 0.075, Step 0.025

LL: h = 0, Step 0.1

GM: h = 0, Step 0.1

Figure 39: “Eye-control” menu.

6.7 The final estimation

If you have set all values in the window for parameters setting (see Figure 37) and if you want to go to the final estimation of the regression function, confirm your setting by button ⑮ . It calls up the *Basic menu*, where all buttons are active already.

By clicking on the button with the estimator’s name (for example ㉟ for the Nadaraya – Watson estimator), the relevant regression estimate is drawn (solid line in the figure). The button “cyclic” shows the regression estimate with using the assumption of cyclic model. By using buttons “L” and “R” we get the estimate near the boundary of the interval obtained by using special boundary kernels (L=left, R=right). Button ㊱ draws confidence intervals (dashed) computed by using the formula (6.14) in [3]. To do this computation, it is necessary to have the *Stats toolbox* installed on your computer. If this toolbox is not installed, the button is not active. Button ㊲ shows original data and the final estimate. You can also save data to variables and then as a file by using buttons ㊳ . Button ㊴ ends the application.

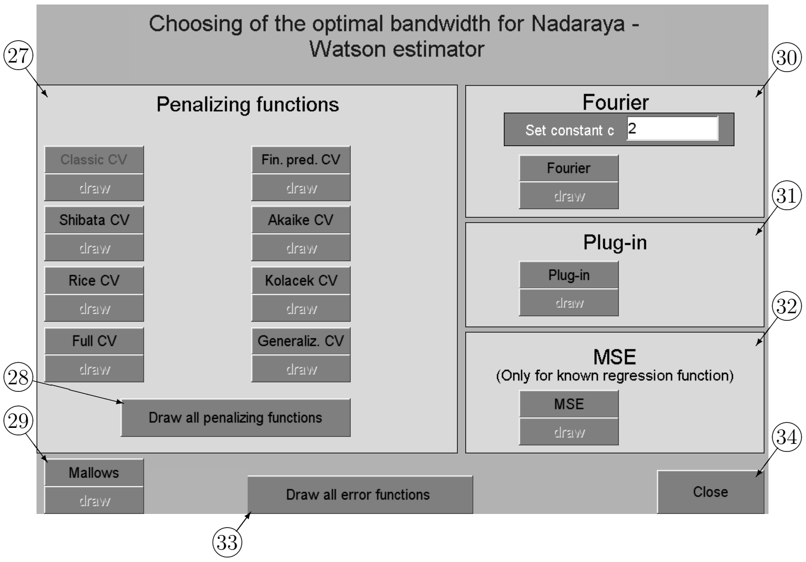


Figure 40: Bandwidth estimators.

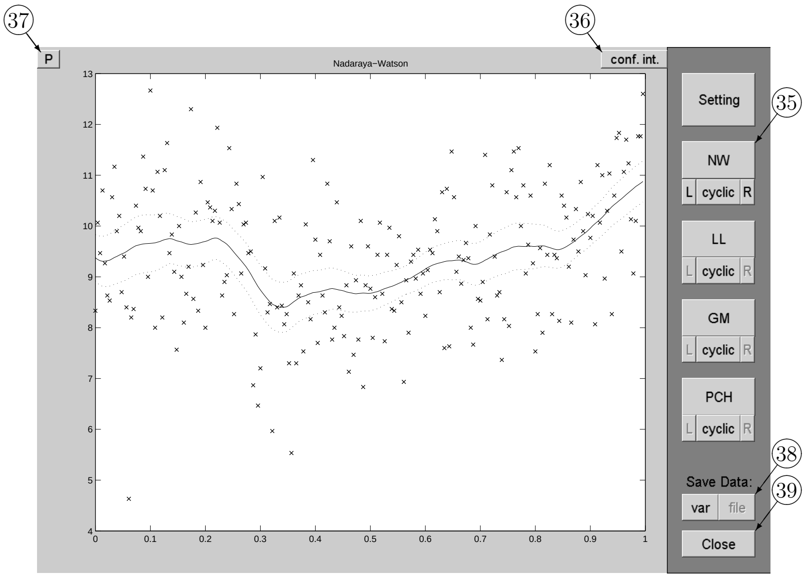


Figure 41: Original data and the final kernel regression estimate.

7 Multivariate kernel density estimation

7.1 Running the program

The *Start menu* (Figure 42) for kernel estimation of two-dimensional density is called up by the command `ksbivardens`.

