Package ‘Kernelheaping’

April 16, 2016

Type Package

Title Kernel Density Estimation for Heaped and Rounded Data

Version 1.6

Date 2016-04-15

Depends R (>= 2.15.0), MASS, ks, sparr

Imports sp, plyr

Author Marcus Gross

Maintainer Marcus Gross <marcus.gross@fu-berlin.de>

Description In self-reported or anonymised data the user often encounters heaped data, i.e. data which are rounded (to a possibly different degree of coarseness). While this is mostly a minor problem in parametric density estimation the bias can be very large for non-parametric methods such as kernel density estimation. This package implements a partly Bayesian algorithm treating the true unknown values as additional parameters and estimates the rounding parameters to give a corrected kernel density estimate. It supports various standard bandwidth selection methods. Varying rounding probabilities (depending on the true value) and asymmetric rounding is estimable as well. Additionally, bivariate non-parametric density estimation for rounded data as well as data aggregated on areas is supported.

License GPL-2 | GPL-3

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-16 00:09:59

R topics documented:

createSim.Kernelheaping ........................................ 2
dbivr .............................................................. 3
dclass ............................................................ 4
dheaping .......................................................... 5
createSim.Kernelheaping

Create heaped data for Simulation

Description

Create heaped data for Simulation

Usage

createSim.Kernelheaping(n, distribution, rounds, thresholds, offset = 0,
downbias = 0.5, Beta = 0, ...)

Arguments

- **n**: sample size
- **distribution**: name of the distribution where random sampling is available, e.g. "norm"
- **rounds**: rounding values
- **thresholds**: rounding thresholds (for Beta=0)
- **offset**: certain value added to all observed random samples
- **downbias**: bias parameter
- **Beta**: acceleration parameter
- **...**: additional attributes handed over to "rdistribution" (i.e. rnorm, rgamma,...)

Value

List of heaped values, true values and input parameters
Bivariate kernel density estimation for rounded data

Usage

dbivr(xrounded, roundvalue, burnin = 2, samples = 5, adaptive = FALSE, gridsize = 1000)

Arguments

xrounded rounded values from which to estimate bivariate density, matrix with 2 columns (x,y)
roundvalue rounding value (side length of square in that the true value lies around the rounded one)
burnin burn-in sample size
samples sampling iteration size
adaptive set to TRUE for adaptive bandwidth
gridsize number of evaluation grid points

Value

The function returns a list object with the following objects (besides all input objects):

Mestimates kde object containing the corrected density estimate
gridx Vector Grid on which density is evaluated (x)
gridy Vector Grid on which density is evaluated (y)
resultDensity Array with Estimated Density for each iteration
resultX Matrix of true latent values X estimates
delaigle Matrix of Delaigle estimator estimates

Examples

# Create Mu and Sigma
mu1 <- c(0, 0)
mu2 <- c(5, 3)
mu3 <- c(-4, 1)
Sigma1 <- matrix(c(4, 3, 3, 4, 2, 2)
Sigma2 <- matrix(c(3, 0.5, 0.5, 1, 2, 2)
Sigma3 <- matrix(c(5, 4, 4, 6, 2, 2)
# Mixed Normal Distribution
mus <- rbind(mu1, mu2, mu3)
dclass <- R
props <- c(1/3, 1/3, 1/3)
## Not run: xtrue=rmvnorm.mixt(n=1000, mus=mus, Sigmas=Sigmas, props=props)
roundvalue=2
xrounded=plyr::round_any(xtrue, roundvalue)
est <- dbivr(xrounded, roundvalue=roundvalue, burnin=5, samples=10)

# Plot corrected and Naive distribution
plot(est, xtrue=xtrue)
# for comparison: plot true density
dens=dmvnorm.mixt(x=expand.grid(est$Mestimates$eval.points[[1]], est$Mestimates$eval.points[[2]]), mus=mus, Sigmas=Sigmas, props=props)
dens=matrix(dens, nrow=length(est$gridx), ncol=length(est$gridy))
contour(dens, x=est$Mestimates$eval.points[[1]], y=est$Mestimates$eval.points[[2]], xlab=c(min(est$gridx), max(est$gridx)), ylab=c(min(est$gridx), max(est$gridx)), main="True Density")

