Monte Carlo permutation test in constrained ordination

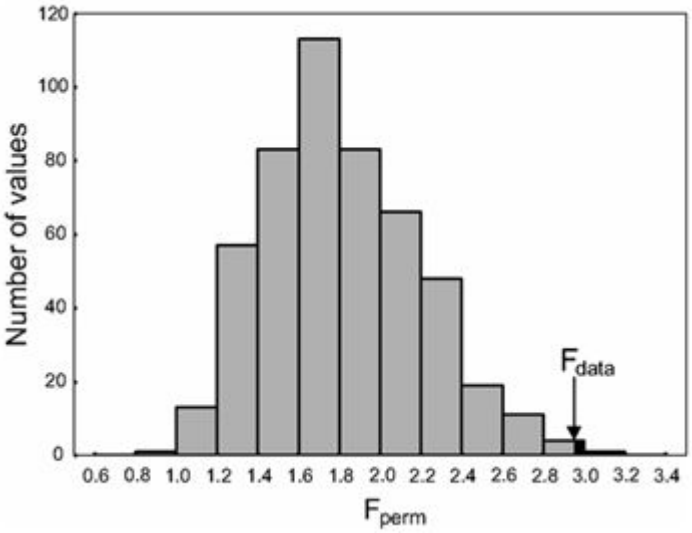
In constrained ordination, variation explained by explanatory variables (after accounting for covariates) is routinely tested, and only those analyses which are significant (or those environmental variables that are significant) are considered for further interpretation. Statistical tests aim to reject the null hypothesis (H0), which is (usually) formulated as *there is no relationship between the response variable and predictor*. By statistical test we estimate the probability of obtaining results as different (or even more different) from those expected under the null hypothesis in our observed data; or, formulated otherwise, we estimate the probability (*P*) of obtaining the value of the test statistic we got from our data in the case that the response variable is independent of the predictor (and the null hypothesis is therefore true). In standard parametric statistical tests, we compare the value of test statistic (e.g. F-value in regression) with the expected null distribution (derived from the assumption about the distribution of the original data - this is why e.g. in regression we expect the normal distribution of response residuals).

How Monte Carlo permutation test works?

If we do not know what is expected null distribution of the test statistic, we need to generate it from our data by permuting (reshuffling) them. **Monte Carlo permutation test** is the way to do that. For example, in linear regression based on least-squares algorithm, we would first calculate the test statistic (e.g. F-value, R^2 or slope of the regression) for our real (unpermuted) data. Then we take one of them (no matter if predictor or response variable) and permute the values, to disconnect possible relationships between the variables and create dataset for which we know that the null hypothesis (of no relationship) is true. We would calculate the same test statistic on the permuted dataset, record it, and repeat it many times (e.g. 499 times). Finally, we would add also the value of test statistic observed on unpermuted data, and the result is the expected null distribution to which we compare our observed test statistic. The result of the statistical test is the **P-value**, the probability of obtaining the value of test statistic equal or more extreme than the observed one, even if the null hypothesis is true. The P-value is calculated as (*the number of permutations where the value of test statistic is* ≥ *observed test statistic* + 1) divided by (*total number of permutations* + 1). The "+ 1" in the formula represents the observed value of the test statistic (on unpermuted data), which was added into the null distribution. Calculated P-value is then compared with a set of arbitrary thresholds (usually 0.05, 0.01 and 0.001), and we conclude whether our analysis is significant at given level (e.g. $P < 0.01$), which means that we rejected the null hypothesis of no relationship between predictor and response variable. In this sense, the lower the P-value, the more significant is the result.

In eigenvalue-based constrained ordination, the test statistic used is the so-called **pseudo-F value**, a multivariate analogue of F-value from linear regression (calculated from the axis eigenvalue, the residual sum of squares of the constrained model, numbers of explanatory variables, covariables and samples in the dataset). If no covariates are used in the analysis, the pseudo-F value can be replaced by the values of (not adjusted) R^2 or the sum of eigenvalues by constrained axes. The resulting expected null distribution of pseudo-F values (F_{perm}) and their comparison with the pseudo-F value on unpermuted data (F_{data} is on Fig. 1.

Figure 1: Observed value of the pseudo-F value (Fdata) compared with the null distribution of expected values (Fperm). From Šmilauer & Lepš (2014).



The formula for calculating P-value is:

$$P = \frac{n_x + 1}{N + 1},$$

where n_x is the number of permutations where the expected value of pseudo-F is higher than observed value ($F_{perm} > F_{data}$), and N is the number of all permutations.

