Package 'permutes'

July 23, 2025

Title Permutation Tests for Time Series Data

Version 2.8

Author Cesko C. Voeten [aut, cre]

Maintainer Cesko C. Voeten <cvoeten@gmail.com>

Description Helps you determine the analysis window to use when analyzing densely-sampled
time-series data, such as EEG data, using permutation testing (Maris & Oostenveld, 2007)
<doi:10.1016/j.jneumeth.2007.03.024>. These permutation tests can help identify the timepoints
where significance of an effect begins and ends, and the results can be plotted in various
types of heatmap for reporting. Mixed-effects models are supported using an implementation of
the approach by Lee & Braun (2012) <doi:10.1111/j.1541-0420.2011.01675.x>.

Depends R (>= 2.10)

Imports plyr, stats, utils

Suggests buildmer (>= 2.3), car, doParallel, ggplot2, glmmTMB, knitr,
lme4, lmPerm, permuco, rmarkdown, viridis

License FreeBSD

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

BugReports https://gitlab.com/cvoeten/permutes/-/issues

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-28 14:00:02 UTC

Contents

clusterperm.glm	*Cluster-based permutation tests for time series data, based on generalized linear models or other* buildmer *models. This is an alias for* clusterperm.lm *provided for discoverability.*

Description

Cluster-based permutation tests for time series data, based on generalized linear models or other buildmer models. This is an alias for clusterperm.lm provided for discoverability.

Usage

```
clusterperm.glm(...)
```

Arguments

...   Arguments to be passed to clusterperm.lm.

See Also

clusterperm.lm, clusterperm.lmer

Examples

```
perms <- clusterperm.glm(Fz ~ Deviant * Session,data=MMN,series.var=~Time)
```

---

clusterperm.glmer	*Cluster-based permutation tests for time series data, based on generalized linear mixed-effects models or other* buildmer *models. This is an alias for* clusterperm.lmer *provided for discoverability.*

Description

Cluster-based permutation tests for time series data, based on generalized linear mixed-effects models or other buildmer models. This is an alias for clusterperm.lmer provided for discoverability.

Usage

```
clusterperm.glmer(...)
```

Arguments

...	Arguments to be passed to clusterperm.lmer.

See Also

clusterperm.lmer

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- perm.glmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),data=MMN)
# Testing a single EEG electrode, with random effects by participants, ANOVA inference
perms <- perm.glmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),data=MMN,type='anova')
```

---

clusterperm.glmmTMB	*Cluster-based permutation tests for time series data, based on* buildglmmTMB.

Description

Cluster-based permutation tests for time series data, based on buildglmmTMB.

Usage

```
clusterperm.glmmTMB(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
  series.var = ~0,
  buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE),
  nperm = 1000,
  type = "regression",
  parallel = FALSE,
  progress = "none"
)
```

Arguments

formula	A normal formula, possibly using `lme4`-style random effects. This can also be a buildmer terms object, provided dep is passed in `buildmerControl`. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.
data	The data.
family	The family.
weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
series.var	A one-sided formula giving the variable grouping the time series.
buildmerControl	
	Options overriding the defaults in `buildmerControl`.
nperm	The number of permutations.
type	A character string of either `'anova'` or `'regression'`. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
parallel	Whether to parallelize the permutation testing using plyr's `parallel` option. Needs some additional set-up; see the plyr documentation.
progress	A plyr `.progress` bar name, see the plyr documentation. If not `'none'` while `parallel=TRUE`, an ad-hoc solution will be used, which will be visible if the cluster nodes were created with `outfile=''`.

Details

If not explicitly overridden in the `buildmerControl` argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting `direction='order'` and `quiet=TRUE`. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the `w2` column will not be included in the resulting `permutes` object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. `attr(x,'perms')` `<- NULL`.

`clusterperm.glmmTMB` is much slower than `clusterperm.lmer`, but it is also more flexible, allowing for things like beta regression and zero-inflation.

Value

A data frame.

Examples

```
# buildglmmTMB is much slower than clusterperm.lmer

perms <- clusterperm.glmmTMB(Fz ~ Deviant * Session + (Deviant * Session | Subject),
data=MMN[MMN$Time > 150 & MMN$Time < 250,],series.var=~Time)
```

---

clusterperm.lm	*Cluster-based permutation tests for time series data, based on mixed-effects models or other* buildmer *models. This is an alias for* clusterperm.lmer*, except that random effects are explicily disallowed.*

Description

Cluster-based permutation tests for time series data, based on mixed-effects models or other `buildmer` models. This is an alias for `clusterperm.lmer`, except that random effects are explicitly disallowed.

Usage

```
clusterperm.lm(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
  series.var,
```

