Permutation tests for regression, ANOVA and comparison of signals : the permuco package

Jaromil Frossard
University of Geneva

Olivier Renaud
University of Geneva

Abstract

Recent methodological researches produced permutation methods to test parameters in presence of nuisance variables in linear models or repeated measures ANOVA. This methods are briefly described in this article. Permutation tests are particularely usefull for the multiple comparisons problem as used to test the effect of factors or variables on signals while controling the family-wise error rate (FWER). This article introduces the **permuco** package that allows several permutation methods as well as functions implementing those methods jointly with cluster-mass tests or threshold-free cluster enhancement (TFCE). The **permuco** package is designed, first, for univariate permutation tests with nuisance variables; and secondly, for comparing signals as required, for example, for the analysis of event-related potential (ERP) of experiments using electroencephalography (EEG). A tutorial for each of this cases is provided.

This articles has been submitted to peer-review journal.

Keywords: projections, EEG, ERP, TFCE, cluster-mass statistics, multiple comparisons.

1. Introduction

Permutation tests are exact for simple models like one-way ANOVA and t-test (Lehmann and Romano 2008, pp. 176-177). Moreover it has been shown that they have some robust propreties under non normality (Lehmann and Romano 2008). However they require the assumption of exchangeability under the null hypothesis and it is not fulfilled in a multifactorial setting. Several authors (Draper and Stoneman 1966; Freedman and Lane 1983; Manly 1991; Kennedy 1995; Huh and Jhun 2001, Dekker, Krackhardt, and Snijders (2007); Kherad Pajouh and Renaud 2010; ter Braak 1992) had proposed ways to handle those models and Winkler, Ridgway, Webster, Smith, and Nichols (2014) give a simple and unique notation to compare those different methods.

Repeated measures ANOVA including one or several within subject effects are the most widely used models in the field of psychology. In the simplest case of one random factor, an exact permutation procedure consists of restricted permutations within the subjects. In a more general case, permutations in repeated measures ANOVA violate the exchangeability assumption. In particular the random effects due to the subjects and its interaction(s) with fixed effects imply a complex structure for the covariance matrix of the observations which means that the second moment is not more conserved after permutations. Kherad-Pajouh and Renaud (2014) proposed several methods to handle those designs.

For linear model, permutation tests are useful when the assumption of normality is violated or when the sample size is to small to apply asymptotic theory. In addition they have been shown to be useful for controlling the family wise error rate (FWER) for multiple comparisons (Troendle 1995; Maris and Oostenveld 2007; Smith and Nichols 2009). Those methods have been successfully applied for comparing conditions in experimental design using functional magnetic resonance imaging (fMRI) or electroencephalography (EEG) because they use the spatial and/or temporal correlation of the data.

The aim of the present article is to provide an overview of the use of permutation tests in all the above settings and explains how it can be used in R (Chambers 2009). Note that the presentation and discussion of the available packages that handle permutation tests in related settings is deferred to section 5.1, when all the notion are introduced. Appendix A show the comparison of the relevant code and output. But first, Section 2 is focused on the fixed effect models. It explains the model used for ANOVA and regression and the various permutation methods proposed in the literature. The Section 3 introduces the methods for repeated measures ANOVA. The Section 4 explains the multiple comparisons procedures used for comparing signals between experimental conditions and how permutation tests are applied in this setting. The Section 5 explains programming details and some of the choices for the default setting in the **permuco** package. The Section 6 introduces real data analyses from a control trial in psychology and from an experiment in neurosciences using EEG and the code to obtain them.

2. The fixed effects model

2.1. Model and notation

For each hypothesis of interest, the fixed effects model (regression or ANOVA) can be written as :

$$y = D\eta + X\beta + \epsilon, \tag{1}$$

where $\underset{n\times 1}{y}$ is the response variable, $\left[\underset{n\times(p-q)}{D} \quad \underset{n\times q}{X} \right]$ is a design matrix split into the nuisance variables D and the variables of interest X associated with the tested hypothesis. We assume without loss of generality that D and X are full rank matrices that may be correlated. The parameters of the full model $\left[\underset{1\times(p-q)}{\eta^\top} \quad \underset{1\times q}{\beta^\top} \right]^\top$ are also split into the parameters associated to the nuisance variables η and the one associated to the interest variables β. ϵ is an error term that follows a distribution $(0, \sigma^2 I_n)$. In this models we are interested by testing the hypothesis :

$$H_0 : \beta = 0 \text{ vs. } H_1 : \beta \neq 0. \tag{2}$$

We will write a permutation of a vector v using Pv and the permutation of the rows of a matrix M using PM where P is a permutation matrix (Gentle 2007, pp. 66-67). The notation for the "hat" matrix of a design matrix M will be $H_M = M(M^\top M)^{-1}M^\top$ and for the "residuals"

Table 1: Permutation methods in the presence of nuisance variables. See text for explanations of the symbols.

`method`/Authors	y^*	D^*	X^*
`manly` (Manly 1991)	Py	D	X
`draper_stoneman` (Draper and Stoneman 1966)	y	D	PX
`dekker`(Dekker *et al.* 2007)	y	D	PR_DX
`kennedy` (Kennedy 1995)	$(PR_D)y$		R_DX
`huh_jhun` (Huh and Jhun 2001)	$(PV_D^\top R_D)y$		$V_D^\top R_DX$
`freedman_lane` (Freedman and Lane 1983)	$(H_D + PR_D)y$	D	X
`terBraak` (ter Braak 1992)	$(H_{X,D} + PR_{X,D})y$	D	X

matrix on the same design will be $R_M = I - M(M^\top M)^{-1}M^\top$ (Greene 2011, pp. 24-25). The full QR-decomposition is :

$$\underset{n\times n}{\begin{bmatrix} M & 0 \end{bmatrix}} = \begin{bmatrix} Q_M & V_M \end{bmatrix} \begin{bmatrix} U_M & 0 \\ 0 & 0 \end{bmatrix}, \tag{3}$$

with $\underset{n\times p}{Q_M}$, $\underset{n\times(n-p)}{V_M}$, two matrices representing together an orthogonal basis of $\mathbb{R}^n$ and $\underset{p\times p}{U_M}$ is interpreted as M in the subspace of Q_M. Then we write $H_M = Q_M Q_M^\top$ and $R_M = V_M V_M^\top$ (Seber and Lee 2012, pp. 340-341).

2.2. Permutation methods for linear model or factorial ANOVA

The permutation methods are function that transform the data and reduce the effect of the nuisances variables. We define them as functions that transform the data through a permutation $P \in \mathscr{P}$ where $\mathscr{P}$ is the set of all n_P distinct permutation matrices of same size. From the observed data $\{y, D, X\}$ we compute the set of permuted data $\{y^*, D^*, X^*\}$ that depend on the observed data, on a permutation matrix P and a permutation method.

The **permuco** provide several permutation methods that are presented in table 1 using a notation inspired by Winkler *et al.* (2014).

The `manly` method simply permute the response (this method is sometimes called raw permutation). `draper_stoneman` permute the design of interest, ignoring the correlation between D and X. The `dekker` method orthogonalizes X with respect to D before permuting the design of interest. The `kennedy` method orthogonalizes all the data with respect to the nuisance variables before permuting the response. The `huh_jhun` method is similar to `kennedy` but it apply a second transformation to the data to ensure exchangeability up to the second moment. The V_D matrix comes from the equation (3) and has a dimension of $n \times (n - (p - q))$. It implies that the P's matrices for the `huh_jhun` method have smaller dimensions. The `freedman_lane` method permutes the residuals of the smaller model and adds them to its fitted values. The `terBraak` is similar to `freedman_lane` but uses the full model. However it uses a different null hypothesis $H_0 : \beta = (X^\top R_D X)^{-1} X^\top R_D Y$, where the right part of the equation correspond to the estimated effect under the full model implicitly using pivotal assumptions. Note that the notation $R_{D,X}$ means that the residuals matrix is based on the concatenation of the matrices D and X. See section 5.2 for advises on the choice of the

method.

