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. Permutation tests are also particularly useful to overcome the multiple comparisons problem as they are used to test the effect of factors or variables on signals while controlling the family-wise error rate (FWER). This article introduces the **permuco** package which implements several permutation methods. They can all be used jointly with multiple comparisons procedures like the cluster-mass tests or threshold-free cluster enhancement (TFCE). The **permuco** package is designed, first, for univariate permutation tests with nuisance variables, like regression and ANOVA; and secondly, for comparing signals as required, for example, for the analysis of event-related potential (ERP) of experiments using electroencephalography (EEG). This article describes the permutation methods and the multiple comparisons procedures implemented. A tutorial for each of theses cases is provided.

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 properties under non normality (Lehmann and Romano 2008). However they require the assumption of exchangeability under the null hypothesis to be fulfilled which is not the case in a multifactorial setting. For these more complex designs, Janssen and Pauls (2003), Janssen (2005), Pauly, Brunner, and Konietschke (2015) and Konietschke, Bathke, Harrar, and Pauly (2015) show that permutation tests based on non exchangeable data can be exact asymptotically if used with studentized statistics. Another approach to handle multifactorial designs is to transform the data before permuting. Several authors (Draper and Stoneman 1966; Freedman and Lane 1983; Kennedy 1995; Huh and Jhun 2001; Dekker, Krackhardt, and Snijders 2007; Kherad-Pajouh and Renaud 2010; ter Braak 1992) have proposed different types of transformations and Winkler, Ridgway, Webster, Smith, and Nichols (2014) gave a simple and unique notation to compare those different methods.

Repeated measures ANOVA including one or more within subject effects are the most widely used models in the field of psychology. In the simplest case of one single random factor, an exact permutation procedure consists in restricting the permutations within the subjects. In more general cases, free permutations in repeated measures ANOVA designs would violate the exchangeability assumption. This is because the random effects associated with subjects and

their interactions with fixed effects imply a complex structure for the (full) covariance matrix of observations. It follows that the second moments are not preserved after permutation. Friedrich, Brunner, and Pauly (2017a) have derived exact asymptotic properties in those designs for a Wald-type statistic and Kherad-Pajouh and Renaud (2015) proposed several methods to transform the data following procedures developed by Kennedy (1995) or Kherad-Pajouh and Renaud (2010).

For linear models, permutation tests are useful when the assumption of normality is violated or when the sample size is too small to apply asymptotic theory. In addition they can be used to control the family wise error rate (FWER) in some multiple comparisons settings (Troendle 1995; Maris and Oostenveld 2007; Smith and Nichols 2009). These methods have been successfully applied for the comparison of experimental conditions in both functional magnetic resonance imaging (fMRI) and electroencephalography (EEG) as they take advantage of 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 methods and multiple comparisons procedures using permutation tests and to explain how it can be used in R (R Core Team 2021) with the package **permuco** (Frossard and Renaud 2019). The package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=permuco`. Note that the presentation and discussion of the available packages that handle permutation tests in related settings is deferred to Section 5.1, where all the notions are introduced. Appendix A shows a comparison of the relevant code and outputs. But first, Section 2 focuses on fixed effect models. It explains the model used for ANOVA and regression and the various permutation methods proposed in the literature. Section 3 introduces the methods for repeated measures ANOVA. Section 4 explains the multiple comparisons procedures used for comparing signals between experimental conditions and how permutation tests are applied in this setting. Section 5 describes additional programming details and some of the choices for the default settings in the **permuco** package. Section 6 treats two real data analyses, one from a control trial in psychology and the second from an experiment in neurosciences using EEG.

2. The fixed effects model

2.1. Model and notation

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

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

where $\underset{n\times 1}{y}$ is the response variable, $\left[\begin{array}{cc} \underset{n\times(p-q)}{D} & \underset{n\times q}{X} \end{array}\right]$ is a design matrix split into the nuisance variable(s) D (usually including the intercept) and the variable(s) of interest X associated with the tested hypothesis. D and X may be correlated and we assume without loss of generality that $\left[\begin{array}{cc} D & X \end{array}\right]$ is a full rank matrix. The parameters of the full model $\left[\begin{array}{cc} \underset{1\times(p-q)}{\eta^\top} & \underset{1\times q}{\beta^\top} \end{array}\right]^\top$ are also split into the parameters associated with the nuisance variable(s) η and the one(s) associated with the variable(s) of interest β. ϵ is an error term that follows a distribution

$(0, \sigma^2 I_n)$. The hypothesis tested writes

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

The permutation test is exact under the null hypothesis for finite samples if the data are exchangeable under the null hypothesis. This assumption is not fulfilled in model in Equation 1 as we cannot control the influence of the nuisance term $D\eta$ when permuting. In fact, under the null hypothesis in Equation 2, the responses follow a distribution $(D\eta, \sigma^2 I_n)$ which are not exchangeable due to the presence of unequal first moments. Pauly *et al.* (2015) show however that permuting the responses and using a Wald-type statistic is an asymptotically exact procedure in factorial designs. Another approach, which is the focus of this paper, is to transform the data prior to the permutation. Those transformation procedures are what will be called permutation methods. They are described in Section 2.2 and are implemented in **permuco**.

The permutation of a vector v is defined as Pv and the permutation of the rows of a matrix M as PM where P is a permutation matrix (Gentle 2007, pp. 66–67). For any design matrix M, its corresponding "hat" matrix is $H_M = M(M^\top M)^{-1}M^\top$ and its corresponding "residuals" matrix is $R_M = I - M(M^\top M)^{-1}M^\top$ (Greene 2011, pp. 24–25). The full QR decomposition is

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

where $\underset{n\times p}{Q_M}$ and $\underset{n\times(n-p)}{V_M}$ define together an orthonormal basis of $\mathbb{R}^n$ and where $\underset{p\times p}{U_M}$ is interpreted as M in the subspace of Q_M. An important property of the QR decomposition is that $H_M = Q_M Q_M^\top$ and $R_M = V_M V_M^\top$ (Seber and Lee 2012, pp. 340–341).

2.2. Permutation methods for linear models and factorial ANOVAs

The discussed permutation methods are functions that transform the data in order to reduce the effect of the nuisance variables. They can be computed for all permutations $P \in \mathscr{P}$ where $\mathscr{P}$ is the set of all n_P distinct permutation matrices of the same size. For any permutation matrix P, a given permutation method will transform the observed data $\{y, D, X\}$ into the permuted data $\{y^*, D^*, X^*\}$. The **permuco** package provides several permutation methods that are summarized in Table 1 using a notation inspired by Winkler *et al.* (2014).