---

dclass

Kernel density estimation for classified data

Description

Kernel density estimation for classified data

Usage

dclass(xclass, classes, burnin = 2, samples = 5, boundary = FALSE, bw = "nrd0", evalpoints = 200, adjust = 1)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xclass</td>
<td>classified values; factor with ordered factor values</td>
</tr>
<tr>
<td>classes</td>
<td>numeric vector of classes; Inf as last value is allowed</td>
</tr>
<tr>
<td>burnin</td>
<td>burn-in sample size</td>
</tr>
<tr>
<td>samples</td>
<td>sampling iteration size</td>
</tr>
<tr>
<td>boundary</td>
<td>TRUE for positive only data (no positive density for negative values)</td>
</tr>
<tr>
<td>bw</td>
<td>bandwidth selector method, defaults to &quot;nrd0&quot; see density for more options</td>
</tr>
<tr>
<td>evalpoints</td>
<td>number of evaluation grid points</td>
</tr>
<tr>
<td>adjust</td>
<td>as in density, the user can multiply the bandwidth by a certain factor such that bw=adjust*bw</td>
</tr>
</tbody>
</table>

Value

The function returns a list object with the following objects (besides all input objects):

- Mestimates: kde object containing the corrected density estimate
- gridx: Vector Grid on which density is evaluated
- resultDensity: Matrix with Estimated Density for each iteration
- resultX: Matrix of true latent values X estimates
**dheaping**

**Kernel density estimation for heaped data**

**Description**

Kernel density estimation for heaped data

**Usage**

```r
dheaping(xheaped, rounds, burnin = 5, samples = 10, setBias = FALSE, weights = NULL, bw = "nrd0", boundary = FALSE, unequal = FALSE, random = FALSE, adjust = 1, recall = F, recallParams = c(1/3, 1/3))
```

**Arguments**

- `xheaped` heaped values from which to estimate density of `x`
- `rounds` rounding values, numeric vector of length >=1
- `burnin` burn-in sample size
- `samples` sampling iteration size
- `setBias` if TRUE a rounding Bias parameter is estimated. For values above 0.5, the respondents are more prone to round down, while for values < 0.5 they are more likely to round up
- `weights` optional numeric vector of sampling weights
- `bw` bandwidth selector method, defaults to "nrd0" see `density` for more options
- `boundary` TRUE for positive only data (no positive density for negative values)
- `unequal` if TRUE a probit model is fitted for the rounding probabilities with log(true value) as regressor
- `random` if TRUE a random effect probit model is fitted for rounding probabilities
- `adjust` as in density, the user can multiply the bandwidth by a certain factor such that bw=adjust*bw
- `recall` if TRUE a recall error is introduced to the heaping model
- `recallParams` recall error model parameters expression(nu) and expression(eta). Default is c(1/3, 1/3)

Examples

```r
x <- rlnorm(500, meanlog = 8, sdlog = 1)
classes <- c(0, 500, 1000, 1500, 2000, 2500, 3000, 4000, 5000, 6000, 8000, 10000, 15000, Inf)
xclass <- cut(x, breaks = classes)
densityEst <- dclass(xclass = xclass, classes = classes, burnin = 2, samples = 5, evalpoints = 1000)
hist(densityEst$xclass, breaks = densityEst$classes)
lines(densityEst$Mestimates ~ densityEst$gridx, col = "purple", lwd = 2)
```
**Value**

The function returns a list object with the following objects (besides all input objects):

- **meanPostDensity**: Vector of Mean Posterior Density
- **gridx**: Vector Grid on which density is evaluated
- **resultDensity**: Matrix with Estimated Density for each iteration
- **resultRR**: Matrix with rounding probability threshold values for each iteration (on probit scale)
- **resultBias**: Vector with estimated Bias parameter for each iteration
- **resultBeta**: Vector with estimated Beta parameter for each iteration
- **resultX**: Matrix of true latent values X estimates

