# Package: ptestR (via r-universe)

July 2, 2026

**Title** Permutation-Based Significance Testing for Regression Models

**Version** 0.1.1

**Description** Wraps glm(), lme4::lmer(), and binomial glm() with a permutation loop to compute nonparametric p-values. For each model, ptestR generates a null distribution of the test statistic by randomly rearranging the outcome variable, then computes p.perm as the proportion of permuted statistics at least as extreme as the observed one. This approach requires far fewer distributional assumptions than standard Wald or likelihood-ratio tests, making it well-suited to neuroimaging, EEG, and other biomedical datasets with repeated measures and small samples.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Depends** R (>= 4.0.0)

**Imports** broom, broom.mixed, dplyr, lme4, modelr, purrr, tidyr

**Suggests** ggplot2, knitr, lmerTest, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**URL** https://github.com/CoDe-Neuro/ptestR, https://code-neuro.github.io/ptestR

**BugReports** https://github.com/CoDe-Neuro/ptestR/issues

**Config/pak/sysreqs** cmake make libicu-dev

**Repository** https://code-neuro.r-universe.dev

**Date/Publication** 2026-05-31 13:24:36 UTC

**RemoteUrl** https://github.com/CoDe-Neuro/ptestR

**RemoteRef** v0.1.0

**RemoteSha** af7815cbc60aa39f3af6bac8576e0e86d79331b1

# Contents

grouped_perm_binoglm . . . . . 2
grouped_perm_glm . . . . . 3
grouped_perm_glmm . . . . . 4

**Index** **6**

---

`grouped_perm_binoglm` *Permutation test for binomial logistic regression*

---

## Description

`grouped_perm_binoglm` is a nonparametric test for binomial logistic regression. It assesses the significance of coefficients by permutation, computing the null distribution of the z-statistic by randomly rearranging the binary outcome variable.

## Usage

```
grouped_perm_binoglm(tbl, formla, var_to_perm, permNum = 1000, seed = 42)
```

## Arguments

| | |
|---|---|
| `tbl` | A data frame or tibble containing all model variables. |
| `formla` | A formula describing the logistic regression model; passed to `stats::glm()` with `family = binomial`. |
| `var_to_perm` | Character. Name of the binary outcome column to permute. |
| `permNum` | Integer. Number of permutations to generate. Default `1000`. |
| `seed` | Integer. Random seed for reproducibility; passed to `base::set.seed()`. Default `42`. |

## Value

A tibble with one row per model term and columns:

`term` Name of the regression term.

`estimate` Estimated log-odds coefficient.

`statistic` Observed z-statistic.

`p.value` Asymptotic two-sided p-value from the fitted model.

`p.perm` Permutation p-value: proportion of permuted |statistics| >= |observed statistic|. A value of `0` means no permuted statistic was as extreme; report as p < `1/permNum`.

## Examples

```
counts <- c(10, 11, 8, 9, 6, 3, 5, 1)
gender <- c(0, 0, 0, 0, 0, 1, 1, 1)
TBL    <- data.frame(counts, gender)
grouped_perm_binoglm(TBL, gender ~ counts, "gender", permNum = 500, seed = 1)
```

`grouped_perm_glm` *Permutation test for generalised linear models*

## Description

`grouped_perm_glm` is a nonparametric test for generalised linear models. It assesses the significance of coefficients by permutation, computing the distribution of the test statistic by randomly rearranging the outcome variable.

## Usage

```
grouped_perm_glm(
  tbl,
  formla,
  var_to_perm,
  family = gaussian,
  permNum = 1000,
  seed = 42
)
```

## Arguments

| | |
|---|---|
| `tbl` | A data frame or tibble containing all model variables. |
| `formla` | A formula describing the regression model to fit; passed directly to `stats::glm()`. |
| `var_to_perm` | Character. Name of the column to permute (typically the outcome variable). |
| `family` | A description of the error distribution and link function; passed to `stats::glm()`. Defaults to `stats::gaussian()`. |
| `permNum` | Integer. Number of permutations to generate. Default `1000`. |
| `seed` | Integer. Random seed for reproducibility; passed to `base::set.seed()`. Default `42`. |

## Value

A tibble with one row per model term and columns:

`term` Name of the regression term.

`estimate` Estimated coefficient.

`statistic` Observed t-statistic.

`p.value` Asymptotic two-sided p-value from the fitted model.

`p.perm` Permutation p-value: proportion of permuted |statistics| >= |observed statistic|. A value of `0` means no permuted statistic was as extreme; report as p < `1/permNum`.

## Examples

```
counts    <- sample(1:100, 9, replace = TRUE)
outcomes  <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
treatment <- gl(3, 3)
TBL <- data.frame(counts, outcomes, treatment)
grouped_perm_glm(TBL, outcomes ~ counts + treatment, "outcomes")
```

---

`grouped_perm_glmm` *Permutation test for linear mixed-effects models*

---

## Description

`grouped_perm_glmm` is a nonparametric test for linear mixed-effects models. It assesses the significance of fixed-effect coefficients by permutation, computing the null distribution of the test statistic by randomly rearranging the outcome variable while preserving the random-effects structure.

## Usage

```
grouped_perm_glmm(tbl, formla, var_to_perm, permNum = 1000, seed = 42)
```

## Arguments

| | |
|---|---|
| `tbl` | A data frame or tibble containing all model variables. |
| `formla` | A formula with both fixed- and random-effects parts; passed directly to `lme4::lmer()`. |
| `var_to_perm` | Character. Name of the column to permute (typically the outcome variable). |
| `permNum` | Integer. Number of permutations to generate. Default `1000`. |
| `seed` | Integer. Random seed for reproducibility; passed to `base::set.seed()`. Default `42`. |

## Value

A tibble with one row per fixed-effect term and columns:

`term` Name of the regression term.

`effect` Always `"fixed"` (random-parameter rows are dropped).

`estimate` Estimated coefficient.

`statistic` Observed t-statistic.

`p.perm` Permutation p-value: proportion of permuted |statistics| >= |observed statistic|. Replaces the conventional `p.value` because `lme4::lmer` does not compute degrees of freedom or p-values by default. A value of `0` means no permuted statistic was as extreme; report as `p < 1/permNum`.

## Examples

```
## Not run:
library(sdamr)
data("anchoring")
grouped_perm_glmm(
  anchoring,
  everest_feet ~ anchor + sex + (1 | referrer),
  "everest_feet"
)

## End(Not run)
```

# Index

base::set.seed(), *2–4*

formula, *2–4*

grouped_perm_binoglm, 2
grouped_perm_glm, 3
grouped_perm_glmm, 4

lme4::lmer(), *4*

stats::gaussian(), *3*
stats::glm(), *2*, *3*