

Package ‘ContourFunctions’

May 4, 2017

Type Package

Title Create Contour Plots from Data or a Function

Version 0.1.0

Description Provides functions for making contour plots.

The contour plot can be created from grid data, a function, or a data set. If non-grid data is given, then a Gaussian process is fit to the data and used to create the contour plot.

License GPL-3

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, mlegp, laGP

VignetteBuilder knitr

URL <https://github.com/CollinErickson/contour>

NeedsCompilation no

Author Collin Erickson [aut, cre]

Maintainer Collin Erickson <collinberickson@gmail.com>

Repository CRAN

Date/Publication 2017-05-04 16:22:19 UTC

R topics documented:

| | |
|----------------------------|---|
| cf | 2 |
| cf_data | 2 |
| cf_func | 3 |
| cf_grid | 4 |
| cf_grid1 | 6 |
| csa | 8 |
| multicolor.title | 8 |

| | |
|--------------|-----------|
| Index | 10 |
|--------------|-----------|

| | |
|----|---|
| cf | <i>Simpler function for making contours with cf package. Won't give argument completion, so all must be specified</i> |
|----|---|

Description

Simpler function for making contours with cf package. Won't give argument completion, so all must be specified

Usage

```
cf(...)
```

Arguments

| | |
|-----|---|
| ... | Arguments to be passed to cf_func or cf_data based on data type of first argument |
|-----|---|

Value

Whatever is returned from other function, probably nothing

Examples

```
cf(function(x){x[1]^2 - x[2]})
x <- runif(20)
y <- runif(20)
z <- exp(-(x-.5)^2-5*(y-.5)^2)# + rnorm(20,0,.05)
cf(x,y,z)
```

| | |
|---------|--|
| cf_data | <i>Makes filled contour plot from data without sidebar by interpolating with Gaussian process, passes model to cf_func to make contour</i> |
|---------|--|

Description

Makes filled contour plot from data without sidebar by interpolating with Gaussian process, passes model to cf_func to make contour

Usage

```
cf_data(x, y = NULL, z = NULL, xlim = NULL, ylim = NULL, xlim = NULL,
        fit = "", ...)
```

Arguments

| | |
|-------|---|
| x | either just x data, x and y data, or x, y and z data |
| y | either y data, z data, or null |
| z | either z data or null |
| xlim | x limits for the contour plot, will be set to data limits +- 5% if not specified |
| ylim | y limits for the contour plot, will be set to data limits +- 5% if not specified |
| xylim | x and y limits for the contour plot |
| fit | Method to fit a model with. Current options are laGP (default) and mlegp. laGP is faster but might cause trouble. |
| ... | passed to cf_func |

References

- [1] filled.contour R function, copied function but removed part for sidebar
- [2] <http://stackoverflow.com/questions/16774928/removing-part-of-a-graphic-in-r>, answer by P La-pointe

Examples

```
x <- runif(20)
y <- runif(20)
z <- exp(-(x-.5)^2-5*(y-.5)^2)
cf_data(x,y,z)
```

| | |
|---------|--|
| cf_func | <i>Makes filled contour plot from function without sidebar, uses cf_grid</i> |
|---------|--|

Description

Makes filled contour plot from function without sidebar, uses cf_grid

Usage

```
cf_func(fn0, n = 100, xlim = c(0, 1), ylim = c(0, 1), xylim = NULL,
        batchmax = 1, out.col.name = NULL, out.name = NULL, pts = NULL, ...)
```

Arguments

| | |
|-------|---|
| fn0 | function to plot, first argument must be two-dimensional |
| n | number of points in each dimension |
| xlim | x limits for the contour plot |
| ylim | y limits for the contour plot |
| xylim | x and y limits for the contour plot, use when both are same #@param mainmin-max whether the min and max values should be shown in the title of plot |

| | |
|--------------|--|
| batchmax | number of datapoints that can be computed at a time |
| out.col.name | if a column needs to be selected from the function, specify it |
| out.name | Selects with a \$ the name from output to be used, for lists and data frames |
| | #@param pretitle Text to be preappended to end of plot title #@param posttitle Text to be appended to end of plot title #@param title Title for the plot #@param mainminmax_minmax Whether [min,max]= should be shown in title or just the numbers |
| pts | Points to plot on top of contour |
| ... | Passed to cf_grid |

References

[1] filled.contour R function, copied function but removed part for sidebar

[2] <http://stackoverflow.com/questions/16774928/removing-part-of-a-graphic-in-r>, answer by P La-pointe

