Package 'fixedpointproperty'

June 13, 2023

Type Package

Title Determine and Test the Fixed-Point Property in Binary Mixture Data

Version 1.0

Date 2023-06-12

Description Determine and test the fixed-point property in binary mixture data. This package was originally developed in the context of detecting mixture of cognitive processing strategies, based on observed response time distributions. The method is explain in more detail by Van Maanen, De Jong, Van Rijn (2014) <doi:10.1371/journal.pone.0106113> and Van Maanen, Couto, Lebreton, (2016) <doi:10.1371/journal.pone.0167377>.

License GPL-2

Depends BayesFactor

Imports methods

NeedsCompilation no

Author Leendert Van Maanen [aut, cre], Joaquina Couto [ctb], Hedderik Van Rijn [ctb], Mael Lebreton [ctb]

Maintainer Leendert Van Maanen <l.vanmaanen@uu.nl>

Repository CRAN

Date/Publication 2023-06-13 08:10:05 UTC

R topics documented:

fpAnova *Function to perform ANOVA on fp objects*

Description

This function computes Bayes Factors and p-values for within-subjects ANOVA designs, encoded as fp objects.

Usage

```
fpAnova(object, stat = "BF", na.rm = TRUE, check = TRUE)
```

Arguments

`object`	a list of objects from class fpp.
`stat`	Either "BF" (default), "p", or "both", specificying what statistic to report.
`na.rm`	Are `NA`s removed?
`check`	Should the data be checked for suitability? A warning will be provided if a check is failed.

Details

The function expects the output of `fpGet`, but in a list.

Value

A list containing the results of either the Bayesian or frequentist analysis, or both:

`BF`	The output of `anovaBF`
`p`	The output of `summary.aov`

Warning

If `check=TRUE`, then warnings will be provided if the data are not suitable for correct inferences.

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

See Also

`fpGet`, `anovaBF`, `fpConditionCheck`, `summary.aov`.

Examples

```
## generate data
p <- c(.1,.5,.9)
rt <- sapply(1:3, function(i) {rnormMix(10000, c(1,2), c(1,1), p[i])})
dat <- data.frame(rt=c(rt), cond=rep(1:3, each=10000), pp=rep(1:50, each=200, times=3))

## compute the list of fpp objects
res <- tapply(1:nrow(dat), dat$pp, function(X) {fpGet(dat[X,], 1000, bw=.75)})

## call fpAnova, with stat="both" to do both a Bayesian and a frequentist test
fpAnova(res, stat="both")
```

`fpConditionCheck` *Checks preconditions before performing fpANOVA*

Description

This function checks whether two conditions are met before performing fpANOVA.

Usage

```
fpConditionCheck(object)
```

Arguments

`object`	a list of objects from class fpp.

Details

Finding support for the fixed-point property will be mute if there is no significant difference between experimental conditions. Whether all conditions differ can be tested using `fpConditionCheck1`, which performs pairwise t-tests. A warning is provided if at least one paire of conditions does not significantly differ (default settings of `pairwise.t.test` are used).

Finding support for the fixed-point property is difficult if the bandwidth of the density estimation is chosen too small. In that case, multiple crossing points of pairs of densities will preclude a precise estimate of the fixed point. `fpConditionCheck2` tests the number of crossing points for each pair of conditions, and provides a warning if more crossing points are detected.

Value

No return value, called for warnings generated by `fpConditionCheck1` and `fpConditionCheck2`

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

See Also

fpAnova, pairwise.t.test

Examples

```
N <- 200  # nr of observations per condition
M <- 50  # nr of participants
p <- seq(0.1, 0.9, 0.4)  # mixture proportions
means <- c(0.3, 0.3)  # means of base distributions are equal, yielding a warning if check=TRUE
sigma <- 5  # scale of base distributions

bw <- 0.01
# kernel bandwidth of the density estimation. Too small values yield a warning if check=TRUE

### generate data
rt <- NULL
for (i in 1:length(p)) {
    rt <- c(rt, ifelse(sample(0:1, N * M, replace = TRUE, prob = c(p[i], 1 - p[i])),
        rnorm(N * M, means[1], sigma), rnorm(N * M, means[2], sigma)))
}
rt <- rt + rep(rnorm(M, sd = 0.1), times = N)  # normally distributed pp random effect
dat <- data.frame(rt = rt, cond = rep(1:length(p), each = N * M), pp = rep(1:M,
    each = N))

### compute crossing points
res <- tapply(1:nrow(dat), dat$pp, function(X) {
    fpGet(dat[X, ], 1000, bw = bw)
})

### test fixed point
fpAnova(res, stat = "both", check=TRUE) # this provides both warnings
```

---

fpDensDiff *Function to compute the crossing point of two kernel-based distribution functions*

Description

This function takes a list of fpp objects, and computes at which point the distributions cross each other.

Usage

```
fpDensDiff(object)
```

Arguments

`object`	a list of `fpp` objects

Value

An m * n-array containing the crossing points of the pairs of distributions, with n the length of the list of `fpp` objects, and m the number of pairs of distributions.

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

See Also

`fpGet`,`fpAnova`

Examples

```
## generate data
p <- c(.1,.5,.9)
rt <- sapply(1:3, function(i) {rnormMix(10000, c(1,2), c(1,1), p[i])})
dat <- data.frame(rt=c(rt), cond=rep(1:3, each=10000), pp=rep(1:50, each=200, times=3))

## compute the list of fpp objects
res <- tapply(1:nrow(dat), dat$pp, function(X) {fpGet(dat[X,], 1000, bw=.75)})

## get the crossing points
crosses=fpDensDiff(res)
boxplot(t(crosses), frame.plot=FALSE,xlab="Crossing point", ylab="Condition pair",
  names=c("1-2","2-3","1-3"), horizontal=TRUE)
```

`fpGet` *Function to compute densities and density differences for three or more data sets*

Description

This function computes standard kernel-based density functions for a response time data set with three or more conditions. In addition, it computes the pairwise differences for each pair of density functions.

