Package 'denstrip'

March 25, 2025

Type Package

Title Density Strips and Other Methods for Compactly Illustrating Distributions

Version 1.5.5

Date 2025-03-25

Maintainer Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Description Graphical methods for compactly illustrating probability distributions, including density strips, density regions, sectioned density plots and varying width strips, using base R graphics. Note that the 'ggdist' package offers a similar set of tools for illustrating distributions, based on 'ggplot2'.

License GPL (>= 2)

Depends R (>= 2.15)

LazyLoad yes

Imports lattice

Enhances survival

NeedsCompilation no

Author Christopher Jackson [aut, cre]

Repository CRAN

Date/Publication 2025-03-25 16:30:02 UTC

Contents

denstrip-package 2
bpstrip 3
cistrip 5
densregion 6
densregion.normal 8
densregion.survfit 10
denstrip 11
denstrip.legend 15
denstrip.normal 17

sectioned.density 18
seqToIntervals 20
vwstrip 21
vwstrip.normal 23

Index **25**

`denstrip-package` *Overview of the denstrip package*

Description

Graphical methods for compactly illustrating and comparing distributions, particularly distributions arising from parameter estimation or prediction.

Details

`denstrip` implements the *density strip* for illustrating a single univariate distribution. The darkness of the density strip at a point is proportional to the density at that point. A shortcut function `denstrip.normal` draws the strip for the given normal distribution.

`densregion` implements the *density region*, which illustrates the uncertainty surrounding a continuously-varying quantity as a two-dimensional shaded region with darkness proportional to the density. There are shortcut functions `densregion.normal` and `densregion.survfit` for computing and drawing the region for normally-distributed predictions and survival curves, respectively.

`sectioned.density` implements the *sectioned density plots* of Cohen and Cohen (2006). These illustrate distributions using occlusion and varying shading. They were developed for summarising data, but can also be used for illustrating known distributions.

`vwstrip` can be used to draw *varying-width strips* to illustrate distributions, in a similar manner to the *violin plot* for summarising data. The width of the strip is proportional to the density. A shortcut function `vwstrip.normal` draws the strip for the given normal distribution.

`bpstrip` adapts the *box-percentile plot* to illustrate a distribution instead of observed data. This strip has width proportional to the probability of a more extreme point.

`cistrip` implements the popular point and line figure for illustrating point and interval estimates, for example from multiple regression.

These methods are discussed in more detail by Jackson (2008).

Each function is designed to add a graphic to an existing set of plot axes. The plots can be added to either base graphics or **lattice** panels.

Similar graphics using the **ggplot2** package are implemented in the **ggdist** package.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

bpstrip *Box-percentile strips*

Description

Box-percentile strips give a compact illustration of a distribution. The width of the strip is proportional to the probability of a more extreme point. This function adds a box-percentile strip to an existing plot.

Usage

```
bpstrip(x, prob, at, width, horiz=TRUE, scale=1, limits=c(-Inf, Inf),
        col="gray", border=NULL, lwd, lty, ticks=NULL, tlen=1, twd, tty,
        lattice=FALSE)
panel.bpstrip(...)
```

Arguments

x	Either the vector of points at which the probability is evaluated (if prob supplied), or a sample from the distribution (if prob not supplied).
prob	Probability, or cumulative density, of the distribution at x. If prob is not supplied, this is estimated from the sample x using ecdf(x).
at	Position of the centre of the strip on the y-axis (if horiz=TRUE) or the x-axis (if horiz=FALSE).
width	Thickness of the strip at its thickest point, which will be at the median. Defaults to 1/20 of the axis range.
horiz	Draw the strip horizontally (TRUE) or vertically (FALSE).
scale	Alternative way of specifying the thickness of the strip, as a proportion of width.
limits	Vector of minimum and maximum values, respectively, at which to terminate the strip.
col	Colour to shade the strip, either as a built-in R colour name (one of colors()) or an RGB hex value, e.g. black is "#000000".
border	Colour of the border, see polygon. Use border=NA to show no border. The default, 'NULL', means to use 'par("fg")' or its **lattice** equivalent.
lwd	Line width of the border (defaults to par("lwd") or its **lattice** equivalent).
lty	Line type of the border (defaults to par("lty") or its **lattice** equivalent).
ticks	Vector of x-positions on the strip to draw tick marks, or NULL for no ticks.
tlen	Length of the ticks, relative to the thickness of the strip.
twd	Line width of these marks (defaults to par("lwd") or its **lattice** equivalent).
tty	Line type of these marks (defaults to par("lty") or its **lattice** equivalent).
lattice	Set this to TRUE to make bpstrip a **lattice** panel function instead of a base graphics function. panel.bpstrip(x,...) is equivalent to bpstrip(x, lattice=TRUE, ...).
...	Other arguments passed to panel.bpstrip.

Details

The box-percentile strip looks the same as the *box-percentile plot* (Esty and Banfield, 2003) which is a generalisation of the boxplot for summarising data. However, `bpstrip` is intended for illustrating distributions arising from parameter estimation or prediction. Either the distribution is known analytically, or an arbitrarily large sample from the distribution is assumed to be available via a method such as MCMC or bootstrapping.

The function `bpplot` in the **Hmisc** package can be used to draw vertical box-percentile plots of observed data.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

Esty, W. W. and Banfield, J. D. (2003) The box-percentile plot. *Journal of Statistical Software* **8**(17).