Examples

```
cf_func(function(x){x[1]*x[2]})
cf_func(function(x)(exp(-(x[1]-.5)^2-5*(x[2]-.5)^2)))
cf_func(function(xx){exp(-sum((xx-.5)^2/.1))}, bar=TRUE)
cf_func(function(xx){exp(-sum((xx-.5)^2/.1))}, bar=TRUE, mainminmax=TRUE)
```

| | |
|---------|--|
| cf_grid | <i>Makes filled contour plot with an optional sidebar, essentially filled.contour function. This version uses the split.screen() function to add the sidebar if bar is TRUE. By default it won't show the bar but will show the min and max values in the plot title along with their colors. Using this function will make other functions such as points() called afterwards not put points where you expect. Pass anything you want added to the plot area to afterplotfunc as a function to get it to work properly.</i> |
|---------|--|

Description

Makes filled contour plot with an optional sidebar, essentially filled.contour function. This version uses the split.screen() function to add the sidebar if bar is TRUE. By default it won't show the bar but will show the min and max values in the plot title along with their colors. Using this function will make other functions such as points() called afterwards not put points where you expect. Pass anything you want added to the plot area to afterplotfunc as a function to get it to work properly.

Usage

```
cf_grid(x = seq(0, 1, length.out = nrow(z)), y = seq(0, 1, length.out =
  ncol(z)), z, xlim = range(x, finite = TRUE), ylim = range(y, finite =
  TRUE), zlim = range(z, finite = TRUE), levels = pretty(zlim, nlevels),
  nlevels = 20, color.palette = cm.colors,
  col = color.palette(length(levels) - 1), plot.title, plot.axes, key.title,
  key.axes, asp = NA, xaxs = "i", yaxs = "i", las = 1, axes = TRUE,
  frame.plot = axes, bar = F, pts = NULL, reset.par = TRUE,
  pretitle = "", posttitle = "", main = NULL, mainminmax = !bar,
  mainminmax_minmax = TRUE, afterplotfunc = NULL,
  cex.main = par()$cex.main, par.list = NULL, ...)
```

Arguments

| | |
|---------------|---|
| x | x values, must form grid with y. If not given, it is assumed to be from 0 to 1. |
| y | y values, must form grid with x. If not given, it is assumed to be from 0 to 1. |
| z | z values at grid locations |
| xlim | x limits for the plot. |
| ylim | y limits for the plot. |
| zlim | z limits for the plot. |
| levels | a set of levels which are used to partition the range of z. Must be strictly increasing (and finite). Areas with z values between consecutive levels are painted with the same color. |
| nlevels | if levels is not specified, the range of z, values is divided into approximately this many levels. |
| color.palette | a color palette function to be used to assign colors in the plot. |
| col | an explicit set of colors to be used in the plot. This argument overrides any palette function specification. There should be one less color than levels |
| plot.title | statements which add titles to the main plot. |
| plot.axes | statements which draw axes (and a box) on the main plot. This overrides the default axes. |
| key.title | statements which add titles for the plot key. |
| key.axes | statements which draw axes on the plot key. This overrides the default axis. |
| asp | the y/x aspect ratio, see plot.window. |
| xaxs | the x axis style. The default is to use internal labeling. |
| yaxs | the y axis style. The default is to use internal labeling. |
| las | the style of labeling to be used. The default is to use horizontal labeling. |
| axes | logical indicating if axes should be drawn, as in plot.default. |
| frame.plot | logical indicating if a box should be drawn, as in plot.default. |
| bar | Should a bar showing the output range and colors be shown on the right? |
| pts | Points to plot on top of contour |

| | |
|-------------------|---|
| reset.par | Should the graphical parameters be reset before exiting? Usually should be unless you need to add something to the plot afterwards and bar is TRUE. |
| pretitle | Text to be preappended to end of plot title |
| posttitle | Text to be appended to end of plot title |
| main | Title for the plot |
| mainminmax | whether the min and max values should be shown in the title of plot |
| mainminmax_minmax | Whether [min,max]= should be shown in title or just the numbers |
| afterplotfunc | Function to call after plotting, such as adding points or lines. |
| cex.main | The size of the main title. 1.2 is default. |
| par.list | List of options to pass to par |
| ... | additional graphical parameters, currently only passed to title(). |

References

- [1] filled.contour R function, copied function but removed part for sidebar
 [2] <http://stackoverflow.com/questions/16774928/removing-part-of-a-graphic-in-r>, answer by P La-pointe

Examples

```
x <- y <- seq(-4*pi, 4*pi, len = 27)
r <- sqrt(outer(x^2, y^2, "+"))
cf_grid(cos(r^2)*exp(-r/(2*pi)))
```

| | |
|----------|---|
| cf_grid1 | <i>Makes filled contour plot without sidebar, essentially filled.contour function. A sidebar can be added by setting bar to TRUE. However, this will cause any other plot functions used afterwards, such as points, to not put points at the correct spot.</i> |
|----------|---|

Description

Makes filled contour plot without sidebar, essentially filled.contour function. A sidebar can be added by setting bar to TRUE. However, this will cause any other plot functions used afterwards, such as points, to not put points at the correct spot.