How many permutations?

How many permutations we should choose depends on several considerations. First, because the observed test statistic is in the end added into the expected null distribution, the value is usually ending with 9 (199, 499, 1999), because after adding the observed value they will be nicely round (200, 500, 2000, respectively).

The lowest P-value (P_{min}) we can reach depends on the number of permutations (N) and can be calculated as P_{min} = 1/(N+1). For example, with N = 49 permutations, the lowest P-value which can be obtained is 1/(49+1) = 0.02, and one cannot, therefore, hope to reject the null hypothesis at P < 0.001. Especially if the multiple testing correction is applied (e.g. in the case of stepwise variable selection, or testing several constrained axes), it may be necessary to increase the number of permutations to a quite high value to make sure that the minimum P-value after adjustment will be lower than the selected significance threshold (example: with 499 permutations, P_{min} = 1/(499-1) = 0.002; if we are conducting 10 tests (n_tests = 10, e.g. because we do forward selection from 10 explanatory variables in constrained ordination) and we apply Bonferroni correction, each calculated P-value gets multiplied by 10, and the minimum achievable P-value after adjustment is $P_{min\text{-}adj}$ = 1/(499-1)*n_tests = 0.02; we may then need to increase the number of permutations to be able to reject the null hypothesis at lower P-values).

The number of possible unique permutations is usually very high (in unrestricted permutation can be calculated as the factorial of the number of samples, e.g. 20! = 2,432,902,008,176,640,000 possible options), and the Monte Carlo permutation test is using only a random subsample of these options (this is why the P-value may slightly change in each run of permutation test). In permutations restricted by design (e.g. in randomized block design or rectangular grid) the number of possible permutations may be much lower, and it is then better to enumerate all of them and calculate **the exact P-value** from all existing permutations (this P-value does not change if recalculated, since the set of permutations is always the same).

What can be tested?

We can test variation explained by one or several explanatory variables (predictors). If more than one predictor is included, there is more than one constrained ordination axis; in that case, we can test the significance of all constrained ordination axes, or only the first one. We can also test the significance of each ordination axis separately, e.g. to decide how many of them are relevant for interpretation (and may need to be displayed by ordination diagrams); when the second or higher axis is tested, all lower axis are included as covariables. We can also test the variation explained by each explanatory variable separately or by each variable while accounting for the variation of all other variables included in the model. When making a stepwise selection of variables into the model, in each step, we test whether the variation brought by the newly included variable significantly improves explained variation of the model (while costing one degree of freedom).

Permutation schemas for restricted permutation tests

If our data come from a sampling design where each sample can be considered as independent from others (they are not located in the same block of the experimental design, not too close to other samples, not on the transect or grid), we can permute data freely without any restriction. But if our data come from sampling design that introduces some type of dependence between the samples, we need to consider this dependence when testing the result. This can be done either by **design-based permutations**, where permutation of samples is restricted according to the sampling design (e.g. samples between blocks are never permuted, only within blocks). The alternative is **model-based permutation**, where design restrictions are included as covariables (e.g. blocks) and permutation of samples is completely free because it is done on residuals. Model-based permutation can offer higher power, as it does not come with drawbacks of restricted permutation (e.g. lower number of possible permutations), but on the other side it is not available for all types of sampling designs (e.g. for split-plot design there is no way how to test the effect of "whole-plot" environmental variable, unless we sum/average values of individual split-plots). In design-based permutation, permuted data should represent the situation when the null hypothesis is true (i.e. permutation broke the relationship between the response variable and predictor) while preserving the dependence between samples. If we break also this dependence, we will most likely end up with an inflated Type I error rate (P-values too good to be true). Below are examples or permutation schemas for some of the sampling designs of manipulative and natural experiments, as described in the section Sampling design of ecological experiments. The example tables always contain the first column with sample numbers, one or several columns specifying the design, and then three columns showing permutation outcome. In section Examples (Example 3) you can see how to generate these examples using function how from the vegan library.

Before describing each schema, let's introduce terminology that will be useful when defining the permutation schemas in R. We can imagine three hierarchical levels at which shuffling of cases can be done: *samples*, *plots*, and *blocks*. Samples are grouped into plots, and plots are grouped into blocks. Samples can be permuted within plots, plots can be permuted among each other, but blocks are never permuted (cases can be permuted only within blocks, never between). I will follow this terminology strictly in the text below (in the rest of the website, I am using terms *cases*, *samples* and *plots* interchangeably).