Using table 1 permutations can be computed under a specific method for several statistics. The **permuco** package gives the choice of a F statistic used in a marginal test (or type III sum of square) (Searle 2006, pp. 53-54) and a t statistic for a univariate $\underset{1\times1}{\beta}$. We write the F statistic as :

$$F = \frac{y^\top H_{R_D X} y}{y^\top R_{D,X} y} \frac{n-p}{p-q}. \tag{4}$$

And when $q = 1$, the t statistic is :

$$t_{St} = \frac{(X^\top R_D X)^{-1} X R_D y}{\sqrt{y^\top R_{D,X} y (X^\top R_D X)^{-1}}} \sqrt{n-p}, \tag{5}$$

where the numerator correspond to the estimate of β and can be simplified by a factor of $(X^\top R_D X)^{-1}$. Those two statistics can be applied to the original data leading to the value $t = t(y, D, X)$ and to the permuted data leading to the values $t^* = t(y^*, D^*, X^*)$. Then the permutation distribution called $\mathscr{T}$ is the set of t^* for all $P \in \mathscr{P}$. We define the p-value, $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(|t^*| \geq |t|\right)$, for a bilateral t-test, $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(t^* \geq t\right)$, for a unilateral right sided t-test or a F-test and finally $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(t^* \leq t\right)$, for a unilateral left sided t-test, where $I(\cdot)$ is the indicator function.

3. Repeated measures ANOVA

3.1. Model and notation

We define the repeated measures ANOVA in a mixed linear form :

$$y = D\eta + X\beta + E^0\kappa + Z^0\gamma + \epsilon \tag{6}$$

where $\underset{n\times1}{y}$ is the response, the fixed part of the design is split into the nuisance variables $\underset{n\times(p_1-q_1)}{D}$, and the variables of interest $\underset{n\times(p_1)}{X}$. The specificity of the model allows us to split the random part into $\underset{n\times(p_2^0-q_2^0)}{E^0}$ and $\underset{n\times q_2^0}{Z^0}$ with the random effects associated with D and X respectively (Kherad-Pajouh and Renaud 2014). The fixed parameters are $\begin{bmatrix} \underset{1\times(p_1-q_1)}{\eta^\top} & \underset{1\times q_1}{\beta^\top} \end{bmatrix}^\top$. The random part is $\begin{bmatrix} \underset{1\times(p_2^0-q_2^0)}{\kappa^\top} & \underset{1\times q_2^0}{\gamma^\top} \end{bmatrix}^\top \sim (0, \Omega)$ and $\epsilon \sim (0, \sigma^2 I)$. The matrices associated with the random effects E^0 and Z^0 can be computed :

$$E^0 = (D^{0\prime}_{within} * Z^{0\prime}_{\Delta})^\top \text{ and } Z^0 = (X^{0\prime}_{within} * Z^{0\prime}_{\Delta})^\top, \tag{7}$$

where D^0_{within} and X^0_{within} are overparametrized matrices and are associated to the within effects in the design D and X, Z^0_{Δ} is the overparametrized design matrix associated to the subjects and $*$ is the column-wise Khatri-Rao product (Khatri and Rao 1968). Since the

Table 2: Permutation methods in the presence of nuisance variables for repeated measures ANOVA.

`method`	y^*	D^*	X^*	E^*	Z^*
`Rd_keradPajouh_renaud` (R_D)	PR_Dy		R_DX		R_DZ
`Rde_keradPajouh_renaud` ($R_{D,E}$)	$PR_{D,E}y$		$R_{D,E}X$		$R_{D,E}Z$

matrices E^0 and Z^0 are overparametrized they are not useful to compute the correct sum of squares associate to random effects. We need to restrict them into the right dimensionality by applying:

$$E = R_{D,X}E^0 \text{ and } Z = R_{D,X}Z^0, \tag{8}$$

where the matrices E and Z are respectively of rank $p_2 - q_2$ and q_2 and are the ones used to compute the F statistic. For an hypothesis on the fixed effect in the model of equation (6), we are interested by testing:

$$H_0 : \beta = 0 \text{ vs. } H_1 : \beta \neq 0. \tag{9}$$

3.2. Permutation methods for repeated measures ANOVA

Similarly to the fixed effects model, we can test hypothesis using permutation methods (Kherad-Pajouh and Renaud 2014). The one that are implemented in the **permuco** package are given in the table 2. The two methods are based on a similar idea. By premultiplying the design and response variables by R_D or $R_{D,E}$, we orthogonalize the model to the nuisance variables. The procedure can be viewed as an extension of the `"kennedy"` procedure (see table 1) to repeated measures ANOVA.

The hypothesis in (9) is tested based on the conventional F statistic for repeated measures ANOVA :

$$F = \frac{y^\top H_{R_DX}y}{y^\top H_Z y}\frac{p_2}{p_1}. \tag{10}$$

Similarly to the fixed effects model we write the statistic as a function of the data $t = t(y, D, X, E, Z)$. Then the permuted statistic $t^* = t(y^*, D^*, X^*, E^*, Z^*)$ is a function of the permuted data under the chosen method. We define the p-value similarly to the fixed case.

4. Signal and multiple comparisons

In EEG data analysis we may be interested by testing the effect of a condition on the event-related potential (ERP). It is a common practice to test it at each time of the ERP. In that kind of experiment we are typically facing thousands of tests (e.g., one measure every $2ms$ over 2 seconds) and the basic multiple hypotheses corrections like Bonferroni are useless as their power is to low.

Troendle (1995) proposed a multiple comparisons method that take into account the correlation between the resampling data. This method do not use specifically the time-neighborhood informations of a signal but uses wisely the general correlation between the statistics and may be used in a more general settings.

Better known, the cluster-mass test (Maris and Oostenveld 2007) has shown to be powerful and controlling the family-wise error rate (FWER) in EEG data analysis. And recently using a similar idea, the threshold-free cluster-enhancement (TFCE) was developed for fMRI data (Smith and Nichols 2009) and EEG data (Pernet, Latinus, Nichols, and Rousselet 2014), but usually presented only for factor.

All those methods use permutations and are compatible with the methods explained in the table 1 and 2, as shown next.

4.1. Model and notation

We can construct a model at each time points $s \in \{1, \dots, k\}$ for the fixed effects design as:

$$y_s = D\eta_s + X\beta_s + \epsilon_s, \tag{11}$$

where each k model is the same as (1) if we define y_s as the response variable for each observation at time s and D, X is the design that is similar for the k tests. The aim is to test simultaneously all the hypotheses $H_0^s : \beta_s = 0$ vs. $H_1^s : \beta_s \neq 0$ for $s \in \{1, \dots, k\}$ while controlling for the FWER through the k tests.

Likewise, the random effects model is written :

$$y_s = D\eta_s + X\beta_s + E^0\kappa_s + Z^0\gamma_s + \epsilon_s, \tag{12}$$

where each k model is defined as in (6) and similarly we want to test the multiple hypotheses $H_0^s : \beta_s = 0$ vs. $H_1^s : \beta_s \neq 0$ for $s \in \{1, \dots, k\}$.

Given the notation introduced previously, the p-values of the fixed and the random effects model can be written using the same approach. For both models we can choose on of the permutation method presented in the tables 1 or 2 and compute the k observed statistics t_s, the k permutation distributions $\mathcal{T}_s$, and the k uncorrected p-values or use the procedures described below to take into account the FWER.

4.2. Troendle's step-wise resampling method

The method developed by Troendle (1995) take advantage of the form of the multivariate resampling distribution of the t_s^*. If we assume that t_s is distributed according to T_s then by ordering the observed statistics t_s we obtain $t_{(1)} \leq \cdots \leq t_{(s)} \leq \cdots \leq t_{(k)}$ with their corresponding k null hypotheses $H_{(1)} \leq \cdots \leq H_{(s)} \leq \cdots \leq H_{(k)}$. Then Troendle (1995) use the following arguments. First, for all s, controlling the FWER with $P_{H_{(1)},\dots,H_{(k)}}\left(\max_{i\in\{1,\dots,k\}} T_{(i)} \leq t_{(s)}\right) < \alpha_{FWER}$ is a conservative approach. Secondely, if we reject $H_{(k)}$ and want to test $H_{(k-1)}$, we can safely assume that $H_{(k)}$ is false for controlling the FWER. Either $H_{(k)}$ was true and we already made a type I error or was wrong and we can go as if $H_{(k)}$was absent. We can then update our decision rule for testing $H_{(k-1)}$ by $P_{H_{(1)},\dots,H_{(k-1)}}\left(\max_{i\in\{1,\dots,k-1\}} T_{(i)} \leq t_{(k-1)}\right) < \alpha_{FWER}$. We continue until the first non-significant result and declare all s with a smaller t statistic as non-significant.

