# Package: sshist (via r-universe)

October 18, 2026

**Type** Package

**Title** Optimal Density Estimation via Shimazaki-Shinomoto Method

**Version** 0.2.6

**Description** Implements the Shimazaki-Shinomoto method for optimizing the bin width of histograms and the bandwidth of kernel density estimators. The framework minimizes the expected Mean Integrated Squared Error (MISE) and supports both 1D and 2D distributions, fixed and locally adaptive estimators, bootstrap confidence intervals, and 'OpenMP'-accelerated 'C++' 'backends'. Ideally suited for time-dependent rate estimation and identifying intrinsic data structures. For more details see Shimazaki and Shinomoto (2007) <doi:10.1162/neco.2007.19.6.1503> and Shimazaki and Shinomoto (2010) <doi:10.1007/s10827-009-0180-4>.

**License** GPL (>= 3)

**URL** https://github.com/celebithil/sshist, https://www.neuralengine.org/res/histogram.html

**BugReports** https://github.com/celebithil/sshist/issues

**Imports** boot, graphics, grDevices, Rcpp, stats

**Suggests** ggplot2, knitr, patchwork, rmarkdown, testthat

**LinkingTo** Rcpp

**VignetteBuilder** knitr, rmarkdown

**Config/roxygen2/version** 8.0.0

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**NeedsCompilation** yes

**Roxygen** list(markdown = TRUE)

**Repository** https://celebithil.r-universe.dev

**Date/Publication** 2026-10-18 14:53:27 UTC

**RemoteUrl** https://github.com/celebithil/sshist

**RemoteRef** HEAD

**RemoteSha** c58a56a5958d523ee64735ac023b0b8095654380

# Contents

plot.sshist . . . 2
plot.sshist_2d . . . 3
plot.sskernel . . . 3
plot.sskernel2d . . . 4
plot.ssvkernel . . . 4
plot.ssvkernel2d . . . 5
print.sshist . . . 5
print.sshist_2d . . . 6
print.sskernel . . . 6
print.sskernel2d . . . 7
print.ssvkernel . . . 7
print.ssvkernel2d . . . 8
sshist . . . 8
sshist_2d . . . 9
sskernel . . . 10
sskernel2d . . . 11
ssvkernel . . . 12
ssvkernel2d . . . 13

**Index** **15**

---

`plot.sshist` *Plot method for sshist objects*

---

## Description

Produces single-panel histogram with a jittered-data strip drawn BELOW the x-axis (in a reserved negative-y band), so that the jittered points do not overlap with the histogram bars.

## Usage

```
## S3 method for class 'sshist'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class `"sshist"`. |
| `...` | Additional arguments passed to `hist`. |

### Value

No return value; called for its side effect of producing a plot.

---

## plot.sshist_2d *Plot method for sshist_2d objects*

---

### Description

Draws the optimal 2D histogram as a heatmap with proper data-coordinate axes.

### Usage

```
## S3 method for class 'sshist_2d'
plot(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "sshist_2d". |
| ... | Additional arguments passed to image. |

### Value

No return value; called for its side effect of producing a plot.

---

## plot.sskernel *Plot method for sskernel objects*

---

### Description

Draws the optimal fixed-bandwidth kernel density curve. When bootstrap confidence intervals are stored in the object (nbs > 0), a shaded 90% band is added automatically. A rug and jittered points are drawn in a reserved strip BELOW y = 0 (so they never overlap the density curve).

### Usage

```
## S3 method for class 'sskernel'
plot(x, col = "#2166ac", band_col = adjustcolor(col, alpha.f = 0.2), ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "sskernel". |
| col | Color of the density curve (default "#2166ac"). |
| band_col | Fill color of the confidence band. |
| ... | Additional arguments passed to plot.default. |

## Value

No return value; called for its side effect of producing a plot.

---

`plot.sskernel2d` *Plot method for sskernel2d objects*

---

## Description

Draws the 2D kernel density estimate as a heatmap with optional data overlay.

## Usage

```
## S3 method for class 'sskernel2d'
plot(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class `"sskernel2d"`. |
| `...` | Additional arguments passed to `image`. |

## Value

No return value; called for its side effect of producing a plot.

