

Package: ofemeantest (via r-universe)

September 20, 2026

Type Package

Title On Farm Experimentation Mean Test

Version 1.0.0.9000

Description Provides tools for comparing treatments in unreplicated on-farm experiments (OFE) using georeferenced data. The package includes functions for constructing spatial grids, estimating the effective sample size from spatial autocorrelation, performing repeated permutation tests, adjusting pairwise comparisons, and visualizing the results. It implements the OFE-mean test described in Córdoba, Paccioretti and Balzarini (2025) <[doi:10.1007/s11119-024-10206-0](https://doi.org/10.1007/s11119-024-10206-0)>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports multcompView, permuco, sf, spatialreg, spdep, withr

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

Depends R (>= 4.1.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://ppaccioretti.github.io/ofemeantest/>,
<https://github.com/PPaccioretti/ofemeantest>

BugReports <https://github.com/PPaccioretti/ofemeantest/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://ppaccioretti.r-universe.dev>

Date/Publication 2026-08-21 13:23:58 UTC

RemoteUrl <https://github.com/ppaccioretti/ofemeantest>

RemoteRef HEAD

RemoteSha cc3ea9db133a3c6f203e32e9585fa59140f26476

Contents

make_ofe_grid	2
ofe_f2	3
ofe_p	4
ofemt	4
plot_grid_selection	7
plot_pvalue_hist	9
print.ofemt_result	10

Index	12
--------------	-----------

make_ofe_grid	<i>Create an OFE grid and select valid cells</i>
---------------	--

Description

Convenience wrapper that builds a grid with [all_cells_grid()] and filters cells using [select_grid()]. It is the main entry point for generating grids for on-farm experiment (OFE) analysis.

Usage

```
make_ofe_grid(
  data,
  x,
  cellsize,
  min_per_cell = 1L,
  angle_deg = 0,
  buffer = 0,
  shift = c(0, 0),
  return_points = FALSE
)
```

Arguments

data	An ‘sf’ object of points. See [all_cells_grid()].
x	Character scalar with the treatment column name in ‘data’. See [select_grid()].
cellsize	Grid cell size. See [all_cells_grid()].
min_per_cell	Minimum observations per cell. See [select_grid()].
angle_deg	Grid rotation in degrees. See [all_cells_grid()].
buffer	Buffer applied to the data bounding box before gridding. See [all_cells_grid()].
shift	Offset applied to the grid origin. See [all_cells_grid()].
return_points	Logical. See [select_grid()].

Value

An 'ofe_grid' object returned by [select_grid()], containing the full grid, the selected cells, and associated metadata.

See Also

[all_cells_grid()], [select_grid()]

Examples

```
grid_ofe <- make_ofe_grid(
  ofe_f2,
  x = "Treatment",
  cellsize = 9,
  min_per_cell = 1L,
  angle_deg = 0,
  buffer = 0,
  shift = c(0, 0),
  return_points = TRUE
)
plot_grid_selection(grid_ofe)
```

ofe_f2

On-Farm single strip treatment trial (Field F2)

Description

The impact of broadcast phosphorus (P) fertilization on corn was evaluated. The Fertilized strip received superphosphate at rates ranging from 250 to 500 kg ha⁻¹ \[0-21-0\], while the Control strip received no P fertilizer (0 kg ha⁻¹). Each strip was 2.2 ha in area, totaling 100 ha.

Usage

```
ofe_f2
```

Format

A sf object with 3070 rows and 3 variables:

Treatment character, identifying Fertilized or Control strip

Yield_tn numeric, corn grain yield in tn/ha

geom sf geometry column

Details

Coordinate reference system is "WGS 84 / UTM zone 20S", EPSG:32720.

ofe_p	<i>On-Farm single strip treatment trial (companion dataset)</i>
-------	---

Description

Companion on-farm experiment used to demonstrate the package on a second field. Same layout as [ofe_f2]: a single fertilized strip compared against an adjacent control strip on dense yield-monitor data, with yields cleaned following Vega et al. (2019).

Usage

```
ofe_p
```

Format

A sf object with 3070 rows and 3 variables:

Treatment character, identifying Fertilized or Control strip

Yield_tn numeric, grain yield in tn/ha

geom sf geometry column

Details

Coordinate reference system is "WGS 84 / UTM zone 20S", EPSG:32720.

ofemt	<i>OFE permutation analysis</i>
-------	---------------------------------

Description

Analyzes unreplicated on-farm experiments to support field-specific inference about treatment effects. Spatial statistical methods are combined with permutation tests to compare the means of two or more treatments.