Then the permuted sets $\mathscr{T}_s$ can be interpreted as a non-parametric distribution of the T_s and based on Troendle (1995), we can use the following algorithm to compute the corrected p-value :

Algorithm 1 Troendle corrected p-value

1: Order the k observed statistics t_s into $t_{(1)} \leq \cdots \leq t_{(s)} \leq \cdots \leq t_{(k)}$
2: **for** $i \in \{1, \dots k\}$ **do**
3: Define the null distribution $\mathscr{S}_{(k-i+1)}$ for $t_{(k-i+1)}$ by :
4: **for** `each` $P \in \mathscr{P}$ **do**
5: `Return` the maximum over the $k-i+1$ first values $t^*_{(s)}$ for $s \in \{1, \dots, k-i+1\}$
6: Define the corrected p-value $p_{(k-i+1)} = \frac{1}{n_P} \sum_{t^* \in \mathscr{S}_{(k-i+1)}} I\left(t^* \geq t_{(k-i+1)}\right)$
7: Control for a stepwise procedure by :
8: **if** $p_{(k-i+1)} < p_{(k-i+2)}$ **and** $i > 1$ **then** $p_{(k-i+1)} \coloneqq p_{(k-i+2)}$

4.3. Cluster-mass statistic

The method proposed by Maris and Oostenveld (2007) for EEG take advantage of the fact that the effect will appears into clusters of adjacent timeframes. Based on individual statistics, we find those clusters using a threshold τ. All the adjacent time points for which the statistics are above this threshold define a cluster C_i for $i \in [1, \dots, n_c]$, where n_c is the number of cluster founded in the k statistics. We assign to each time point in the same cluster C_i, the same cluster-mass statistics $m_i = f(C_i)$ where f is a function that aggregates the statistics of the whole cluster into a scalar; typically the sum of F statistics or the sum of squared of the t statistics. The cluster-mass null distribution $\mathscr{M}$ is computed by repeating the process described above for each permutation. The contribution of a permutation to the cluster-mass null distribution is the maximum over all cluster-masses for this permutation. This process is described in the algorithm 2.

Algorithm 2 Cluster-mass null distribution $\mathscr{M}$

1: **for** `each` $P \in \mathscr{P}$ **do**
2: Compute the k permuted statistics t^*_s for $s \in \{1, \dots, k\}$.
3: Compute n^*_c clusters C^*_i as the set of adjacent time points which statistic is above τ.
4: Compute the cluster-mass for each cluster $m^*_i = f(C^*_i)$
5: `Return` the maximum value over the n^*_c values m^*_i.

To test the significance of an observed cluster C_i, we compare its cluster-mass $m_i = f(C_i)$ with the cluster-mass null distribution $\mathscr{M}$. The p-value of the effect at each time of the cluster C_i is the p-value of its cluster $p_i = \frac{1}{n_P} \sum_{m^* \in \mathscr{M}} I(m^* \geq m_i)$.

In addition of the good properties of this procedure (Maris and Oostenveld 2007), this method makes sense for EEG data analysis because if a difference of cerebral activity is believed to happen at a time s for a given factor, it is very likely that the time $s+1$ (or $s-1$) will show this difference too.

4.4. Threshold-free cluster-enhancement

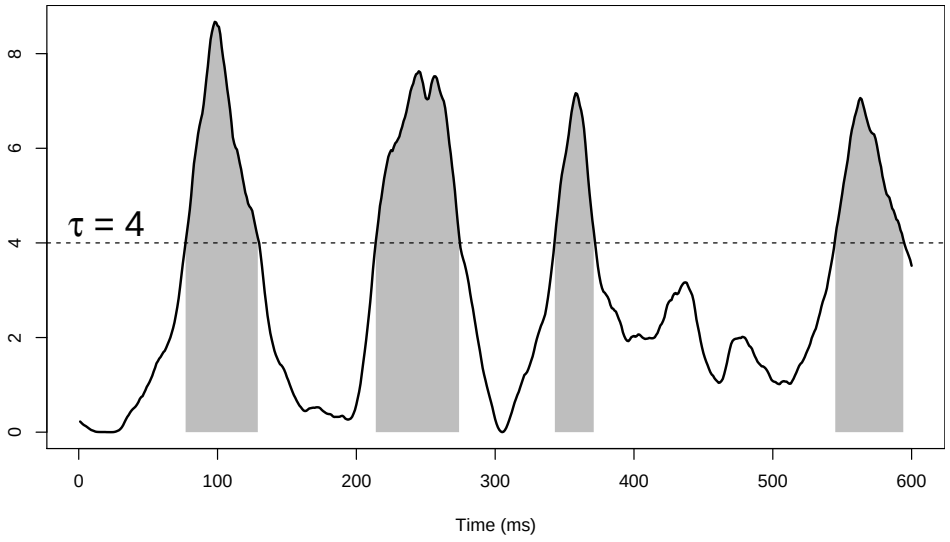

Figure 1: 4 clusters on 600 statistics using a threshold $\tau = 4$. Using the sum to aggregate the statistics, the grey areas underneath the curve represent the cluster-masses m_i.

Despite its advantages, the cluster-mass statistic is sensible to the choice of the threshold. The TFCE (Smith and Nichols 2009) is closely related to the cluster-mass but get rid of this seemingly arbitrary choice. It is defined at each time $s \in [1, \dots, k]$ for the statistics t_s as :

$$u_s = \int_{h=t_0}^{h=t_s} e(h)^E h^H dh, \tag{13}$$

where $e(h)$ is the extend at the height h and it is interpreted as the length of a cluster for a threshold of h. E and H are free parameters named the extend power, and the height power respectively. t_0 is set close to zero. The figure 2 illustrate how the TFCE statistics is computed for a given time point s.

We construct the TFCE null distribution $\mathscr{U}$ by applying the formula in (13) at each time-point of the permuted statistics t_s^* for $s \in \{1, \dots, k\}$ to produce for each permutation, k values u_s^*. Then the contribution of a permutation to $\mathscr{U}$ is the maximum over all k values u_s^*. Practically the integral in (13) is approximated numerically using small $dh \leq 0.1$, (Smith and Nichols 2009, Pernet *et al.* (2014)).

At the time s, the statistics t_s will be modified using the formula in (13). The formula can be viewed as a function of values in the grey area.

Algorithm 3 Threshold-free cluster-enhancement null distribution $\mathscr{U}$

1: **for** `each` $P \in \mathscr{P}$ **do**
2: Compute the k permuted statistics t_s^* for $s \in \{1, \dots, k\}$
3: Compute the k enhanced statistics u_s^* using a numerical approximation of (13)
4: `Return` the maximum over the k value u_s^*

To test the significance of a time point s we compare its enhanced statistics u_s with the

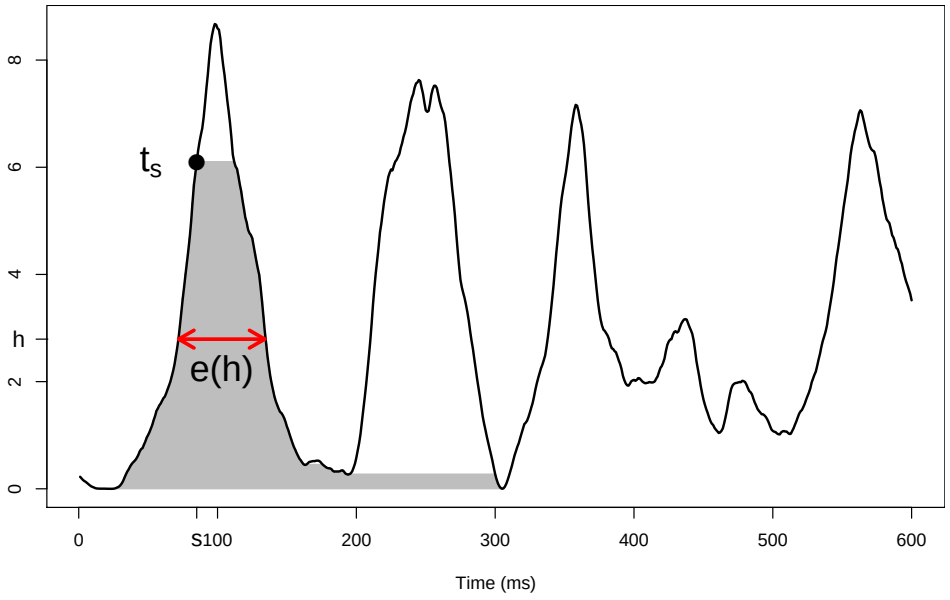

Figure 2: The TFCE transforms the statistic t_s using formula in (13). The extend $e(h)$, in red, is shown for a given height h. The TFCE statistics u_s at s can be viewed as a function of values in the grey area.

threshold-free cluster-enhancement null distribution $\mathscr{U}$. For a F-test we define the p-value as $p_s = \frac{1}{n_P} \sum_{u^* \in \mathscr{U}} I(u^* \geq u_s)$.

