

Package ‘DENSaftertransform’

October 19, 2026

Title Estimating Density after Logarithmic or Power Transformation of Data

Version 1.0

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

Description Functions for computing: (1) the adaptive normal plug-in(PI) density estimate for data after the logarithmic transformation; (2) the adaptive log-normal PI density estimate for data after the power transformation with an exponent ranging from zero to one; (3) the single-bandwidth PI density estimate for data after the logarithmic transformation; (4) the single-bandwidth PI density estimate for data after the power transformation; (5) the data-driven bandwidth for the single-bandwidth PI estimator for data after the logarithmic transformation; (6) the rule-of-thumb bandwidth for the single-bandwidth PI estimator for data after the power transformation. For details, see the articles: (1) Savchuk, O. (2026) (submitted). Adaptive log-normal plug-in density estimator for power transformations. (2) Savchuk, O. (2026). Density estimation for log-transformed data. Computational Statistics, 41(98) <[doi:10.1007/s00180-026-01766-y](https://doi.org/10.1007/s00180-026-01766-y)>. (3) Savchuk, O., Schick A. (2013). Density estimation for power transformations. Journal of Nonparametric Statistics, 25(3), 545-559 <[doi:10.1080/10485252.2013.811788](https://doi.org/10.1080/10485252.2013.811788)>.

License GPL-2

Encoding UTF-8

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Olga Savchuk [aut, cre, cph]

Repository CRAN

Date/Publication 2026-10-19 02:30:15 UTC

Contents

h_ln_PI	2
h_pi_datadriven_power	3
KDE_ln_PI	4
KDE_PI_adapt_lognorm_power	5
KDE_PI_adapt_norm	6
KDE_power_PI	7

Index**9**

h_ln_PI	<i>The bandwidth for the single-bandwidth PI density estimator in the case of the logarithmic transformation.</i>
---------	---

Description

Bandwidth for the PI estimator for estimating g , the true pdf of the transformed sample.

Usage

```
h_ln_PI(x)
```

Arguments

`x` vector of the original positive x-data.

Value

The bandwidth for the PI estimator for estimating g , the density after the logarithmic transformation.

References

Savchuk, O. (2026). Density estimation for log-transformed data. *Computational Statistics*, 41(98).

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))
```

h_pi_datadriven_power	<i>The bandwidth for the single-bandwidth PI density estimator for data after the power transformation with an exponent ranging from zero to one based on the Gaussian kernel.</i>
-----------------------	--

Description

Bandwidth for the PI estimator for estimating g , the true pdf of the transformed sample.

Usage

```
h_pi_datadriven_power(x, alpha)
```

Arguments

x	vector of the original nonnegative-valued x-data.
alpha	exponent of the power transformation (ranges between 0 and 1).

Value

The bandwidth for the PI density estimator for estimating g , the density after the power transformation.

References

Savchuk, O. (2026, submitted). Adaptive log-normal plug-in density estimator for power transformations.

See Also

[KDE_power_PI](#), [KDE_PI_adapt_lognorm_power](#).

Examples

```
# Example. Finding bandwidth for the single-bandwidth PI density estimator for the transformed
# data on river lengths of 141 major North American rivers with alpha=1/5.
xdat=rivers
alpha=1/5
n=length(xdat)
h_PI_power=h_pi_datadriven_power(xdat,alpha)
u=seq(2,5.5,len=1000) #estimation points
dens_PI=KDE_power_PI(alpha,h_PI_power,xdat,u)
dens_TD=density(xdat^alpha,bw="ucv",from=2,to=5.5,n=1000)
dev.new()
oldpar=par(mar=c(6.5, 5, 2, 2))
plot(u,dens_PI,'l',lwd=2,xlab=expression(length^{frac(1,5)}),ylab="density",cex.lab=1.7,
cex.axis=1.7)
lines(dens_TD,lwd=2,col="red")
legend("topright",legend=as.expression(c(bquote(n==.(n)),bquote(alpha==frac(1,5))),
```