The default method of **permuco** is the `freedman_lane` method that works as follows: we first fit the "small" model which only uses the nuisance variables D as predictors. Then, we permute its residuals and add them to the fitted values. Theses steps produce the permuted response variable y^* which constitutes the "new sample". It is fitted using the unchanged design D and X. In this procedure, only the residuals are permuted and they are supposed to share the same expectation (of zero) under the null hypothesis. For each permutation, the effect of nuisance variables is hence reduced. Using the above notation, the fitted values of the "small" model can be written as $H_D y$ and its residuals $R_D y$. Its permuted version is pre-multiplied by a permutation matrix, e.g., $PR_D y$. The permuted response variable is therefore simply written as $y^* = H_D y + PR_D y = (H_D + PR_D)y$, as displayed in Table 1. The permuted statistics (e.g., t or F statistics) are then computed using y^* and the unchanged design matrices $D^* = D$ and $X^* = X$.

`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

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

All the remaining permutation methods are also summarized by the transformation of y, D and X into y^*, X^* and D^* and are explained next. The `manly` method simply permutes the response (this method is sometimes called raw permutations). Even if this method does not take into account the nuisance variables, it still has good asymptotic properties when using studentized statistics. `draper_stoneman` permutes the design of interest (note that without nuisance variables permuting the design is equivalent to permuting the response variable). However, this method ignores the correlation between D and X that is typically present in regressions or unbalanced designs. For the `dekker` method, we first orthogonalize X with respect to D, then we permute the design of interest. This transformation reduces the influence of the correlation between D and X and is more appropriate for unbalanced design. The `kennedy` method orthogonalizes all of the elements (y, D and X) with respect to the nuisance variables, removing the nuisance variables in the equation, and then permutes the obtained response. Doing so, all the design matrices lie in the span of X, a sub-space of observed design X and D. However this projection modifies the distribution of the residuals that lose exchangeability ($R_Dy \sim (0, R_D\sigma^2)$ for original IID data). The `huh_jhun` method is similar to `kennedy` but it applies a second transformation ($V_D^\top$) to the data to ensure exchangeability (up to the second moment, $V_D^\top R_Dy \sim (0, I_{n-(p-q)}\sigma^2)$). 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 `terBraak` method is similar to `freedman_lane` but uses the residuals of the full model. This permutation method creates a new response variable y^* which assumes that the observed value of the estimate $\hat{\beta}|y$ is the true value of β. Computing the statistic using y^*, X, D would not produce a permutation distribution under the null hypothesis. To circumvent this issue, the method changes the null hypothesis when computing the statistics at each permutation to $H_0 : \beta = \hat{\beta}|y = (X^\top R_DX)^{-1}X^\top R_Dy|y$. The right part of this new hypothesis corresponds to the observed estimate of the parameters of interest under the full model, and implicitly uses a pivotal assumption. Note that `terBraak` is the only method where the statistic computed with the identity permutation is different from the observed statistic. 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.

For each of the methods presented in Table 1, permutation tests can be computed using different statistics. For univariate or multivariate β parameters, the **permuco** package implemented a F statistic that constitutes a marginal test (or "type III" sum of square) (Searle

2006, pp. 53–54). For a univariate $\underset{1\times1}{\beta}$, one- and two-sided tests (based on a t-statistic) are also implemented. We write the F statistic as

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

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 is the estimate of β under the full model. Note that the statistic can be simplified by a factor of $(X^\top R_D X)^{-1/2}$. The two statistics are function of data. They lead to the general notation $t = t(y, D, X)$ when applied to the observed data and to $t^* = t(y^*, D^*, X^*)$ when applied to the permuted data. The permuted statistics constitute the set $\mathscr{T}$ which contains the t^* for all $P \in \mathscr{P}$. We define the permuted p value as $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(|t^*| \geq |t|\right)$, for a two-tailed t test, $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(t^* \geq t\right)$, for an upper-tailed t test or an F test and finally $p = \frac{1}{n_P} \sum_{t^* \in \mathscr{T}} I\left(t^* \leq t\right)$, for a lower-tailed t test, where $I(\cdot)$ is the indicator function.

3. Repeated measures ANOVA

3.1. Model and notation

We write the repeated measures ANOVA model in a linear mixed effects 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 variable(s) $\underset{n\times(p_1-q_1)}{D}$, and the variable(s) of interest $\underset{n\times(p_1)}{X}$. The specificity of the repeated measures ANOVA model allows us to split the random part into $\underset{n\times(p_2^0-q_2^0)}{E^0}$ and $\underset{n\times q_2^0}{Z^0}$ which are the random effects associated with D and X respectively (Kherad-Pajouh and Renaud 2015). 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 using

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

where D^0_{within} and X^0_{within} are overparametrized matrices and are associated with the within effects in the design matrices 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 matrices E^0 and Z^0 are overparametrized and colinear to the intercept or between-participant effects they cannot directly be used to compute their corresponding sums of squares. We need versions that are constrained into their respective appropriate sub-spaces:

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

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

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

The matrices E and Z are respectively of rank $p_2 - q_2$ and q_2 and are the ones used to compute F statistics. Formally, the hypothesis of interest associated with Equation 6 writes:

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

3.2. Permutation methods for repeated measures ANOVA

Similarly to the fixed effects model, we can test hypotheses using permutation methods (Kherad-Pajouh and Renaud 2015). The ones that are implemented in the **permuco** package are given in Table 2. The two methods are based on a similar idea. By pre-multiplying the design and response variables by R_D or $R_{D,E}$, we orthogonalize the model to the nuisance variables. This procedure can be viewed as an extension of the `kennedy` procedure (see Table 1) to repeated measures ANOVA.

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

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

As for the fixed effects model, the statistic is written as a function of the data $t = t(y, D, X, E, Z)$ and the permuted statistic $t^* = t(y^*, D^*, X^*, E^*, Z^*)$ is a function of the permuted data under the chosen method. The p value is defined as in the fixed effect case.

Here is a small example of the creation of the matrices for the F statistic in repeated measures ANOVA. In a balanced design with 12 participants, 1 between-participants factor B_2 with 2 levels and 1 within-participants factor W_3 with 3 levels, assuming the test of the main effect of B_2, the denominator of Equation 10 represents the sum of squares associated to the participants. The matrix Z^0 has one column for each participant coding with 0 and 1 for the participant. It is overparametrized as it has a dimension 36×12 and a rank of 12. However, the matrix Z^0 is not orthogonal the fixed part of the design, especially the intercept and the main effect of B_2. Computing the sum of squares using directly Z^0 would also consider the effect of the intercept and of B_2 in addition of the effect of the participants. If we only want the sum of squares associated to the participants, we must reduce the rank of Z^0 which means, geometrically, orthogonalizing Z^0 to the intercept and to the matrix associated to B_2. Moreover, we are not interested by the estimations of the parameters γ but only by the projection of y into Z^0 which means that any matrices spanning the appropriate space is a potential candidate for B_2. Hence, we only have to orthogonalize Z^0 to the fixed part of the design which is done using Equation 8. It creates the matrix Z with a dimension of 36×12 but a rank of 10. Note that most of the columns of $[D\ X]$ are not useful when computing $R_{D,X}$ as the matrix Z^0 is already orthonogonal to the part of the design coding the effects of W_3 and the interaction between B_2 and W_3.