See Also

`vwstrip`, `cistrip`, `denstrip`

Examples

```
x <- seq(-4, 4, length=1000)
prob <- pnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-5, 5), xlab="x", ylab="x", type="n")
bpstrip(x, prob, at=1, ticks=qnorm(c(0.25, 0.5, 0.75)))

## Terminate the strip at specific outer quantiles
bpstrip(x, prob, at=2, limits=qnorm(c(0.025, 0.975)))
bpstrip(x, prob, at=3, limits=qnorm(c(0.005, 0.995)))

## Compare with density strip
denstrip(x, dnorm(x), at=0)

## Estimate the density from a large sample
x <- rnorm(10000)
bpstrip(x, at=4)
```

`cistrip` *Line drawings of point and interval estimates*

Description

Adds one or more points and lines to a plot, representing point and interval estimates.

Usage

```
cistrip(x, at, d, horiz=TRUE, pch = 16, cex = 1, lattice=FALSE, ...)
panel.cistrip(...)
```

Arguments

`x`	Either a vector of three elements corresponding to point estimate, lower limit and upper limit of the interval estimate, respectively, or a numeric matrix or data frame with three columns representing point estimates, lower and upper limits.
`at`	Position of the line on the y-axis (if `horiz=TRUE`) or the x-axis (if `horiz=FALSE`).
`d`	Length of the serifs at each end of the line. Defaults to 1/60 of the axis range.
`horiz`	Draw the line horizontally (`TRUE`) or vertically (`FALSE`).
`pch`	Character to draw at the point estimate, see `points`. By default this is a small solid circle, `pch=16`.
`cex`	Expansion factor for the character at the point estimate, for. A vector can be supplied here, one for each estimate, as in `pch`. Useful for meta-analysis forest plots.
`lattice`	Set this to `TRUE` to make `cistrip` a **lattice** panel function instead of a base graphics function. `panel.cistrip(x,...)` is equivalent to `cistrip(x, lattice=TRUE, ...)`.
`...`	Further arguments passed to the `points` and `segments` functions or their **lattice** equivalents. For example `lty,lwd` to set the style and thickness of the line.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

`denstrip`, `vwstrip`, `bpstrip`

Examples

```
## One estimate
x <- c(0.1, -2, 2)
plot(0, type="n", xlim=c(-5, 5), ylim=c(-5, 5), xlab="", ylab="")
abline(h=0, lty=2, col="lightgray")
abline(v=0, lty=2, col="lightgray")
```

```
cistrip(x, at=-0.1)
cistrip(x, at=0.2, lwd=3, d=0.1)
cistrip(x, at=-4, horiz=FALSE, lwd=3, d=0.2)

## Double / triple the area of the central point, as in forest plots
cistrip(x, at=2, d=0.2, pch=22, bg="black")
cistrip(x, at=2.5, d=0.2, pch=22, bg="black", cex=sqrt(2))
cistrip(x, at=3, d=0.2, pch=22, bg="black", cex=sqrt(3))

## Several estimates
x <- rbind(c(0.1, -2, 2), c(1, -1, 2.3),
           c(-0.2, -0.8, 0.4), c(-0.3, -1.2, 1.5))
plot(0, type="n", xlim=c(-5, 5), ylim=c(-5, 5), xlab="", ylab="")
cistrip(x, at=1:4)
abline(v=0, lty=2, col="lightgray")
cistrip(x, at=1:4, horiz=FALSE, lwd=3, d=0.2)
abline(h=0, lty=2, col="lightgray")
```

`densregion` *Density regions*

Description

A density region uses shading to represent the uncertainty surrounding a continuously-varying quantity, such as a survival curve or a forecast from a time series. The darkness of the shading is proportional to the (posterior, predictive or fiducial) density. This function adds a density region to an existing plot.

Usage

```
densregion(x, ...)
## Default S3 method:
densregion(x, y, z, pointwise=FALSE, nlevels=100,
                  colmax=par("fg"), colmin="white", scale=1, gamma=1,
                  contour=FALSE, ...)
```

Arguments

`x`	Suppose the continuously-varying quantity varies over a space S. `x` is a vector of the points in S at which the full posterior / predictive / fiducial distribution will be evaluated.
`y`	Vector of ordinates at which the density of the distribution will be evaluated for every point in `x`.
`z`	Matrix of densities on the grid defined by `x` and `y`, with rows corresponding to elements of `x` and columns corresponding to elements of `y`.
`pointwise`	If `TRUE` then the maximum density at each `x` is shaded with `colmax` (default black), and the shading intensity is proportional to the density within each `x`. If `FALSE` then the maximum density *over all* `x` is shaded with `colmax`, and the shading is proportional to the density over all `x`.

nlevels	Number of distinct shades to use to illustrate the varying densities. The default of 100 should result in a plot with smoothly-varying shading.
colmax	Colour to shade the maximum density, either as a built-in R colour name (one of colors()) or an RGB hex value. Defaults to par("fg") which is normally "black", or "#000000".
colmin	Colour to shade the minimum density, likewise. Defaults to "white". If this is set to "transparent", and the current graphics device supports transparency (see rgb), then multiple regions drawn on the same plot will merge smoothly.
scale	Proportion of colmax to shade the maximum density, for example scale=0.5 with colmax="black" for a mid-grey colour.
gamma	Gamma correction to apply to the colour palette, see denstrip.
contour	If TRUE then contours are added to illustrate lines of constant density.
...	Further arguments passed to or from other methods, such as the contour function for drawing contours.

Details

The plot is shaded by interpolating the value of the density between grid points, using the algorithm described by Cleveland (1993) as implemented in the filled.contour function.

