Package ‘DENSaftertransform’

June 2, 2026

Title Estimating Density after Logarithmic or Power Transformation of Data

Version 0.1

Description Functions for computing: (1) the adaptive normal PI estimate for data after the logarithmic transformation; (2) single-bandwidth PI density estimate for data after the logarithmic transformation; (3) single bandwidth PI estimate for data after the power transformation. See the articles: (1) Savchuk, O. (2026, under review). Density estimation for log-transformed data; (2) Savchuk, O., Schick A. (2013). Density estimation for power transformations. Journal of Nonparametric Statistics, 25(3), 545-559 <doi:10.1080/10485252.2013.811788>.

License GPL-2

Encoding UTF-8

RoxygenNote 8.0.0

NeedsCompilation no

Author Olga Savchuk [aut, cre]

Maintainer Olga Savchuk <olga.y.savchuk@gmail.com>

Repository CRAN

Date/Publication 2026-06-02 09:40:07 UTC

Contents

`h_ln_PI` *Bandwidth for the PI estimator in the case of the logarithmic transformation.*

Description

Bandwidth for the PI estimator for estimating g, the true probability density function (pdf), corresponding to the logarithmic transformation of data based on replacing g by the reference $N(\bar{y}, s_Y^2)$ distribution. See Section 7 of the article by Savchuk (2026).

Usage

```
h_ln_PI(x)
```

Arguments

`x`	vector of the original X-data.

Value

The bandwidth for the PI estimator for estimating g, the density after the logarithmic transformation, obtained by replacing g by the reference $N(\bar{y}, s_Y^2)$ density.

References

Savchuk, O. (2026, under review). Density estimation for log-transformed data.

See Also

KDE_ln_PI.

Examples

```
# Example. Finding bandwidth for the PI estimator for the log-transformed sample of size n=100
#originated from a gamma distribution
#with the parameters: shape=2, scale=2.
xdat=rgamma(100,2,scale=2)
paste("PI bandwidth for the PI estimator=", h_ln_PI(xdat))
```

KDE_ln_PI — *The PI kernel density estimate in the case of the logaritjmic transformation.*

Description

Computing the PI kernel density estimate for the log-transformed data based on the Gaussian kernel.

Usage

```
KDE_ln_PI(h, x, u)
```

Arguments

h	the bandwidth, scalar,
x	vector of the original X-data,
u	vector of the estimation points.

Value

Vector of density estimates at the corresponding values of the argument (u).

References

Savchuk, O. (2026, under review). Density estimation for log-transformed data.

See Also

h_ln_PI.

Examples

```
#Example(Gamma(shape=2,scale=2) density). Estimating the pdf of the density corresponding to the
#logarithmic transformation of sample of size n=100 originated from the gamma density.
xdat=rgamma(100,2,scale=2)  #original data
n=length(xdat)  #sample size
u=seq(-3,5,len=1000) #estimation points
dens_lntransform=exp(2*u-exp(u)/2)/4  #density after log-transformation
h_PI=h_ln_PI(xdat)
dev.new()
dens_PI=KDE_ln_PI(h_PI,xdat,u)
plot(u,dens_PI,'l',ylim=c(0,1.05*max(dens_PI)),lwd=2,cex.lab=1.7,cex.axis=1.7,cex.main=1.5,
main="Gamma dens. after ln-transform. and its PI estimate",xlab="u",ylab="density")
lines(u,dens_lntransform,lwd=2,lty="dashed",col="red",bty="n")
legend(-3,0.5,legend=c("true density","PI estimate"),lwd=c(2,2),col=c("red","black"),
lty=c("dashed","solid"),cex=1.5,bty="n")
text(3.25,0.525,paste("n=",n),cex=2)
text(3.85,0.425,bquote(h[PI] == .(sprintf("%.4f", h_PI))),cex=2)
```

KDE_PI_adapt_norm — *Adaptive normal PI estimate in the case of the logarithmic transformation.*

Description

Computing the adaptive normal PI kernel density estimate for the log-transformed data based on the Gaussian kernel.

Usage

```
KDE_PI_adapt_norm(x, u)
```

Arguments

x	vector of the original X-data,
u	vector of the estimation points.

Value

Vector of density estimates at the corresponding values of the argument (u).

References

Savchuk, O. (2026, under review). Density estimation for log-transformed data.

See Also

KDE_ln_PI.

