Package: poissonexact (via r-universe)

November 9, 2026

Title Exact Conditional Inference for Poisson Regression

Version 0.1.0

Description Compute exact conditional maximum likelihood estimates, exact p-values, and exact confidence intervals for categorical effects in Poisson regression models with an optional offset and stratification. Nuisance parameters (stratum intercepts and other model effects not under test) are eliminated by conditioning on their sufficient statistics rather than relying on asymptotic approximations, which matters when case counts are small or sparse and standard Wald/likelihood-ratio inference is unreliable. Three algorithms are provided for different problem shapes: a fast closed-form method for a single two-level effect with strata as the only nuisance structure, following Guess, Lydick, Small and Miller (1987) <doi:10.1093/oxfordjournals.aje.a114535> and Guess and Thomas (1990) <doi:10.1097/00001648-199001000-00016>; a multivariate shift algorithm for arbitrary categorical covariates and strata, generalizing Hirji, Mehta and Patel (1987) <doi:10.2307/2289388> to bounded Poisson counts; and a network algorithm offering more aggressive mid-computation pruning for larger problems, generalizing Mehta, Patel and Senchaudhuri (1992) <doi:10.1080/10618600.1992.10474574> to bounded Poisson counts.

License Apache License (>= 2)

URL https://github.com/GSK-Biostatistics/poissonexact, https://gsk-biostatistics.github.io/poissonexact/

BugReports https://github.com/GSK-Biostatistics/poissonexact/issues

Depends R (>= 4.1.0)

Imports rlang, stats

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

VignetteBuilder knitr

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

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

Date/Publication 2026-11-09 14:45:18 UTC

RemoteUrl https://github.com/GSK-Biostatistics/poissonexact

RemoteRef HEAD

RemoteSha 8154c779d677f0f016ec0a4851454c12de0b195f

Contents

ep_closed_form 2
ep_multi_shift 4
ep_network 7
recompute_exact_inference 9

Index **12**

`ep_closed_form` *Exact conditional Poisson regression (fast, closed-form algorithm)*

Description

Computes the exact conditional maximum likelihood estimate, exact confidence interval, and exact p-value for the coefficient of a single two-level (treatment vs. control) factor in a Poisson regression with log offset and, optionally, one or more stratification variables acting purely as nuisance intercepts. This is the special case handled by `ep_multi_shift()` when the model contains exactly one exact effect with two levels and no other categorical covariates; in that special case, the exact distribution can be obtained in closed form, stratum by stratum, following Guess, Lydick, Small and Miller (1987) and Guess and Thomas (1990), instead of by the general per-observation network algorithm. See the accompanying theory note for why this is faster in this scenario, and use `ep_multi_shift()` whenever the model contains additional categorical covariates to adjust for, or more than one exact effect.

Usage

```
ep_closed_form(
  formula,
  data,
  offset = NULL,
  offset_log = TRUE,
  strata = NULL,
  ref_levels = list(),
  alpha = 0.05,
```

```
  ci_type = c("two-sided", "one-sided"),
  verbose = TRUE
)
```

Arguments

formula	A model formula of the form `response ~ group`, where `group` is the two-level factor of interest and is the formula's only right-hand-side term. Strata and the offset are supplied separately (as in `ep_multi_shift()`), not as terms in this formula.
data	A data frame. One row per observation; rows belonging to the same stratum and the same level of `group` are summed internally, so `data` may be individual-level or already aggregated.
offset	Name of the person-time / exposure variable in `data`, or `NULL` (default). If `NULL`, every observation is treated as contributing one unit of exposure (offset = 0 on the log scale), matching `ep_multi_shift()`. Note that, unlike the general algorithms, supplying `offset` here never imposes a per-observation cap on any observation's count (see Details).
offset_log	If `TRUE` (default, matching `ep_multi_shift()`), `offset` already holds log(exposure). If `FALSE`, `offset` holds raw exposure and is logged internally.
strata	Optional character vector naming one or more stratification variables in `data`. If `NULL` (default), all observations form a single stratum, i.e. an unstratified two-sample exact Poisson test.
ref_levels	Optional named list giving the reference (control) level of the formula's group variable, e.g. `list(trt = "0")`; the other level is treated as the treatment group. Defaults to R's usual first level.
alpha	Type I error rate for the confidence interval (default 0.05).
ci_type	`"two-sided"` or `"one-sided"`.
verbose	If `TRUE` (default), prints a short note identifying this as the closed-form two-group/strata-only algorithm.

