

# Package ‘bdots’

September 26, 2015

**Type** Package

**Title** Bootstrapped Differences of Time Series

**Version** 0.1.2

**Date** 2015-09-25

**Author** Michael Seedorff, Jacob Oleson, Grant Brown, Joseph Cavanaugh, and Bob McMurray

**Maintainer** Michael Seedorff <michael-seedorff@uiowa.edu>

**Depends** nlme, mvtnorm

**Imports** doParallel, doRNG, foreach

**LazyData** FALSE

**Description**

Analyze differences among time series curves with Oleson et al's modified p-value technique.

**License** GPL (>= 3)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-09-26 01:22:08

## R topics documented:

Boostrap Step . . . . .	2
ci . . . . .	3
ests.plot . . . . .	4
Fitting Step . . . . .	5
Refitting Step . . . . .	6
replot . . . . .	7
subs.plot . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

**Description**

Bootstrap on the fitted parameters, plot the estimates, and highlight the significant regions

**Usage**

```
doubleGauss.boot(part1.list, seed = new.seed(), alpha = 0.05, paired = FALSE,
N.iter = 1000, cores = 1, p.adj = "oleson", test.spots = NULL,
time.test = NULL, test.params = FALSE)
logistic.boot(part1.list, seed = new.seed(), alpha = 0.05, paired = FALSE,
N.iter = 1000, cores = 1, p.adj = "oleson", test.spots = NULL,
time.test = NULL, test.params = FALSE)
```

**Arguments**

<code>part1.list</code>	list. Output from <code>doubleGauss.fit</code>
<code>seed</code>	integer. What to set seed at
<code>alpha</code>	numeric (Between 0 and 1). Probability of familywise Type I Error
<code>paired</code>	boolean. Whether the same subjects are in both data sets
<code>N.iter</code>	numeric (positive integer). Number of bootstrap iterations to run
<code>cores</code>	integer. Number of cores on the localhost to use
<code>p.adj</code>	Options: oleson, fdr, none
<code>test.spots</code>	numeric. Specify specific x-values for testing at
<code>time.test</code>	numeric. Specify individual time points to conduct t-tests at without any p-value correction
<code>test.params</code>	boolean. Whether to test for significant differences in group mean parameter estimates. Performs a 2-sample t-test with equal variance assumption if <code>paired = FALSE</code> . If <code>paired = TRUE</code> , performs a paired t-test.

**Details**

Bootstrap on the fitted parameters, plot the estimates, and highlight the significant regions

**Value**

List: for input in `replot`

**Note**

There are no further notes

## Examples

```
## Not run:
data(ci)
ci.1 <- subset(ci, ci$LookType == "Target")
ci.1$Group <- ci.1$protocol
out.1 <- logistic.fit(ci.1, 4)
out.2 <- logistic.boot(out.1)
replot(out.2, bucket.lim = c(0, 1))

ci.2 <- subset(ci, ci$LookType == "Cohort" | ci$LookType == "Unrelated")
ci.2$Group <- ci.2$protocol
ci.2$Curve <- ifelse(ci.2$LookType == "Cohort", 1, 2)
out.1 <- doubleGauss.fit(ci.2, 4, diffs = TRUE)
out.1 <- doubleGauss.refit(out.1, subj = c(13, 23), group = c(2, 2),
  curves = c(2, 2), cor = c(FALSE, FALSE))
out.2 <- doubleGauss.boot(out.1)
replot(out.2, ylim = c(-0.01, 0.1), bucket.lim = c(0, 0.08))

## End(Not run)
```

---

ci	<i>Eyetracking Data from Normal Hearing Individuals and those with Cochlear Implants</i>
----	--

---

## Description

This data set has stuff on CIs and NHs. Provide more info here.