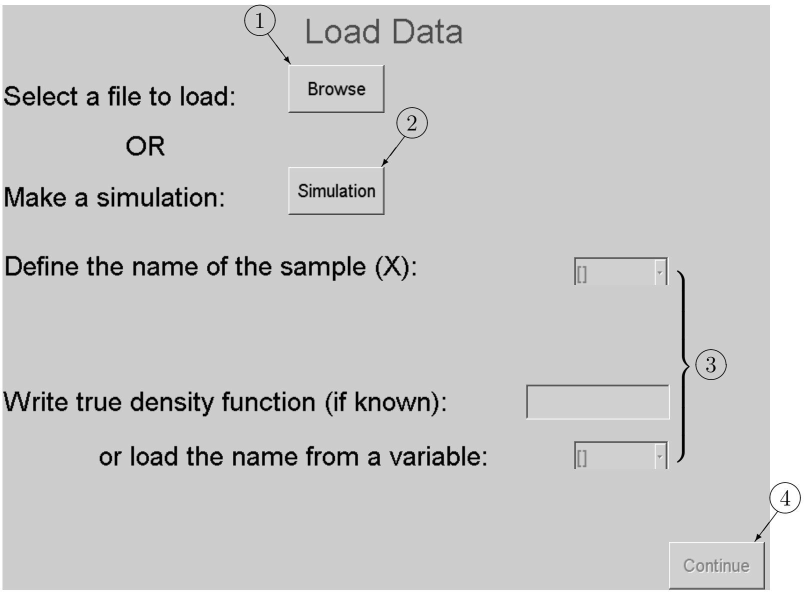

①
Load Data
Select a file to load:
Browse
②
OR
Make a simulation:
Simulation
Define the name of the sample (X):
[]
③
Write true density function (if known):
or load the name from a variable:
[]
④
Continue

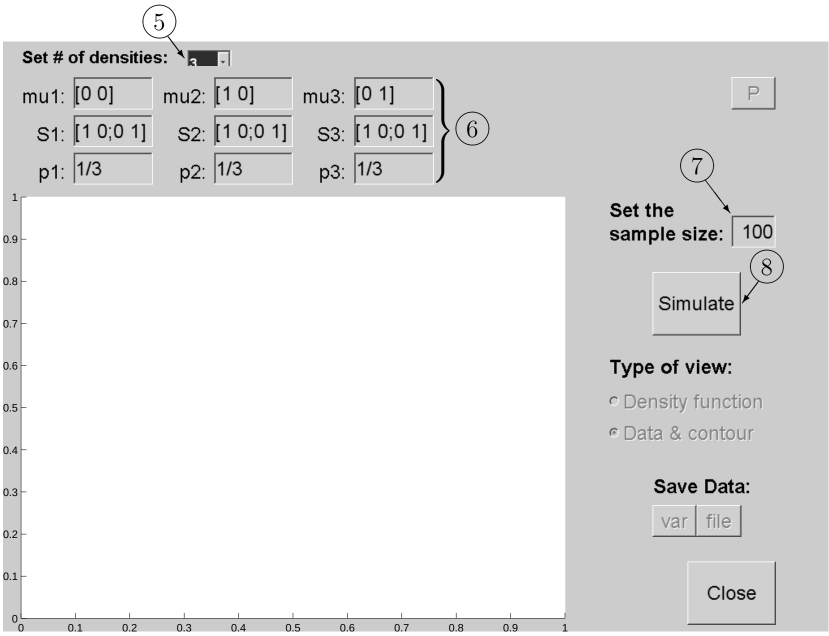
Figure 42: Start menu.

You can skip this menu by typing input data as an argument `ksbivardens(X)`, where the matrix X should have the size $2 \times n$, where n is the sample size. If we know also the original density f (for example for simulated data), we can set it as the next argument. For more see `help ksbivardens`. After the execution of this command, the window in Figure 45 is called up directly.

In the *Start menu*, you have several possibilities how to define input data. You can load it from a file (button ①) or simulate data (button ②). In the fields ③ you can list your variables in the current workspace to define input data. If your workspace is empty, these fields are nonactive. If you know the true density of the sample, you can write it to the text field or load it from a variable. If you need to simulate a sample, press button ② . Then the menu for simulation (Figure 43) is called up. This application generates a random sample from a two-dimensional normal mixture density.

