Package: stepwedgepower (via r-universe)

July 10, 2026

Type Package

Title Tools for Stepped-Wedge Clinical Trial Analysis and Power Simulation

Version 0.1.0

Maintainer Lin (Amanda) Li <`amandali14124277@gmail.com`>

Description Refactors an academic stepped-wedge clinical trial analysis script into reusable functions for physician-level data preparation, specialty-level rate modeling, and simulation-based power calculations with a random provider effects. The package was designed to support Eli Lilly Lp(a) grant support and more general stepped-wedge planning workflows.

License MIT + file LICENSE

Encoding UTF-8

Imports lme4

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Lin (Amanda) Li [aut, cre], Florin Vaida [aut] (degree: PhD)

Repository https://cran.r-universe.dev

Date/Publication 2026-07-10 20:50:08 UTC

RemoteUrl https://github.com/cran/stepwedgepower

RemoteRef HEAD

RemoteSha 226a004b116b6ab7c0265d6771c30a942cac3e53

Contents

`stepwedgepower-package` *stepwedgepower: Tools for stepped-wedge clinical trial analysis and power simulation*

Description

The **stepwedgepower** package refactors a one-off academic analysis script into reusable tools for physician-level data preparation, specialty-level rate modeling, and simulation-based planning for stepped-wedge studies.

`analyze_lpa_outcomes` *Reproduce the core Lp(a) outcome analyses*

Description

Fits the main outcome models from the original script for both overall Lp(a) testing and Lp(a) testing among patients with elevated LDL.

Usage

```
analyze_lpa_outcomes(
  data,
  provider_var = "prov_id",
  specialty_var = "specialty",
  outcomes = list(
    overall = list(successes = "n_lpa_pat", trials = "n_total_pat"),
    high_ldl = list(successes = "n_ldl_lpa_pat", trials = "n_ldl_pat")
  ),
  links = c("logit", "identity"),
  nAGQ = 10
)
```

Arguments

`data`	A physician-level analysis data frame.
`provider_var`	Provider identifier column.
`specialty_var`	Specialty column.
`outcomes`	Named list defining success and trial columns for each outcome.
`links`	Character vector of links to fit.
`nAGQ`	Number of quadrature points for `glmer()`.

Value

A nested list containing fitted models and specialty-rate tables.

`estimate_power` *Estimate power by repeated stepped-wedge simulation*

Description

Repeats the stepped-wedge simulation-and-analysis workflow and estimates power.

Usage

```
estimate_power(
  n_simulations = 100,
  alpha = 0.05,
  effect_size_or = 2,
  n_providers_per_specialty = c(40, 40, 40, 40) * 0.25,
  specialty_names = c("Cardiol", "IntMed", "FamMed", "Neurol"),
  tau_provider = 1.21,
  base_probs = c(0.07, 0.04, 0.03, 0.02),
  pts_per_step = 100/5,
  n_steps = length(n_providers_per_specialty) + 1L,
  fit_link = c("logit", "identity"),
  seed = NULL,
  nAGQ = 1
)
```

Arguments

`n_simulations`	Number of simulations.
`alpha`	Significance threshold.
`effect_size_or`	Odds ratio under the data-generating model.
`n_providers_per_specialty`	Provider counts by specialty.

`specialty_names`	Labels for the specialty groups.
`tau_provider`	Standard deviation of the provider random intercept.
`base_probs`	Baseline testing probabilities by specialty.
`pts_per_step`	Patients per provider per study step.
`n_steps`	Number of study steps.
`fit_link`	Link used when fitting the analysis model.
`seed`	Optional random seed.
`nAGQ`	Number of quadrature points for the fitted mixed model.

Value

A list with the estimated power and the vector of p-values.

`estimate_specialty_rates`

Estimate specialty-specific probabilities from a fitted model

Description

Extracts specialty-level probabilities from a fitted specialty model.

Usage

```
estimate_specialty_rates(
  model,
  specialty_levels = NULL,
  specialty_var = "specialty",
  link = c("logit", "identity"),
  approximate_marginal = TRUE,
  logit_scale_factor = 0.346
)
```

Arguments

`model`	A fitted model returned by `fit_specialty_rate_model`.
`specialty_levels`	Optional vector of specialty levels.
`specialty_var`	Name of the specialty column used in the model.
`link`	Link function for the fitted model.
`approximate_marginal`	Logical; whether to apply the random-intercept logit approximation.
`logit_scale_factor`	Approximation constant used in the shrinkage factor.

Value

A data frame with specialty-level linear predictors and probabilities.

`estimate_type1_error` *Estimate type I error by repeated stepped-wedge simulation*

Description

A convenience wrapper around `estimate_power` that sets the treatment odds ratio to 1.