Usage

```
fpGet(dat, n = 512, bw = "nrd0")
```

Arguments

`dat`	n*2 dataframe or matrix with in col 1: RT (the values for which to compute the density); col 2: condition (an index)
`n`	the number of equally spaced points at which the density is to be estimated. See `density` for details.
`bw`	the smoothing bandwidth to be used. See `density` for details.

Value

an object of class `fpp`, with the following components.

`dens`	list of objects from class `density`
`diff`	dataframe of the density differences
`dat`	dataframe with the input data

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

See Also

`density`.

Examples

```
### one data set or participant
## generate data
p <- c(.1,.5,.9)
rt <- sapply(1:3, function(i) {rnormMix(1000, c(1,2), c(1,1), p[i])})
dat <- data.frame(rt=c(rt), cond=rep(1:3, each=1000))

## compute one fp object
fpobject <- fpGet(dat, 1000, bw=.75)


### multiple participants
## generate data
p <- c(.1,.5,.9)
rt <- sapply(1:3, function(i) {rnormMix(10000, c(1,2), c(1,1), p[i])})
dat <- data.frame(rt=c(rt), cond=rep(1:3, each=10000), pp=rep(1:50, each=200, times=3))

## compute the list of fpp objects
res <- tapply(1:nrow(dat), dat$pp, function(X) {fpGet(dat[X,], 1000, bw=.75)})
```

`fpp-class` *Class* `"fpp"`

Description

Objects of the fpp class are used by methods and functions of the `fp` package, to visualize and compute the fixed-point property in response time data.

Objects from the Class

Objects can be created by calls of the form `new("fpp", ...)`.

Slots

`dens`: Object of class `"array"`. This is an array of objects of class `"density"`.

`diff`: Object of class `"data.frame"`. This is a dataframe with the paired differences between densities.

`dat`: Object of class `"data.frame"`. This is a dataframe with the input data.

Methods

plot `signature(x = "fpp")`: ...

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

Examples

```
showClass("fpp")
```

---

normMix *Gaussian binary mixture distribution*

Description

Density, distribution, function, quantile function, and random generation for the mixture of two Gaussian distributions with mixture proportion `p` and `1-p`.

Usage

```
dnormMix(x, mean=c(0,1), sd=c(1,1), p=1)
pnormMix(x, mean=c(0,1), sd=c(1,1), p=1)
qnormMix(x, mean=c(0,1), sd=c(1,1), p=1)
rnormMix(n, mean=c(0,1), sd=c(1,1), p=1)
```

Arguments

`x`	vector of quantiles or probabilities.
`n`	number of observations.
`mean`	vector of two means.
`sd`	vector of standard deviations.
`p`	mixture proportion of the first distribution (the second has proportion *1-p*).

Value

`dnormMix` gives the density, `pnormMix` gives the distribution function, `qnormMix` gives the quantile function, and `rnormMix` generates random deviates.

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

Examples

```
## the default value for p reduces this function to a normal distribution
dnormMix(1) == dnorm(1)

## plot that illustrates the fixed-point property
p <- c(.8,.5,.2)
m <- c(0,1); s <- c(1,2)
plot(function(X) {dnormMix(X, m, s, p[1])}, -5,10, ylab="Density",xlab="x", bty='L')
for (i in 2:3) {
  plot(function(X) {dnormMix(X, m, s, p[i])},-5,10, add=TRUE, lty=i, lwd=2)
}
legend("topright", legend=paste("p=",p,sep=''), lty=1:3, lwd=2, bty='n')
```

`plot.fpp` *Function to plot the distributions and differences of multiple binary mixture distributions.*

Description

This function plots kernel-based densities as well as density differences for three or more data sets (fpp objects).

Usage

```
## S3 method for class 'fpp'
plot(x, ylab = c("Density", "Density difference"), xlim = NULL, ...)
```

Arguments

`x`	an fpp object
`ylab`	y-axis labels of the two plots
`xlim`	the x limits (x1, x2) of the two plots
`...`	additional arguments to pass to both `plot.density` and `matplot`

Details

Generates two plots, one showing the estimated densities (based on `plot.density`) and one showing the density differences (based on `matplot`)

Value

No return value, called for side effects

Author(s)

Leendert van Maanen (l.vanmaanen@uu.nl)

References

Van Maanen, L., De Jong, R., Van Rijn, H (2014). How to assess the existence of competing strategies in cognitive tasks: A primer on the fixed-point property. PLOS One, 9, e106113

Van Maanen, L. Couto, J. & Lebetron, M. (2016). Three boundary conditions for computing the fixed-point property in binary mixture data. PLOS One, 11, e0167377.

See Also

fpGet,fpDensDiff,plot.density,matplot

Examples

```
## generate data
p <- c(.1,.5,.9)
rt <- sapply(1:3, function(i) {rnormMix(1000, c(1,2), c(1,1), p[i])})
dat <- data.frame(rt=c(rt), cond=rep(1:3, each=1000))

## compute fpp object
fpobject <- fpGet(dat, 1000, bw=.75)

## plot it
op <- par(mfrow=c(1,2))
plot(fpobject)
par(op)
```

Index