With **lattice** graphics, similar plots can be implemented using the contourplot or levelplot functions.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

Cleveland, W. S. (1993) *Visualizing Data*. Hobart Press, Summit, New Jersey.

See Also

densregion.survfit, densregion.normal, denstrip

Examples

```
## Predictive uncertainty around a hypothetical regression line

x <- 1:10
nx <- length(x)
est <- seq(0, 1, length=nx)
lcl <- seq(-1, 0, length=nx)
ucl <- seq(1, 2, length=nx)
se <- (est - lcl)/qnorm(0.975)

y <- seq(-3, 3, length=100)
```

```
z <- matrix(nrow=nx, ncol=length(y))
for(i in 1:nx)
  z[i,] <- dnorm(y, est[i], se[i])
plot(x, type="n", ylim=c(-5.5, 2.5))
densregion(x, y, z, colmax="darkgreen")
lines(x, est)
lines(x, lcl, lty=2)
lines(x, ucl, lty=2)
box()

## On graphics devices that support transparency, specify
## colmin="transparent" to allow adjacent regions to overlap smoothly
densregion(x, y-1, z, colmax="magenta", colmin="transparent")


## or automatically choose the y points to evaluate the density

plot(x, type="n", ylim=c(-1.5, 2.5))
densregion.normal(x, est, se, ny=50, colmax="darkgreen")
lines(x, est)
lines(x, lcl, lty=2)
lines(x, ucl, lty=2)
```

`densregion.normal` *Density regions based on normal distributions*

Description

Adds a density region to an existing plot of a normally-distributed quantity with continuously-varying mean and standard deviation, such as a time series forecast. Automatically computes a reasonable set of ordinates to evaluate the density at, which span the whole forecast space.

Usage

```
## S3 method for class 'normal'
densregion(x, mean, sd, ny=20, ...)
```

Arguments

`x`	Suppose the continuously-varying quantity varies over a space S. `x` is a vector of the points in S at which the posterior / predictive / fiducial distribution will be evaluated.
`mean`	Vector of normal means at each point in `x`.
`sd`	Vector of standard deviations at each point in `x`.
`ny`	Minimum number of points to calculate the density at for each `x`. The density is calculated for at least `ny` equally spaced normal quantiles for each point. The density is actually calculated at the union over `x` of all such points, for each `x`.
`...`	Further arguments passed to `densregion`.

Details

The plot is shaded by interpolating the value of the density between grid points, using the algorithm described by Cleveland (1993) as implemented in the `filled.contour` function.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

Cleveland, W. S. (1993) *Visualizing Data*. Hobart Press, Summit, New Jersey.

See Also

`densregion`, `densregion.survfit`, `denstrip`

Examples

```
## Time series forecasting

(fit <- arima(USAccDeaths, order = c(0,1,1),
              seasonal = list(order=c(0,1,1))))
pred <- predict(fit, n.ahead = 36)
plot(USAccDeaths, xlim=c(1973, 1982), ylim=c(5000, 15000))

## Compute normal forecast densities automatically (slow)

## Not run:
densregion.normal(time(pred$pred), pred$pred, pred$se,
                  pointwise=TRUE, colmax="darkgreen")
lines(pred$pred, lty=2)
lines(pred$pred + qnorm(0.975)*pred$se, lty=3)
lines(pred$pred - qnorm(0.975)*pred$se, lty=3)

## End(Not run)

## Compute forecast densities by hand (more efficient)

nx <- length(pred$pred)
y <- seq(5000, 15000, by=100)
z <- matrix(nrow=nx, ncol=length(y))
for(i in 1:nx)
    z[i,] <- dnorm(y, pred$pred[i], pred$se[i])
plot(USAccDeaths, xlim=c(1973, 1982), ylim=c(5000, 15000))
densregion(time(pred$pred), y, z, colmax="darkgreen", pointwise=TRUE)
lines(pred$pred, lty=2)
lines(pred$pred + qnorm(0.975)*pred$se, lty=3)
lines(pred$pred - qnorm(0.975)*pred$se, lty=3)
```

```
densregion(time(pred$pred), y+2000, z, colmax="darkblue", pointwise=TRUE)
```

---

densregion.survfit *Density regions for survival curves*

Description

Adds a density region to a survival plot. The shading of the region has darkness proportional to the fiducial density of the point. This distribution is assumed to be normal with standard deviation calculated using the lower confidence limit stored in the survival curve object.

Usage

```
## S3 method for class 'survfit'
densregion(x, ny=20, ...)
```

Arguments

x	Survival curve object, returned by survfit. Confidence intervals must have been calculated, using conf.type.
ny	Minimum number of points to calculate the density at for each event time. The default of 20 should be sufficient to obtain smooth-looking plots.
...	Further arguments passed to densregion.default.

Details

The density is calculated at a grid of points, and interpolated using the method referred to in densregion.