Usage

```
ofemt(
  data,
  y,
  x,
  cellsize = 10,
  min_per_cell = 4,
  n_p = 1000,
  n_s = 200,
  alpha = 0.05,
```

```

shift = c(0, 0),
p_adjust_method = c("none", "bonferroni", "holm", "BH"),
crs = NULL,
keep_components = c("none", "light", "full"),
grid = NULL,
angle_deg = 0,
buffer = 0,
seed = 7L
)

```

Arguments

<code>data</code>	An 'sf' object containing point geometries and the response and treatment columns specified in 'y' and 'x'.
<code>y</code>	response column (numeric).
<code>x</code>	treatment column (factor/character).
<code>cellsize</code> , <code>shift</code> , <code>angle_deg</code> , <code>buffer</code> , <code>min_per_cell</code>	grid settings (used when 'grid' is NULL).
<code>n_p</code>	number of permutations per ANOVA run.
<code>n_s</code>	number of sampling runs.
<code>alpha</code>	significance threshold for letters.
<code>p_adjust_method</code>	p-value adjustment method across pairwise comparisons. The adjustment is applied *within each sampling run*, across the 'choose(k, 2)' comparisons of that run; the reported 'p_adj' is the median of those per-run adjusted values (see Details).
<code>crs</code>	optional target projected CRS if 'data' is in lon/lat.
<code>keep_components</code>	What spatial components to embed in the result. One of: "none" (default) Nothing spatial is stored. Smallest object, but [plot_grid_selection()] and [plot.ofemt_result()] cannot be used on it. "light" Adds 'grid': the complete 'ofe_grid' object used for the analysis ('grid_all', 'grid_sel', 'cell_stats', 'params'). Enough to redraw the full grid and the selected cells, but the observations themselves are *not* stored, so no points can be overlaid. "full" Everything in "light", plus 'cell_medians' (one point per selected cell carrying the per-cell median response, its treatment and the ANOVA residual used for the spatial diagnostics) and 'points_joined' (every observation that fell inside a selected cell, with its 'CellID'). This is the option that lets 'plot()' overlay the raw points on the grid, which is the useful view when tuning 'shift', 'angle_deg', 'buffer' or 'cellsize'. Roughly, "light" costs one polygon per grid cell and "full" adds one row per observation.
<code>grid</code>	optional: an 'ofe_grid' (e.g., from [make_ofe_grid()]). If 'NULL', the grid is generated internally using 'cellsize', 'min_per_cell', 'shift', 'angle_deg' and 'buffer', and an informative message is emitted.

seed integer; controls reproducibility of the permutation sampling (the grid itself is deterministic from its construction arguments). Set 'seed = NULL' to let results vary across runs.

Details

The OFE-mean test accounts for spatial dependence when comparing treatments in unreplicated on-farm experiments. The procedure involves:

1. Aggregating the georeferenced observations within grid cells.
2. Estimating the spatial autocorrelation of the treatment-adjusted residuals.
3. Calculating the effective sample size (ESS) from the estimated spatial dependence.
4. Drawing repeated balanced subsamples whose size is determined by the ESS.
5. Performing pairwise permutation analysis of variance tests for each subsample.
6. Generating an empirical distribution of p-values for each treatment comparison.

The median of each empirical p-value distribution is reported as the p-value associated with the null hypothesis of no treatment effect.

For experiments with more than two treatments, all pairwise treatment comparisons are performed.

Multiplicity adjustment

When 'p_adjust_method != "none"', multiplicity adjustment is performed separately within each sampling run. [stats::p.adjust()] is applied to the 'choose(k, 2)' pairwise p-values obtained in that run, where 'k' is the number of treatments. The reported adjusted p-value is the median of the resulting empirical distribution of adjusted p-values.

The 'p_adj' column in 'perm_runs' contains the adjusted p-value from each sampling run. Therefore, [plot_pvalue_hist()] displays the empirical distribution used to calculate the reported median.

Compact letter display

Treatments are ordered by decreasing response before the compact letter display is generated. Treatments that do not share a letter are considered significantly different at the significance level specified by 'alpha'. Treatment labels are internally recoded before calling [multcompView::multcompLetters()] and subsequently restored. Consequently, labels containing spaces or special characters are preserved in the results.

Value

An object of class 'ofemt_result': a list with

'General information' One-row data frame with the cell size, the number of cells in the full grid and in the selection, the min/median/max number of observations per selected cell, the number of cells entering the analysis ('n'), the effective sample size ('ESS'), the spatial autocorrelation estimate ('Rho') and Moran's I^* .

'Cells per treatment' 'table' of selected cells per treatment.

'ANOVA permutation test' One row per pairwise comparison, with the median p-value across runs ('p_value') and its multiplicity-adjusted counterpart ('p_adj').

'Means comparison' One row per treatment, sorted by decreasing median response, with the compact letter display.