5. Comparison of packages, parameters choice and implementation details

5.1. Comparison of packages

Several packages for permutations tests are available for R in CRAN. Since permutation tests have such a variety of applications, we only review packages – or the part of packages – that handle regression, ANOVA or comparison of signals.

For testing one factor, the **perm** (Fay and Shaw 2010), **wPerm** (Weiss 2015) and **coin** (Hothorn, Hornik, Van De Wiel, Zeileis, and others 2008) packages produce permutation tests of differences of locations between two or several groups. The latter can also test the difference within groups or block, corresponding to a one within factor ANOVA.

The package **lmPerm** (Wheeler and Torchiano 2016) produces tests for factorial ANOVA and repeated measures ANOVA. It computes sequential (or Type I) and marginal (or Type III) tests for factorial ANOVA and ANCOVA but only the sequential is implemented with repeated measures, even after setting the parameter `seqs = FALSE`. The order of the factors will therefore matter in this case. The permutation method is to permute the raw data, irrespective of the presence of nuisance variables, which correspond to the `"manly"` method, see table 1. For repeated measure designs, data are first projected into the `"Error()"` strata and then permuted, a method that has not been validated (to our knowledge) in any peer-reviewed journal. Additionally, **lmPerm** by default uses a stopping rules based on current

p-value to define the number of permutations. By default, the permutations are not randomly sampled but modified sequentially merely on a pair of observations. This speeds up the code but the effect on the obtained p-value are not well documented.

The **flip** package (Finos, Basso, Solari, Goeman, and Rinaldo 2014) produces permutation and rotation tests (Langsrud 2005) for fixed effect and handle nuisance variables based on methods similar to `"huh_juhn"` in table 1. It performs tests in design with random effects only for singular models (e.g. repetition of measures by subjects in each condition) with method based on Basso and Finos (2012) and Finos and Basso (2014) to handle nuisance variables.

The **GFD** package Friedrich, Sarah, Konietschke, Frank, and Pauly, Markus (2017) produces marginal permutation tests for pure factorial design (without covariates) with a Wald-type statistic that take into account heteroscedasticity between groups. The permutation method is `"manly"`.

To our knowledge, only the **permuco** provide tests for comparison of signals.

The code and output for packages that perform ANOVA/ANCOVA are given in Appendix and in Appendix for repeated measures. For fixed effects, this illustrates that **permuco**, **flip** and **lmPerm** handle covariates and are based on the same statistic (F) whereas **GFD** uses the Wald-type statistic. It also shows that **flip** is testing one factor at a time (main effect of `sex` in this case) whereas the other packages produce directly tests for the all the effects. Also, the nuisance variables in **flip** must be carefully implemented using the appropriate coding variables in case of factors. Note that **lmPerm** centers the covariates using the default setting and that it can provide both marginal (Type III) or sequential (Type I) tests.

Concerning permutation methods, only the `"manly"` method is used for both **lmPerm** and **GFD**, the **flip** package uses the `"huh_jhun"` method, whereas multiple methods can be set by users using the **permuco** package. Note also that different default choices for the V matrix implemented in **flip** (based on eigen value decomposition) and **permuco** (based on QR decomposition) package do not allow to replicate identically the results (see table 1 for more informations on permutation method).

Finally, concerning repeated measures designs, the **flip** cannot handle cases where measures are not repeated in each condition for each subject, and therefore cannot be compared in Appendix . As already said, **lmPerm** produces sequential tests in repeated measure designs and **permuco** produces marginal tests. This explains why with unbalanced data, only the last interaction term in each strata produces the same statistic.

5.2. Permutation method

For the fixed effects model, simulations (Kherad Pajouh and Renaud 2010; Winkler *et al.* 2014) show that the method `freedman_lane`, `dekker`, `huh_jhun` and `terBraak` perform well, whereas `manly`, `draper_stoneman` and `kennedy` can be either liberal or conservative Moreover Kherad Pajouh and Renaud (2010) provide a proof for an exact test of the `huh_jhun` method under sphericity. However `huh_jhun` will reduce the dimensionality of the data and if $n - (p - q) \leq 7$ the number of permutations may be too low. Based on all the above literature the default method is set to `freedman_lane`.

For the random effects model, Kherad-Pajouh and Renaud (2014) shows that a more secure approach is to choose the `Rde_keradPajouh_renaud` method.

All $n!$ permutations are not feasible already for moderate sized datasets. A large subset of

permutation is used instead and it can be tuned with the `np` argument. The default value is `np = 5000`. Winkler, Ridgway, Douaud, Nichols, and Smith (2016) recall that with `np = 5000` the 0.95% confidence interval around $p = 0.05$ is relatively small : $[0.0443; 0.0564]$. For replicability purpose, the `P` argument can be used instead of the `np` argument. The `P` argument needs a `Pmat` object which stores the permutations. For small datasets, if the `np` argument is greater than the number of possible permutation ($n!$), the tests will be done on all permutations. This can be also be selected manually by setting `type = "unique"` in the `Pmat` functions.

Given the inequality sign in the formulas for the p-value described at the end of section 2.2, the minimal p-value is $1/$ `np`. Moreover this implies that the sum of the two unilateral p-values is slightly greater than 1.

The `huh_jhun` method is based on a random rotation that can be set by a random $n \times n$ matrix in the `rnd_rotation` argument. This random matrix will be orthogonalized by a QR decomposition to produce the proper rotation. Note that the random rotation in the `huh_jhun` method allows us to test the intercept, which is not available for the other methods.

5.3. Multiple comparisons

The `multcomp` argument can be set to `"bonferroni"` for the Bonferroni correction, to `"holm"` for the Holm correction (Holm 1979), `"benjamini_hocheberg"` for the Benjamini-Hochenberg method (Benjamini and Hochberg 1995), to `"troendle"`, see chapter 4.2, to `"clustermass"`, see chapter 4.3 and to `"tfce"`, see chapter 4.4. Those 6 methods are only available for the p-value computed by permutation in the **permuco** package. The first 3 methods are general procedure that could also be used in a parametric setting and the 3 lasts need resampling techniques.

For the `"clustermass"` method, the `threshold` parameter of the cluster-mass statistic is usually chosen by default at the 0.95 quantile of the statistics to match the univariate parametric significance; but the FWER is preserved for any a priori value of the `threshold` that the user may set. The mass function is specified by the `aggr_FUN` argument. It is set by default to the sum of squares for a t statistic and the sum for a F. It should be a function that returns a positive scalar which will be large for uncommon event under the null hypothesis (e.g., use the sum of absolute value of t statistics instead of the sum). It can be tuned depending on the expected signal. For the t statistic, typically, the sum of squares will detect more efficiently high peaks and the sum of absolute values will detect more efficiently the longer clusters.

For the `"tfce"` method, the default value for the extend is `E` $= 0.5$ and for the height is `H` $= 2$ for t test and, for F test, it is `E` $= 0.5$ and `H` $= 1$ following the recommendations of Smith and Nichols (2009) and Pernet *et al.* (2014). The `ndh` parameter controls the number of terms used in the approximation of the integral in (13) and is set to 500 by default.

The argument `return_distribution` is set by default to `FALSE` but can be set to `TRUE` to return the large matrices ($n_P \times k$) with the value of the permuted statistics.

The algorithm and formula presented in the previous sections may not be efficient for very large size of data. The permuted statistics are computed through QR decomposition using the `qr`, `qr.fitted`, `qr.resid` or `qr.coef` functions.