Note

In general, this approach can only illustrate one survival curve per plot. Though if the graphics device supports transparency (e.g. PDF) multiple curves can be made to overlap smoothly - see the example below.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

See Also

densregion, densregion.normal, denstrip

Examples

```
if (requireNamespace("survival", quietly=TRUE)){

library(survival)
fit <- survfit(Surv(time, status) ~ 1, data=aml, conf.type="log-log")
plot(fit, col=0)
densregion(fit)
lines(fit, lwd=3, conf.int=FALSE, lty=1)
lines(fit, lwd=3, conf.int=TRUE, lty=2)

## Wider CIs based on log survival
fit <- survfit(Surv(time, status) ~ 1, data=aml, conf.type="log")
plot(fit, col=0)
densregion(fit) # Big variation in maximum density
plot(fit, col=0)
densregion(fit, pointwise=TRUE, colmax="maroon4")
par(new=TRUE)
plot(fit)

## Narrower CIs based on untransformed survival.
## Normal assumption probably unrealistic
fit <- survfit(Surv(time, status) ~ 1, data=aml, conf.type="plain")
plot(fit, col=0)
densregion(fit, pointwise=TRUE, colmax="darkmagenta")
par(new=TRUE)
plot(fit)

## Multiple survival curves on same axes
## Should overlap smoothly on devices that allow transparency
fit2 <- survfit(Surv(time, status) ~ x, data=aml, conf.type="log-log")
fit2x1 <- survfit(Surv(time, status) ~ 1, data=aml,
                  conf.type="log-log", subset=(x=="Maintained"))
fit2x0 <- survfit(Surv(time, status) ~ 1, data=aml,
                  conf.type="log-log", subset=(x=="Nonmaintained"))
plot(fit2, lwd=3, xlab="Weeks", ylab="Survival", xlim=c(0, 60),
     lty=1:2, col=c("red", "blue"), conf.int=TRUE, mark.time=TRUE)
densregion(fit2x1, colmax="red", gamma=2)
densregion(fit2x0, colmax="blue", gamma=2)

}
```

`denstrip` *Density strips*

Description

The density strip illustrates a univariate distribution as a shaded rectangular strip, whose darkness at a point is proportional to the probability density. The strip is darkest at the maximum density and

fades into the background at the minimum density. It may be used to generalise the common point-and-line drawing of a point and interval estimate, by representing the entire posterior or predictive distribution of the estimate. This function adds a density strip to an existing plot.

Usage

```
denstrip(x, dens, at, width, horiz=TRUE, colmax, colmin="white",
         scale=1, gamma=1, ticks=NULL, tlen=1.5, twd, tcol, mticks=NULL,
         mlen=1.5, mwd, mcol, lattice=FALSE, ...)
panel.denstrip(...)
```

Arguments

x	Either the vector of points at which the density is evaluated (if dens supplied), or a sample from the distribution (if dens not supplied).
dens	Density at x. If dens is not supplied, the density of the sample x is estimated by kernel density estimation, using density(x,...).
at	Position of the centre of the strip on the y-axis (if horiz=TRUE) or the x-axis (if horiz=FALSE).
width	Thickness of the strip, that is, the length of its shorter dimension. Defaults to 1/30 of the axis range.
horiz	Draw the strip horizontally (TRUE) or vertically (FALSE).
colmax	Colour at the maximum density, either as a built-in R colour name (one of colors()) or an RGB hex value. Defaults to par("fg") which is normally "black", or "#000000". Or in **lattice**, defaults to trellis.par.get("add.line")$col.
colmin	Colour to shade the minimum density, likewise. Defaults to "white". If this is set to "transparent", and the current graphics device supports transparency (see rgb), then overlapping strips will merge smoothly.
scale	Proportion of colmax to shade the maximum density, for example scale=0.5 with colmax="black" for a mid-grey colour.
gamma	Gamma correction to apply to the colour palette. The default of 1 should give an approximate perception of darkness proportional to density, but this may need to be adjusted for different displays. Values of gamma greater than 1 produce colours weighted towards the lighter end, and values of between 0 and 1 produce darker colours.
ticks	Vector of x-positions on the strip to draw tick marks, or NULL for no ticks.
tlen	Length of these tick marks relative to the strip width.
twd	Line thickness of these marks (defaults to par("lwd"), or in **lattice**, to trellis.par.get("add.line")$lwd*2.).
tcol	Colour of the tick marks. Defaults to colmax.
mticks	x-position to draw a thicker tick mark or tick marks (for example, at the mean or median).
mlen	Length of this mark relative to the strip width.
mwd	Line thickness of this mark (defaults to par("lwd")*2, or in **lattice**, to trellis.par.get("add.line")$

mcol	Colour of this mark. Defaults to colmax.
lattice	Set this to TRUE to make denstrip a **lattice** panel function instead of a base graphics function. panel.denstrip(x,...) is equivalent to denstrip(x, lattice=TRUE, ...).
...	Additional arguments supplied to density(x,...), if the density is being estimated. For example, bw to change the bandwidth of the kernel.

In other software

Similar graphics using the **ggplot2** package are implemented in the **ggdist** package, termed "gradient intervals".

In OpenBUGS, density strips are available via the Inference/Compare menu.

See this blog post: http://blogs.sas.com/content/graphicallyspeaking/2012/11/03/density-strip-plot/, for density strips in SAS.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

See Also

denstrip.legend, densregion.