```
bquote(h[PI]==.(round(h_PI_power,2))),bquote(h[CV]==.(round(dens_TD$bw,2))))),
col=c(NA,NA,"black","red"),lwd=c(NA,NA,2,2),cex=1.7,bty="n",y.intersp=1.2)
legend("topleft",legend=c("PI","TD"),lwd=c(2,2),col=c("black","red"),bty="n",cex=1.7)
title("Density of river lengths",cex=1.5)
par(oldpar)
```

KDE_ln_PI

The PI kernel density estimate in the case of the logarithmic 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

<code>h</code>	the bandwidth, scalar,
<code>x</code>	vector of the original positive x-data,
<code>u</code>	vector of the estimation points.

Value

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

References

Savchuk, O. (2026). Density estimation for log-transformed data. *Computational Statistics*, 41(98).

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=1000 originated from the gamma density.
xdat=rgamma(1000,2,scale=2) #original data
n=length(xdat) #sample size
u=seq(-3,5,len=1000) #estimation points
dens_lnttransform=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_lnttransform,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_lognorm_power

Adaptive log-normal plug-in density estimator for power transformations.

Description

Computing the adaptive log-normal PI kernel density estimate for nonnegative data after a power transformation with exponent ranging from 0 to 1 based on the Gaussian kernel.

Usage

```
KDE_PI_adapt_lognorm_power(x, alpha, u)
```

Arguments

x	vector of the original positive x-data,
alpha	the exponent of the power transformation ($0 < \alpha < 1$),
u	vector of nonnegative estimation points.

Value

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

References

Savchuk, O. (2026, submitted). Adaptive log-normal plug-in density estimator for power transformations.

See Also

[KDE_power_PI](#).

Examples

```

#Example (standard exponential density, n=1000, alpha=1/5)
# Plotting the true density g of Y = X^alpha, where X ~ Exp(1) (black)
# together with the adaptive PI estimate(blue) and the
# single-bandwidth PI estimate(olive).
n=1000
alpha=1/5
u=seq(0.001,2,len=1000) #estimation points

```

```
x_exp=rexp(n)
h_PI_exp=h_pi_datadriven_power(x_exp,alpha)
#true density after the power transformation of the standard exponential distribution
g_exp=u^(1/alpha-1)*exp(-u^(1/alpha))/alpha #true density after the power transformation
KDE_exp_adapt_PI=KDE_PI_adapt_lognorm_power(x_exp,alpha,u)
KDE_exp_PI=KDE_power_PI(alpha,h_PI_exp,x_exp,u)
dev.new()
plot(u,g_exp,'l',lwd=2,col="black",xlab="y",ylab="density",cex.lab=1.7,cex.axis=1.7,cex.main=1.7,
main="Standard exponential distribution",ylim=c(0,1.25*max(KDE_exp_adapt_PI)))
lines(u,KDE_exp_adapt_PI,lty="solid",lwd=2,col="blue")
lines(u,KDE_exp_PI,lty="dashed",lwd=2,col="forestgreen")
legend("topleft", legend = c("Adaptive PI estimate", "PI estimate", "density g"),
lty=c("solid","dashed","solid"), col=c("blue","forestgreen","black"),lwd=2,bty="n",cex=1.7)
legend("topright",legend=expression(n==1000,alpha==frac(1,5)),cex=1.7,bty="n",y.intersp=1.3)
```

KDE_PI_adapt_norm	<i>Adaptive normal PI estimate in the case of the logarithmic transformation.</i>
-------------------	---

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 positive 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). Density estimation for log-transformed data. *Computational Statistics*, 41(98).

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 logarithmic 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

1. Savchuk, O. and Schick, A. (2013). Density estimation for power transformations. *Journal of Nonparametric Statistics*, 25(3), 545–559. doi:[10.1080/10485252.2013.811788](https://doi.org/10.1080/10485252.2013.811788)
2. Savchuk, O. (2026, submitted). Adaptive log-normal plug-in density estimator for power transformations.

See Also

[KDE_ln_PI](#), [h_pi_datadriven_power](#).

Examples

```
#Example(original data is standard 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

`h_ln_PI`, [2](#), [4](#)

`h_pi_datadriven_power`, [3](#), [8](#)

`KDE_ln_PI`, [2](#), [4](#), [6](#), [8](#)

`KDE_PI_adapt_lognorm_power`, [3](#), [5](#)

`KDE_PI_adapt_norm`, [6](#)

`KDE_power_PI`, [3](#), [5](#), [7](#)