4. Signal and multiple comparisons

In EEG experiments, researchers are often interested in testing the effect of conditions on the event-related potential (ERP). It is a common practice to test the signals at each time point of the ERP. In that kind of experiments, thousands of tests are typically carried out (e.g., one measure every 2 ms over 2 seconds) and the basic multiple hypotheses corrections like Bonferroni (Dunn 1958) are useless as their power is too low.

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

Better known, the cluster-mass test (Maris and Oostenveld 2007) has shown to be powerful while 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 2015), but usually presented only with one factor.

All these approaches use permutations and are compatible with the methods displayed in Tables 1 and 2, as shown next. In addition to multiple comparisons procedures that use permutation, the well-known Bonferroni and Holm (Holm 1979) corrections and the control of the false positive rate by Benjamini and Hochberg (1995) are also implemented in **permuco**.

4.1. Model and notation

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

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

where y_s is the response variable for all observations at time s and each of the k models are the same as Equation 1. D and X, the design matrices, are then identical over the k time points. The aim is to test simultaneously all k 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 of the k models are defined as in Equation 6 and, similarly, we are interested to test the k hypotheses $H_0^s : \beta_s = 0$ vs. $H_1^s : \beta_s \neq 0$ for $s \in \{1, \dots, k\}$.

For both models, we choose one of the permutation methods presented in Tables 1 or 2 and compute the k observed statistics t_s, the k sets of permutated statistics $\mathscr{T}_s$, which lead to k raw or uncorrected p values.

To correct them, the k sets of permutated statistics $\mathscr{T}_s$ can be analyzed as one set of multivariate statistic. It is done simply by combining the k univariate permutation-based distributions into a single k-variate distribution which maintains the correlation between tests. For each permutation, we simply combine all k univariate permuted statistics $t_1^*, \dots, t_k^*$ into one multivariate permuted statistic $\mathbf{t}^* = [t_1^* \ \dots \ t_k^*]^\top$. The three multiple comparisons procedures described below are all based on this multivariate distribution and take advantage of the correlation structure between the tests.

Algorithm 1 Troendle corrected p value.

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

4.2. Troendle's step-wise resampling method

The method developed by Troendle (1995) takes 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. Secondly, if we reject $H_{(k)}$ and want to test $H_{(k-1)}$, we can safely assume that $H_{(k)}$ is false while controlling the FWER. Either $H_{(k)}$ is 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.

This procedure is valid in a general setting and is easly implemented for permutation tests. The permuted sets $\mathscr{T}_s$ is interpreted as a nonparametric distribution of the T_s and based on Troendle (1995), we use Algorithm 1 to compute the corrected p value.

4.3. Cluster-mass statistic

The method proposed by Maris and Oostenveld (2007) for EEG rely on a continuity argument that implies that an effect will appear into clusters of adjacent timeframes. Based on all time-specific statistics, we form these clusters using a threshold τ as follows (see Figure 1). All the adjacent time points for which the statistics are above this threshold define one cluster C_i for $i \in [1, \dots, n_c]$, where n_c is the number of clusters found in the k statistics. We assign to each time point in the same cluster C_i, the same cluster-mass statistic $m_i = f(C_i)$ where f is a function that aggregates the statistics of the whole cluster into a scalar; typically the sum of the 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 Algorithm 2.

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 within a cluster C_i is the p value associated with this cluster, i.e., $p_i = \frac{1}{n_P} \sum_{m^* \in \mathscr{M}} I(m^* \geq m_i)$.

In addition to the theoretical properties of this procedure (Maris and Oostenveld 2007), this method makes sense for EEG data analysis because if a difference of cerebral activity is

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 $s \in \{1, \ldots, k\}$.
3: Find the n_c^* clusters C_i^* as the sets 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^*.

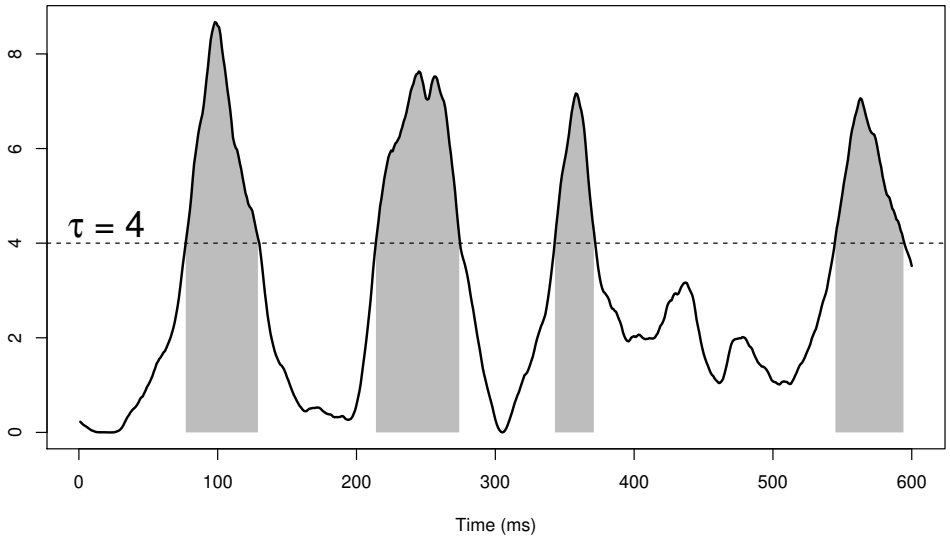

Figure 1: Display of the 600 statistics corresponding to the tests on 600 time points. Here 4 clusters are found using a threshold $\tau = 4$. Using the sum to aggregate the statistics, for each cluster i, the shaded area underneath the curve represents its cluster-mass m_i.

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

Although it controls (weakly) the FWER for any a priori choice of threshold, the result of the cluster-mass procedure is sensitive to this choice. The TFCE (Smith and Nichols 2009) is closely related to the cluster-mass but gets rid of this seemingly arbitrary choice. It is defined at each time $s \in [1, \ldots, k]$ 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. Figure 2 illustrates how the TFCE statistic is computed for a given time point s.