Examples

```
## Illustrate a known standard normal distribution
## Various settings to change the look of the plot

x <- seq(-4, 4, length=10000)
dens <- dnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-5, 5), xlab="x", ylab="x", type="n")
denstrip(x, dens, at=0) # default width
denstrip(x, dens, width=0.5, at=0)
denstrip(x, dens, at=-4, ticks=c(-2, 0, 2))
denstrip(x, dens, at=-3, ticks=c(-2, 2), mticks=0)
denstrip(x, dens, at=-2, ticks=c(-2, 2), mticks=0, mlen=3,
         mwd=4, colmax="#55AABB")
denstrip(x, dens, at=1, ticks=c(-2, 2), tlen=3, twd=3)
denstrip(x, dens, at=-4, ticks=c(-2, 2), mticks=0, colmax="darkgreen",
         horiz=FALSE)
x <- rnorm(1000) # Estimate the density
denstrip(x, width=0.2, at=-3, ticks=c(-2, 2), mticks=0, colmax="darkgreen",
         horiz=FALSE)
denstrip(x, at=2, width=0.5, gamma=2.2)
denstrip(x, at=3, width=0.5, gamma=1/2.2)
```

```
### Specifying colour of minimum density
par(bg="lightyellow")
plot(x, xlim=c(-5, 5), ylim=c(-5, 5), xlab="x", ylab="x", type="n")
x <- seq(-4, 4, length=10000)
dens <- dnorm(x)
## Equivalent ways of drawing same distribution
denstrip(x, dens, at=-1, ticks=c(-2, 2), mticks=0, colmax="darkmagenta")
denstrip(x, dens, at=-2, ticks=c(-2, 2), mticks=0, colmax="darkmagenta",
         colmin="lightyellow")
## ...though the next only works if graphics device supports transparency
denstrip(x, dens, at=-3, ticks=c(-2, 2), mticks=0, colmax="darkmagenta",
         colmin="transparent")
denstrip(x, dens, at=-4, ticks=c(-2, 2), mticks=0, colmax="#8B008B", colmin="white")

## Alternative to density regions (\link{densregion.survfit}) for
## survival curves - a series of vertical density strips with no
## interpolation

if (requireNamespace("survival", quietly=TRUE)){

library(survival)
fit <- survfit(Surv(time, status) ~ 1, data=aml, conf.type="log-log")
plot(fit, col=0)
lse <- (log(-log(fit$surv)) - log(-log(fit$upper)))/qnorm(0.975)
n <- length(fit$time)
lstrip <- fit$time - (fit$time-c(0,fit$time[1:(n-1)])) / 2
rstrip <- fit$time + (c(fit$time[2:n], fit$time[n])-fit$time) / 2
for (i in 1:n) {
    y <- exp(-exp(qnorm(seq(0,1,length=1000)[-c(1,1000)],
                        log(-log(fit$surv))[i], lse[i])))
    z <- dnorm(log(-log(y)), log(-log(fit$surv))[i], lse[i])
    denstrip(y, z, at=(lstrip[i]+rstrip[i])/2,
                 width=rstrip[i]-lstrip[i],
                 horiz=FALSE, colmax="darkred")
}
par(new=TRUE)
plot(fit, lwd=2)

}

## Use for lattice graphics (first example from help(xyplot))

library(lattice)
Depth <- equal.count(quakes$depth, number=8, overlap=.1)
xyplot(lat ~ long | Depth, data = quakes,
       panel = function(x, y) {
           panel.xyplot(x, y)
           panel.denstrip(x, horiz=TRUE, at=-10, ticks=mean(x))
           panel.denstrip(y, horiz=FALSE, at=165, ticks=mean(y))
       }
       )
```

```
## Lattice example data: heights of singing voice types

bwplot(voice.part ~ height, data=singer, xlab="Height (inches)",
       panel=panel.violin, xlim=c(50,80))
bwplot(voice.part ~ height, data=singer, xlab="Height (inches)",
       panel = function(x, y) {
           xlist <- split(x, factor(y))
           for (i in seq(along=xlist))
               panel.denstrip(x=xlist[[i]], at=i)
       },
       xlim=c(50,80)
       )
```

`denstrip.legend` *Add a legend to a density strip or shaded region*

Description

Add a legend to an existing plot with a density strip or shaded region, indicating the mapping of colours to densities.

Usage

```
denstrip.legend(x, y, width, len, colmax, colmin="white", gamma=1,
                horiz=FALSE, max=1, nticks = 5, ticks, value.adj = 0,
                cex, main = "Density", lattice=FALSE)
panel.denstrip.legend(...)
```

Arguments

`x`	Central x position of the legend.
`y`	Central y position of the legend.
`width`	Width of the legend strip, that is, the length of its shorter dimension. Defaults to 1/30 of the axis range.
`len`	Length of the legend strip, that is, the length of its longer dimension. Defaults to 1/4 of the axis range.
`colmax`	Colour at the maximum density, either as a built-in R colour name (one of `colors()`) or an RGB hex value. Defaults to `par("fg")` or its **lattice** equivalent, which is `"black"` by default.
`colmin`	Colour to shade the minimum density, likewise. Defaults to "white". If this is set to `"transparent"`, and the current graphics device supports transparency (see `rgb`), then overlapping strips will merge smoothly.
`gamma`	Gamma correction to apply to the colour palette, see `denstrip`.
`horiz`	Legend strip drawn vertically (`FALSE`) or horizontally (`TRUE`).