6. Tutorial

To load the **permuco** package :

```
R> install.packages("permuco")
R> library("permuco")
```

6.1. Fixed effects model

The `emergencycost` dataset contains information from 176 patients from an emergency service (Heritier, Cantoni, Copt, and Victoria-Feser 2009). The variables are the sex, the age (in years), the type of insurance (private/semiprivate or public), the length of the stay (`LOS`) and the cost. These observational data allow us to test which variables influence the cost of the stay of the patients. In this example we will investigate the effect of the sex and of the type of insurance on the cost and we will control those effects by the length of the stay. In this setting we perform an ANCOVA and must first center the covariate.

```
R> emergencycost$LOSc <- scale(emergencycost$LOS, scale = F)
```

The permutation tests can be assessed with the `aovperm` function. The `np` argument allows us to set the number of permutations. We choose to set a high number of permutations (`np = 100000`) to reduce the variablity of the permutation p-values so that they can safely be compared to the parametric ones. The `aovperm` function automatically convert the coding of factors with the `contr.sum` which allows us to test the main effect of factors and their interactions.

```
R> mod_cost_0 <- aovperm(cost ~ LOSc * sex * insurance, data = emergencycost,
+                        np = 100000)
R> mod_cost_0

Anova Table
Permutation test using freedman_lane to handle nuisance variables and
 1e+05 permutations.

                          SS  df        F parametric P(>F)
LOSc               2.162e+09   1 483.4422           0.0000
sex                1.463e+07   1   3.2714           0.0723
insurance          6.184e+05   1   0.1383           0.7105
LOSc:sex           8.241e+06   1   1.8427           0.1765
LOSc:insurance     2.911e+07   1   6.5084           0.0116
sex:insurance      1.239e+05   1   0.0277           0.8680
LOSc:sex:insurance 1.346e+07   1   3.0091           0.0846
Residuals          7.514e+08 168
                   permutation P(>F)
LOSc                         0.0000
sex                          0.0763
```

```
insurance                    0.6794
LOSc:sex                     0.1576
LOSc:insurance               0.0233
sex:insurance                0.8537
LOSc:sex:insurance           0.0847
Residuals
```

The interaction LOSc:insurance is significant both using the parametric p-value 0.0116 and the permutation one 0.0233. The difference between the 2 p-values is 0.0117 which is high enough to lo led to different conclusions (e.g., in case of correction for multiple tests or a smaller α level).

If we are interested in difference between the groups for a high value of the covariate, we center the covariate to the third quantile (14 days) and re-run the analysis.

```
R> emergencycost$LOS14 <- emergencycost$LOS - 14
R> mod_cost_14 <- aovperm(cost ~ LOS14 * sex * insurance, data = emergencycost,
+                         np = 100000)
R> mod_cost_14

Anova Table
Permutation test using freedman_lane to handle nuisance variables and
 1e+05 permutations.

                          SS  df        F parametric P(>F)
LOS14               2.162e+09   1 483.4422           0.0000
sex                 2.760e+07   1   6.1703           0.0140
insurance           9.864e+05   1   0.2206           0.6392
LOS14:sex           8.241e+06   1   1.8427           0.1765
LOS14:insurance     2.911e+07   1   6.5084           0.0116
sex:insurance       7.722e+05   1   0.1727           0.6783
LOS14:sex:insurance 1.346e+07   1   3.0091           0.0846
Residuals           7.514e+08 168
                    permutation P(>F)
LOS14                      0.0000
sex                        0.0224
insurance                  0.6082
LOS14:sex                  0.1576
LOS14:insurance            0.0233
sex:insurance              0.6540
LOS14:sex:insurance        0.0847
Residuals
```

For a long length of stay, the effect of sex is significant using the parametric p-value p = 0.014 and the permutation one p = 0.0224.

If the researcher has an a priori oriented alternative hypothesis $H_A : \beta_{sex=M} > \beta_{sex=F}$, the `lmperm` function produces unilateral t test. To run the same models as previously, we first need

to set the coding of the factors with the `contr.sum` function before running the permutation tests.

```
R> contrasts(emergencycost$insurance) <- contr.sum
R> contrasts(emergencycost$sex) <- contr.sum
R> contrasts(emergencycost$insurance)

             [,1]
public          1
semi_private   -1

R> contrasts(emergencycost$sex)

  [,1]
F    1
M   -1

R> modlm_cost_14 <- lmperm(cost ~ LOS14 * sex * insurance,
+                          data = emergencycost, np = 100000)
R> modlm_cost_14

Table of marginal t-test of the betas
Permutation test using freedman_lane to handle nuisance variables and
 100000 permutations.

                     Estimate Std. Error t value parametric Pr(>|t|)
(Intercept)           14217.0     360.17 39.4730              0.0000
LOS14                   845.5      38.45 21.9873              0.0000
sex1                   -894.7     360.17 -2.4840              0.0140
insurance1              169.1     360.17  0.4696              0.6392
LOS14:sex1              -52.2      38.45 -1.3575              0.1765
LOS14:insurance1         98.1      38.45  2.5512              0.0116
sex1:insurance1        -149.7     360.17 -0.4155              0.6783
LOS14:sex1:insurance1   -66.7      38.45 -1.7347              0.0846
                     permutation Pr(<t) permutation Pr(>t)
(Intercept)
LOS14                           1.0000             0.0000
sex1                            0.0152             0.9848
insurance1                      0.6823             0.3177
LOS14:sex1                      0.0796             0.9204
LOS14:insurance1                0.9868             0.0132
sex1:insurance1                 0.3337             0.6663
LOS14:sex1:insurance1           0.0395             0.9605
                     permutation Pr(>|t|)
(Intercept)
LOS14                              0.0000
```

```
sex1                          0.0224
insurance1                    0.6082
LOS14:sex1                    0.1576
LOS14:insurance1              0.0233
sex1:insurance1               0.6540
LOS14:sex1:insurance1         0.0847
```

The effect sex1 is significant for both the parametric unilateral p-value p = 0.007 and the permutation unilateral p-value p = 0.0152. Which indicate that when the length of the stay is high, men have a positive influence on the cost in comparison to women.

To test the effect of the sex within the public insured persons (called simple effect), we code the factors using the `contr.treatment` function and use the argument `coding_sum = FALSE` to disable the recoding of factors.

```
R> contrasts(emergencycost$insurance) <- contr.treatment
R> contrasts(emergencycost$sex) <- contr.sum
R> emergencycost$insurance <- relevel(emergencycost$insurance, ref = "public")
R> contrasts(emergencycost$insurance)

            semi_private
public                 0
semi_private           1

R> contrasts(emergencycost$sex)

  [,1]
F    1
M   -1

R> mod_cost_se <- aovperm(cost ~ LOSc * sex * insurance, data = emergencycost,
+                         np = 100000, coding_sum = FALSE)
R> mod_cost_se

Anova Table
Permutation test using freedman_lane to handle nuisance variables and
 1e+05 permutations.

                          SS  df         F parametric P(>F)
LOSc               9.512e+09   1 2126.7539           0.0000
sex                6.092e+07   1   13.6210           0.0003
insurance          6.184e+05   1    0.1383           0.7105
LOSc:sex           1.510e+08   1   33.7708           0.0000
LOSc:insurance     2.911e+07   1    6.5084           0.0116
sex:insurance      1.239e+05   1    0.0277           0.8680
LOSc:sex:insurance 1.346e+07   1    3.0091           0.0846
Residuals          7.514e+08 168
```

```
                  permutation P(>F)
LOSc                        0.0000
sex                         0.0004
insurance                   0.6794
LOSc:sex                    0.0000
LOSc:insurance              0.0233
sex:insurance               0.8537
LOSc:sex:insurance          0.0847
Residuals
```

The sex row can be interpreted as the effect of sex for the `public` insured persons for an average length of stay. Both the parametric p = 0.0003 and permutation p-value $p = 0.0004$ show significant effect of sex within the public insured persons.

Given the skewness of the data for each case where the permutation test differs from the parametric result, we tend to put more faith on the permutation result since it does not rely on assumption of normality.

6.2. Repeated measures ANCOVA

The `jpah2016` dataset contains a subset of a control trial in impulsive approach tendencies toward physical activity or sedentary behaviors. It contains several predictors like, the body mass index, the age, the sex, and the experimental conditions. For the latter, the subjects were asked to perform different tasks : to approach physical activity and avoid sedentary behavior (`ApSB_AvPA`), to approach sedentary behavior and avoid physical activity (`ApPA_AvSB`) and a control task. The dependent variables are measures of impulsive approach toward physical activity (`iapa`) or sedentary behavior (`iasb`). See Cheval, Sarrazin, Pelletier, and Friese (2016) for details on the experiment. We will analyze here only a part of the data.

```
R> jpah2016$bmic <- scale(jpah2016$bmi, scale = F)
```

We perform the permutation tests by running the `aovperm` function. The within subject factor should be written using `+ Error(...)` similarly to the `aov` function from the **stats** package :

```
R> mod_jpah2016 <- aovperm(iapa ~ bmic * condition * time + Error(id/(time)),
+                          data = jpah2016, method = "Rd_kheradPajouh_renaud")
```

The results are shown in an ANOVA table by printing the object :

```
R> mod_jpah2016
```

```
Permutation test using Rd_kheradPajouh_renaud to handle nuisance
variables and 5000 permutations.

                          SSn dfn      SSd dfd       MSEn      MSEd
bmic                  18.6817   1 106883.5  13    18.6817  8221.808
```

