

Package ‘mvctm’

August 1, 2018

Type Package

Title Multivariate Variance Components Tests for Multilevel Data

Version 1.2

Date 2018-08-01

Author Denis Larocque

Maintainer Denis Larocque <denis.larocque@hec.ca>

Depends R (>= 3.5.0)

Imports Formula, MNM, nlme, quantreg, Rfit, SpatialNP

Description Permutation tests for variance components for 2-level, 3-level and 4-level data with univariate or multivariate responses.

License GPL-3

NeedsCompilation no

Repository CRAN

RoxygenNote 6.1.0

Date/Publication 2018-08-01 19:00:03 UTC

R topics documented:

mvctm-package	2
mvctm	2
permcluster	5
toydata	7

Index	8
--------------	----------

mvctm-package

Multivariate Variance Components Tests for Multilevel Data

Description

Permutation tests for variance components for 2-level, 3-level and 4-level data with univariate or multivariate responses.

Details

Package: mvctm
Type: Package
Version: 1.1
Date: 2017-03-08

Author(s)

Denis Larocque

Maintainer: Denis Larocque <denis.larocque@hec.ca>

References

Larocque, D., Nevalainen, J. and Oja, H. (2018). Multivariate Variance Components Tests for Multilevel Data. Les Cahiers du GERAD G-2018-58.

mvctm

Multivariate Variance Components Tests for Multilevel Data

Description

This function performs a permutation test for a variance component for 2-level, 3-level or 4-level data. The response can be univariate or multivariate.

Usage

```
mvctm(fixed, cluster, data, leveltested, method = "ls", npermut = 1000,  
weight = "observation", affequiv = TRUE)
```

Arguments

<code>fixed</code>	An object of class "formula" describing the fixed effects part of the model using the variables in the data frame <code>data</code> .
<code>cluster</code>	A vector giving the name of the variables in the data frame <code>data</code> to specify the clustering configuration. The order is important. For 2-level data it is a vector of dimension 1 specifying the level 1 cluster. For 3-level data, it is a vector of dimension 2. The first element specifies the level 1 (outer) cluster and the second one specifies the level 2 (inner) cluster. For 4-level data, it is a vector of dimension 3. The first element specifies the level 1 (outer) cluster, the second one specifies the level 2 (middle) cluster, and the last one specifies the level 3 (inner) cluster.
<code>data</code>	A data frame containing the data.
<code>leveltested</code>	An integer giving the level to be tested. It must be 1 for 2-level data, 1 or 2 for 3-level data, and 1, 2 or 3 for 4-level data. It corresponds to the element in <code>cluster</code> .
<code>method</code>	The scores to be used. The four choices "ls", "mixed", "rank" and "sign" are available. The default is "ls". The choice "mixed" is only available for a univariate response.
<code>npermut</code>	The number of random permutation used to perform the test. The default is 1000.
<code>weight</code>	The weight function to be used. The three choices "pair", "observation" and "cluster" are available. The default is "observation".
<code>affequiv</code>	Whether or not we want to use the affine-equivariant version of the tests. This is only relevant for a multivariate response and <code>method="rank"</code> or <code>"sign"</code> . The default is TRUE.

Details

With `method="ls"`, the fixed effects are estimated by ordinary least-squares. Then the test is performed on the residuals from this fit. With `method="mixed"`, the fixed effects are estimated with a linear mixed model. Then the test is performed on the marginal (population) residuals from this fit. With `method="rank"`, a rank-based method is used to estimate the fixed effects. Then the test is performed on the ranks of the residuals from this fit. Finally, with `method="sign"`, a sign-based method is used to estimate the fixed effects. Then the test is performed on the signs of the residuals from this fit. For multivariate data, spatial ranks and signs are used.

With a univariate response, `method="sign"` is not recommended because the test might be liberal.

With, `weight="pair"`, observations in larger clusters at the level `leveltested` will have more weights. With, `weight="cluster"`, the same weight is given to each cluster at the level `leveltested`. As a compromise between these two, the default `weight="observation"` gives an equal weight to each individual observation, with respect to the clusters at level `leveltested`.

Value

A list with the following two elements:

<code>pvalue</code>	The p-value of the test.
<code>statistic</code>	The value of the test statistic computed on the original data.

Author(s)

Denis Larocque <denis.larocque@hec.ca>

References

Larocque, D., Nevalainen, J. and Oja, H. (2018). Multivariate Variance Components Tests for Multilevel Data. Les Cahiers du GERAD G-2018-58.