max	Maximum density on the legend, which is represented by colmax. With the default of 1, the legend indicates the mapping of colours to proportions of the maximum density.
nticks	Number of tick marks on the axis adjacent to the legend, if ticks not supplied.
ticks	Positions of numbered ticks on the axis adjacent to the legend. Defaults to nticks equally spaced ticks between 0 and the maximum density.
value.adj	Extra adjustment for the axis labels to the right (if horiz=FALSE) or downwards (if horiz=TRUE).
cex	Text expansion. Defaults to par("cex") * 0.75 or trellis.par.get("axis.text")$cex * 0.75.
main	Text to place above the legend.
lattice	Set this to TRUE to make denstrip.legend a **lattice** panel function instead of a base graphics function. panel.denstrip.legend(x,...) is equivalent to denstrip.legend(x, lattice=TRUE, ...).
...	Other arguments passed to panel.denstrip.legend.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

denstrip, densregion

Examples

```
if (requireNamespace("survival", quietly=TRUE)){

library(survival)
fit <- survfit(Surv(time, status) ~ 1, data=aml, conf.type="log-log")
plot(fit, col=0)
densregion(fit)
denstrip.legend(100, 0.8)

    ### TODO if max not supplied - ticks are not meaningful.
    ### In help example, find actual max dens used for densregion

denstrip.legend(120, 0.8, width=3, len=0.4, value.adj=5)
denstrip.legend(40, 0.9, horiz=TRUE)
denstrip.legend(60, 0.7, horiz=TRUE, width=0.02, len=50, value.adj=0.04)


}
```

`denstrip.normal` *Density strip for a normal or log-normal distribution*

Description

Draws a density strip for a normal or log-normal distribution with the given mean and standard deviation, based on computing the density at a large set of equally-spaced quantiles.

Usage

```
denstrip.normal(mean, sd, log=FALSE, nx=1000, ...)
panel.denstrip.normal(...)
```

Arguments

`mean`	Mean of the normal distribution.
`sd`	Standard deviation of the normal distribution.
`log`	If `TRUE` then the strip for a log-normal distribution, with mean and SD on the log scale `mean` and `sd`, respectively, is plotted. This may be useful for illustrating hazard ratios or odds ratios.
`nx`	Number of points to evaluate the density at.
`...`	Further arguments passed to `denstrip`, for example, `at` to position the strip on the y-axis, or `lattice=TRUE` to use as a lattice panel function. `panel.denstrip.normal(x,...)` is equivalent to `denstrip.normal(x, lattice=TRUE,...)`.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

`denstrip`

Examples

```
x <- seq(-4, 4, length=10000)
dens <- dnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-1, 2), xlab="x", ylab="", type="n", axes=FALSE)
axis(1)
denstrip(x, dens, at=0, width=0.3)
denstrip.normal(0, 1, at=1, width=0.3)

### log-normal distribution
sdlog <- 0.5
x <- rlnorm(10000, 0, sdlog)
plot(x, xlim=c(0, 5), ylim=c(-2, 4), xlab="x", , ylab="", type="n",
     axes=FALSE)
```

```
axis(1)
abline(v=1, lty=2, col="lightgray")
denstrip(x, at=0, ticks=exp(-sdlog^2), width=0.4) # tick at theoretical maximum density
denstrip(x, at=1, bw=0.1, ticks=exp(-sdlog^2), width=0.4)
denstrip.normal(0, sdlog, log=TRUE, at=3, nx=1000,
                ticks=exp(-sdlog^2), width=0.4)
```

`sectioned.density` *Sectioned density plots*

Description

Sectioned density plots (Cohen and Cohen, 2006) use shading and occlusion to give a compact illustration of a distribution, such as the empirical distribution of data.

Usage

```
sectioned.density(x, dens, at, width, offset, ny,
                  method=c("kernel","frequency"), nx, horiz=TRUE,
                  up.left = TRUE, colmax, colmin="white", gamma=1,
                  lattice=FALSE, ...)
panel.sectioned.density(...)
```

Arguments

`x`	Either the vector of points at which the density is evaluated (if `dens` supplied), or a sample from the distribution (if `dens` not supplied).
`dens`	Density at `points`. If `dens` is not supplied, the density of the distribution underlying `x` is estimated using the method specified in `method`.
`at`	Position of the bottom of the plot on the y-axis (if `horiz=TRUE`) or position of the right of the plot on the x-axis (if `horiz=FALSE`) (required).
`ny`	Number of fixed-width intervals for categorising the density.
`width`	Width of individual rectangles in the plot. Defaults to the range of the axis divided by 20.
`offset`	Offset for adjacent rectangles. Defaults to `width/3`.
`method`	Method of estimating the density of `x`, when `dens` is not supplied. If `"kernel"` (the default) then kernel density estimation is used, via `density(x,...)`. If `"frequency"` then the density is estimated as the relative frequency in a series of bins, as in Cohen and Cohen (2006). This method is controlled by the number of data bins `nx`.
`nx`	Number of data bins for the `"frequency"` density estimation method. The default uses Sturges' formula (see `nclass.Sturges`, `hist`).
`horiz`	If `horiz=TRUE`, then the plot is horizontal and points upwards. If `horiz=FALSE` then the plot is vertical and points leftwards, as the illustrations in Cohen and Cohen (2006).

up.left	If changed to FALSE, then horizontal plots point downwards and vertical plots point rightwards.
colmax	Darkest colour, either as a built-in R colour name (one of colors()) or an RGB hex value. Defaults to par("fg") or its **lattice** equivalent, which is normally "black", or "#000000".
colmin	Lightest colour, either as a built-in R colour name (one of colors()) or an RGB hex value. Defaults to white.
gamma	Gamma correction to apply to the colour palette, see denstrip.
lattice	Set this to TRUE to make sectioned.density a **lattice** panel function instead of a base graphics function. panel.sectioned.density(x,...) is equivalent to sectioned.density(x, lattice=TRUE, ...).
...	Additional arguments supplied to density(x,...), if method="kernel".