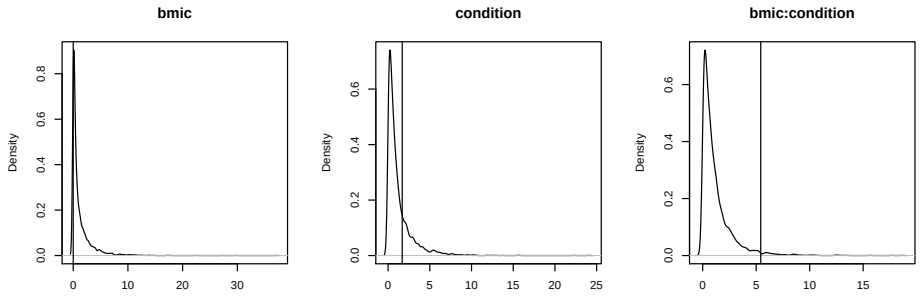

Figure 3: The permutation distributions of the F statistics for the effects `bmic`, `condition` and `bmic:condition`. The vertical lines indicate the observed statistics.

```
condition           27878.1976   2 106883.5  13 13939.0988  8221.808
bmic:condition      89238.4780   2 106883.5  13 44619.2390  8221.808
time                  268.8368   1 167304.9  13   268.8368 12869.607
bmic:time             366.4888   1 167304.9  13   366.4888 12869.607
condition:time      21159.7735   2 167304.9  13 10579.8867 12869.607
bmic:condition:time 29145.7201   2 167304.9  13 14572.8601 12869.607
                         F parametric P(>F) permutation P(>F)
bmic                0.0023           0.9627            0.9602
condition           1.6954           0.2217            0.2282
bmic:condition      5.4269           0.0193            0.0216
time                0.0209           0.8873            0.8806
bmic:time           0.0285           0.8686            0.8714
condition:time      0.8221           0.4611            0.4526
bmic:condition:time 1.1323           0.3521            0.3438
```

This analysis reveals a significant p-value for the effect of the interaction `bmic:condition` with a statistic $F = 5.4269$, which led to a permutation p-value $p = 0.0216$ not far for the parametric one. For this example, the permutation tests backs the parametric analysis. The permutation distributions can be viewed using the `plot` function like in figure 3.

```
R> plot(mod_jpah2016, effect = c("bmic", "condition", "bmic:condition"))
```

6.3. EEG experiment in attention shifting

The **permuco** package provides the sets `attentionshifting_signal` and `attentionshifting_design`. It comes from an EEG recording of an experiment using 15 participants watching images of either neutral or angry faces (Tipura, Renaud, and Pegna 2017). Those faces were shown at a different visibility : subliminal (`16ms`) and supraliminal (`166ms`) and were displayed to the left or to the right of a screen. The recording is at 1024Hz for 800ms. Time 0 is when the image appears (event-related potential or ERP). The `attentionshifting_signal` dataset contains the ERP of the electrode O1. The design of experiment is given in the

Table 3: Variables in the `attentionshifting_design` dataset

Variable name	Description	Levels
`id`	number of identification	15 subjects
`visibility`	time that the image is shown	`16ms` `166ms`
`emotion`	emotion of the shown faces	`angry`, `neutral`
`direction`	position of the faces on the screen	`left`, `right`
`laterality_id`	measure of the laterality of the subjects	scale from 25 to 100
`age`	age of the subjects	from 18 to 25
`sex`	sex of the subjects	`male`, `female`
`STAIS_state`	state anxiety score of the subjects	
`STAIS_trait`	trait anxiety score of the subjects	

`attentionshifting_design` dataset along with the laterality, sex, age, and 2 measures of anxiety of each subjects.

This experiment is designed for a repeated measures ANOVA. Using the **permuco** package, we can test each time points of the ERP for the main effects and the interactions of the variables `visibility`, `emotion` and `direction`. We perform F tests using a threshold at the 95% quantile, the sum as a cluster-mass statistics and 5000 permutations. We handle nuisance variables with the method `Rd_kheradPajouh_renaud` :

```
R> electrod_O1 <-
+    clusterlm(attentionshifting_signal ~ visibility * emotion * direction
+              + Error(id/(visibility * emotion * direction)),
+              data = attentionshifting_design)
```

The `plot` method produced a graphical representation of the tests that allows us to see quickly the significant time frames corrected by `clustermass`. The results are shown in the figure 4.

```
R> plot(electrod_O1)
```

Only one significant result appears for the main effect of visibility. This cluster is corrected using the `clustermass` method. Printing the `clusterlm` object gives more information about all clusters for the main effect of visibility, whether they are significant or not :

```
R> print(electrod_O1, effect = "visibility")
```

```
Cluster fisher test using Rd_kheradPajouh_renaud to handle nuisance variables
 with 5000 permutations and the sum as mass function.

Alternative Hypothesis : bilateral.

visibility, threshold = 4.60011.
```

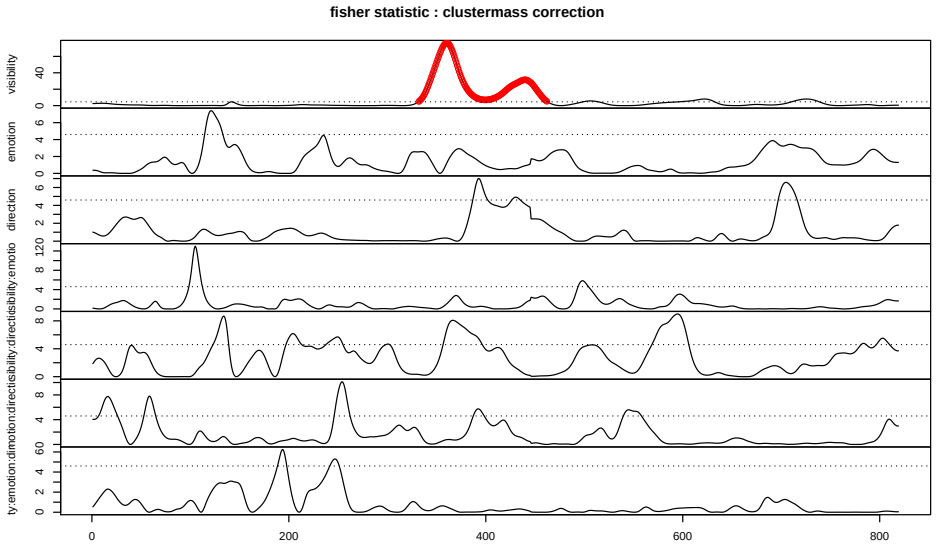

Figure 4: The `plot` method on a `clusterlm` object displays the observed statistics of the three main effects and their interactions. One cluster is significant for the main effect of visibility using the `clustermass` method, as shown by the red part. The `print` method will specified the details.

```
  start end cluster mass P(>mass)
1   142 142     4.634852   0.5048
2   332 462  3559.149739   0.0018
3   499 514    85.019645   0.4060
4   596 632   234.877913   0.2290
5   711 738   191.576178   0.2680
```

The only significant effect appears between the measures 332 and 462 that correspond to the 123.7ms and 250.9ms after the event. The cluster-mass statistic is 3559.1 with a p-value of 0.0018. The threshold is set to 4.60011 which is the 95% quantile of the F statistic. If we want to use other multiple comparisons procedures, we use `multcomp` argument :

```
R> full_electrod_O1 <-
+    clusterlm(attentionshifting_signal ~ visibility * emotion * direction
+              + Error(id/(visibility * emotion * direction)),
+              data = attentionshifting_design, P = electrod_O1[["P"]]
+              method = "Rde_kheradPajouh_renaud",
+              multcomp = c("troendle", "tfce", "clustermass",
+                           "bonferroni", "holm", "benjaminin_hochberg"))
```

Note that we retrieve the exact permutation from the previous model usint the `P` argument. The computation time for those tests is reasonable: it takes less than 12 minutes on a desktop computer (i7 3770CPU 3.4GHz, 8Go RAM) to compute the 7 permutation tests with all the