Details

Unlike `ep_multi_shift()` and `ep_network()`, this function has no `exactmax` argument and never imposes a per-observation cap beyond what conditioning on the stratum total already implies (SAS's own "Rule 2"; see the theory vignette, Sections 5.2 and 10). In particular, supplying `offset` here does *not* trigger the offset-implied per-observation cap ("Rule 3": $S_i = \max(\lceil e^{o_i} \rceil, 1)$) that the general algorithms apply automatically whenever `offset` is supplied without `exactmax`. When that cap would bind, small exposures relative to the observed counts, typically in unstratified or lightly-stratified two-group data, `ep_closed_form()` and the general algorithms are, by design, answering subtly different questions, and can legitimately disagree on the estimate, confidence interval, and p-value even given identical call arguments. This is expected, documented behaviour, not a bug in either algorithm; see the theory vignette for a worked numerical example and `tests/testthat/test-cross-algorithm-agreement.R` for a regression test that locks in the expected divergence.

Value

An object of class `c("ep_closed_form", "ep_fit")`: a list with elements `exactparmest` (a one-row data frame with `Parameter`, `Estimate` (the coefficient beta = log rate ratio, not the rate ratio), `StdError`, `Alpha`, `LowerCL`, `UpperCL` (beta scale), `PValue`, and `SuffStat`), `OUTDIST` (the exact distribution of the sufficient statistic), and `call`.

References

Guess HA, Lydick EG, Small RD, Miller LP, Exact binomial confidence intervals for the relative risk in follow-up studies with sparsely stratified incidence density data, American Journal of Epidemiology, Volume 125, Issue 2, February 1987, Pages 340-347, https://doi.org/10.1093/oxfordjournals.aje.a114535

Guess HA, Thomas JE. A rapidly converging algorithm for exact binomial confidence intervals about the relative risk in follow-up studies with stratified incidence-density data. Epidemiology. 1990 Jan;1(1):75-7. doi: 10.1097/00001648-199001000-00016. PMID: 2081244.

Examples

```
df <- data.frame(
  cases = c(3, 32, 1, 19, 2, 24, 4, 40),
  fu = c(100, 100, 120, 120, 150, 150, 170, 170),
  trt = c(1, 0, 1, 0, 1, 0, 1, 0),
  strata = c(1, 1, 2, 2, 3, 3, 4, 4)
)
ep_closed_form(
  formula = cases ~ trt,
  data = df,
  offset = "fu",
  offset_log = FALSE,
  strata = "strata"
)
```

---

ep_multi_shift *Exact conditional Poisson regression (multivariate-shift algorithm)*

Description

Exact conditional inference for one or more categorical effects in a Poisson regression with a log link, an optional offset, and optional stratification. Nuisance parameters (stratum intercepts and every model effect not listed in `exact_vars`) are eliminated by conditioning on their sufficient statistics, and the resulting exact conditional distribution of each requested effect's sufficient statistic is obtained by the multivariate-shift algorithm: a per-stratum dynamic program combined with explicit cross-stratum convolution, generalizing the multivariate shift algorithm of Hirji, Mehta, and Patel (1987) for exact logistic regression to bounded Poisson counts, following the `EXACTMAX/OFFSET` rules documented for SAS `PROC GENMOD`'s `EXACT` statement.

Usage

```
ep_multi_shift(
  formula,
  data,
  exact_vars,
  offset = NULL,
  offset_log = TRUE,
  exactmax = NULL,
  strata = NULL,
  ref_levels = list(),
  alpha = 0.05,
  ci_type = c("two-sided", "one-sided"),
  chunk_size = 4000L,
  condition_on = c("model", "exact"),
  verbose = TRUE
)
```