In the *Simulation menu*, first set the number of components of the mixture by ⑤ . You can choose from 1 to 5 components of the normal mixture density. In the fields ⑥ specify the parameters (mean, variance matrix and proportion) of each component. By ⑦ specify the sample size. You can simulate a sample by pressing button ⑧ and then see the result (Figure 44).

By clicking on ⑨ you can print the current plot to a new figure. In the fields ⑩ you switch the type of view between the data plot with contours and the 3-D plot of the den-



Set # of densities: 3

mu1: [0 0] mu2: [1 0] mu3: [0 1]

S1: [1 0;0 1] S2: [1 0;0 1] S3: [1 0;0 1]

p1: 1/3 p2: 1/3 p3: 1/3

Set the sample size: 100

Type of view:

○ Density function

● Data & contour

Figure 43: Simulation menu.

sity function. You can also save the obtained data to variables and then as a file by using buttons ⑪ . If you have done the simulation, press button ⑫ . The *Simulation menu* will be closed and you will be returned to the *Start menu*. In this menu, you can redefine the input data. If you want to continue, press button ④ . The menu will be closed and the *Basic menu* (see the next subsection) will be called up.

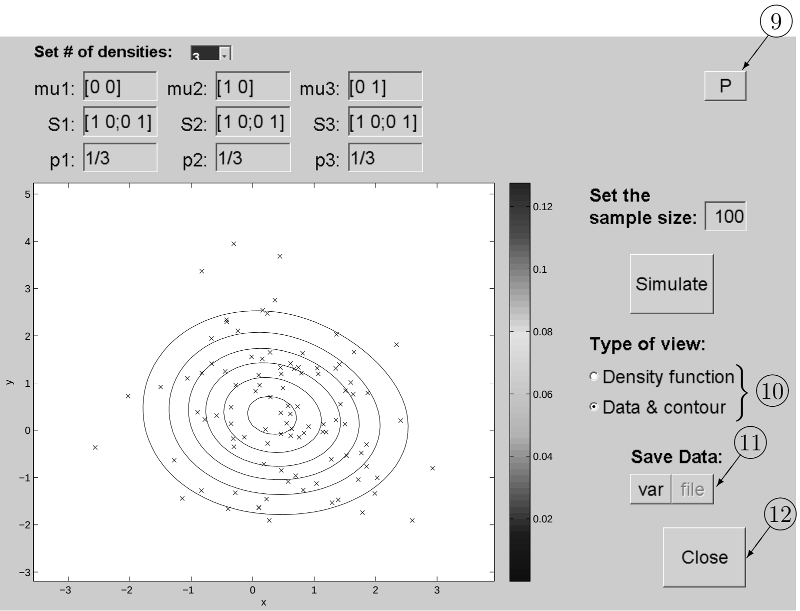
7.2 Main figure

This menu (Figure 45) was called up from the *Start menu* or directly from the command line (see `help ksbivardens`). If the original density is known, in ⑭ you can switch the type of view between the data plot with contours and the 3-D plot of the density function. By clicking on ⑮ you can show or hide contours of original density. If the original density is unknown, only data are plotted. Use button ⑬ to continue.

7.3 Setting the parameters

Button ⑬ calls up the menu for setting the parameters (Figure 46) which will be used for bivariate kernel density estimation.

In the array ⑯ , you can set a type of the kernel. Implicit setting is the Gaussian kernel or it can be changed to polynomial kernel. In this case, set the type of kernel (product or spherically symmetric) and set the order of the kernel (ν, k), where $\nu + k$ should be even, for density estimation $\nu = 0$ is used. The parameter μ is the smoothness



Set # of densities: 3

mu1: [0 0] mu2: [1 0] mu3: [0 1]

S1: [1 0;0 1] S2: [1 0;0 1] S3: [1 0;0 1]

p1: 1/3 p2: 1/3 p3: 1/3

Set the sample size: 100

Type of view:

○ Density function

● Data & contour

Figure 44: Simulation menu – results.