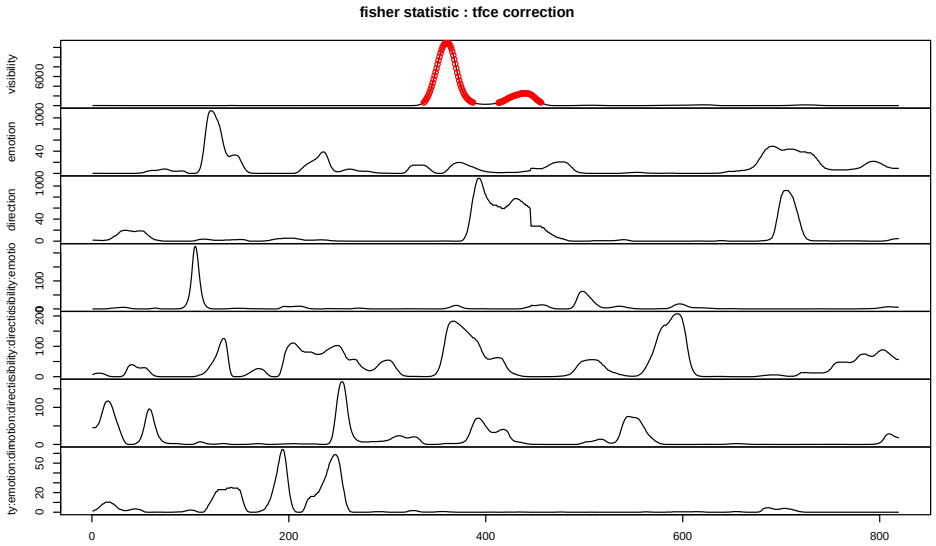

Figure 5: Setting the `multcomp` to `"tfce"` in the `plot` function will display the TFCE p-values. The argument `enhanced_stat = TRUE` shows the TFCE statistics u_s of equation (13)

multiple comparison procedures available. To see quickly the results of the threshold-free cluster enhancement-procedure, we set the `multcomp` argument of `plot` to `"tfce"` as shown in figure 5.

```
R> plot(full_electrod_O1, multcomp = "tfce", enhanced_stat = TRUE)
```

This procedure gets approximately a similar significant period for the same effect. However we get two smaller and separated significant periods rather than one longer. If the lines in the plot showing the TFCE statistics happen to show some small steps (which is not the case in 5) it may be because of a too small number of terms in the approximation of the integral of the `tfce` statistics of equation (13). In that case it would be reasonable to increase the value of the parameter `ndh`.

7. Conclusion

This article presents recent methodological advances in permutations tests and their implementation in the **permco** package. Hypotheses in linear models framework or repeated measures ANOVA can be tested using several methods to handle nuisance variables using **permuco**. Moreover permutations tests can solve the multiple comparisons problem and control the FWER trough cluster-mass tests or TFCE, and the `clusterlm` function is implements those procedures for analysis of signal like EEG data. Section 6 shows readers some real data example of tests that can be performed for regression, repeated measures ANCOVA and ERP signal comparison.

We hope that further developments of **permuco** expand cluster-mass tests to multidimensional adjacency (space and time) to handle full scalp ERP tests that control the FWER over all electrodes. Another evolution will concern permutation procedures for mixed effects models to allows researchers to performs tests in models containing subject and item specific random effects.

A. Comparisons of existing packages

A.1. ANOVA and ANCOVA

```
R> install.packages("lmPerm")
R> install.packages("flip")
R> install.packages("GFD")
R> library("lmPerm")
R> library("flip")
R> library("GFD")

R> emergencycost$LOSc <- scale(emergencycost$LOS, scale = FALSE)
R> contrasts(emergencycost$sex) <- contr.sum
R> contrasts(emergencycost$insurance) <- contr.sum
R>
R> X <- model.matrix( ~ sex+insurance, data = emergencycost)[, -1]
R> colnames(X) <- c("sex_num", "insurance_num")
R> emergencycost <- data.frame(emergencycost, X)
R>
R> anova_permuco <- aovperm(cost ~ sex * insurance, data = emergencycost)
R> anova_GFD <- GFD(cost ~ sex * insurance, data = emergencycost,
+                   CI.method = "perm", nperm  = 5000)
R>
R> ancova_permuco <- aovperm(cost ~ LOSc * sex * insurance, data = emergencycost,
+                            method = "huh_jhun")
R> ancova_flip <- flip(cost ~1, X = ~ sex_num, Z = ~ LOSc * insurance_num * sex_num
+                      - sex_num, data = emergencycost, statTest = "ANOVA",
+                      perms = 5000)
R> ancova_lmPerm <- aovp(cost ~ LOS * sex * insurance, data = emergencycost,
+                        seqs = FALSE, nCycle = 1)

R> anova_permuco

Anova Table
Permutation test using freedman_lane to handle nuisance variables and
 5000 permutations.

                      SS  df      F parametric P(>F)
sex             60470803   1 0.7193           0.3975
```

```
insurance       598973609   1 7.1249           0.0083
sex:insurance   334349436   1 3.9771           0.0477
Residuals     14459666504 172
              permutation P(>F)
sex                        0.3978
insurance                  0.0120
sex:insurance              0.0508
Residuals
```

R> anova_GFD

```
Call:
cost ~ sex * insurance

Wald-Type Statistic (WTS):
              Test statistic df    p-value p-value WTPS
sex                ⌋0.6397413  1 0.42380448       0.4662
insurance          6.3367469  1 0.01182616       0.0584
sex:insurance      3.5371972  1 0.06000678       0.0730

ANOVA-Type Statistic (ATS):
              Test statistic df1      df2   p-value
sex                ⌋0.6397413   1 5.743756 0.4556003
insurance          6.3367469   1 5.743756 0.0471947
sex:insurance      3.5371972   1 5.743756 0.1112178
```

R> ancova_permuco

```
Anova Table
Permutation test using huh_jhun to handle nuisance variables and
 5000, 5000, 5000, 5000, 5000, 5000, 5000 permutations.

                          SS  df        F parametric P(>F)
LOSc               2162110751   1 483.4422           0.0000
sex                  14630732   1   3.2714           0.0723
insurance              618366   1   0.1383           0.7105
LOSc:sex              8241073   1   1.8427           0.1765
LOSc:insurance       29107536   1   6.5084           0.0116
sex:insurance          123892   1   0.0277           0.8680
LOSc:sex:insurance   13457877   1   3.0091           0.0846
Residuals           751350616 168
                   permutation P(>F)
LOSc                         0.0002
sex                          0.0736
insurance                    0.7224
LOSc:sex                     0.1756
```

```
LOSc:insurance                0.0102
sex:insurance                 0.8704
LOSc:sex:insurance            0.0820
Residuals
```

```
R> summary(ancova_lmPerm)
```

```
Component 1 :
                    Df   R Sum Sq  R Mean Sq Iter            Pr(Prob)
LOS                  1 2162110751 2162110751 5000 <0.0000000000000002
sex                  1   14630732   14630732 4159               0.0236
LOS:sex              1    8241073    8241073 1525               0.0616
insurance            1     618366     618366   94               0.5213
LOS:insurance        1   29107536   29107536 5000               0.0010
sex:insurance        1     123892     123892   80               0.5625
LOS:sex:insurance    1   13457877   13457877 2238               0.0429
Residuals          168  751350616    4472325

LOS                ***
sex                *
LOS:sex            .
insurance
LOS:insurance      ***
sex:insurance
LOS:sex:insurance  *
Residuals
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R> ancova_flip
```

```
     Test  Stat tail p-value
cost    F 3.271    >  0.0724
```

A.2. Repeated measures ANOVA

```
R> jpah2016$id <- as.factor(jpah2016$id)
R> jpah2016$bmic <- scale(jpah2016$bmi,scale = FALSE)
R> contrasts(jpah2016$time) <- contr.sum
R> contrasts(jpah2016$condition) <- contr.sum
R>
R> rancova_permuco <- aovperm(iapa ~ bmic * condition * time + Error(id/(time)),
+                             data = jpah2016)
R> rancova_lmPerm <- aovp(iapa ~ bmic * condition * time + Error(id/(time)),
+                         data = jpah2016, nCycle = 1, seqs = FALSE)
```

```
R> rancova_permuco

Permutation test using Rd_kheradPajouh_renaud to handle nuisance
variables and 5000 permutations.

                          SSn dfn      SSd dfd       MSEn      MSEd
bmic                  18.6817   1 106883.5  13    18.6817  8221.808
condition          27878.1976   2 106883.5  13 13939.0988  8221.808
bmic:condition     89238.4780   2 106883.5  13 44619.2390  8221.808
time                 268.8368   1 167304.9  13   268.8368 12869.607
bmic:time            366.4888   1 167304.9  13   366.4888 12869.607
condition:time     21159.7735   2 167304.9  13 10579.8867 12869.607
bmic:condition:time 29145.7201   2 167304.9  13 14572.8601 12869.607
                       F parametric P(>F) permutation P(>F)
bmic                0.0023           0.9627            0.9660
condition           1.6954           0.2217            0.2180
bmic:condition      5.4269           0.0193            0.0248
time                0.0209           0.8873            0.8856
bmic:time           0.0285           0.8686            0.8666
condition:time      0.8221           0.4611            0.4392
bmic:condition:time 1.1323           0.3521            0.3528

R> summary(rancova_lmPerm)


Error: id
Component 1 :
              Df R Sum Sq R Mean Sq Iter Pr(Prob)
bmic           1     3270      3270   51   0.8824
condition      2    20000     10000  801   0.3009
bmic:condition 2    89238     44619 4863   0.0255 *
Residuals     13   106884      8222
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Error: id:time
Component 1 :
                   Df R Sum Sq R Mean Sq Iter Pr(Prob)
time                1     1047    1047.4   51   0.9412
bmic:time           1       31      31.5   51   0.8039
condition:time      2    29793   14896.4  320   0.3875
bmic:condition:time 2    29146   14572.9  419   0.3914
Residuals          13   167305   12869.6
```