Arguments

formula	A model formula.
data	A data frame.
exact_vars	Character vector of model *terms* to compute exact results for. Each entry must name a term that appears in formula, exactly as SAS's EXACT statement requires. An entry can be a main effect ("region") or an interaction of any order, named by joining every variable in it with * or : ("region:gender" and "region*gender" both work, and the variables may be listed in either order). Interaction columns are matched via the model's own terms()/assign structure, not by name-guessing, so a term is never confused with a different term that happens to share a text prefix (e.g. requesting "region" in a model that also contains region:gender selects only region's own main-effect columns). Requesting a term does *not* automatically request any other term built from the same variable(s): a main effect and its interactions are entirely independent entries, so testing region while leaving region:gender untested keeps the interaction as an ordinary nuisance covariate (conditioned on under condition_on = "model", never tested); to test the interaction itself, add it as its own entry, e.g. exact_vars = c("region", "region:gender").
offset	Name of an offset variable in data, or NULL (default). If NULL, every observation is treated as contributing one unit of exposure (offset = 0 on the log scale).
offset_log	If TRUE (default), offset already holds log(exposure); if FALSE, it holds raw exposure.
exactmax	NULL (not specified), TRUE (specified without a variable), or a character variable name (SAS EXACTMAX=).
strata	Optional character vector of stratification variable names in data.
ref_levels	Optional named list of reference levels for categorical variables.
alpha	Significance level for confidence intervals.

ci_type	"two-sided" (default) or "one-sided", matching ep_closed_form()'s argument of the same name. For "one-sided", only the confidence limit on the informative side is computed for each parameter (the other is left at its boundary, -Inf or Inf); which side is informative is determined per parameter from the two one-sided tail probabilities at the null (see two_sided_from_tails()'s direction), exactly as in ep_closed_form().
chunk_size	Maximum number of rows processed at once while combining two strata's coefficient tables. Combining strata requires pairing every row of one stratum's table with every row of the other's; chunk_size bounds how many such pairs are materialized simultaneously, trading memory for speed. The default is generous for typical problems; lower it if combining strata runs out of memory, or raise it if memory is plentiful and speed matters more.
condition_on	"model" (default; condition on every other model effect, matching SAS) or "exact" (condition only on the other requested exact_vars).
verbose	If TRUE (default), prints progress and which SAS rule determined the per-observation caps.

Details

formula may contain interaction terms (a*b, a:b, or higher-order interactions) freely, in two distinct roles. An interaction can be a plain adjustment covariate, left out of exact_vars entirely, in which case it is conditioned on like any other nuisance term (under the default condition_on = "model") but never itself tested. Or an interaction can be the effect of interest, including an interaction of the effect of interest with another covariate, by listing it as its own entry in exact_vars; see that argument's documentation for the exact naming rules. The two roles are independent: listing a main effect in exact_vars never implicitly tests its interactions, and listing an interaction never implicitly tests the main effects it is built from, each term to be tested must be named explicitly.

Value

An object of class c("ep_multi_shift", "ep_fit"): a list with elements exactparmest (a one-row data frame with Parameter, Estimate (the coefficient beta = log rate ratio, not the rate ratio), StdError, Alpha, LowerCL, UpperCL (beta scale), PValue, and SuffStat), OUTDIST (the exact distribution of the sufficient statistic), and call.

References

Hirji KF, Mehta CR, Patel NR. Computing Distributions for Exact Logistic Regression. JASA, 82(400):1110-1117, 1987.

Mehta CR, Patel NR. Exact logistic regression: theory and examples. Statistics in Medicine, 14(19):2143-2160, 1995.

Examples

```
df <- data.frame(
  group = c(1L, 1L, 1L, 0L, 0L, 0L),
  age = c(50L, 60L, 70L, 50L, 60L, 70L),
  subjects = c(3492, 2141, 1711, 3525, 2166, 1724),
  cases = c(3, 2, 1, 10, 7, 5),
```

```
  futime = c(50, 60, 40, 51, 62, 42)
)

df$lnfutime <- log(df$futime)

ep_multi_shift(
  formula = cases ~ group,
  data = df,
  offset = "lnfutime",
  offset_log = TRUE,
  strata = "age",
  exact_vars = "group",
  ref_levels = list(group="0", age="50"),
  alpha = 0.05,
  verbose = TRUE,
  ci_type = "two-sided"
)
```