---

`plot.ssvkernel` *Plot method for ssvkernel objects*

---

## Description

Draws the locally adaptive kernel density curve. When bootstrap confidence intervals are stored (`nbs > 0`), a shaded 90% band is added. A rug and jittered points are drawn in a reserved strip BELOW y = 0 (so they never overlap the density curve).

## Usage

```
## S3 method for class 'ssvkernel'
plot(
  x,
  col = "#d6604d",
  bw_col = "#4d9221",
  band_col = adjustcolor(col, alpha.f = 0.2),
  ...
)
```

### Arguments

| | |
|---|---|
| x | An object of class "ssvkernel". |
| col | Color of the density curve (default "#d6604d"). |
| bw_col | Color of the local bandwidth line (default "#4d9221"). |
| band_col | Fill color of the confidence band. |
| ... | Additional arguments passed to plot.default. |

### Value

No return value; called for its side effect of producing a plot.

---

## plot.ssvkernel2d *Plot method for ssvkernel2d objects*

---

### Description

Draws the locally adaptive 2D kernel density as a heatmap with optional data point overlay.

### Usage

```
## S3 method for class 'ssvkernel2d'
plot(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "ssvkernel2d". |
| ... | Additional arguments passed to image. |

### Value

No return value; called for its side effect of producing a plot.

---

## print.sshist *Print method for sshist objects*

---

### Description

Print method for sshist objects

### Usage

```
## S3 method for class 'sshist'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "sshist". |
| ... | Additional arguments passed to print. |

### Value

Returns x invisibly.

---

## print.sshist_2d *Print method for sshist_2d objects*

---

### Description

Print method for sshist_2d objects

### Usage

```
## S3 method for class 'sshist_2d'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class sshist. |
| ... | Additional arguments passed to print. |

### Value

Returns the input object x invisibly. The method is called for its side effect of printing a summary of the 2D Shimazaki-Shinomoto histogram optimization results, including the optimal number of bins for both X and Y dimensions, bin widths, and minimum cost value.

---

## print.sskernel *Print method for sskernel objects*

---

### Description

Print method for sskernel objects

### Usage

```
## S3 method for class 'sskernel'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "sskernel". |
| ... | Additional arguments passed to print. |

### Value

Returns x invisibly.

---

## print.sskernel2d *Print method for sskernel2d objects*

---

### Description

Print method for sskernel2d objects

### Usage

```
## S3 method for class 'sskernel2d'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "sskernel2d". |
| ... | Additional arguments passed to print. |

### Value

Returns x invisibly.

---

## print.ssvkernel *Print method for ssvkernel objects*

---

### Description

Print method for ssvkernel objects

### Usage

```
## S3 method for class 'ssvkernel'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "ssvkernel". |
| ... | Additional arguments passed to print. |

### Value

Returns x invisibly.

---

`print.ssvkernel2d` *Print method for ssvkernel2d objects*

---

### Description

Print method for ssvkernel2d objects

### Usage

```
## S3 method for class 'ssvkernel2d'
print(x, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class "ssvkernel2d". |
| ... | Additional arguments passed to print. |

### Value

Returns x invisibly.

---

`sshist` *Optimal Histogram Binning (Shimazaki-Shinomoto Method)*

---

### Description

Computes the optimal bin width and number of bins using the Shimazaki-Shinomoto (2007) method. This implementation performs an exhaustive search over candidate bin counts, exactly matching the original Python/MATLAB reference algorithms.

### Usage

```
sshist(x, n_max = NULL, sn = 30, ncores = getOption("sshist.ncores", 1L))
```

### Arguments

| | |
|---|---|
| x | A numeric vector of data. NA values are silently removed. |
| n_max | Integer or NULL. Maximum number of bins to consider. When NULL (default) the limit is set automatically as min(500, floor(range / (2 * min_resolution)), n_samples). |
| sn | Integer. Number of histogram shifts used in the shift-average (default 30). |
| ncores | Integer. Number of OpenMP threads to use. Defaults to 1 for CRAN compliance. |

## Value

An object of class `"sshist"` (a named list) containing:

`opt_n` Optimal number of bins (integer).

`opt_d` Optimal bin width ($= \text{range}/N_{\text{opt}}$).

`edges` Numeric vector of $N_{\text{opt}} + 1$ break points for the optimal histogram.

`data` Cleaned (NA-removed) input data.

## References

Shimazaki, H. and Shinomoto, S. (2007). A method for selecting the bin size of a time histogram. *Neural Computation*, **19**(6), 1503–1527.

---

`sshist_2d` *Optimal 2D Histogram Binning (Shimazaki-Shinomoto Method)*

---

## Description

Computes the optimal number of bins for a 2-dimensional histogram using the Shimazaki-Shinomoto (2007) cost function.

## Usage