**Examples**

```r
# Simple Rounding
-------------
xtrue <- rnorm(3000)
xrounded <- round(xtrue)
est <- dheaping(xrounded, rounds=1, burnin=20, samples=50)
plot(est, trueX=xtrue)

# Heaping
###
# Real Data Example

# Student learning hours per week
data(students)
xheaped <- as.numeric(omit(students$StudyHrs))
## Not run: est <- dheaping(xheaped, rounds=c(1,2,5,10), boundary=TRUE, unequal=TRUE, burnin=20, samples=50)
plot(est)
supply(est)
## End(Not run)

# Simulate Data
-------------
Sim1 <- createSim_KERNELheaping(n=500, distribution="norm", rounds=c(1,10,100),
thresholds=c(-0.5244005, 0.5244005), sd=100)
## Not run: est <- dheaping(Sim1$xheaped, rounds=Sim1$rounds)
plot(est, trueX=Sim1$x)
## End(Not run)

# Biased Rounding
Sim2 <- createSim_KERNELheaping(n=500, distribution="gamma", rounds=c(1,2,5,10),
thresholds=c(-1.2815516, -0.6744898, 0.3853205), downbias=0.2,
shape=4, scale=8, offset=45)
## Not run: est <- dheaping(Sim2$xheaped, rounds=Sim2$rounds, setBias=T, bw="SJ")
plot(est, trueX=Sim2$x)
supply(est)
tracePlots(est)
## End(Not run)
```
dshapebivr

Bivariate Kernel density estimation for data classified in polygons or shapes

Description

Bivariate Kernel density estimation for data classified in polygons or shapes

Usage

dshapebivr(data, burnin = 2, samples = 5, adaptive = FALSE, shapefile, gridsize = 200)

Arguments

data matrix with at least 3 columns: x-coordinate, y-coordinate (i.e. center of polygon) and number of observations in area Optional fourth column: ID-Variable if area consists of more than 1 polygon

burnin burn-in sample size

samples sampling iteration size

adaptive TRUE for adaptive kernel density estimation

shapefile shapefile with number of polygons equal to nrow(data)

gridsize number of evaluation grid points

Value

The function returns a list object with the following objects (besides all input objects):

Mestimates kde object containing the corrected density estimate

gridx Vector Grid of x-coordinates on which density is evaluated

gridy Vector Grid of y-coordinates on which density is evaluated

resultDensity Matrix with Estimated Density for each iteration

resultX Matrix of true latent values X estimates
Kernelheaping

*Kernel Density Estimation for Heaped Data*

**Description**

In self-reported or anonymized data the user often encounters heaped data, i.e. data which are rounded (to a possibly different degree of coarseness). While this is mostly a minor problem in parametric density estimation the bias can be very large for non-parametric methods such as kernel density estimation. This package implements a partly Bayesian algorithm treating the true unknown values as additional parameters and estimates the rounding parameters to give a corrected kernel density estimate. It supports various standard bandwidth selection methods. Varying rounding probabilities (depending on the true value) and asymmetric rounding is estimable as well. Additionally, bivariate non-parametric density estimation for rounded data is supported.

**Details**

The most important function is `dheaping`. See the help and the attached examples on how to use the package.

---

**plot.bivrounding**

*Plot Kernel density estimate of heaped data naively and corrected by partly bayesian model*

**Description**

Plot Kernel density estimate of heaped data naively and corrected by partly bayesian model

**Usage**

```r
## S3 method for class 'bivrounding'
plot(x, trueX = NULL, ...)
```

**Arguments**

- `x` bivrounding object produced by `dbivr` function
- `trueX` optional, if true values X are known (in simulations, for example) the 'Oracle' density estimate is added as well
- `...` additional arguments given to standard plot function

**Value**

plot with Kernel density estimates (Naive, Corrected and True (if provided))
plot.Kernelheaping