Usage

```
cf_grid1(x = seq(0, 1, length.out = nrow(z)), y = seq(0, 1, length.out =
ncol(z)), z, xlim = range(x, finite = TRUE), ylim = range(y, finite =
TRUE), zlim = range(z, finite = TRUE), levels = pretty(zlim, nlevels),
nlevels = 20, color.palette = cm.colors,
col = color.palette(length(levels) - 1), plot.title, plot.axes, key.title,
key.axes, asp = NA, xaxs = "i", yaxs = "i", las = 1, axes = TRUE,
frame.plot = axes, bar = F, pts = NULL, reset.par = T, ...)
```

Arguments

| | |
|---------------|---|
| x | x values, must form grid with y |
| y | y values, must form grid with x |
| z | z values at grid locations |
| xlim | x limits for the plot. |
| ylim | y limits for the plot. |
| zlim | z limits for the plot. |
| levels | a set of levels which are used to partition the range of z. Must be strictly increasing (and finite). Areas with z values between consecutive levels are painted with the same color. |
| nlevels | if levels is not specified, the range of z, values is divided into approximately this many levels. |
| color.palette | a color palette function to be used to assign colors in the plot. |
| col | an explicit set of colors to be used in the plot. This argument overrides any palette function specification. There should be one less color than levels |
| plot.title | statements which add titles to the main plot. |
| plot.axes | statements which draw axes (and a box) on the main plot. This overrides the default axes. |
| key.title | statements which add titles for the plot key. |
| key.axes | statements which draw axes on the plot key. This overrides the default axis. |
| asp | the y/x aspect ratio, see plot.window. |
| xaxs | the x axis style. The default is to use internal labeling. |
| yaxs | the y axis style. The default is to use internal labeling. |
| las | the style of labeling to be used. The default is to use horizontal labeling. |
| axes | logical indicating if axes should be drawn, as in plot.default. |
| frame.plot | logical indicating if a box should be drawn, as in plot.default. |
| bar | Should a bar showing the output range and colors be shown on the right? |
| pts | Points to plot on top of contour |
| reset.par | Should the graphical parameters be reset before exiting? Usually should be. |
| ... | additional graphical parameters, currently only passed to title(). |

References

- [1] filled.contour R function, copied function but removed part for sidebar
- [2] <http://stackoverflow.com/questions/16774928/removing-part-of-a-graphic-in-r>, answer by P La-pointe

Examples

```
x <- y <- seq(-4*pi, 4*pi, len = 27)
r <- sqrt(outer(x^2, y^2, "+"))
cf_grid1(cos(r^2)*exp(-r/(2*pi)))
```

| | |
|-----|---|
| csa | <i>Closes the screens open if plotting with split.screen is interrupted. Happens often when there is a plotting error, then when you try to plot the next thing it gives an error. Running this function will reset. It just does close.screen(all.screens=TRUE) but is faster to type.</i> |
|-----|---|

Description

Closes the screens open if plotting with split.screen is interrupted. Happens often when there is a plotting error, then when you try to plot the next thing it gives an error. Running this function will reset. It just does close.screen(all.screens=TRUE) but is faster to type.

Usage

```
csa()
```

Examples

```
# Split screen into fourths
split.screen(c(2,2))
hist(rnorm(100))
screen(2)
hist(runif(100))
# Use csa() to go back to normal plotting
csa()
hist(rexp(100))
```

| | |
|------------------|---|
| multicolor.title | <i>Makes plot title using specified colors for the text</i> |
|------------------|---|

Description

Makes plot title using specified colors for the text

Usage

```
multicolor.title(main, col.main, collapse = "", cex.main = par()$cex.main)
```

Arguments

| | |
|----------|---|
| main | Text to put in main title of plot |
| col.main | Colors to use for the text |
| collapse | What to put between elements of main, defaults to "" but " " might be appropriate |
| cex.main | The size of the main title. 1.2 is default. |

Examples

```
plot(1:4)  
multicolor.title(c('Black, ', 'red, ', 'green'),c(1,2,3))
```

Index

`cf`, 2
`cf_data`, 2
`cf_func`, 3
`cf_grid`, 4
`cf_grid1`, 6
`csa`, 8

`multicolor.title`, 8