‘perm_runs’ The raw per-run output: ‘n_s * choose(k, 2)’ rows with columns ‘Trt_1’, ‘Trt_2’, ‘Comparison’, ‘p_value’ (the permutation p-value of that comparison in that run), ‘p_adj’ (the same value after adjusting within the run) and ‘run’ (run index, ‘1:n_s’). This is the empirical p-value distribution the method is built on; it is what [plot_pvalue_hist()] draws and what you would use to inspect the run-to-run variability behind the reported medians.

‘params’ The settings actually used (grid geometry, ‘n_p’, ‘n_s’, ‘alpha’, ‘p_adjust_method’, ‘seed’, and ‘grid_source’, which records whether the grid was supplied or built internally).

‘grid’, ‘cell_medians’, ‘points_joined’ Optional spatial components; see ‘keep_components’.

References

Córdoba, M., Paccioletti, P. and Balzarini, M. (2025). A new method to compare treatments in unreplicated on-farm experimentation. *Precision Agriculture*, 26, Article 4. doi:10.1007/s11119-024102060

See Also

[make_ofe_grid()], [plot_grid_selection()], [plot_pvalue_hist()]

Examples

```
res <- ofemt(ofe_f2, y = "Yield_tn", x = "Treatment",
            cellsize = 10, min_per_cell = 4, alpha = 0.05)

# Keep the geometries to inspect how the grid lands on the points
res <- ofemt(ofe_f2, y = "Yield_tn", x = "Treatment",
            cellsize = 10, min_per_cell = 4,
            keep_components = "full")
plot(res)
```

plot_grid_selection *Plot the grid, the selected cells and the observations*

Description

Draws the three layers that matter when tuning a grid, on one set of axes: the **full grid** in light grey, the **selected cells** (those that passed the single-treatment and ‘min_per_cell’ filters) shaded and outlined in black, and the **observations** as points. Seeing the points against the cell boundaries is the quickest way to judge the effect of ‘cellsize’, ‘shift’, ‘angle_deg’ and ‘buffer’.

Usage

```
plot_grid_selection(
  x,
  data = NULL,
  points = TRUE,
  main = NULL,
```

```

    legend = TRUE,
    legend_pos = NULL,
    point_size = NULL,
    engine = c("ggplot2", "base"),
    ...
)

## S3 method for class 'ofe_grid'
plot(x, ...)

## S3 method for class 'ofemt_result'
plot(x, ...)

```

Arguments

<code>x</code>	Either an <code>'ofe_grid'</code> (from <code>[make_ofe_grid()]</code>) or an <code>'ofemt_result'</code> (from <code>[ofemt()]</code>) that was run with <code>'keep_components = "light"'</code> or <code>"full"</code> . With an <code>'ofemt_result'</code> every layer is taken from the object itself, so no extra arguments are needed; the points are only available under <code>'keep_components = "full"'</code> .
<code>data</code>	Optional <code>'sf'</code> points to overlay. Rarely needed: the observations are taken from the object itself — <code>'points_sel'</code> for an <code>'ofe_grid'</code> built with <code>'make_ofe_grid(return_points = TRUE)'</code> , <code>'points_joined'</code> for an <code>'ofemt_result'</code> run with <code>'keep_components = "full"'</code> . Pass <code>'data'</code> when the object carries no points, or to override the stored ones.
<code>points</code>	Logical; set to <code>'FALSE'</code> to skip the point layer. When <code>'TRUE'</code> (the default) and no observations are available, a message explains how to obtain them rather than silently drawing a grid without points.
<code>main</code>	Plot title. Defaults to a one-line summary of the grid parameters, which is what makes successive calls comparable.
<code>legend</code>	Logical; draw the legend. Default <code>'TRUE'</code> .
<code>legend_pos</code>	Where to place the legend. Defaults to the engine's own sensible choice: <code>"right"</code> (outside the panel) for <code>"ggplot2"</code> , and <code>"topleft"</code> for <code>"base"</code> . Base-style keywords are translated for the ggplot2 engine, so <code>"bottomright"</code> works with either; <code>"none"</code> hides it.
<code>point_size</code>	Size of the observation dots. Defaults to <code>'0.15'</code> for the ggplot2 engine and <code>'0.35'</code> (as <code>'cex'</code>) for the base engine. Yield-monitor data runs to tens of thousands of points, where the default can still read as a solid mass — lower it to see the cell boundaries underneath.
<code>engine</code>	Which graphics system to draw with. <code>"ggplot2"</code> (the default) places the legend outside the plotting panel, so it can never sit on top of the data and the result does not depend on the device size. <code>"base"</code> uses base graphics and draws the legend inside the panel. If <code>**ggplot2**</code> is not installed the function falls back to <code>"base"</code> with a message.
<code>...</code>	Further arguments passed to the underlying <code>[plot()]</code> call for the full-grid layer. Base engine only; ignored by the ggplot2 engine.