```
  buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE, ddf = "lme4"),
   nperm = 1000,
   type = "regression",
   parallel = FALSE,
   progress = "none"
)
```

Arguments

formula	A normal formula without random effects. This can also be a buildmer terms object, provided dep is passed in buildmerControl. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.
data	The data.
family	The family.
weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
series.var	A one-sided formula giving the variable grouping the time series.
buildmerControl	
	Options overriding the defaults in buildmerControl.
nperm	The number of permutations.
type	A character string of either 'anova' or 'regression'. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
parallel	Whether to parallelize the permutation testing using plyr's parallel option. Needs some additional set-up; see the plyr documentation.
progress	A plyr .progress bar name, see the plyr documentation. If not 'none' while parallel=TRUE, an ad-hoc solution will be used, which will be visible if the cluster nodes were created with outfile=''.

Details

If not explicitly overridden in the buildmerControl argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting direction='order' and quiet=TRUE. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the w2 column will not be included in the resulting permutes object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. attr(x,'perms') <- NULL.

Value

A data frame.

See Also

clusterperm.lmer

Examples

```
perms <- clusterperm.lm(Fz ~ Deviant * Session,data=MMN,series.var=~Time)
```

---

clusterperm.lmer	*Cluster-based permutation tests for time series data, based on mixed-effects models or other* buildmer *models.*

Description

Cluster-based permutation tests for time series data, based on mixed-effects models or other buildmer models.

Usage

```
clusterperm.lmer(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
  series.var = ~0,
 buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE, ddf = "lme4"),
  nperm = 1000,
  type = "regression",
  parallel = FALSE,
  progress = "none"
)
```

Arguments

formula	A normal formula, possibly using lme4-style random effects. This can also be a buildmer terms object, provided dep is passed in buildmerControl. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.
data	The data.
family	The family.

weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
series.var	A one-sided formula giving the variable grouping the time series.
buildmerControl	
	Options overriding the defaults in `buildmerControl`.
nperm	The number of permutations.
type	A character string of either `'anova'` or `'regression'`. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
parallel	Whether to parallelize the permutation testing using plyr's `parallel` option. Needs some additional set-up; see the plyr documentation.
progress	A plyr `.progress` bar name, see the plyr documentation. If not `'none'` while `parallel=TRUE`, an ad-hoc solution will be used, which will be visible if the cluster nodes were created with `outfile=''`.

Details

If not explicitly overridden in the `buildmerControl` argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting `direction='order'` and `quiet=TRUE`. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the w2 column will not be included in the resulting `permutes` object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. `attr(x,'perms')` `<- NULL`.

Value

A data frame.

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- clusterperm.lmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
data=MMN,series.var=~Time)
# Testing a single EEG electrode, with random effects by participants, ANOVA inference
perms <- clusterperm.lmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
data=MMN,series.var=~Time,type='anova')
```

MMN   *ERP data from Jager (in prep.). See the vignettes for details.*

Description

ERP data from Jager (in prep.). See the vignettes for details.

Usage

```
data(MMN)
```

Format

A standard data frame.

---

perm.glm   *A general permutation test for mixed-effects models or other* buildmer *models. This is an alias for* perm.lm *provided for discoverability.*

Description

A general permutation test for mixed-effects models or other buildmer models. This is an alias for perm.lm provided for discoverability.

Usage

```
perm.glm(...)
```

Arguments

```
...             Arguments to be passed to perm.lm.
```

See Also

perm.lm, perm.lmer

Examples

```
perms <- perm.glm(Fz ~ Deviant * Session,data=MMN[MMN$Time > 150 & MMN$Time < 250,])
```

---

perm.glmer   *A general permutation test for mixed-effects models or other* buildmer
                           models. This is an alias for perm.lmer *provided for discoverability.*

Description

A general permutation test for mixed-effects models or other buildmer models. This is an alias for
perm.lmer provided for discoverability.

Usage

```
perm.glmer(...)
```

Arguments

...   Arguments to be passed to perm.lmer.

See Also

perm.lmer

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- perm.glmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
                    data=MMN[MMN$Time > 150 & MMN$Time < 250,])
# Testing a single EEG electrode, with random effects by participants, ANOVA inference
perms <- perm.glmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
                    data=MMN[MMN$Time > 150 & MMN$Time < 250,],type='anova')
```

---

perm.glmmTMB   *A general permutation test for* glmmTMB *models.*

Description

A general permutation test for glmmTMB models.

Usage