We construct the TFCE null distribution $\mathscr{U}$ by applying the formula in Equation 13 at each time-point of the permuted statistics t_s^* for $s \in \{1, \ldots, k\}$ to produce for each permutation, k values u_s^*. Then the contribution of a permutation to $\mathscr{U}$ is the maximum of all k values u_s^*

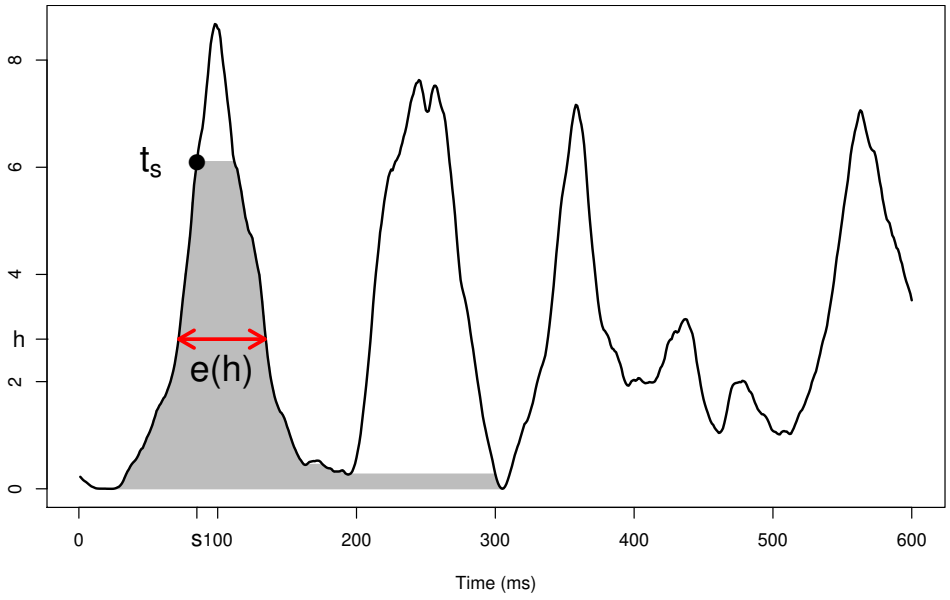

Figure 2: The TFCE transforms the statistic t_s using formula in Equation 13. The extend $e(h)$, in red, is shown for a given height h. The TFCE statistics u_s at s can be viewed as a function of characteristics in the grey area.

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 $s \in \{1, \ldots, k\}$
3: Compute the k enhanced statistics u_s^* using a numerical approximation of the integral in Equation 13
4: `Return` the maximum over the k value u_s^*

(see Algorithm 3). In practice, the integral in Equation 13 is approximated numerically using small $dh \leq 0.1$ (Smith and Nichols 2009; Pernet *et al.* 2015).

At time s, the statistic t_s will be modified using the formula in Equation 13. The formula can be viewed as a function of characteristics in the grey area (its area in the special case where both E and H are set to 1).

To test the significance of a time point s we compare its enhanced statistics u_s with the threshold-free cluster-enhancement null distribution $\mathscr{U}$. For an F test we define the p value as $p_s = \frac{1}{n_P} \sum_{u^* \in \mathscr{U}} I(u^* \geq u_s)$.

4.5. Interpreting cluster based inference

The cluster-mass test and the TFCE are methods based on clustering the data and the interpretation of significant findings is then not intuitive. First, note that the Troendle's method is not based on clustering and does not have these issues. Its interpretation is straight-forwards as we can interpret individually each discovery. For the cluster-mass test the interpretation should be done at a cluster level: a significant cluster is a cluster which contains at least one significant time-point. It follows that the cluster-mass test does not allows the interpretation of the precise time location of clusters (Sassenhagen and Draschkow 2019). Intuitively, the

cluster-mass test is a two steps procedure: first, it aggregates time-points into clusters, and then summarizes them using the cluster-mass. The inference is only performed at the second step which looses any information on the shape and size of the clusters. It implies that the interpretation of individual time-point is proscribed. Finally, the transformation of the TFCE statistic is an integration over all thresholds of cluster statistics (Smith and Nichols 2009). Therefore, the TFCE does not allow an interpretation of each time-point individually either as it also summarizes statistics using the concept of clusters. Thus, the interpretation of individual time-point must also involves it. Therefore, a significant time-point must be interpreted as a time-point being part of at least one significant cluster (among all clusters formed using all thresholds), where a significant cluster contains at least one significant time-point.

5. Comparison of implementations

5.1. Comparison of packages

Several packages for permutation 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 *et al.* 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 multifactorial 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 for repeated measures, even when setting the parameter `seqs = FALSE`. The order of the factors will therefore matter in this case. The permutation method consists in permuting the raw data even in the presence of nuisance variables, which correspond to the `manly` method, see Table 1. For repeated measures 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 rule 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 single pair of observations. This speeds up the code but the quality of the obtained p value is not well documented.

The **flip** package (Finos 2018) produces permutation and rotation tests (Langsrud 2005) for fixed effects and handles nuisance variables based on methods similar to the `huh_juhn` method of Table 1. It performs tests in designs 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, Konietschke, and Pauly 2017b) produces marginal permutation tests for pure factorial design (without covariates) with a Wald-type statistic. The permutation method is `manly`. This method has been shown to be asymptotically exact even under heteroscedastic conditions (Pauly *et al.* 2015). Moreover, Friedrich, Konietschke, and Pauly (2021) generalize these tests to multivariate data like MANOVA models.

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

The codes and outputs for packages that perform ANOVA/ANCOVA are given in Appendix A.1 and in Appendix A.2 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 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 provides 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 as implemented in **flip** (based on eigendecomposition) and **permuco** (based on QR decomposition) packages lead to slightly different results (see Table 1 for more information on the permutation methods).

Finally, concerning repeated measures designs, **flip** cannot handle cases where measures are not repeated in each condition for each subject, and therefore cannot be compared in Appendix A.2. As already said, **lmPerm** produces sequential tests in repeated measures 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 methods

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. Note that `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 for the **permuco** package is set to `freedman_lane`.

For the random effects model, Kherad-Pajouh and Renaud (2015) show 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 all permutations. For small datasets, if the `np` argument is greater than the number of possible permutations ($n!$), 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`, which is a good practice for permutation tests. Moreover this implies that the sum of the two one-sided 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 (Dunn 1958), to `"holm"` for the Holm correction (Holm 1979), `"benjamini_hochberg"` for the Benjamini-Hochberg method (Benjamini and Hochberg 1995), to `"troendle"`, see Section 4.2, to `"clustermass"`, see Section 4.3 and to `"tfce"`, see Section 4.4. Note that in the **permuco** package, these 6 methods are available in conjunction with permutation, although the first 3 methods are general procedures that could also be used in a parametric setting.

For the `"clustermass"` method, the `threshold` parameter of the cluster-mass statistic is usually chosen by default at the 0.95 quantile of the corresponding univariate parametric distribution; 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 an F. It should be a function that returns a positive scalar which will be large for an 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 wider clusters.

For the `"tfce"` method, the default value for the extend parameter is `E` $= 0.5$ and for the height `H` $= 2$ for t tests and, for F test, it is `E` $= 0.5$ and `H` $= 1$ following the recommendations of Smith and Nichols (2009) and Pernet *et al.* (2015). The `ndh` parameter controls the number of steps used in the approximation of the integral in Equation 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. When available, they are implemented in a more efficient way in **permuco**. For example, to reduce the computing time, the permuted statistics are computed through a 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 adjust those effects by the length of the stay. To this end, we perform an ANCOVA and need to center the covariate.