Completely randomized design (Fig. 2) here is just to show the case of unrestricted permutation, where samples are permuted completely randomly.

Completely randomized design (unrestricted)

sample	rand1	rand2	rand3	randn
1	1	10	12	...
2	3	1	5	...
3	12	2	3	...
4	7	4	1	...
5	5	8	8	...
6	4	6	6	...
7	11	12	2	...
8	6	3	7	...
9	2	5	10	...
10	9	9	9	...
11	10	11	4	...
12	8	7	11	...

Figure 2: Completely randomized design.

Randomized block design (Fig. 3) groups cases into blocks, and we are interested in differences between cases within each block, but not in differences between blocks. Permutation of cases is done only within block, and block should be also used as a covariable in the constrained ordination model (as we are not interested in the effect of the block on the response variable, we need to partial this effect out). The possible number of permutations is limited by the number of treatments within the block and the number of blocks; if we have four treatments in the block, and we have 3 blocks, the possible number is $4!^3 = 13{,}824$.

Randomized block design

sample	block	treatment	rand1	rand2	rand3	randn
1	A	T1	2	1	3	...
2	A	T2	4	4	4	...
3	A	T3	3	3	1	...
4	A	T4	1	2	2	...
5	B	T1	7	7	8	...
6	B	T2	8	6	7	...
7	B	T3	6	8	6	...
8	B	T4	5	5	5	...
9	C	T1	10	11	9	...
10	C	T2	11	12	12	...
11	C	T3	12	9	11	...
12	C	T4	9	10	10	...

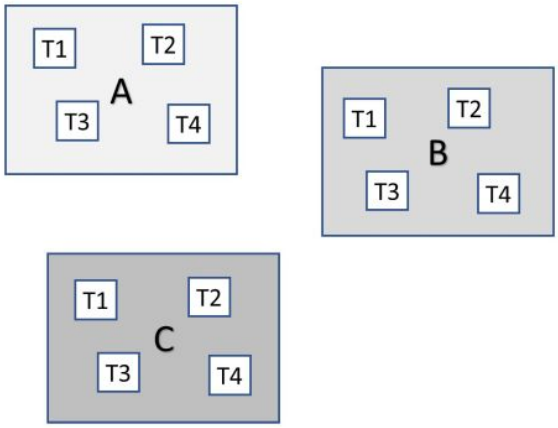


Figure 3: Randomized block design.

Hierarchical design (or nested design, split-plot design, Fig. 4) include hierarchically nested cases, *samples* nested within *plots*, or, in terminology of split-plot design, “split-plots” nested within “whole-plots”. The permutation can be either only within whole-plots, or the permutation of whole-plots without permuting split-plots, or combining both. With three samples within four plots, we get $3!^4 = 1296$ permutations if split-plots only are permuted, $4! = 24$ permutations if whole-plots only are permuted, and $3!^4*4 = 5184$ permutations if both are permuted. The number of whole-plots is clearly a limiting factor here.

Split-plot design

			Only whole-plots randomized				*Only split-plots randomized*				*Both whole- & split-plots randomized*			
sample	whole-plot	split-plot	rand1	rand2	rand3	randn	rand1	rand2	rand3	randn	rand1	rand2	rand3	randn
1	I	a	7	1	4	...	1	2	1	...	4	12	4	...
2	I	b	8	2	5	...	3	1	2	...	6	11	5	...
3	I	c	9	3	6	...	2	3	3	...	5	10	6	...
4	II	a	4	7	10	...	6	4	6	...	12	5	9	...
5	II	b	5	8	11	...	5	5	4	...	11	4	8	...
6	II	c	6	9	12	...	4	6	5	...	10	6	7	...
7	I	a	1	10	1	...	8	9	9	...	3	9	12	...
8	I	b	2	11	2	...	9	8	8	...	1	7	10	...
9	I	c	3	12	3	...	7	7	7	...	2	8	11	...
10	II	a	10	4	7	...	10	10	12	...	9	2	1	...
11	II	b	11	5	8	...	12	12	11	...	8	1	3	...
12	II	c	12	6	9	...	11	11	10	...	7	3	2	...