Usage

```
estimate_type1_error(
  n_simulations = 100,
  alpha = 0.05,
  n_providers_per_specialty = c(40, 40, 40, 40) * 0.25,
  specialty_names = c("Cardiol", "IntMed", "FamMed", "Neurol"),
  tau_provider = 1.21,
  base_probs = c(0.07, 0.04, 0.03, 0.02),
  pts_per_step = 100/5,
  n_steps = length(n_providers_per_specialty) + 1L,
  fit_link = c("logit", "identity"),
  seed = NULL,
  nAGQ = 1
)
```

Arguments

`n_simulations`	Number of simulations.
`alpha`	Significance threshold.
`n_providers_per_specialty`	Provider counts by specialty.
`specialty_names`	Labels for the specialty groups.
`tau_provider`	Standard deviation of the provider random intercept.
`base_probs`	Baseline testing probabilities by specialty.
`pts_per_step`	Patients per provider per study step.
`n_steps`	Number of study steps.
`fit_link`	Link used when fitting the analysis model.
`seed`	Optional random seed.
`nAGQ`	Number of quadrature points for the fitted mixed model.

Value

A list like `estimate_power` with the type I error estimate added.

`fit_specialty_rate_model`

Fit a specialty-level testing-rate model

Description

Fits either a binomial GLM or a provider-random-intercept binomial GLMM for aggregated success/trial data.

Usage

```
fit_specialty_rate_model(
  data,
  successes,
  trials,
  specialty_var = "specialty",
  provider_var = NULL,
  link = c("logit", "identity"),
  random_intercept = !is.null(provider_var),
  nAGQ = 10
)
```

Arguments

`data`	A data frame containing counts and grouping variables.
`successes`	Name of the success-count column.
`trials`	Name of the trial-count column.
`specialty_var`	Name of the specialty column.
`provider_var`	Optional provider identifier column.
`link`	Link function. Supported values are `"logit"` and `"identity"`.
`random_intercept`	Logical; whether to include a provider random intercept.
`nAGQ`	Number of adaptive Gauss-Hermite quadrature points for `lme4::glmer()`.

Value

A fitted `glm` or `merMod` object.

`prepare_physician_data` *Prepare physician-level stepped-wedge analysis data*

Description

Filters the input data to the specialties of interest, applies panel-size thresholds, removes extreme outliers, and sorts the output.

Usage

```
prepare_physician_data(
  data,
  specialties = c("CARDIOLOGY", "FAMILY MEDICINE", "INTERNAL MEDICINE", "NEUROLOGY"),
  min_patients = 100,
  max_patients = 10000,
  specialty_var = "specialty",
  patient_var = "n_total_pat",
  provider_name_var = "PROV_NAME"
)
```

Arguments

`data`	A data frame.
`specialties`	Character vector of specialties to keep.
`min_patients`	Minimum total number of patients required.
`max_patients`	Maximum total number of patients allowed.
`specialty_var`	Name of the specialty column.
`patient_var`	Name of the total-patient count column.
`provider_name_var`	Name of the provider name column used for ordering.

Value

A filtered and sorted data frame.

`read_example_physician_data` *Read the bundled example physician data*

Description

Reads a small synthetic physician-level example dataset bundled with the package.

Usage

```
read_example_physician_data()
```

Value

A data frame.

`run_stepwedge_analysis` *Fit the stepped-wedge analysis model to a simulated dataset*

Description

Fits the mixed-effects stepped-wedge analysis model used in the original script.

Usage

```
run_stepwedge_analysis(
  sim_data,
  fit_link = c("logit", "identity"),
  nAGQ = 1
)
```

Arguments

`sim_data`	A compatible aggregated provider-step dataset.
`fit_link`	Link function used in the fitted model.
`nAGQ`	Number of quadrature points for `lme4::glmer()`.

Value

A list with the fitted model, coefficient table, and treatment p-value.

`simulate_stepwedge_trial`

Simulate one stepped-wedge trial dataset

Description

Generates aggregated provider-by-step binomial data for a sequential stepped-wedge design.

Usage

```
simulate_stepwedge_trial(
  effect_size_or = 1.5,
  n_providers_per_specialty = c(40, 40, 40, 40),
  specialty_names = c("Cardiol", "IntMed", "FamMed", "Neurol"),
  tau_provider = 1.21,
  base_probs = c(0.06, 0.04, 0.03, 0.02),
  pts_per_step = 20,
  n_steps = length(n_providers_per_specialty) + 1L,
  seed = NULL
)
```

Arguments

`effect_size_or`	Odds ratio for treatment under the data-generating model.
`n_providers_per_specialty`	Provider counts by specialty.
`specialty_names`	Labels for the specialty groups.
`tau_provider`	Standard deviation of the provider random intercept.
`base_probs`	Baseline testing probabilities for each specialty.
`pts_per_step`	Number of patients per provider per study step.
`n_steps`	Number of study steps.
`seed`	Optional random seed.

Value

A data frame with one row per provider-step combination.

`summarize_by_specialty` *Summarize physician counts by specialty*

Description

Computes common summary statistics for one or more numeric variables within each specialty.

Usage

```
summarize_by_specialty(
  data,
  specialty_var = "specialty",
  vars = c("n_total_pat", "n_ldl_pat"),
  na.rm = TRUE
)
```

Arguments

`data`	A data frame.
`specialty_var`	Name of the specialty column.
`vars`	Character vector of numeric variable names to summarize.
`na.rm`	Logical; whether to remove missing values.

Value

A data frame with one row per specialty-variable combination.

Index