```
perm.glmmTMB(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
 buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE, ddf = "lme4"),
  nperm = 1000,
  type = "regression",
  progress = TRUE
)
```

Arguments

formula	A normal formula, possibly using lme4-style random effects. This can also be a buildmer terms object, provided dep is passed in buildmerControl. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.
data	The data.
family	The family.
weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
buildmerControl	
	Options overriding the defaults in buildmerControl.
nperm	The number of permutations.
type	A character string of either 'anova' or 'regression'. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
progress	Logical indicating whether to print progress messages during the permutation testing.

Details

If not explicitly overridden in the buildmerControl argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting direction='order' and quiet=TRUE. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the w2 column will not be included in the resulting permutes object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute

a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. `attr(x,'perms')` `<- NULL`.

Value

A data frame.

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- perm.glmmTMB(Fz ~ Deviant * Session,data=MMN[MMN$Time > 150 & MMN$Time < 250,])
```

---

perm.lm	*A general permutation test for mixed-effects models or other* buildmer *models. This is an alias for* perm.lmer, *except that random effects are explicily disallowed.*

Description

A general permutation test for mixed-effects models or other buildmer models. This is an alias for perm.lmer, except that random effects are explicily disallowed.

Usage

```
perm.lm(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
  buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE, ddf = "lme4"),
  nperm = 1000,
  type = "regression",
  progress = TRUE
)
```

Arguments

formula   A normal formula, possibly using lme4-style random effects. This can also be a buildmer terms object, provided dep is passed in buildmerControl. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.

data   The data.

family	The family.
weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
buildmerControl	
	Options overriding the defaults in `buildmerControl`.
nperm	The number of permutations.
type	A character string of either `'anova'` or `'regression'`. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
progress	Logical indicating whether to print progress messages during the permutation testing.

Details

If not explicitly overridden in the `buildmerControl` argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting `direction='order'` and `quiet=TRUE`. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the `w2` column will not be included in the resulting `permutes` object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. `attr(x,'perms') <- NULL`.

Value

A data frame.

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- perm.lm(Fz ~ Deviant * Session,data=MMN[MMN$Time > 150 & MMN$Time < 250,])
# Testing a single EEG electrode, with random effects by participants, ANOVA inference
perms <- perm.lm(Fz ~ Deviant * Session,data=MMN[MMN$Time > 150 & MMN$Time < 250,],type='anova')
```

perm.lmer	*A general permutation test for mixed-effects models or other* buildmer *models.*

Description

A general permutation test for mixed-effects models or other buildmer models.

Usage

```
perm.lmer(
  formula,
  data = NULL,
  family = gaussian(),
  weights = NULL,
  offset = NULL,
 buildmerControl = list(direction = "order", crit = "LRT", quiet = TRUE, ddf = "lme4"),
  nperm = 1000,
  type = "regression",
  progress = TRUE
)
```

Arguments

formula	A normal formula, possibly using lme4-style random effects. This can also be a buildmer terms object, provided dep is passed in buildmerControl. Only a single response variable is supported. For binomial models, the cbind syntax is not supported; please convert your dependent variable to a proportion and use weights instead.
data	The data.
family	The family.
weights	An optional vector of prior case weights. This vector is evaluated in the calling environment, not in the formula or data frame!
offset	An optional vector containing a prior offset term. This vector is evaluated in the calling environment, not in the formula or data frame!
buildmerControl	
	Options overriding the defaults in buildmerControl.
nperm	The number of permutations.
type	A character string of either 'anova' or 'regression'. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
progress	Logical indicating whether to print progress messages during the permutation testing.

Details

If not explicitly overridden in the `buildmerControl` argument, the function changes a few buildmer defaults to make more sense for permutation testing. This includes setting `direction='order'` and `quiet=TRUE`. Set these options explicitly to override this behavior. Because model comparisons are used for efficient permutation testing, REML is not supported. Omega squared is not available for mixed-effects ANOVA; the w2 column will not be included in the resulting `permutes` object. The returned object provides the raw bootstrap samples in the pb attribute. These are used to compute a corrected p-value using Maris & Oostenveld's (2007) cluster mass statistic, but can also be used by the user to compute other such statistics. These obviously significantly increase the size of the R object; if this becomes a problem, there is no harm in deleting them using e.g. `attr(x,'perms')` `<- NULL`.

Value

A data frame.

Examples

```
# Testing a single EEG electrode, with random effects by participants
perms <- perm.lmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
                   data=MMN[MMN$Time > 150 & MMN$Time < 250,])
# Testing a single EEG electrode, with random effects by participants, ANOVA inference
perms <- perm.lmer(Fz ~ Deviant * Session + (Deviant * Session | Subject),
                   data=MMN[MMN$Time > 150 & MMN$Time < 250,],type='anova')
```

`permu.test`	*Permutation tests for time series data, based on classic linear regression or ANOVA This is a legacy function that is primarily provided for backward compatibility. You should probably use* `perm.lm` *or* `perm.lmer` *instead.* `permu.test` *is the only function to support multivariate responses, although multivariate data can trivially be recoded into a univariate model.* `permu.test` *does not support random effects or corrected p-values (e.g. the cluster mass test), which are supported by* `permutelm`*.*

Description

Permutation tests for time series data, based on classic linear regression or ANOVA This is a legacy function that is primarily provided for backward compatibility. You should probably use `perm.lm` or `perm.lmer` instead. `permu.test` is the only function to support multivariate responses, although multivariate data can trivially be recoded into a univariate model. `permu.test` does not support random effects or corrected p-values (e.g. the cluster mass test), which are supported by `permutelm`.

Usage