```
sshist_2d(x, y = NULL, n_min = 2L, n_max = 200L)
```

## Arguments

| | |
|---|---|
| `x` | Numeric vector (X coordinates) or a 2-column matrix/data.frame. |
| `y` | Numeric vector (Y coordinates). Ignored if `x` is a matrix. |
| `n_min` | Integer. Minimum number of bins per axis (default `2`). |
| `n_max` | Integer or `NULL`. Maximum number of bins per axis (default `200`). Automatically clamped to the data resolution limit and the sample size. |

## Value

An object of class `"sshist_2d"` containing:

`opt_nx`, `opt_ny` Optimal bin counts.

`opt_dx`, `opt_dy` Optimal bin widths.

`data` Cleaned input data.

## References

Shimazaki, H. and Shinomoto, S. (2007). A method for selecting the bin size of a time histogram. *Neural Computation*, **19**(6), 1503-1527. doi:10.1162/neco.2007.19.6.1503

## Examples

```
set.seed(42)
x <- rnorm(500); y <- rnorm(500)

res <- sshist_2d(x, y)
plot(res)

# ggplot2
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(data.frame(x = x, y = y), aes(x, y)) +
    geom_bin2d(bins = c(res$opt_nx, res$opt_ny)) +
    scale_fill_viridis_c() +
    ggtitle(sprintf("Optimal 2D Bins: %dx%d", res$opt_nx, res$opt_ny)) +
    theme_minimal()
}
```

---

# sskernel *Optimal 1D Kernel Density Estimation (Fixed Bandwidth)*

---

## Description

Computes the optimal global bandwidth for 1D kernel density estimation using the Shimazaki-Shinomoto method. Minimizes the expected Mean Integrated Squared Error (MISE).

## Usage

```
sskernel(
  x,
  tin = NULL,
  W = NULL,
  nbs = 0,
  ncores = getOption("sshist.ncores", 1L)
)
```

## Arguments

| | |
|---|---|
| x | Numeric vector of sample data. Missing values (NA) will be removed. |
| tin | Optional numeric vector specifying the grid of evaluation points. |
| W | Optional numeric vector of bandwidths to search. If provided, the grid search is skipped. |
| nbs | Integer specifying the number of bootstrap samples for calculating confidence intervals. |
| ncores | Integer specifying the number of CPU cores to use for bootstrap (default: 1). |

## Value

An object of class "sskernel" containing:

**x** The evaluation points (same as tin).

**y** The optimized kernel density estimate.

**optw** The optimal global bandwidth.

**data** The original evaluated data.

**confb95** (If nbs > 0) A matrix with 5th and 95th percentile bootstrap confidence intervals.

**yb** (If nbs > 0) A matrix of all bootstrap density samples.

---

sskernel2d *Optimal 2D Kernel Density Estimation (Fixed Bandwidth)*

---

## Description

Computes the optimal global bandwidth for a 2D kernel density estimate based on the exact L2 risk (Mean Integrated Squared Error) minimization. Uses a two-stage search: a logarithmic grid (10 points, auto-expanding) followed by golden section refinement on the logexp-transformed scale.

## Usage

```
sskernel2d(
  x,
  y = NULL,
  W = NULL,
  n_grid = 100,
  ncores = getOption("sshist.ncores", 1L)
)
```

## Arguments

| | |
|---|---|
| x | Numeric vector for X coordinates, or a 2-column matrix containing X and Y. |
| y | Numeric vector for Y coordinates (required if x is a vector). |
| W | Optional numeric vector of bandwidths to evaluate. If NULL, a default log-spaced grid (10 pts) with golden section refinement is used. |
| n_grid | Integer specifying the number of grid points for the output density matrix (default: 100). |
| ncores | Integer. Number of OpenMP threads to use. Defaults to 1 for CRAN compliance. |

## Value

An object of class "sskernel2d" containing:

**x_grid** Numeric vector of the X-axis grid points.

**y_grid** Numeric vector of the Y-axis grid points.

**z** Numeric matrix of the estimated 2D density.

**opt_wx** The optimal global bandwidth for the X dimension.

**opt_wy** The optimal global bandwidth for the Y dimension.

**data** A list containing the original X and Y data.

---

`ssvkernel` *Locally Adaptive 1D Kernel Density Estimation (Shimazaki-Shinomoto)*

---

## Description

Computes locally adaptive bandwidths for 1D distributions following the method of Shimazaki & Shinomoto (2010). Optimizes the stiffness constant gamma via a multi-start (K=5) golden section search over GS-based intervals, accelerated with C++/OpenMP parallelization.

## Usage