Plot Kernel density estimate of heaped data naively and corrected by partly bayesian model

Description

Plot Kernel density estimate of heaped data naively and corrected by partly bayesian model

Usage

## S3 method for class 'Kernelheaping'
plot(x, trueX = NULL, ...)

Arguments

x Kernelheaping object produced by dheaping function
trueX optional, if true values X are known (in simulations, for example) the 'Oracle'
density estimate is added as well
...
additional arguments given to standard plot function

Value

plot with Kernel density estimates (Naive, Corrected and True (if provided))

sim.Kernelheaping

Simulation of heaping correction method

Description

Simulation of heaping correction method

Usage

sim.Kernelheaping(simRuns, n, distribution, rounds, thresholds,
downbias = 0.5, setBias = FALSE, Beta = 0, unequal = FALSE,
burnin = 5, samples = 10, bw = "nrd0", offset = 0, boundary = FALSE,
adjust = 1, ...)
Arguments

- **simRuns**: number of simulations runs
- **n**: sample size
- **distribution**: name of the distribution where random sampling is available, e.g. "norm"
- **rounds**: rounding values, numeric vector of length >=1
- **thresholds**: rounding thresholds
- **downbias**: Bias parameter used in the simulation
- **setBias**: if TRUE a rounding Bias parameter is estimated. For values above 0.5, the respondents are more prone to round down, while for values < 0.5 they are more likely to round up
- **Beta**: Parameter of the probit model for rounding probabilities used in simulation
- **unequal**: if TRUE a probit model is fitted for the rounding probabilities with log(true value) as regressor
- **burnin**: burn-in sample size
- **samples**: sampling iteration size
- **bw**: bandwidth selector method, defaults to "nrd0" see density for more options
- **offset**: location shift parameter used simulation in simulation
- **boundary**: TRUE for positive only data (no positive density for negative values)
- **adjust**: as in density, the user can multiply the bandwidth by a certain factor such that bw=adjust*bw
- ... additional attributes handed over to createSim.Kernelheaping

Value

List of estimation results

Examples

```r
## Not run: simS1 <- sim.Kernelheaping(simRuns=2, n=500, distribution="norm",
rounds=c(1,10,100), thresholds=c(0.3,0.4,0.3), sd=100)
## End(Not run)
```

---

**Simulation Summary**

Description

Simulation Summary

Usage

```r
simSummary.Kernelheaping(sim, coverage = 0.9)
```
students

Arguments

sim          Simulation object returned from sim.Kernelheaping
coverage     probability for computing coverage intervals

Value

list with summary statistics

students      Student0405

Description

Data collected during 2004 and 2005 from students in statistics classes at a large state university in the northeastern United States.

Author(s)

Jessica M. Utts, Robert F. Heckard

Source

http://mathfaculty.fullerton.edu/mori/Math120/Data/readme

summary.Kernelheaping  Prints some descriptive statistics (means and quantiles) for the estimated rounding, bias and acceleration (beta) parameters

Description

Prints some descriptive statistics (means and quantiles) for the estimated rounding, bias and acceleration (beta) parameters

Usage

## S3 method for class 'Kernelheaping'
summary(object, ...)

Arguments

object  Kernelheaping object produced by dheaping function
...

Value

Prints summary statistics
tracePlots

Plots some trace plots for the rounding, bias and acceleration (beta) parameters

Description

Plots some trace plots for the rounding, bias and acceleration (beta) parameters

Usage

tracePlots(x, ...)

Arguments

x Kernelheaping object produced by dheaping function
...

additional arguments given to standard plot function

Value

Prints summary statistics
Index

createSim.Kernelheaping, 2

dbivr, 3
dclass, 4
dheaping, 5, 8
dshapebivr, 7

Kernelheaping, 8
Kernelheaping-package (Kernelheaping), 8

plot.bivrounding, 8
plot.Kernelheaping, 9

sim.Kernelheaping, 9
simSummary.Kernelheaping, 10
students, 11
summary.Kernelheaping, 11

tracePlots, 12