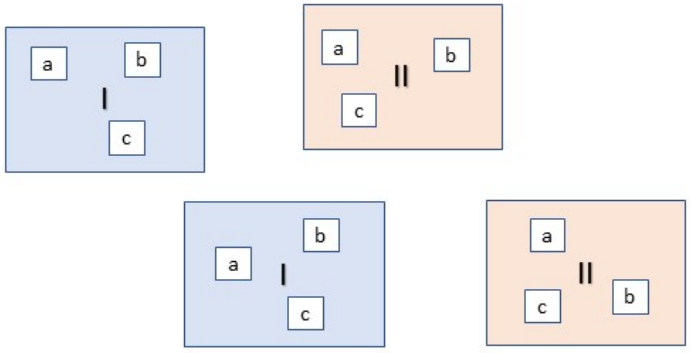
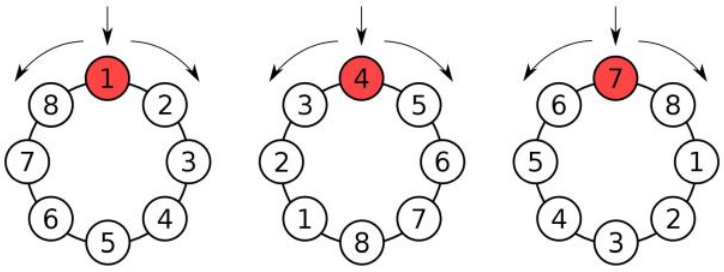


Figure 4: Split-plot design (hierarchical design, nested design).

In **linear transects** (observational studies, Fig. 5), samples located next to each other in space (or in time) are likely to be more similar than samples far from each other, imposing spatial (or temporal) dependence. To preserve this dependence, samples are reshuffled using circular permutation; ends of the transect are connected, and permutation is done by choosing the start by randomly rotating the circle. If the series is stationary, we can also mirror the permutations (e.g. in the transect with samples 1, 2, 3, 4, one circular permutation would be e.g. 3, 4, 1, 2, and one mirrored circular permutation would be e.g. 3, 2, 1, 4). The number of possible permutations equals to the number of samples if permutation is not mirrored, and to double of the number of samples if it is mirrored.

Linear transect (“series”, circular shift)

sample	rand1	rand2	rand3	randn
1	5	5	3	...
2	6	4	4	...
3	7	3	5	...
4	8	2	6	...
5	9	1	7	...
6	10	12	8	...
7	11	11	9	...
8	12	10	10	...
9	1	9	11	...
10	2	8	12	...
11	3	7	1	...
12	4	6	2	...



Circular shift (schema by G. Simpson)

Figure 5: Linear transect (series) design.

In **rectangular grid design** (Fig. 6), samples are located in grid with even distances to the neighbour in the same column and the same row. Permutation schema of regular rectangular grid is called toroidal shift; the opposite sides of the grid are folded into torus, and the torus is rotated in both directions. The number of possible permutations equals to the number of samples if mirroring is not allowed, and can be increased four times if mirroring is allowed.

Spatial grid design (toroidal shift)

sample	row	col	rand1	rand2	rand3	randn
1	1	1	2	11	6	...
2	2	1	3	10	4	...
3	3	1	1	12	5	...
4	1	2	5	2	3	...
5	2	2	6	1	1	...
6	3	2	4	3	2	...
7	1	3	8	5	12	...
8	2	3	9	4	10	...
9	3	3	7	6	11	...
10	1	4	11	8	9	...
11	2	4	12	7	7	...
12	3	4	10	9	8	...

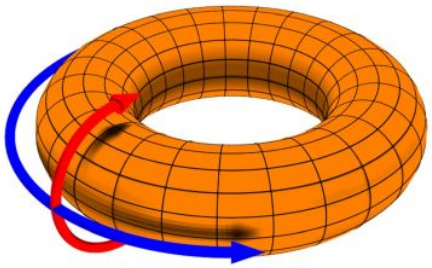
Position of original and randomized samples in the grid 3 x 4 cells

orig	col1	col2	col3	col4
row1	1	4	7	10
row2	2	5	8	11
row3	3	6	9	12

rand1	col1	col2	col3	col4
row1	2	5	8	11
row2	3	6	9	12
row3	1	4	7	10

rand2	col1	col2	col3	col4
row1	11	2	5	8
row2	10	1	4	7
row3	12	3	6	9

rand3	col1	col2	col3	col4
row1	6	3	12	9
row2	4	1	10	7
row3	5	2	11	8



Toroidal shift (schema by G. Simpson)

Figure 6: Rectangular grid design (lattice, grid).

Individual permutation schemas are also in this XLSX file:

permutation_schemas_2.xlsx

From:
https://anadat-r.davidzeleny.net/ - **Analysis of community ecology data in R**

Permanent link:
https://anadat-r.davidzeleny.net/doku.php/en:monte_carlo

Last update: **2026/02/18 05:54**