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

The permutation tests are obtained with the `aovperm` function. The `np` argument sets 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 converts the coding of factors with the `contr.sum` which allows us to test the main effects of factors and their interactions.

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

Anova Table
Resampling 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
                    resampled 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 using a 5% level. However, the difference between these 2 p values is 0.0117 which is high enough to lead to different conclusions e.g., in case of correction for multiple tests or a smaller α level.

If we are interested in the 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
Resampling 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
                    resampled P(>F)
LOS14                       0.0000
sex                         0.0214
insurance                   0.6101
LOS14:sex                   0.1550
LOS14:insurance             0.0229
sex:insurance               0.6530
LOS14:sex:insurance         0.0827
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 $p = 0.0214$.

If the researcher has an a priori oriented alternative hypothesis $H_A : \beta_{sex=M} > \beta_{sex=F}$, the `lmperm` function produces one-sided t tests. 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$insurance) <- contr.sum
R> contrasts(emergencycost$insurance)

             [,1]
public          1
semi_private   -1

R> contrasts(emergencycost$sex) <- contr.sum
R> contrasts(emergencycost$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
Resampling 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
                     resampled Pr(<t) resampled Pr(>t)
(Intercept)
LOS14                          1.0000           0.0000
sex1                           0.0141           0.9859
insurance1                     0.6778           0.3222
LOS14:sex1                     0.0788           0.9212
LOS14:insurance1               0.9878           0.0122
sex1:insurance1                0.3345           0.6655
LOS14:sex1:insurance1          0.0408           0.9592
                     resampled Pr(>|t|)
(Intercept)
LOS14                            0.0000
sex1                             0.0211
insurance1                       0.6104
LOS14:sex1                       0.1572
```

```
LOS14:insurance1                0.0232
sex1:insurance1                 0.6526
LOS14:sex1:insurance1           0.0835
```

The effect `sex1` is significant for both the parametric one-sided p value, $p = 0.007$, and the permutation one-sided p value, $p = 0.0211$. It indicates that when the length of the stay is high, men have a shorter cost than women.

To test the effect of the sex within the public insured persons (called simple effect), we change the coding of the factors inside the `data.frame` using the `contr.treatment` function and disable the automatic recoding using the argument `coding_sum = FALSE`.

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

            semi_private
public                 0
semi_private           1

R> contrasts(emergencycost$sex) <- contr.sum
R> contrasts(emergencycost$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
Resampling 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
                   resampled P(>F)
LOSc                          0.0000
sex                           0.0003
```

```
insurance                 0.6829
LOSc:sex                  0.0000
LOSc:insurance            0.0231
sex:insurance             0.8519
LOSc:sex:insurance        0.0836
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, $p = 0.0003$, 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 = FALSE)
```

We perform the permutation tests by running the `aovperm` function. The within subject factors 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")
```

```
Warning message:
In checkBalancedData(fixed_formula = formula_f, data = cbind(y,  :
  The data are not balanced, the results may not be exact.
```

A warning message is issued if the design is not fully balanced, as some exactness properties of the tests are no longer warranted. However, the method from Kherad-Pajouh and Renaud (2015) can still be applied as the within-subject factor (`time`) is balanced. The results are shown in an ANOVA table by printing the object:

```
R> mod_jpah2016
```

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

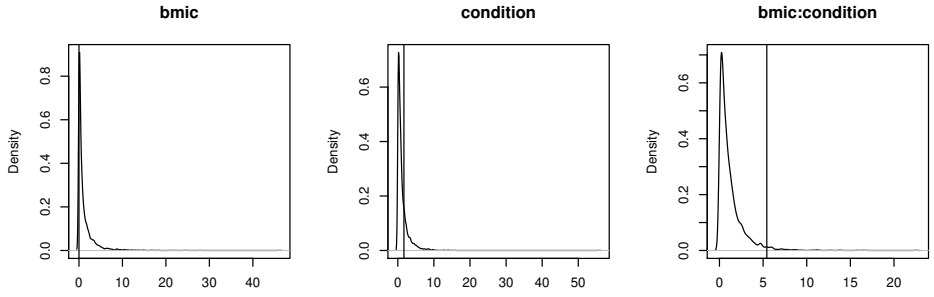

Figure 3: The permutation distributions of the F statistics for the effects `bmic`, `condition` and `bmic:condition`. The vertical lines indicate the observed statistics.

```
                         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) resampled P(>F)
bmic                0.0023           0.9627          0.9610
condition           1.6954           0.2217          0.2234
bmic:condition      5.4269           0.0193          0.0224
time                0.0209           0.8873          0.8848
bmic:time           0.0285           0.8686          0.8696
condition:time      0.8221           0.4611          0.4432
bmic:condition:time 1.1323           0.3521          0.3554
```

This analysis reveals a significant p value for the effect of the interaction `bmic:condition` with a statistic $F = 5.4269$, which lead to a permutation p value $p = 0.0224$ not far from 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> par(pin = c(1.5, 1.5))
R> plot(mod_jpah2016, effect = c("bmic", "condition", "bmic:condition"))
```

6.3. EEG experiment in attention shifting

`attentionshifting_signal` and `attentionshifting_design` are data provided in the **permuco** package. They come from an EEG recording of 15 participants watching images of either neutral or angry faces (Tipura, Renaud, and Pegna 2019). 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 1024 Hz for 800 ms. Time 0 is when the image appears (event-related potential or ERP). The `attentionshifting_signal`

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	

Table 3: Variables in the `attentionshifting_design` dataset.

dataset contains the ERP of the electrode O1. The design of experiment is given in the `attentionshifting_design` dataset along with the laterality, sex, age, and 2 measures of anxiety of each subjects, see Table 3.

As almost any ERP experiment, the data is designed for a repeated measures ANOVA. Using the **permuco** package, we test each time points of the ERP for the main effects and the interactions of the variables `visibility`, `emotion` and `direction` while controlling for the FWER. 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 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. The summary of the `clusterlm` object gives more information about all clusters for the main effect of visibility, whether they are driving the significant effect or not:

```
R> summary(electrod_O1)$visibility

Effect: visibility.
Alternative Hypothesis: two.sided.
Statistic: fisher(1, 14).
Resampling Method: Rd_kheradPajouh_renaud.
Type of Resampling: permutation.
Number of Dependant Variables: 819.
```