```
ssvkernel(
  x,
  tin = NULL,
  M = 80,
  WinFunc = "Boxcar",
  nbs = 0,
  ncores = getOption("sshist.ncores", 1L)
)
```

## Arguments

| | |
|---|---|
| `x` | Numeric vector of sample points. Missing values (NA) will be removed. |
| `tin` | Optional numeric vector of evaluation points. |
| `M` | Integer, number of bandwidths to examine (default: 80). |
| `WinFunc` | Character string specifying the window function for local weights: "Gauss", "Boxcar", "Laplace", or "Cauchy" (default: "Boxcar"). |
| `nbs` | Integer, number of bootstrap samples for confidence intervals. Set to 0 to skip (default: 0). |
| `ncores` | Integer specifying the number of CPU cores to use for C++/OpenMP gamma optimization and bootstrap (default: 1). |

## Value

An object of class "ssvkernel" containing:

**x** The evaluation points (same as tin).

**y** The optimized adaptive density estimate.

**optw_local_min** Numeric vector of locally adaptive bandwidths evaluated at x.

**optw_local_max** Numeric vector of locally adaptive bandwidths evaluated at x.

**gamma** The optimal stiffness constant.

**data** The original evaluated data.

**confb95** (If nbs > 0) A matrix with 5th and 95th percentile bootstrap confidence intervals.

**yb** (If nbs > 0) A matrix of all bootstrap density samples.

---

# ssvkernel2d *Locally Adaptive 2D Kernel Density Estimation (Abramson's Method)*

---

## Description

Computes a 2D kernel density estimate using Abramson's square-root adaptive bandwidths. The procedure is initialized by finding the optimal global bandwidth via sskernel2d.

## Usage

```
ssvkernel2d(
  x,
  y = NULL,
  n_grid = 100,
  sensitivity = 0.5,
  ncores = getOption("sshist.ncores", 1L)
)
```

## Arguments

| | |
|---|---|
| x | Numeric vector for X coordinates, or a 2-column matrix containing X and Y. |
| y | Numeric vector for Y coordinates (required if x is a vector). |
| n_grid | Integer specifying the number of grid points for the output density matrix (default: 100). |
| sensitivity | Numeric scalar controlling the sensitivity of local bandwidths to the pilot density. A value of 0.5 (default) corresponds to Abramson's original inverse square-root law. |
| ncores | Integer. Number of OpenMP threads to use. Defaults to 1 for CRAN compliance. |

## Value

An object of class `"ssvkernel2d"` containing:

**x_grid** Numeric vector of the X-axis grid points.

**y_grid** Numeric vector of the Y-axis grid points.

**z** Numeric matrix of the estimated adaptive 2D density.

**pilot_wx** The global optimal pilot bandwidth for the X dimension.

**pilot_wy** The global optimal pilot bandwidth for the Y dimension.

**lambda_factors** Numeric vector of local scaling factors applied to each data point.

**data** A list containing the original X and Y data.

# Index

hist, *2*

image, *3–5*

plot.default, *3*, *5*
plot.sshist, 2
plot.sshist_2d, 3
plot.sskernel, 3
plot.sskernel2d, 4
plot.ssvkernel, 4
plot.ssvkernel2d, 5
print, *6–8*
print.sshist, 5
print.sshist_2d, 6
print.sskernel, 6
print.sskernel2d, 7
print.ssvkernel, 7
print.ssvkernel2d, 8

sshist, 8
sshist_2d, 9
sskernel, 10
sskernel2d, 11, *13*
ssvkernel, 12
ssvkernel2d, 13