## Usage

```
ci
```

## Format

A matrix of values

## References

Farris-Trimble, A., McMurray, B., Cigrand, N. and Tomblin, J.B. (2014) The process of spoken word recognition in the face of signal degradation: Cochlear implant users and normal-hearing listeners. *Journal of Experimental Psychology: Human Perception and Performance*, 40(1), 308-327

---

`ests.plot`*Plot Parameter Estimates*

---

**Description**

Plots of a histogram of the parameter estimates

**Usage**

```
ests.plot(part1.list)
```

**Arguments**

`part1.list`      Output from `doubleGauss.fit` or `logistic.fit`

**Details**

Plots of a histogram of the parameter estimates

**Value**

NULL

**Note**

There are no further notes

**Examples**

```
## Not run:
data(ci)
ci.1 <- subset(ci, ci$LookType == "Target")
ci.1$Group <- ci.1$protocol
out.1 <- logistic.fit(ci.1, 4)ests.plot(out.1)
ests.plot(out.1)

## End(Not run)
```

---

Fitting Step

*Fit Subjects Individual Curves*


---

**Description**

Fit Subjects from 2 groups with the 6-parameter Double Gaussian or the 4-parameter Logistic

**Usage**

```
doubleGauss.fit(data, col, concave = TRUE, diffs = FALSE,
rho.0 = 0.9, cor = TRUE, cores = 1)
logistic.fit(data, col, diffs = FALSE, rho.0 = 0.9, cor = TRUE, cores = 1)
```

**Arguments**

data	data.frame. A data.frame with the columns 'Subject', 'Time', and 'Group'. 'Subject' designates the subject number (numeric), 'Time' designates the time (numeric, should be ordered from low to high), and 'Group' designates the group (should be 2 unique groups)
col	numeric. The column in the data.frame that corresponds to the eyetracking
concave	boolean. TRUE indicates concave UP, FALSE indicates concave DOWN. Only for Double Gaussian.
diffs	boolean. If the each group is calculating the difference of 2 logistic curves, set to TRUE. In this case, there needs to be a numeric 'Curve' column (with only 1s and 2s) designating the secondary curve (2) to subtract from the primary curve (1).
rho.0	numeric (Between 0 and 1). Assumed autocorrelation of errors for individual subject's curve
cor	boolean. If TRUE assumes an AR1 autocorrelation structure among the residuals for fitting
cores	integer. Number of cores on the localhost to use

**Details**

Fit Subjects from 2 groups with the 6-parameter Double Gaussian or the 4-parameter Logistic

**Value**

List: for input into refit, boot, plot.ests, and plot.subjs functions

**Note**

There are no further notes

**Examples**

```
## Not run:
data(ci)
ci.1 <- subset(ci, ci$LookType == "Target")
ci.1$Group <- ci.1$protocol
out.1 <- logistic.fit(ci.1, 4)
out.2 <- logistic.boot(out.1)
replot(out.2, bucket.lim = c(0, 1))

ci.2 <- subset(ci, ci$LookType == "Cohort" | ci$LookType == "Unrelated")
ci.2$Group <- ci.2$protocol
ci.2$Curve <- ifelse(ci.2$LookType == "Cohort", 1, 2)
out.1 <- doubleGauss.fit(ci.2, 4, diffs = TRUE)
out.1 <- doubleGauss.refit(out.1, subj = c(13, 23), group = c(2, 2),
  curves = c(2, 2), cor = c(FALSE, FALSE))
out.2 <- doubleGauss.boot(out.1)
replot(out.2, ylim = c(-0.01, 0.1), bucket.lim = c(0, 0.08))

## End(Not run)
```

---

 Refitting Step

*Refit Subjects Individual Curves*


---

**Description**

Refit Subjects from 2 groups with the 6-parameter Double Gaussian or the 4-parameter Logistic.  
Can specify starting parameters

**Usage**

```
doubleGauss.refit(part1.list, subj, group, curves = NULL,
  params = NULL, cor=NULL, rho.0 = NULL)
logistic.refit(part1.list, subj, group, curves = NULL,
  params = NULL, cor = NULL, rho.0 = NULL)
```