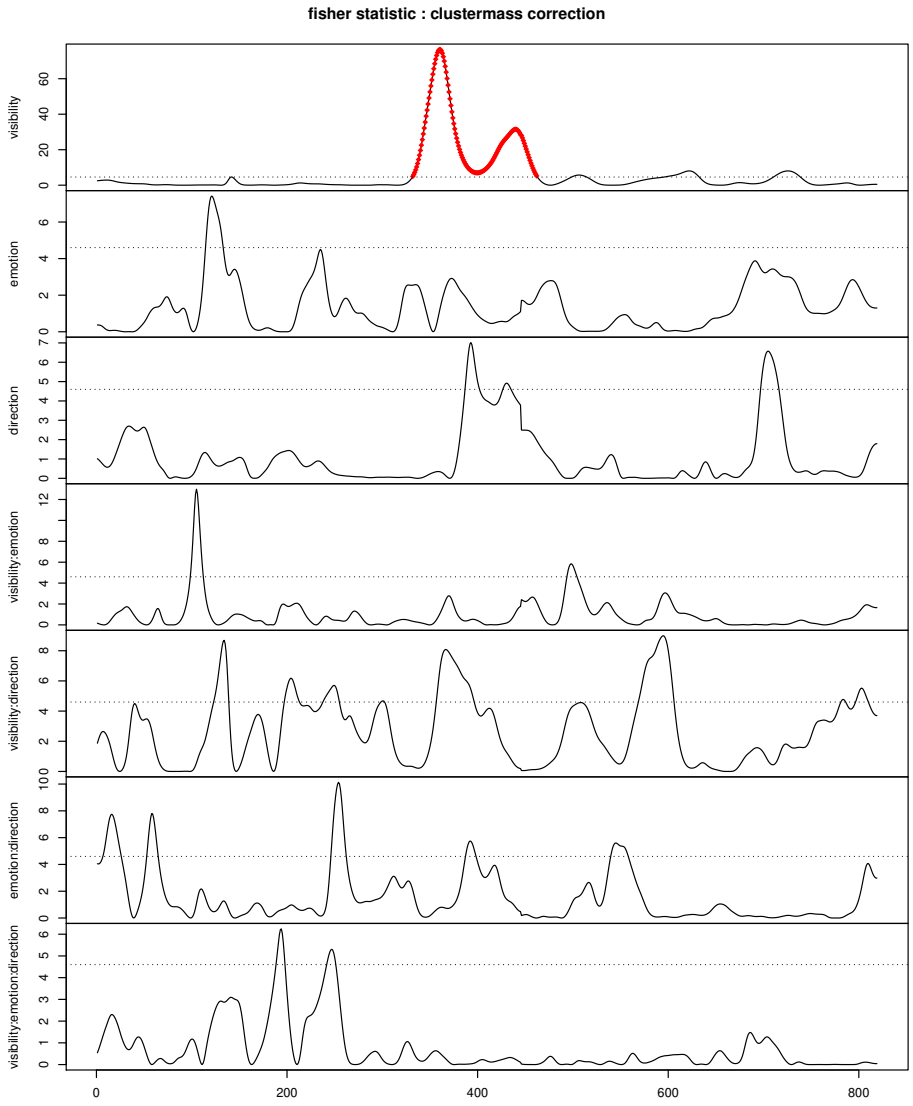

Figure 4: The `plot` method on a `clusterlm` object displays the observed statistics of the three main effects and their interactions. The dotted horizontal line represents the threshold which is set by default to the 95% percentile of the statistic. For this dataset, one cluster is significant for the main effect of visibility using the `clustermass` method, as shown by the red part. The `summary` method gives more details.

```
Number of Resamples: 5000.
Multiple Comparisons Procedure: clustermass.
Threshold: 4.60011.
Mass Function: the sum.
Table of clusters.

  start end cluster mass P(>mass)
1   142 142     4.634852   0.4980
```

```
2   332 462  3559.149739   0.0012
3   499 514    85.019645   0.3994
4   596 632   234.877913   0.2334
5   711 738   191.576178   0.2736
```

There is a significant difference between the two levels of visibility. This difference is driven by one cluster that appears between the measures 332 and 462 which correspond to the 123.7 ms and 250.9 ms after the event. Its cluster-mass statistic is 3559.1 with an associated p-value of 0.0012. The threshold is set to 4.60011 which is the 95% percentile 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 very same permutations as previous model by using the `P` argument. The computation time for those tests is reasonably low: 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 multiple comparisons 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)
```

The TFCE procedure gets approximately a similar effect. However the time-points around 400 (190 ms) are not part of significant effect. If the curves in the TFCE plot happen to to show some small steps (which is not the case in Figure 5) it may be because of a 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`.

Finally, to be able to interpret individually each time-point, we can use the `troendle` multiple comparisons procedure whose results are visualized by plotting the `full_electrod_O1` object. A similar period is detected for the main effect of `visibility`.

```
R> plot(full_electrod_O1, multcomp = "troendle")
```

To interpret individually each time-point in Figure 6, we extract the significant time-points (with an α level of 5%) using the `summary` method, setting the `multcomp` parameter to `"troendle"`. We find that the main effect of `visibility` begin at 129.6 ms after the event.

```
R> summary(full_electrod_O1, multcomp = "troendle")$visibility
```

```
Effect: visibility.
Alternative Hypothesis: two.sided.
Statistic: fisher(1, 14).
```

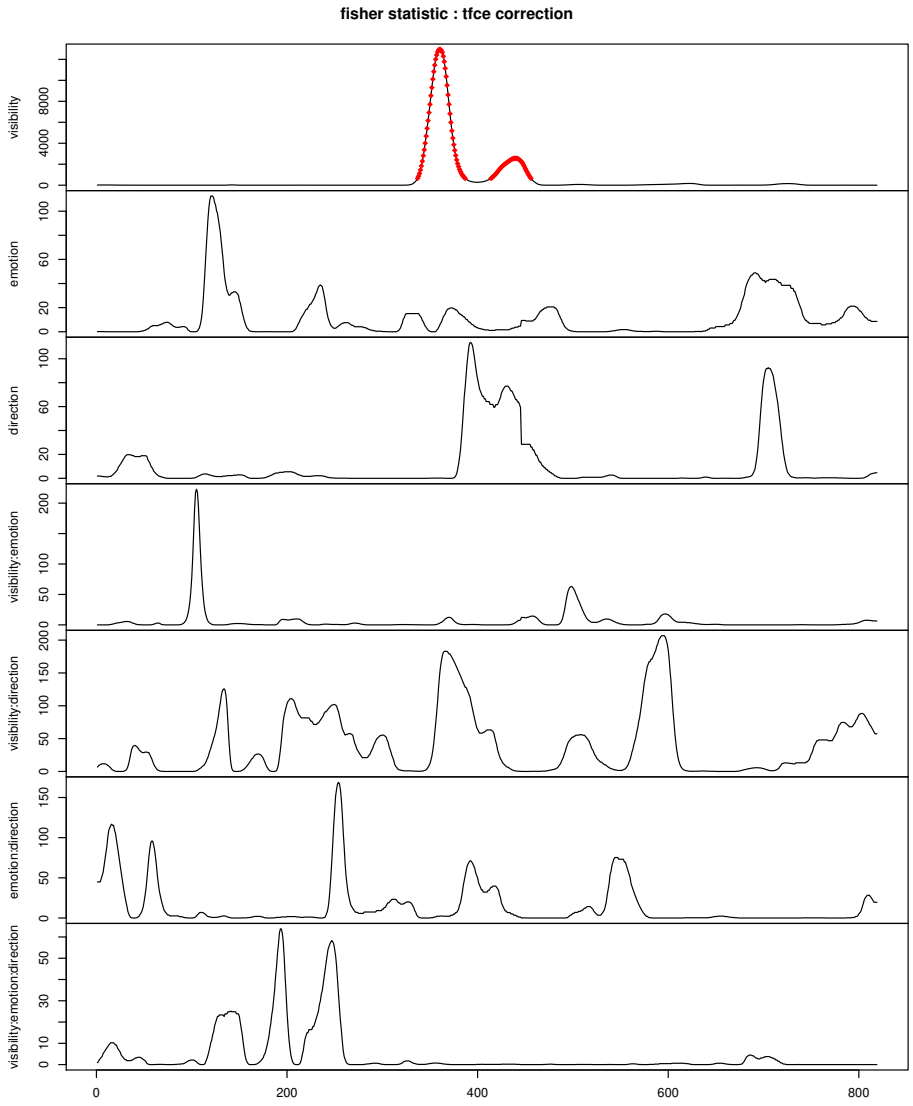

Figure 5: Setting the `multcomp` argument to `"tfce"` in the `plot` function will display the TFCE p values. The argument `enhanced_stat = TRUE` shows the TFCE statistics u_s of Equation 13.