Examples

```
#Example(skewed bimodal density). Estimating the density corresponding to the logarithmic
#transformation of a sample of size n=1000 originated from the skewed bimodal density
#(density #8 of Marron and Wand (1992)).
#Density after the logarithmi transformation
g_skew=function(u)
  3/4*dnorm(u)+1/4*dnorm(u,mean=3/2,sd=1/3)
skew_realiz=function(n){
  # n is the sample size
  mix_comp=runif(n)<0.25
  dat=numeric(n)
  dat[mix_comp]=rnorm(sum(mix_comp),mean=3/2,sd=1/3)
  dat[!mix_comp]=rnorm(sum(!mix_comp))
 dat
}
#Plotting an estimate
n=1000
u=seq(-4,4,len=1000)
```

```
y_skew=skew_realiz(n)
x_skew=exp(y_skew)
h_PI=h_ln_PI(x_skew)
KDE_skew_PI=KDE_ln_PI(h_PI,x_skew,u)
KDE_skew_adapt_norm=KDE_PI_adapt_norm(x_skew,u)
dev.new()
plot(u,g_skew(u),'l',lwd=2,ylim=c(0,1.05*max(KDE_skew_PI)),col="red",xlab="y",ylab="density",
cex.lab=1.7,cex.axis=1.7,cex.main=1.7,main="Skewed bimodal distribution")
lines(u,KDE_skew_PI,lty="longdash",lwd=2)
lines(u,KDE_skew_adapt_norm,lty="solid",lwd=2,col="blue")
legend(-4.5,0.45,legend=c("PI estimate","Adaptive normal","density g"),lty=c("longdash",
"solid","solid"),col=c("black","blue","red"),lwd=c(3,3,3),bty="n",cex=1.7)
legend(1.25,0.45,legend=paste("n=",n),cex=2,bty="n")
```

`KDE_power_PI` *The PI kernel density estimate in the case of the power transformation.*

Description

Computing the PI kernel density estimate for data after a power transformation based on the Gaussian kernel.

Usage

```
KDE_power_PI(alpha, h, x, u)
```

Arguments

`alpha`	the exponent of the power transformation (alpha>0),
`h`	the bandwidth, scalar,
`x`	vector of the original X-data,
`u`	vector of the estimation points.

Value

Vector of density estimates at the corresponding values of the argument (u).

References

Savchuk, O., Schick A. (2013). Density estimation for power transformations. *Journal of Nonparametric Statistics*, 25(3), 545-559.

See Also

`KDE_ln_PI`.

Examples

```
#Example(original data is sandard normal, n=1000 and (1)alpha=0.5 (Case 1), (2) alpha=2 (Case 2))
n=1000
xdat=rnorm(n)
h_P=bw.SJ(xdat) #Sheather-Jones PI for original x-data

#Case1: alpha=0.5
alpha=0.5
u=seq(-2.5,2.5,len=1000)
g_0.5=1/alpha/sqrt(2*pi)*abs(u)^(1/alpha-1)*exp(-abs(u)^(2/alpha)/2) #true density
PI_power_est_0.5=KDE_power_PI(alpha,h_P,xdat,u)
dev.new()
plot(u,g_0.5,lwd=2,'l',col="red",ylim=c(0,1.125*max(PI_power_est_0.5)),xlab="y",ylab="density",
cex.axis=1.7,cex.main=1.7,cex.lab=1.7,main=expression(paste("Norm density, ",alpha,"=0.5")))
lines(u,PI_power_est_0.5,lwd=2,lty="dashed")
legend(-2.75,1.2*max(max(PI_power_est_0.5),0.52),legend=c("true density","PI estimate"),
lty=c("solid","dashed"),col=c("red","black"),lwd=c(2,2),bty="n",cex=1.5)
legend(0.75,0.5,legend=paste("n=",n),cex=2,bty="n")

#Case 2: alpha=2
alpha=2
y=sign(xdat)*abs(xdat)^alpha
u=seq(-1.5,1.5,len=1000)
g_2=1/alpha/sqrt(2*pi)*abs(u)^(1/alpha-1)*exp(-abs(u)^(2/alpha)/2) #true density
PI_power_est_2=KDE_power_PI(alpha,h_P,xdat,u)
dev.new()
plot(u,g_2,lwd=2,'l',col="red",ylim=c(0,1.1*max(PI_power_est_2)),xlab="y",ylab="density",
cex.axis=1.7,cex.main=1.7,cex.lab=1.7,main=expression(paste("Norm density, ",alpha,"=2")))
lines(u,PI_power_est_2,lwd=2,lty="dashed")
legend(0,2,legend=c("true density","PI estimate"),lty=c("solid","dashed"),col=c("red","black"),
lwd=c(2,2),cex=1.5,bty="n")
legend(-1.5,5,legend=paste("n=",n),cex=2,bty="n")
```

Index