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk> (R implementation)

References

Cohen, D. J. and Cohen, J. The sectioned density plot. *The American Statistician* (2006) **60**(2):167–174

Examples

```
## Fisher's iris data
## Various settings to change the look of the plot
hist(iris$Sepal.Length, nclass=20, col="lightgray")
sectioned.density(iris$Sepal.Length, at=0.2)
sectioned.density(iris$Sepal.Length, at=5)
sectioned.density(iris$Sepal.Length, at=10, width=0.5)
hist(iris$Sepal.Length, nclass=20, col="lightgray")
sectioned.density(iris$Sepal.Length, at=7, width=0.5,
                  offset=0.1, colmax="darkmagenta")
sectioned.density(iris$Sepal.Length, at=9, width=0.5,
                  offset=0.1, ny=15, colmin="lemonchiffon")
## frequency method less smooth than kernel density
sectioned.density(iris$Sepal.Length, at=12, width=0.5, offset=0.1,
                  method="frequency")
sectioned.density(iris$Sepal.Length, at=13.5, width=0.5, offset=0.1,
                  method="frequency", nx=20)

## Illustrate a known distribution
x <- seq(-4, 4, length=1000)
dens <- dnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-5, 5), xlab="x", ylab="x", type="n")
sectioned.density(x, dens, ny=8, at=0, width=0.3)
sectioned.density(x, dens, ny=16, at=2, width=0.1)
sectioned.density(x, dens, at=-3, horiz=FALSE)
sectioned.density(x, dens, at=4, width=0.3, horiz=FALSE)
```

`seqToIntervals` *Find contiguous sequences in a vector of integers*

Description

Get all sequences of contiguous values in a vector of integers.

Usage

```
seqToIntervals(x)
```

Arguments

`x`	A vector of integers, for example, representing indices. `x` is coerced to integer, sorted, and unique values extracted, if necessary, before finding the contiguous sequences.

Value

A matrix with one row for each sequence, and two columns containing the start and the end of the sequence, respectively.

Author(s)

Chris Jackson <chris.jackson@mrc-bsu.cam.ac.uk>. Thanks to Tobias Verbeke for the suggestion.

See Also

`sectioned.density`

Examples

```
seqToIntervals(1:10)  # [1 10]
seqToIntervals(c(1:10, 15:18, 20))  # [1 10; 15 18; 20 20]
# vectorised, so efficient for large vectors x
seqToIntervals(sample(1:1000000, size=999990))
```

vwstrip *Varying-width strips*

Description

Varying-width strips give a compact illustration of a distribution. The width of the strip is proportional to the density. This function adds a varying-width strip to an exising plot.

Usage

```
vwstrip(x, dens, at, width, horiz=TRUE, scale=1, limits=c(-Inf, Inf),
        col="gray", border=NULL, lwd, lty, ticks=NULL, tlen=1, twd, tty,
        lattice=FALSE,...)
panel.vwstrip(...)
```

Arguments

x	Either the vector of points at which the density is evaluated (if dens supplied), or a sample from the distribution (if dens not supplied).
dens	Density at x. If dens is not supplied, the density of the sample x is estimated by kernel density estimation, using density(x,...).
at	Position of the centre of the strip on the y-axis (if horiz=TRUE) or the x-axis (if horiz=FALSE).
width	Thickness of the strip at the maximum density, that is, the length of its shorter dimension. Defaults to 1/20 of the axis range.
horiz	Draw the strip horizontally (TRUE) or vertically (FALSE).
scale	Alternative way of specifying the thickness of the strip, as a proportion of width.
limits	Vector of minimum and maximum values, respectively, at which to terminate the strip.
col	Colour to shade the strip, either as a built-in R colour name (one of colors()) or an RGB hex value, e.g. black is "#000000".
border	Colour of the border, see polygon. Use border=NA to show no border. The default, 'NULL', means to use 'par("fg")' or its **lattice** equivalent
lwd	Line width of the border (defaults to par("lwd") or its **lattice** equivalent).
lty	Line type of the border (defaults to par("lty") or its **lattice** equivalent).
ticks	Vector of x-positions on the strip to draw tick marks, or NULL for no ticks.
tlen	Length of the ticks, relative to the thickness of the strip.
twd	Line width of these marks (defaults to par("lwd") or its **lattice** equivalent).
tty	Line type of these marks (defaults to par("lty") or its **lattice** equivalent).
lattice	Set this to TRUE to make vwstrip a **lattice** panel function instead of a base graphics function. panel.vwstrip(x,...) is equivalent to vwstrip(x, lattice=TRUE, ...).
...	Additional arguments supplied to density(x,...), if the density is being estimated.

Details

Varying-width strips look like *violin plots*. The difference is that violin plots are intended to summarise data, while `vwstrip` is intended to illustrate a distribution arising from parameter estimation or prediction. Either the distribution is known analytically, or an arbitrarily large sample from the distribution is assumed to be available via a method such as MCMC or bootstrapping.

Illustrating outliers is important for summarising data, therefore violin plots terminate at the sample minimum and maximum and superimpose a box plot (which appears like the bridge of a violin, hence the name). Varying-width strips, however, are used to illustrate known distributions which may have unbounded support. Therefore it is important to think about where the strips should terminate (the `limits` argument). For example, the end points may illustrate a particular pair of extreme quantiles of the distribution.

The function `vioplot` in the **vioplot** package and `panel.violin` in the **lattice** package can be used to draw violin plots of observed data.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. (2008) Displaying uncertainty with shading. *The American Statistician*, 62(4):340-347.