```
Resampling Method: Rde_kheradPajouh_renaud.
Type of Resampling: permutation.
Number of Dependant Variables: 819.
Number of Resamples: 5000.
Multiple Comparisons Procedure: troendle.
Table of pseudo-clusters.

  start end P(>)
1     1 337 n.s.
2   338 386 sign
```

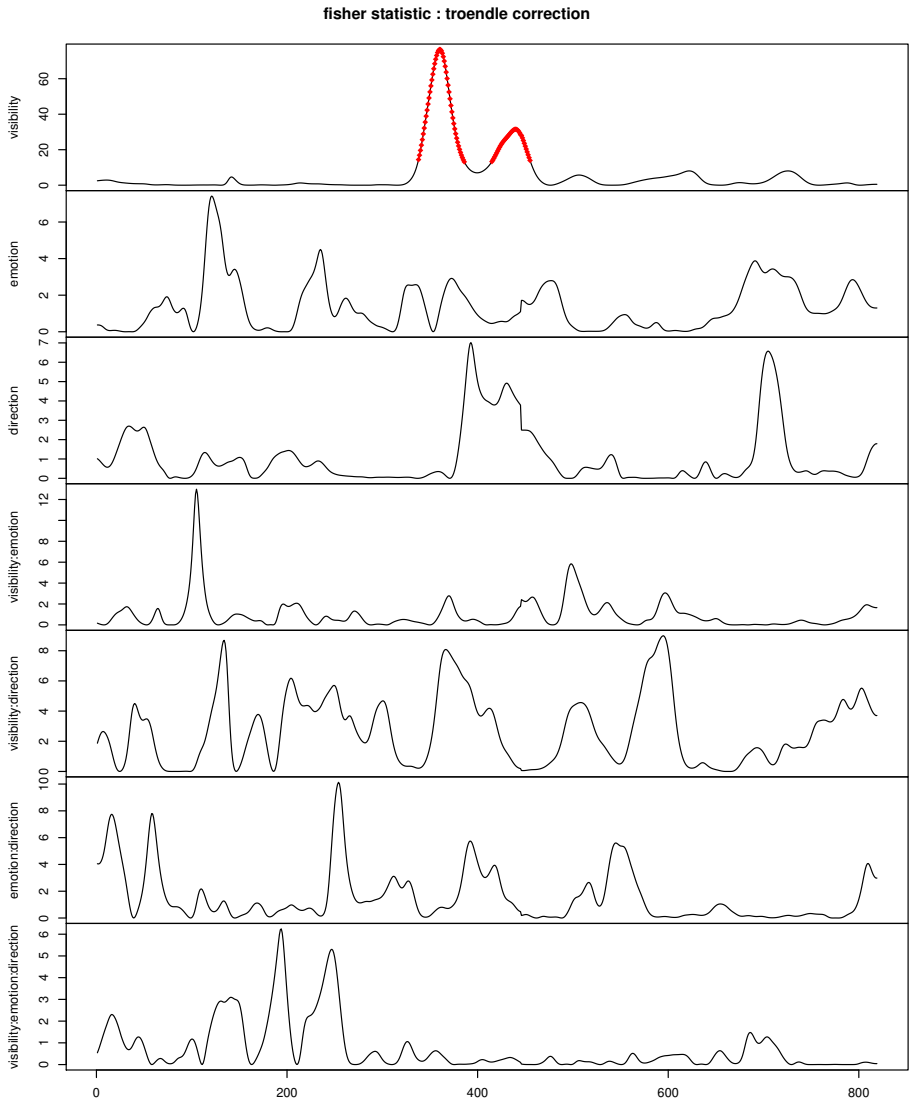

Figure 6: Setting the `multcomp` to `"troendle"` will display the `troendle` correction which allows an interpretation of each time-point individually.