Examples

```

data(toydata)

# Bivariate 2-level model.
# Classroom as the clusters.
# Only an intercept is in the fixed part of the model.
# Test based on 200 permutations
mvctm(fixed=cbind(y1,y2)~1,cluster=c("classroom"),
data=toydata,leveltested=1,npermut=200)

# Same as above but The two covariates are in the fixed part of the model.
# Test based on 1000 permutations (default).
## Not run:
mvctm(fixed=cbind(y1,y2)~x1+x2,cluster=c("classroom"),
data=toydata,leveltested=1)

## End(Not run)

# Same as above but the rank scores are used.
## Not run:
mvctm(fixed=cbind(y1,y2)~x1+x2,cluster=c("classroom"),
data=toydata,leveltested=1, method="rank")

## End(Not run)

# Univariate 4-level model.
# Classrooms, nested within schools, nested within regions.
# The variance component at the region level is tested.
# The fixed effects are estimated with a linear mixed model.
## Not run:
mvctm(fixed=y1~x1+x2,cluster=c("region","school","classroom"),
data=toydata,leveltested=1,method="mixed")

## End(Not run)

# Same as above but the variance component at the school level is tested.
## Not run:
mvctm(fixed=y1~x1+x2,cluster=c("region","school","classroom"),
data=toydata,leveltested=2,method="mixed")

## End(Not run)

```

```

# Same as above but the variance component at the classroom level is tested.
## Not run:
mvctm(fixed=y1~x1+x2,cluster=c("region","school","classroom"),
data=toydata,leveltested=3,method="mixed")

## End(Not run)

# Univariate 3-level model.
# The variance component at the classroom level is tested.
# The fixed effects are removed with an M-estimator with the rlm function
#   in the MASS package.
# Then the residuals from this fit are used to perform the test.
# The ~0 in the formula tells mvctm to use mresid directly to perform
#   the test without any centering or transformation.
## Not run:
library("MASS")
toydata[, "mresid"]=rlm(y1~x1+x2,data=toydata)$residuals
mvctm(fixed=mresid~0,cluster=c("school","classroom"),
data=toydata,leveltested=2)

## End(Not run)

```

permcluster

Generates a Permuted Data Set

Description

This function generates a permuted data set that can be used to perform a permutation test for a variance component for 2-level, 3-level or 4-level data.

Usage

```
permcluster(cluster, data, leveltested)
```

Arguments

cluster	A vector giving the name of the variables in the data frame data to specify the clustering configuration. The order is important. For 2-level data it is a vector of dimension 1 specifying the level 1 cluster. For 3-level data, it is a vector of dimension 2. The first element specifies the level 1 (outer) cluster and the second one specifies the level 2 (inner) cluster. For 4-level data, it is a vector of dimension 3. The first element specifies the level 1 (outer) cluster, the second one specifies the level 2 (middle) cluster, and the last one specifies the level 3 (inner) cluster.
data	A data frame containing the data.
leveltested	An integer giving the level to be tested. It must be 1 for 2-level data, 1 or 2 for 3-level data, and 1, 2 or 3 for 4-level data. It corresponds to the element in cluster.

Details

This is a utility function that allows to perform a permutation test with another test statistic than the one in the function `mvctm`.

Value

A data frame that contains the original variables plus 1, 2 or 3 new columns. For 2-level data, a single new column called `clusperm1` contains the permuted indices for the level 1 cluster. For 3-level data, two new columns called `clusperm1` and `clusperm2` contain the permuted indices for the level 1 and level 2 clusters, respectively. For 4-level data, three new columns called `clusperm1`, `clusperm2` and `clusperm3` contain the permuted indices for the level 1, level 2, and level 3 clusters, respectively.

Author(s)

Denis Larocque <denis.larocque@hec.ca>

References

Larocque, D., Nevalainen, J. and Oja, H. (2018). Multivariate Variance Components Tests for Multilevel Data. Les Cahiers du GERAD G-2018-58.

Examples

```
data(toydata)

# generates a permuted data set with 2-level data

permcluster(cluster=c("classroom"),data=toydata,leveltested=1)

# generates a permuted data set with 4-level data to test for level 2

permcluster(cluster=c("region","school","classroom"),
data=toydata,leveltested=2)

# performing a permutation test with another
# test statistic might look like this

## Not run:
pval=0
# compute the statistic with the original data. Call in stato.
for (i in 1:npermut)
{
pdata=permcluster(cluster=c("classroom"),data=toydata,leveltested=1)
# compute the statistic with pdata using the new column
# clusperm1 instead of classroom as the cluster index.
# call it statp.
pval=pval+(statp>stato)
}
pval/npermut
```

```
## End(Not run)
```

toydata	<i>Artificial 4-level data set</i>
---------	------------------------------------

Description

A 4-level data set with two responses and two covariates to illustrate the package mvctm. The classroom is nested within the school which is nested within the region.

Usage

```
data(toydata)
```

Format

A data frame with 150 observations on the following 7 variables.

y1 The first response

y2 The second response

x1 The first covariate

x2 The second covariate

region The level 1 cluster (outer)

school The level 2 cluster (middle)

classroom The level 3 cluster (inner)

Examples

```
data(toydata)
```

Index

*Topic **datasets**

toydata, [7](#)

mvctm, [2](#)

mvctm-package, [2](#)

permcluster, [5](#)

toydata, [7](#)