Hintze, J.L. and Nelson, R.D. (1998) Violin plots: a box plot - density trace synergism. *The American Statistician* **52**(2),181–184.

See Also

`denstrip`, `bpstrip`, `cistrip`.

Examples

```
x <- seq(-4, 4, length=10000)
dens <- dnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-5, 5), xlab="x", ylab="x", type="n")
vwstrip(x, dens, at=1, ticks=qnorm(c(0.025, 0.25,0.5, 0.75, 0.975)))

## Terminate the strip at specific outer quantiles
vwstrip(x, dens, at=2, limits=qnorm(c(0.025, 0.975)))
vwstrip(x, dens, at=3, limits=qnorm(c(0.005, 0.995)))

## Compare with density strip
denstrip(x, dens, at=0)

## Estimate the density from a large sample
x <- rnorm(10000)
vwstrip(x, at=4)
```

vwstrip.normal *Varying width strip for a normal or log-normal distribution*

Description

Draws a varying width strip for a normal or log-normal distribution with the given mean and standard deviation, based on computing the density at a large set of equally-spaced quantiles.

Usage

```
vwstrip.normal(mean, sd, log=FALSE, nx=1000, ...)
panel.vwstrip.normal(...)
```

Arguments

mean	Mean of the normal distribution.
sd	Standard deviation of the normal distribution.
log	If TRUE then the strip for a log-normal distribution, with mean and SD on the log scale mean and sd, respectively, is plotted. This may be useful for illustrating hazard ratios or odds ratios.
nx	Number of points to evaluate the density at.
...	Further arguments passed to vwstrip, for example, at to position the strip on the y-axis, or lattice=TRUE to use as a lattice panel function. panel.vwstrip.normal(x,...) is equivalent to vwstrip.normal(x, lattice=TRUE,...).

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

vwstrip

Examples

```
x <- seq(-4, 4, length=10000)
dens <- dnorm(x)
plot(x, xlim=c(-5, 5), ylim=c(-1, 2), xlab="x", ylab="",
     type="n", axes=FALSE)
axis(1)
vwstrip(x, dens, at=0, width=0.4, limits=qnorm(c(0.005, 0.995)))
vwstrip.normal(0, 1, at=1, width=0.4, limits=qnorm(c(0.005, 0.995)))

### log-normal distribution
sdlog <- 0.5
x <- rlnorm(10000, 0, sdlog)
plot(x, xlim=c(0, 5), ylim=c(-1, 3), xlab="x", ylab="",
```

```
     type="n", axes=FALSE)
axis(1)
abline(v=1, lty=2, col="lightgray")
vwstrip(x, at=0, width=0.4, ticks=exp(-sdlog^2),
        limits=qlnorm(c(0.005,0.975),0,sdlog)) # tick at theoretical maximum density
vwstrip(x, at=1, width=0.4, bw=0.1, ticks=exp(-sdlog^2),
        limits=qlnorm(c(0.005,0.975),0,sdlog))
vwstrip.normal(0, sdlog, log=TRUE, at=2.5, width=0.4, nx=1000,
               ticks=exp(-sdlog^2), limits=qlnorm(c(0.005,0.975),0,sdlog))
```

Index

∗ **aplot**
  bpstrip, 3
  cistrip, 5
  densregion, 6
  densregion.normal, 8
  densregion.survfit, 10
  denstrip, 11
  denstrip.legend, 15
  denstrip.normal, 17
  sectioned.density, 18
  vwstrip, 21
  vwstrip.normal, 23
∗ **arith**
  seqToIntervals, 20
∗ **color**
  densregion, 6
  densregion.normal, 8
  denstrip, 11
  denstrip.legend, 15
∗ **manip**
  seqToIntervals, 20
∗ **package**
  denstrip-package, 2
∗ **survival**
  densregion.survfit, 10

bpstrip, *2, 3*, 3, *4, 5*, *22*

cistrip, *2*, *4, 5*, 5, *22*
colors, *3*, *7*, *12*, *19*, *21*
contour, *7*
contourplot, *7*

density, *12, 13*, *18, 19*, *21*
densregion, *2*, 6, *8–10*, *13*, *16*
densregion.default, *10*
densregion.normal, *2*, *7*, 8, *10*
densregion.survfit, *2*, *7*, *9*, 10
denstrip, *2*, *4, 5*, *7*, *9, 10*, 11, *13*, *15–17*, *19*, *22*
denstrip-package, 2
denstrip.legend, *13*, 15, *16*
denstrip.normal, *2*, 17

ecdf, *3*

filled.contour, *7*, *9*

hist, *18*

levelplot, *7*

nclass.Sturges, *18*

panel.bpstrip *(*bpstrip*)*, 3
panel.cistrip *(*cistrip*)*, 5
panel.denstrip *(*denstrip*)*, 11
panel.denstrip.legend *(*denstrip.legend*)*, 15
panel.denstrip.normal *(*denstrip.normal*)*, 17
panel.sectioned.density *(*sectioned.density*)*, 18
panel.violin, *22*
panel.vwstrip *(*vwstrip*)*, 21
panel.vwstrip.normal *(*vwstrip.normal*)*, 23
par, *3*, *12*, *16*, *19*, *21*
points, *5*
polygon, *3*, *21*

rgb, *7*, *12*, *15*

sectioned.density, *2*, 18, *19, 20*
segments, *5*
seqToIntervals, 20
survfit, *10*

trellis.par.get, *12*, *16*

vwstrip, *2*, *4, 5*, *21*, 21, *22, 23*
vwstrip.normal, *2*, 23