```
3    387 414 n.s.
4    415 455 sign
5    456 819 n.s.
```

7. Conclusion

This article presents recent methodological advances in permutations tests and their implementation in the **permuco** package. Hypotheses in linear models framework or repeated measures ANOVA are tested using several methods to handle nuisance variables. Moreover permutations tests can solve the multiple comparisons problem and control the FWER trough

cluster-mass tests or TFCE, and the `clusterlm` function implements those procedures for the analysis of signals, like EEG data. Section 6 illustrates some real data example of tests that can be performed for regression, repeated measures ANCOVA and ERP signals 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. An early version of the functions are already available in the the following repository: https://github.com/jaromilfrossard/permuco4brain. Another evolution will concern permutation procedures for mixed effects models to allows researchers to perform tests in models containing participants and stimuli specific random effects. Indeed, we plan to include in **permuco** the re-sampling test presented by Bürki, Frossard, and Renaud (2018) as they show that, first, using F statistic (by averaging over the stimuli) in combination with cluster-mass procedure increases the FWER and, secondely, that a re-sampling method based on the quasi-F statistic (Clark 1973; Raaijmakers, Schrijnemakers, and Gremmen 1999) keeps it much closer to the nominal level of 5%.

Acknowledgments

We are particularly grateful for the assistance given by Eda Tipura, Guillaume Rousselet and Elvezio Ronchetti that greatly improved this manuscript. Eda Tipura provided original EEG data and all three gave many comments coming from their extended reading of the paper; although any errors are our own.

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 B*, **57**(1), 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

Bürki A, Frossard J, Renaud O (2018). "Accounting for Stimulus and Participant Effects in Event-Related Potential Analyses to Increase the Replicability of Studies." *Journal of Neuroscience Methods*, **309**, 218–227. doi:10.1016/j.jneumeth.2018.09.016.

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.

Clark HH (1973). "The Language-as-Fixed-Effect Fallacy: A Critique of Language Statistics in Psychological Research." *Journal of Verbal Learning and Verbal Behavior*, **12**(4), 335–359. doi:10.1016/s0022-5371(73)80014-3.

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.

Dunn OJ (1958). "Estimation of the Means of Dependent Variables." *The Annals of Mathematical Statistics*, **29**(4), 1095–1111. doi:10.1214/aoms/1177706443.

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), 1–34. doi:10.18637/jss.v036.i02.

Finos L (2018). ***flip**: Multivariate Permutation Tests*. R package version 2.5.0, URL https://CRAN.R-project.org/package=flip.

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.

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 S, Brunner E, Pauly M (2017a). "Permuting Longitudinal Data in Spite of the Dependencies." *Journal of Multivariate Analysis*, **153**, 255–265. doi:10.1016/j.jmva.2016.10.004.

Friedrich S, Konietschke F, Pauly M (2017b). "**GFD**: An R Package for the Analysis of General Factorial Designs." *Journal of Statistical Software*, **79**(1), 1–18. doi:10.18637/jss.v079.c01.

Friedrich S, Konietschke F, Pauly M (2021). ***MANOVA.RM**: Analysis of Multivariate Data and Repeated Measures Designs*. R package version 0.5.1, URL https://CRAN.R-project.org/package=MANOVA.RM.

Frossard J, Renaud O (2019). ***permuco**: Permutation Tests for Regression, (Repeated Measures) ANOVA/ANCOVA and Comparison of Signals*. R package version 1.1.0, URL https://CRAN.R-project.org/package=permuco.

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

Greene W (2011). *Econometric Analysis*. Prentice Hall.

Heritier S, Cantoni E, Copt S, Victoria-Feser MP (2009). *Robust Methods in Biostatistics*. John Wiley & Sons. doi:10.1002/9780470740538.

Holm S (1979). "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics*, **6**(2), 65–70. doi:10.2307/2532027.

Hothorn T, Hornik K, Van De Wiel MA, Zeileis A, *et al.* (2008). "Implementing a Class of Permutation Pests: The **coin** Package." *Journal of Statistical Software*, **28**(8), 1–23. doi:10.18637/jss.v028.i08.

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.

Janssen A (2005). "Resampling Student's t-Type Statistics." *Annals of the Institute of Statistical Mathematics*, **57**(3), 507–529. doi:10.1007/bf02509237.

Janssen A, Pauls T (2003). "How Do Bootstrap and Permutation Tests Work?" *The Annals of Statistics*, **31**(3), 768–806. doi:10.1214/aos/1056562462.

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. doi:10.1002/9781118165676.ch5.

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 (2015). "A General Permutation Approach for Analyzing Repeated Measures ANOVA and Mixed-Model Designs." *Statistical Papers*, **56**(4), 947–967. doi:10.1007/s00362-014-0617-3.

Konietschke F, Bathke AC, Harrar SW, Pauly M (2015). "Parametric and Nonparametric Bootstrap Methods for General MANOVA." *Journal of Multivariate Analysis*, **140**, 291–301. doi:10.1016/j.jmva.2015.05.001.

Langsrud Ø (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*. Springer-Verlag.

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

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.

Pauly M, Brunner E, Konietschke F (2015). "Asymptotic Permutation Tests in General Factorial Designs." *Journal of the Royal Statistical Society B*, **77**(2), 461–473. doi:10.1111/rssb.12073.

Pernet CR, Latinus M, Nichols TE, Rousselet GA (2015). "Cluster-Based Computational Methods for Mass Univariate Analyses of Event-Related Brain Potentials/Fields: A Simulation Study." *Journal of Neuroscience Methods*, **250**, 85–93. doi:10.1016/j.jneumeth.2014.08.003.

Raaijmakers JGW, Schrijnemakers JMC, Gremmen F (1999). "How to Deal with "The Language-as-Fixed-Effect Fallacy": Common Misconceptions and Alternative Solutions." *Journal of Memory and Language*, **41**(3), 416–426. doi:10.1006/jmla.1999.2650.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Sassenhagen J, Draschkow D (2019). "Cluster-Based Permutation Tests of MEG/EEG Data Do Not Establish Significance of Effect Latency or Location." *Psychophysiology*, **56**(6), 1–8. doi:10.1111/psyp.13335.

Searle SR (2006). *Linear Models for Unbalanced Data*. John Wiley & Sons.

Seber GAF, Lee AJ (2012). *Linear Regression Analysis*. John Wiley & Sons. doi:10.1002/0471725315.

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*, pp. 79–85. Springer-Verlag. doi:10.1007/978-3-642-48850-4_10.

Tipura E, Renaud O, Pegna AJ (2019). "Attention Shifting and Subliminal Cueing under High Attentional Load: An EEG Study Using Emotional Faces." *Neuroreport*. doi:10.1097/wnr.0000000000001349.

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*, **141**, 502–516. 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.

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> set.seed(42)
R> emergencycost$LOSc <- scale(emergencycost$LOS, scale = FALSE)
R> contrasts(emergencycost$sex) <- contr.sum
R> contrasts(emergencycost$insurance) <- contr.sum
R> X <- model.matrix( ~ sex+insurance, data = emergencycost)[, -1]
R> colnames(X) <- c("sex_num", "insurance_num")
R> emergencycost <- data.frame(emergencycost, X)
R> anova_permuco <- aovperm(cost ~ sex*insurance, data = emergencycost)
R> anova_GFD <- GFD(cost ~ sex*insurance, data = emergencycost,
+    CI.method = "perm", nperm  = 5000)
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
Resampling test using freedman_lane to handle nuisance variables and
 5000 permutations.

                     SS  df      F parametric P(>F) resampled P(>F)
sex              60470803   1 0.7193           0.3975          0.3978
insurance       598973609   1 7.1249           0.0083          0.0120
sex:insurance   334349436   1 3.9771           0.0477          0.0508
Residuals     14459666504 172

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
Resampling 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
                   resampled 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> rancova_permuco <- aovperm(iapa ~ bmic*condition*time + Error(id/(time)),
+    data = jpah2016)

Warning message:
In checkBalancedData(fixed_formula = formula_f, data = cbind(y,  :
  The data are not balanced, the results may not be exact.

R> rancova_lmPerm <- aovp(iapa ~ bmic*condition*time + Error(id/(time)),
+    data = jpah2016, nCycle = 1, seqs = FALSE)
R> rancova_permuco

Resampling 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) resampled 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  840   0.3009
bmic:condition  2    89238     44619 5000   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  240   0.3875
bmic:condition:time  2    29146   14572.9  345   0.3914
Residuals           13   167305   12869.6
```

Affiliation:

Jaromil Frossard, Olivier Renaud
University of Geneva
Boulevard du Pont d'Arve 40, 1204 Geneva, Switzerland
E-mail: jaromil.frossard@unige.ch, olivier.renaud@unige.ch
URL: http://www.unige.ch/fapse/mad/