References

Basso D, Finos L (2012). "Exact Multivariate Permutation Tests for Fixed Effects in Mixed-Models." *Communications in Statistics - Theory and Methods*, **41**(16-17), 2991–3001. doi:10.1080/03610926.2011.627103.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing." *Journal of the royal statistical society. Series B (Methodological)*, pp. 289–300.

Chambers JM (2009). *Software for Data Analysis: Programming with R*. 1st ed. 2008. corr. 2nd printing 2009 edition edition. Springer-Verlag, New York. ISBN 978-0-387-75935-7.

Cheval B, Sarrazin P, Pelletier L, Friese M (2016). "Effect of Retraining Approach-Avoidance Tendencies on an Exercise Task: A Randomized Controlled Trial." *Journal of Physical Activity and Health*, **13**(12), 1396–1403. doi:10.1123/jpah.2015-0597.

Dekker D, Krackhardt D, Snijders TAB (2007). "Sensitivity of MRQAP Tests to Collinearity and Autocorrelation Conditions." *Psychometrika*, **72**(4), 563–581. doi:10.1007/s11336-007-9016-1.

Draper NR, Stoneman DM (1966). "Testing for the Inclusion of Variables in Linear Regression by a Randomisation Technique." *Technometrics*, **8**(4), 695. doi:10.2307/1266641.

Fay MP, Shaw PA (2010). "Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: the **interval** R Package." *Journal of statistical software*, **36**(2).

Finos L, Basso D (2014). "Permutation tests for between-unit fixed effects in multivariate generalized linear mixed models." *Statistics and Computing*, **24**(6), 941–952. doi:10.1007/s11222-013-9412-6.

Finos L, Basso wcbD, Solari A, Goeman J, Rinaldo M (2014). ***flip**: Multivariate Permutation Tests*. R package version 2.4.3, URL https://CRAN.R-project.org/package=flip.

Freedman D, Lane D (1983). "A Nonstochastic Interpretation of Reported Significance Levels." *Journal of Business & Economic Statistics*, **1**(4), 292. doi:10.2307/1391660.

Friedrich, Sarah, Konietschke, Frank, Pauly, Markus (2017). "**GFD**: An R Package for the Analysis of General Factorial Designs." *Journal of Statistical Software*, **79**(1). doi:10.18637/jss.v079.c01.

Gentle J (2007). *Matrix Algebra : Theory, Computations, and Applications in Statistics*. Springer-Verlag.

Greene W (2011). *Econometric Analysis*. 7 edition edition. Prentice Hall, Boston.

Heritier S, Cantoni E, Copt S, Victoria-Feser MP (2009). *Robust Methods in Biostatistics*. John Wiley & Sons.

Holm S (1979). "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics*, **6**(2), 65–70.

Hothorn T, Hornik K, Van De Wiel MA, Zeileis A, others (2008). "Implementing a class of permutation tests: the **coin** package." *Journal of Statistical Software*, **28**(8).

Huh MH, Jhun M (2001). "Random Permutation Testing in Multiple Linear Regression." *Communications in Statistics - Theory and Methods*, **30**(10), 2023–2032. doi:10.1081/STA-100106060.

Kennedy PE (1995). "Randomization Tests in Econometrics." *Journal of Business & Economic Statistics*, **13**(1), 85. doi:10.2307/1392523.

Khatri CG, Rao CR (1968). "Solutions to some Functional Equations and their Applications to Characterization of Probability Distributions." *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 167–180.

Kherad Pajouh S, Renaud O (2010). "An Exact Permutation Method for Testing any Effect in Balanced and Unbalanced Fixed Effect ANOVA." *Computational Statistics & Data Analysis*, **54**, 1881–1893. doi:10.1016/j.csda.2010.02.015.

Kherad-Pajouh S, Renaud O (2014). "A General Permutation Approach for Analyzing Repeated Measures ANOVA and Mixed-Model Designs." *Statistical Papers*, (August), 1–21. doi:10.1007/s00362-014-0617-3.

Langsrud y (2005). "Rotation Tests." *Statistics and computing*, **15**(1), 53–60. doi:10.1007/s11222-005-4789-5.

Lehmann EL, Romano JP (2008). *Testing Statistical Hypotheses*. 3rd edition edition. Springer-Verlag, New York.

Manly BFJ (1991). *Randomization, Bootstrap and Monte Carlo Methods in Biology*. CRC Press.

Maris E, Oostenveld R (2007). "Nonparametric Statistical Testing of EEG- and MEG-data." *Journal of Neuroscience Methods*, **164**(1), 177–190. doi:10.1016/j.jneumeth.2007.03.024.

Pernet C, Latinus M, Nichols T, Rousselet G (2014). "Cluster-based Computational Methods for Mass Univariate Analyses of Event-Related Brain Potentials/fields: A Simulation Study." *Journal of Neuroscience Methods*. doi:10.1016/j.jneumeth.2014.08.003.

Searle SR (2006). *Linear Models for Unbalanced Data*. 1 edition edition. John Wiley & Sons, Hoboken, N.J.

Seber GAF, Lee AJ (2012). *Linear Regression Analysis*. John Wiley & Sons.

Smith S, Nichols T (2009). "Threshold-free Cluster Enhancement: Addressing Problems of Smoothing, Threshold Dependence and Localisation in Cluster Inference." *NeuroImage*, **44**(1), 83–98. doi:10.1016/j.neuroimage.2008.03.061.

ter Braak CJF (1992). "Permutation Versus Bootstrap Significance Tests in Multiple Regression and Anova." In KH Jöckel, G Rothe, W Sendler (eds.), *Bootstrapping and Related Techniques: Proceedings of an International Conference, Held in Trier, FRG, June 4–8, 1990*, pp. 79–85. Springer-Verlag, Berlin, Heidelberg. DOI: 10.1007/978-3-642-48850-4_10.

Tipura E, Renaud O, Pegna A (2017). "Attention Shifting and Subliminal Cueing: an EEG Study Using Emotional Faces (Submitted)."

Troendle JF (1995). "A Stepwise Resampling Method of Multiple Hypothesis Testing." *Journal of the American Statistical Association*, **90**(429), 370–378. doi:10.1080/01621459.1995.10476522.

Weiss NA (2015). ***wPerm**: Permutation Tests*. R package version 1.0.1, URL https://CRAN.R-project.org/package=wPerm.

Wheeler B, Torchiano M (2016). ***lmPerm**: Permutation Tests for Linear Models*. R package version 2.1.0, URL https://CRAN.R-project.org/package=lmPerm.

Winkler AM, Ridgway GR, Douaud G, Nichols TE, Smith SM (2016). "Faster Permutation Inference in Brain Imaging." *NeuroImage*. doi:10.1016/j.neuroimage.2016.05.068.

Winkler AM, Ridgway GR, Webster MA, Smith SM, Nichols TE (2014). "Permutation Inference for the General Linear Model." *NeuroImage*, **92**, 381–397. doi:10.1016/j.neuroimage.2014.01.060.

Affiliation:

Jaromil Frossard
University of Geneva
Boulevard du Pont d'Arve 40, 1204 Geneva
E-mail: jaromil.frossard@unige.ch
URL: http://unige.ch

Olivier Renaud
University of Geneva
Boulevard du Pont d'Arve 40, 1204 Geneva
E-mail: olivier.renaud@unige.ch
URL: http://unige.ch