---

ep_network *Exact conditional Poisson regression (network algorithm)*

Description

A structurally different, more aggressively pruned alternative to ep_multi_shift(), generalizing the network algorithm of Mehta, Patel, and Senchaudhuri (1992), SAS's own EXACTOPTIONS METHOD=NETWORK, to Poisson counts. This algorithm avoids constructing separate per-stratum tables and cross-joining them. Instead, it processes all observations across every stratum in a single, continuous forward pass. To optimize performance, it instantly discards any partial state the moment it becomes mathematically impossible to hit the observed nuisance total (either because it cannot reach the target or will inevitably overshoot it). These mid-stratum pruning boundaries are calculated just once per stratum using a closed-form backward induction, bypassing the need to wait for stratum boundaries or explicit cross-joins. Every argument, cap rule, interaction-naming convention, and output format is identical to ep_multi_shift(), this function reuses the same data-preparation, cap-rule, stratum-ordering, compute_stratum_max_add(), CI, p-value, and CMLE helpers; only the table-construction strategy differs. See ep_multi_shift()'s documentation for the full description of every argument. Unlike ep_multi_shift(), this function has no chunk_size argument: it never materializes a cross-join between two strata's tables for chunk_size to bound.

Usage

```
ep_network(
  formula,
  data,
  exact_vars,
  offset = NULL,
  offset_log = TRUE,
  exactmax = NULL,
  strata = NULL,
```

```
  ref_levels = list(),
  alpha = 0.05,
  ci_type = c("two-sided", "one-sided"),
  condition_on = c("model", "exact"),
  verbose = TRUE
)
```

Arguments

formula	A model formula.
data	A data frame.
exact_vars	Character vector of model *terms* to compute exact results for. Each entry must name a term that appears in formula, exactly as SAS's EXACT statement requires. An entry can be a main effect ("region") or an interaction of any order, named by joining every variable in it with * or : ("region:gender" and "region*gender" both work, and the variables may be listed in either order). Interaction columns are matched via the model's own terms()/assign structure, not by name-guessing, so a term is never confused with a different term that happens to share a text prefix (e.g. requesting "region" in a model that also contains region:gender selects only region's own main-effect columns). Requesting a term does *not* automatically request any other term built from the same variable(s): a main effect and its interactions are entirely independent entries, so testing region while leaving region:gender untested keeps the interaction as an ordinary nuisance covariate (conditioned on under condition_on = "model", never tested); to test the interaction itself, add it as its own entry, e.g. exact_vars = c("region", "region:gender").
offset	Name of an offset variable in data, or NULL (default). If NULL, every observation is treated as contributing one unit of exposure (offset = 0 on the log scale).
offset_log	If TRUE (default), offset already holds log(exposure); if FALSE, it holds raw exposure.
exactmax	NULL (not specified), TRUE (specified without a variable), or a character variable name (SAS EXACTMAX=).
strata	Optional character vector of stratification variable names in data.
ref_levels	Optional named list of reference levels for categorical variables.
alpha	Significance level for confidence intervals.
ci_type	"two-sided" (default) or "one-sided", matching ep_closed_form()'s argument of the same name. For "one-sided", only the confidence limit on the informative side is computed for each parameter (the other is left at its boundary, -Inf or Inf); which side is informative is determined per parameter from the two one-sided tail probabilities at the null (see two_sided_from_tails()'s direction), exactly as in ep_closed_form().
condition_on	"model" (default; condition on every other model effect, matching SAS) or "exact" (condition only on the other requested exact_vars).
verbose	If TRUE (default), prints progress and which SAS rule determined the per-observation caps.

Value

An object of class `c("ep_network", "ep_multi_shift", "ep_fit")`, in the same format as `ep_multi_shift()`.

References

Mehta CR, Patel NR, Senchaudhuri P. Exact Stratified Linear Rank Tests for Ordered Categorical and Binary Data. Journal of Computational and Graphical Statistics, 1(1):21-40, 1992.

Examples

```
df <- data.frame(
  group = c(1L, 1L, 1L, 0L, 0L, 0L),
  age = c(50L, 60L, 70L, 50L, 60L, 70L),
  subjects = c(3492, 2141, 1711, 3525, 2166, 1724),
  cases = c(3, 2, 1, 10, 7, 5),
  futime = c(50, 60, 40, 51, 62, 42)
)

df$lnfutime <- log(df$futime)

ep_network(
  formula = cases ~ group,
  data = df,
  offset = "lnfutime",
  offset_log = TRUE,
  strata = "age",
  exact_vars = "group",
  ref_levels = list(group="0", age="50"),
  alpha = 0.05,
  verbose = TRUE,
  ci_type = "two-sided"
)
```

---

recompute_exact_inference

Recompute a p-value and/or confidence interval from an already-fitted exact distribution