Value

With `engine = "ggplot2"`, a `'ggplot'` object. In non-interactive contexts (e.g. scripts or inside `'pdf()'`), call `'print()'` on the returned object to render it. With `engine = "base"`, invisibly `'NULL'` — the function is called for the plot it draws.

See Also

`[make_ofe_grid()]`, `[ofemt()]`

Examples

```
g <- make_ofe_grid(ofe_f2, x = "Treatment", cellsize = 9, min_per_cell = 4)
plot_grid_selection(g, data = ofe_f2)
plot(g, data = ofe_f2)           # same thing

res <- ofemt(ofe_f2, y = "Yield_tn", x = "Treatment", cellsize = 9,
             keep_components = "full")
plot(res)                       # grid + selection + points, no extra args

# Base graphics instead, or a ggplot you keep customising
plot(res, engine = "base")
plot(res) + ggplot2::labs(subtitle = "Lote 2")
```

plot_pvalue_hist	<i>Plot histogram(s) of permutation p-values per comparison</i>
------------------	---

Description

Produces one histogram per pairwise comparison showing the empirical distribution of permutation **p*-values across the `'n_s'` sampling runs of an `[ofemt_result]`. Two reference lines are drawn on each panel: a solid line at the ***median*** of the distribution — the value `[ofemt()]` reports for that comparison — and a dashed line at the significance threshold ***alpha***.

Usage

```
plot_pvalue_hist(
  results,
  which = c("auto", "adjusted", "raw"),
  bins = 30,
  engine = c("ggplot2", "base")
)
```

Arguments

<code>results</code>	An object of class <code>'ofemt_result'</code> , typically obtained from <code>[ofemt()]</code> . Must contain <code>'perm_runs'</code> and <code>'params\$alpha'</code> .
----------------------	--

which	Which p-values to plot. "auto" (the default) uses the multiplicity-adjusted values whenever the analysis was run with 'p_adjust_method != "none"', and the raw values otherwise. "adjusted" and "raw" force one or the other. The adjustment is applied within each run before the histogram is built, so the median line coincides with the 'p_adj' reported in the 'ANOVA permutation test' table.
bins	Number of histogram bins. Default 30.
engine	Which graphics system to draw with, "ggplot2" (the default) or "base". If ggplot2 is not installed the function falls back to "base" with a message.

Value

With 'engine = "ggplot2"', a 'ggplot' object. With 'engine = "base"', invisibly 'NULL'.

Examples

```
res <- ofemt(ofe_f2, y = "Yield_tn", x = "Treatment", cellsize = 9,
            p_adjust_method = "bonferroni")
plot_pvalue_hist(res)           # adjusted p-values
plot_pvalue_hist(res, which = "raw")
plot_pvalue_hist(res, engine = "base")
```

print.ofemt_result	<i>Print method for 'ofemt_result' objects</i>
--------------------	--

Description

Displays a concise summary of an ['ofemt_result'] object, including general information about the experimental grid, descriptive statistics, and permutation-based test results.

Usage

```
## S3 method for class 'ofemt_result'
print(x, ...)
```

Arguments

x	An object of class ['ofemt_result'], typically returned by [ofemt()]. The object must contain, at minimum, a "General information" data frame. Optionally, it may include components such as "Means comparison" and "ANOVA permutation test".
...	Additional arguments passed to or from other methods (ignored).

Details

This print method provides a human-readable summary in the console:

- ****General information:**** Cell size, number of selected cells, and per-cell observation counts (minimum, median, maximum).
- ****Spatial statistics:**** Sample size ('n'), effective sample size ('ESS'), spatial correlation ('Rho'), and Moran's I^* .
- ****Means comparison:**** Shows the first 10 rows of the table of comparisons between treatment means.
- ****ANOVA permutation test:**** Displays pairwise tests summarizing median p^* -values and corrected p^* -values across permutation runs.

If any of these components are missing, they are simply skipped in the printed output.

Value

Invisibly returns the input object 'x', unchanged.

Examples

```
res <- ofemt(ofe_f2, y = "Yield_tn", x = "Treatment")
print(res)
```

Index

* datasets

ofe_f2, [3](#)

ofe_p, [4](#)

make_ofe_grid, [2](#)

ofe_f2, [3](#)

ofe_p, [4](#)

ofemt, [4](#)

plot.ofe_grid(plot_grid_selection), [7](#)

plot.ofemt_result
 (plot_grid_selection), [7](#)

plot_grid_selection, [7](#)

plot_pvalue_hist, [9](#)

print.ofemt_result, [10](#)