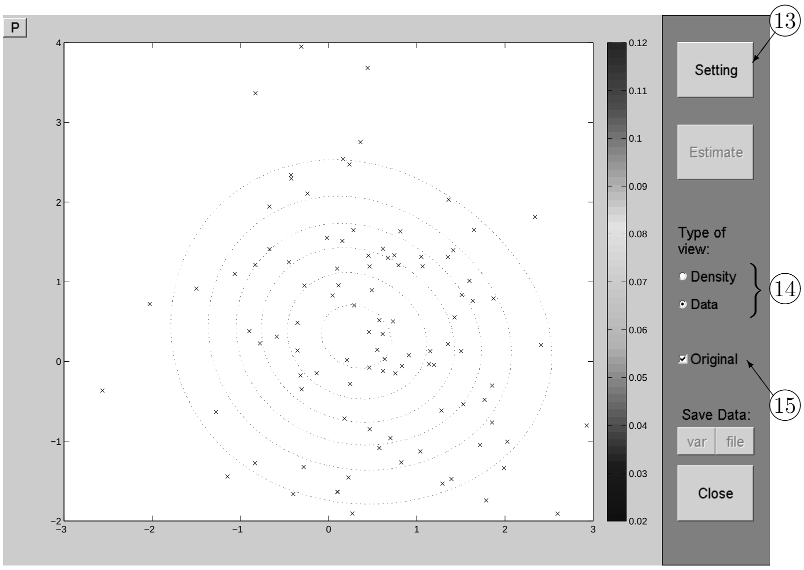
of this kernel. If you want to draw the kernel, use button ⑱ . In the array ⑰ , specify the bandwidth matrix **H**. The matrix should be symmetric and positive definite. If these conditions are not satisfied, the application writes an error message. The implicit setting is

$$\widehat{\mathbf{H}} = \widehat{\mathbf{\Sigma}} n^{-1/3}, \tag{1}$$

where $\widehat{\mathbf{\Sigma}}$ is the empirical estimate of the covariance matrix and n is the number of observations. This formula (known as “Reference rule”) is based on the assumption of normal density and Gaussian kernel (see §7.3.3 in [3]). The terms of the bandwidth matrix can be set manually or you can use one of the buttons in the array ⑲ . There are two groups of methods:

1. **Diagonal matrix** – in this case, it is supposed that the bandwidth matrix is diagonal. Because of computational aspects, the used methods are developed for product polynomial kernels. For other types of kernel the buttons are not active. There are five buttons:

   - *Pseudo-likelihood CV* – represents the pseudo-likelihood cross-validation method described in [2]. This bivariate method is a straightforward extension of univariate pseudo-likelihood CV (see [1]).
   - *Least squares CV* – the method is based on the paper [4], where some research on LSCV selector for diagonal matrix was carried out.
   - *Iterative method 1* – it represents the proposed $M1$ method described in §7.4.1 in [3].



Type of view:

○ Density

● Data

☑ Original

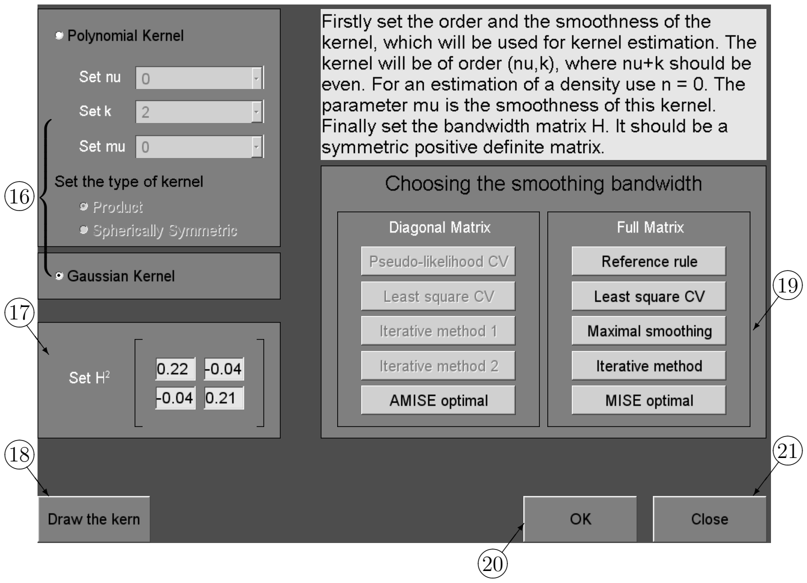
Figure 45: Basic menu.

- *Iterative method 2* – this means the second proposed iterative method $M2$, which is explained in §7.4.2 in [3].
- *AMISE optimal* – this button is active only in the case of known original density, it finds the AMISE optimal diagonal bandwidth matrix by the formula (7.17) in [3].