```
permu.test(
  formula,
  data,
  subset = NULL,
  type = "anova",
  parallel = FALSE,
  progress = "text",
  ...
)
```

Arguments

formula	A formula of the following form: outcome ~ predictors | series variable. Multivariate outcomes (e.g. 32 EEG electrodes) are supported; use cbind(Fp1,Fp2,etc) ~ predictors | series.
data	The dataset referencing these predictors.
subset	If specified, will only analyze the specified subset of the data.
type	A character string of either 'anova' or 'regression'. The former runs an analysis of variance and returns F-values and p-values based on the explained variance of each factor in the design. The latter runs a linear-regression analysis and returns t-values and p-values based on individual effects. When running ANOVA, it is advised to use orthogonal predictors, as type III sums of squares are used.
parallel	Whether to parallelize the permutation testing using plyr's parallel option. Needs some additional set-up; see the plyr documentation.
progress	A plyr .progress bar name, see the plyr documentation. Ignored if parallel=TRUE.
...	Other arguments to be passed to lmp/aovp

Value

A data frame.

See Also

clusterperm.lm, clusterperm.lmer

Examples

```
# EEG data example using the MMN dataset

# Run permutation tests on all electrodes and timepoints, reporting p-values for the three
# manipulated factors
perms <- permu.test(cbind(Fp1,AF3,F7,F3,FC1,FC5,C3,CP1,CP5,P7,P3,Pz,PO3,O1,Oz,O2,PO4,P4,
P8,CP6,CP2,C4,FC6,FC2,F4,F8,AF4,Fp2,Fz,Cz) ~ Deviant * Session | Time,data=MMN)

# Run the tests in parallel on two CPU threads
```

```
# first, set up the parallel backend
library(doParallel)
cl <- makeCluster(2)
registerDoParallel(cl)
perms <- permu.test(cbind(Fp1,AF3,F7,F3,FC1,FC5,C3,CP1,CP5,P7,P3,Pz,PO3,O1,Oz,O2,PO4,P4,
P8,CP6,CP2,C4,FC6,FC2,F4,F8,AF4,Fp2,Fz,Cz) ~ Deviant * Session | Time,data=MMN,
parallel=TRUE)
stopCluster(cl)

# Plot the results by F-value, removing points that were not significant in the
# permutation tests
plot(perms,sig='p')

# t-values instead of F-values
perms <- permu.test(cbind(Fp1,AF3,F7,F3,FC1,FC5,C3,CP1,CP5,P7,P3,Pz,PO3,O1,Oz,O2,PO4,P4,
P8,CP6,CP2,C4,FC6,FC2,F4,F8,AF4,Fp2,Fz,Cz) ~ Deviant * Session | Time,data=MMN,
type='regression')
```

---

plot.permutes   *Create a heatmap of the results of cluster-based permutation testing.*

Description

Create a heatmap of the results of cluster-based permutation testing.

Usage

```
## S3 method for class 'permutes'
plot(
  x,
  type = c("LRT", "F", "t", "beta", "w2", "cluster_mass"),
  breaks = NULL,
  sig = NULL,
  ...
)
```

Arguments

x   Output of permu.test or clusterperm.lmer or associated functions. You may
                want to subset it if you want to simulate zooming in.

type   The quantity to plot. For ANOVA, the options are 'LRT' (default; this is the
                permutation statistic, which is a likelihood-ratio-test statistic), 'F', 'p' (this is
                the permutation p-value of the permutation statistic), or 'w2' (omega squared;
                not available for mixed-effects models). For regression, the options are 't',
                'beta', or 'p'.

breaks   The granularity of the labels of the x axis. Pass 'unique(x[,2])' to get a tick for
                  every timepoint. Combine this trick with subsetting of your dataset, and perhaps
                  averaging over all your dependent variables, to 'zoom in' on your data to help
                  you determine precisely where significance begins and stops to occur.

sig   An optional column name. If specified, only cases with a value of <.05 in this
                  column will be plotted. NULL by default, but suggested to be set to `p.cluster_mass`
                  for easy visualization of the cluster-mass-test results, or `p` to directly use the
                  permutation p-values.

...   Other arguments, which will be ignored (the ellipsis is provided for consistency
                  with the generic plot() method).

Value

A ggplot2 object.

Index