

# Package ‘distfree.cr’

June 15, 2018

**Type** Package

**Title** Distribution-Free Confidence Region

**Version** 1.5.1

**Date** 2018-06-13

**Author** Zhiqiu Hu, Rong-cai Yang

**Maintainer** Zhiqiu Hu <zhiqiu.hu@gmail.com>

**Description** Constructs confidence regions without the need to know the sampling distribution of bi-variate data. The method was proposed by Zhiqiu Hu & Rong-cai Yang (2013) <doi:10.1371/journal.pone.0081179.g001>.

**Depends** R (>= 2.10)

**License** GPL (>= 2)

**URL** <http://statgen.ualberta.ca>

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-06-15 14:21:16 UTC

## R topics documented:

distfree.cr-package . . . . .	2
distfree.cr . . . . .	2
plot.distfree.cr . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

distfree.cr-package     *Distribution-free confidence region (distfree.cr)*

---

### Description

The distfree.cr package was developed to implement a novel geometry-based method introduced by Zhiqiu Hu and Rong-cai Yang for constructing confidence regions without the need to know the sampling distribution of estimated parameters for two or more variables.

### Details

Package: distfree.cr  
Type: Package  
Version: 1.0  
Date: 2012-11-23  
License: GPL (>2.0)

### Author(s)

Zhiqiu Hu and Rong-cai Yang  
Maintainer: Zhiqiu Hu <zhiqiu.hu@gmail.com>

---

distfree.cr                     *distfree.cr*

---

### Description

Constructs empirical confidence regions for bivariate data based on the method proposed by Zhiqiu Hu and Rong-cai Yang(2013) <doi:10.1371/journal.pone.0081179.g001>.

### Usage

```
distfree.cr(x, y, alpha = 0.05, alpha.min.diff = 0.5/NROW(x), nknots = 40,  
          xlab = deparse(substitute(x)), ylab = deparse(substitute(y)),  
          col = c("red", "black", "gray"), draw = T)
```

### Arguments

x                     numeric vector, of dimensions *nobs* \* 1. If a data frame or a two-column numeric matrix of x and y is supplied here, the second option y of the function needs to be ignored.

<code>y</code>	numeric vector, of dimensions $nobs * 1$ . This option needs to be ignored if users provided both <code>x</code> and <code>y</code> in the first option of the function.
<code>alpha</code>	Significant level. By default <code>alpha</code> is set to be 0.05.
<code>alpha.min.diff</code>	minimum difference is allowed for calculating confidence region. This option is not suggested for most users. The default value is set to be $alpha/10$ .
<code>nknots</code>	number of knots that will be used to enclose the confidence region. The default value <code>nknots=40</code> is recommended for all users.
<code>xlab</code>	define the label of x axis of the plot.
<code>ylab</code>	define the label of y axis of the plot.
<code>col</code>	define colors of the scatter points and lines of the plot. The default setting <code>col=("red", "black", "gray")</code> are the colors for the lines enclosed the region, the points within the region and the points outside of the region, respectively.
<code>draw</code>	a logical indicator. Users may disable plotting by setting the option to <code>FALSE</code>

### Details

This function constructs a distribution-free confidence region based on the method proposed by Zhiqiu Hu and Rong-cai Yang.

### Value

<code>alpha.realized</code>	Realized-alpha, which is defined as the proportion of the total points outside the confidence region.
<code>polygon</code>	'data.frame' of x,y providing the apexes of the lines.
<code>polygon.smooth1</code>	'data.frame' of x,y providing the apexes of the smoothed polygon 1.
<code>polygon.smooth2</code>	'data.frame' of x,y providing the apexes of the smoothed polygon 2.
<code>data</code>	'data.frame', of dimension $nobs * 3$ , the first two columns are input data of x and y values and the third column <code>data\$pi</code> are indicators of whether the points are within (1) or outside (0) the confidence region.
<code>alpha,xlab,ylab,col</code>	values assigned by users.

### Note

A smooth confidence region can be achieved by setting up a big number for input variable `nknots`, and this in turn requires large sample sizes.

### Author(s)

Zhiqiu Hu and Rong-cai Yang

**Examples**

```
library(distfree.cr)
dat=data.frame(x=c(rnorm(3000), rnorm(3000, mean=1, sd=2.5)),
               y=c(rnorm(3000), rnorm(3000, mean=1, sd=2.5)))
pt=distfree.cr(dat, draw=TRUE, alpha=0.05)
pt=distfree.cr(x=dat$x, y=dat$y, draw=FALSE)
plot(pt)
```

---

*plot.distfree.cr*      *plot.distfree.cr*

---

**Description**

Plot an object that is returned by the `distfree.cr` function.

**Usage**

```
## S3 method for class 'distfree.cr'
plot(x, show.points = T, ...)
```

**Arguments**

<code>x</code>	An object returned by the <code>distfree.cr</code> function.
<code>show.points</code>	A logical indicator of whether or not the original data are plotted.
<code>...</code>	Other parameters that can be passed to the <code>plot</code> function.

# Index

\*Topic **package**

distfree.cr-package, [2](#)

distfree.cr, [2](#)

distfree.cr-package, [2](#)

plot.distfree.cr, [4](#)