2. **Full matrix** – in this case, the full bandwidth matrix and Gaussian kernel $\phi_{\mathbf{I}}$ are supposed. There are also five buttons:

   - *Reference rule* – represents the implicit setting based on the multiplication of the covariance matrix estimate, see the formula (7.22) in [3].
   - *Least squares CV* – it minimizes the CV error function (7.9) in [3] for $r = 0$.
   - *Maximal smoothing* – represents the maximal smoothing principle described in §7.3.4 in [3].
   - *Iterative method* – it represents the proposed iterative method for full matrix described in §7.3.5 in [3].
   - *MISE optimal* – this button is active only in the case of known mixture of normal distributions as an original density. It estimates the MISE optimal bandwidth matrix by minimizing (7.5) in [3].

To confirm the setting use ⑳ , to close the window, use ㉑ .



Firstly set the order and the smoothness of the kernel, which will be used for kernel estimation. The kernel will be of order (nu,k), where nu+k should be even. For an estimation of a density use n = 0. The parameter mu is the smoothness of this kernel. Finally set the bandwidth matrix H. It should be a symmetric positive definite matrix.

○ Polynomial Kernel

Set nu 0

Set k 2

Set mu 0

Set the type of kernel

○ Product

○ Spherically Symmetric

● Gaussian Kernel

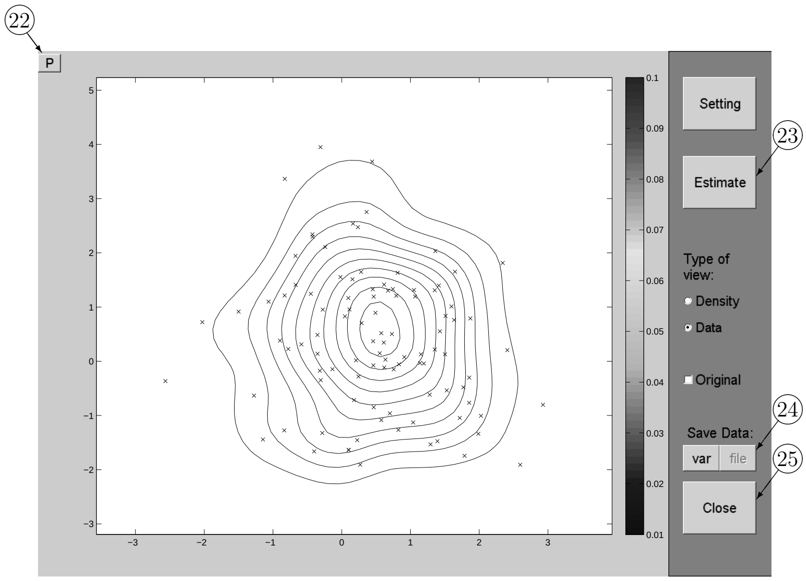
Set H^2 $\begin{bmatrix} 0.22 & -0.04 \\ -0.04 & 0.21 \end{bmatrix}$

Figure 46: Setting of parameters.

7.4 The final estimation

If you set all the values in the window for setting parameters (see Figure 46) and if you want to go to the final estimation of the density, confirm your setting by button ⑳ . It calls up the *Basic menu*, where all buttons are active already.

By clicking on button ㉓ the relevant kernel density estimate is drawn. You can again add contours for the known original density by ⑮ and switch between types of view in ⑭ . By clicking on ㉒ you can print the current plot to a new figure. You can also save data to variables and then as a file by using buttons ㉔ . Button ㉕ ends all applications.



Type of view:
○ Density
● Data

□ Original

Figure 47: Final kernel density estimate.

References

[1] Cao, R., Cuevas, A. and González Manteiga, W. (1994). A comparative study of several smoothing methods in density estimation, *Computational Statistics and Data Analysis* **17**, 2, pp. 153–176.

[2] Horová, I., Koláček, J. and Vopatová, K. (2009). Bandwidth matrix choice for bivariate kernel density estimates, in *Book of short papers* (MU Brno), pp. 22–25.

[3] Horová, I., Koláček, J. and Zelinka, J. (2012). *Kernel smoothing in MATLAB*, World Scientific.

[4] Sain, S., Baggerly, K. and Scott, D. (1994). Cross-validation of multivariate densities, *Journal of the American Statistical Association* **89**, 427, pp. 807–817.