---

Description

Answers a *different* question about a parameter already fit by `ep_multi_shift()`, `ep_network()`, or `ep_closed_form()`; a different null value `beta0`, a different significance level `alpha`, or a different `ci_type`/`direction`, directly from that fit's `OUTDIST` table, without rebuilding the joint distribution. This is possible, and exact (not an approximation), because every algorithm's reference distribution is the same one-dimensional exponential-tilt family, $\Pr_\beta(T = t) \propto C(t)\, e^{\beta t}$, which sums to one at every value of β: re-tilting an already-built `C(t)` table at a new `beta0`, or re-deriving a confidence limit at a new `alpha`, only ever needs that one table, not a new enumeration over the model's stratification and nuisance structure. Building the joint table in the first place is the expensive part of every algorithm in this package; this function is the cheap part, made reusable.

Usage

```
recompute_exact_inference(
  fit,
  parm,
  beta0 = 0,
  alpha = 0.05,
  ci_type = c("two-sided", "one-sided"),
  direction = c("upper", "lower")
)
```

Arguments

`fit`	A fitted object returned by `ep_multi_shift()`, `ep_network()`, or `ep_closed_form()` (any object of class `"ep_fit"`), i.e. anything with an `OUTDIST` list and an `exactparmest` data frame in the format those three functions return.
`parm`	The name of one parameter already present in `fit$OUTDIST/fit$exactparmest$Parameter` (e.g. `"armvaccinated"`); matches the column name `exactparmest` reports for that term.
`beta0`	The null value of `beta` (the log rate ratio) to test against; `0` (the default) reproduces the same null the original fit itself tested. Supply a non-zero value to test a different margin, e.g. `log(0.5)` for a 50% rate-ratio margin.
`alpha`	Significance level for the confidence interval; need not match the `alpha` the original fit used.
`ci_type`	`"two-sided"` (default) or `"one-sided"`; need not match the original fit's own `ci_type`.
`direction`	For `ci_type = "one-sided"`: `"upper"` reports only the upper confidence limit (lower limit fixed at `-Inf`); `"lower"` reports only the lower limit (upper limit fixed at `Inf`). Ignored for `ci_type = "two-sided"`.

Value

A list with:

- `beta0`: the null value tested, echoed back.
- `p_lower`: the lower tail probability at `beta0`, $\Pr_{\beta_0}(T \leq t_{obs})$.
- `p_upper`: the upper tail probability at `beta0`, $\Pr_{\beta_0}(T \geq t_{obs})$.
- `pvalue_two_sided`: twice the smaller one-sided tail probability, capped at 1 (the package's two-sided exact p-value convention, matching SAS's `PROC GENMOD EXACT` statement).
- `pvalue_one_sided`: `min(p_lower, p_upper)`.
- `alpha`: the significance level used, echoed back.
- `LowerCL`, `UpperCL`: the recomputed confidence limits, on the same scale (`beta`, the log rate ratio) as `fit$exactparmest$Estimate`.

See Also

`ep_multi_shift()`, `ep_network()`, `ep_closed_form()`, and `vignette("Theory-Implementation")`'s discussion of reusing an already-built exact distribution.

Examples

```
df <- data.frame(
  group = c(1L, 1L, 1L, 0L, 0L, 0L),
  age = c(50L, 60L, 70L, 50L, 60L, 70L),
  subjects = c(3492, 2141, 1711, 3525, 2166, 1724),
  cases = c(3, 2, 1, 10, 7, 5),
  futime = c(50, 60, 40, 51, 62, 42)
)
df$lnfutime <- log(df$futime)

fit <- ep_multi_shift(
  formula = cases ~ group,
  data = df,
  offset = "lnfutime",
  offset_log = TRUE,
  strata = "age",
  exact_vars = "group",
  ref_levels = list(group = "0", age = "50"),
  verbose = FALSE
)

# Same question the fit itself already answered (beta0 = 0, alpha = 0.05):
# pvalue_two_sided/LowerCL/UpperCL should match fit$exactparmest exactly.
recompute_exact_inference(fit, "group1")

# A different confidence level, reusing the same OUTDIST table:
recompute_exact_inference(fit, "group1", alpha = 0.10)
```

Index

ep_closed_form, 2
ep_closed_form(), *6*, *8–10*
ep_multi_shift, 4
ep_multi_shift(), *2, 3*, *7*, *9, 10*
ep_network, 7
ep_network(), *3*, *9, 10*

recompute_exact_inference, 9