**Arguments**

part1.list	Output from fitting step
subj	numeric vector. Subject numbers (within their group) that you want to refit
group	numeric vector. Group numbers corresponding to subject vector
curves	numeric vector. Curve numbers if it's a differences of logistics fit
params	list of numeric vectors (length 4 or 6). Parameter estimates for the 6 parameter DoubleGauss in the order mu, height, variance 1, variance 2, base 1, base 2. Parameter estimate for the 4 parameter Logistic in the order min, peak, slope, crossover
cor	logical vector. If TRUE assumes an correlation structure of AR(rho) and if FALSE assumes no correlation structure
rho.0	numeric. assumed autocorrelation of errors for subject's curve

**Details**

Refit Subjects from 2 groups with the 6-parameter Double Gaussian or the 4-parameter Logistic.  
Can specify starting parameters

**Value**

List

**Note**

There are no further notes

**Examples**

```
## Not run:
data(ci)
ci.2 <- subset(ci, ci$LookType == "Cohort" | ci$LookType == "Unrelated")
ci.2$Group <- ci.2$protocol
ci.2$Curve <- ifelse(ci.2$LookType == "Cohort", 1, 2)
out.1 <- doubleGauss.fit(ci.2, 4, diffs = TRUE)
out.1 <- doubleGauss.refit(out.1, subj = c(13, 23), group = c(2, 2),
  curves = c(2, 2), cor = c(FALSE, FALSE))
out.2 <- doubleGauss.boot(out.1)
replot(out.2, ylim = c(-0.01, 0.1), bucket.lim = c(0, 0.08))

## End(Not run)
```

---

replot

*Replot Bootstrapped Output*


---

**Description**

Plot the bootstrapped output with different parameters than the default ones.

**Usage**

```
replot(part2.list, xlim = NULL, ylim = c(0, 1), main = "Curve",
  legend.location = "topleft", bucket.lim = c(0, .9))
```

**Arguments**

part2.list	list. Output from doubleGauss.boot or logistic.boot
xlim	numeric vector (length = 2). Start and end point of x-axis. If NULL, takes the full time course
ylim	numeric vector (length = 2). Start and end point of y-axis
main	string. Title

legend.location      string. Location of the legend

bucket.lim          numeric vector (length = 2). How far the yellow significant region goes on the y axis

### Details

Plot the bootstrapped output with different parameters than the default ones.

### Value

NULL

### Note

Options for legend.location include "topleft", "top", "topright", "right", "bottomright", "bottom", "bottomleft", "left"

### Examples

```
## Not run:
data(ci)
ci.1 <- subset(ci, ci$LookType == "Target")
ci.1$Group <- ci.1$protocol
out.1 <- logistic.fit(ci.1, 4)
out.2 <- logistic.boot(out.1)
replot(out.2, bucket.lim = c(0, 1))

## End(Not run)
```

---

subs.plot

*Plot subjects raw data along with function fits*

---

### Description

Plot subjects raw data along with function fits

### Usage

```
subs.plot(part1.list, legend.spot = "topright")
```

### Arguments

part1.list          Output from doubleGauss.fit or logistic.fit

legend.spot        string. Location of the legend

### Details

Plot subjects raw data along with function fits



**Value**

NULL

**Note**

Options for legend.location include "topleft", "top", "topright", "right", "bottomright", "bottom", "bottomleft", "left"

**Examples**

```
## Not run:  
data(ci)  
ci.1 <- subset(ci, ci$LookType == "Target")  
ci.1$Group <- ci.1$protocol  
out.1 <- logistic.fit(ci.1, 4)ests.plot(out.1)  
subs.plot(out.1)  
  
## End(Not run)
```

# Index

\*Topic **datasets**

ci, 3

\*Topic **htest**

Bootstrap Step, 2

ests.plot, 4

Fitting Step, 5

Refitting Step, 6

replot, 7

subs.plot, 8

Bootstrap Step, 2

ci, 3

doubleGauss.boot (Bootstrap Step), 2

doubleGauss.fit (Fitting Step), 5

doubleGauss.refit (Refitting Step), 6

ests.plot, 4

Fitting Step, 5

logistic.boot (Bootstrap Step), 2

logistic.fit (Fitting Step), 5

logistic.refit (Refitting Step), 6

Refitting Step, 6

replot, 7

subs.plot, 8