Package: OdysseusSurvivalModule (via r-universe)

October 26, 2026

Title Cohort-Based Single-Event Survival Utilities

Version 0.1.0

Description Tools to build single-event survival datasets from ``OMOP CDM'' cohorts and estimate survival outcomes. The package supports Kaplan-Meier, Cox proportional hazards, and parametric accelerated-failure-time models, with optional stratification by gender and age groups.

License Apache License (== 2.0)

Imports DatabaseConnector, ggsurvfit, SqlRender, survival

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Alexander Alexeyuk [aut, cre]

Maintainer Alexander Alexeyuk <AlexanderAlexeyuk@gmail.com>

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

Date/Publication 2026-04-03 16:59:47 UTC

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

RemoteRef HEAD

RemoteSha ace979ebe6ac631a53c28e25df67746429e5c2b7

Contents

singleEventSurvival 2

Index **5**

`singleEventSurvival` *Kaplan–Meier survival by simple strata (gender and/or age groups)*

Description

Compute Kaplan–Meier (KM) survival curves overall and, optionally, by simple strata derived from an existing `gender` column and/or an `age_group` built from list-of-range breaks using `age_years`. Results are returned per stratum plus an `"overall"` entry. Additionally, log-rank tests (overall and pairwise) are computed when strata are specified.

Usage

```
singleEventSurvival(
  survivalData,
  timeScale = "days",
  model = "km",
  covariates = NULL,
  strata = NULL,
  ageBreaks = list(c(0, 18), c(19, 45), c(46, 65), c(66, Inf)),
  times = NULL,
  probs = c(0.75, 0.5, 0.25),
  confInt = 0.95,
  confType = "log"
)
```

Arguments

`survivalData`	A `data.frame` with required columns: • `subject_id` (unique id) • `time` (numeric follow-up in **days**; finite) • `status` (0/1; 1 = event) Optional columns for stratification/age grouping: `gender`, `age_years`. Additional columns may be present but are currently unused.
`timeScale`	One of `"days"`, `"weeks"`, `"months"`, or `"years"`. Used only to scale the reported time axis; input `time` is assumed to be days.
`model`	Survival estimator to fit. One of `"km"` (Kaplan–Meier), `"cox"` (Cox PH, baseline hazard), `"weibull"`, `"exponential"`, `"lognormal"`, `"loglogistic"` (AFT parametric models via `survival::survreg()`).
`covariates`	Optional character vector of covariate column names used in Cox and parametric models. Ignored for `model = "km"`.
`strata`	Optional character vector of stratifying variables. Allowed: `"gender"`, `"age_group"`. If both are supplied, they are applied independently (gender OR age_group).
`ageBreaks`	A list of numeric length-2 vectors defining age ranges for auto-stratification, e.g. `list(c(0, 18), c(19, 45), c(46, 65), c(66, Inf))` -> `0-18`, `19-45`, `46-65`, `66+`. Used only if `"age_group"` is in `strata`.

times	Reserved for future enhancements; currently unused.
probs	Numeric vector of probabilities used to extract quantiles from KM curves. Default is c(0.75, 0.5, 0.25).
confInt	Numeric confidence level for KM intervals (e.g., 0.95); passed as conf.int to survival::survfit().
confType	Character string for KM CI type, one of "log", "log-log", "plain", "arcsin", "none"; passed as conf.type to survival::survfit().

Details

- Input follow-up time is supplied in **days** and internally rescaled to the requested timeScale for reporting ("days", "weeks", "months", "years").
- If "age_group" is included in strata, you must provide an age_years column. Age-group labels are generated from ageBreaks (e.g., 0-18, 19-45, 46-65, 66+), where each element is a numeric range c(min_age, max_age). Use Inf for open-ended upper bounds.
- Stratification is **simple**: groups are created from observed levels of gender and/or derived age_group. If both are requested, they are handled **separately** (gender OR age_group), not jointly. A one-sample KM curve is fit for each non-empty group, plus an "overall" curve for the full data.
- Confidence intervals are controlled by confType and confInt and are passed to survival::survfit().
- The model argument controls which survival estimator is fitted:
  - "km": non-parametric Kaplan–Meier estimate via ggsurvfit::survfit().
  - "cox": Cox PH model via survival::coxph() + survival::survfit(). Without covariates the Breslow baseline hazard is used. When covariates are provided, the survival curve is evaluated at the covariate means.
  - "weibull", "exponential", "lognormal", "loglogistic": AFT parametric models via survival::survreg(). S(t) is evaluated analytically at observed event times. Pointwise CIs are not available for parametric models (lower/upper are NA).
- covariates is used only for Cox and parametric models.
- times and probs control quantile extraction; probs defaults to c(0.75, 0.5, 0.25) (q75, median, q25).
- When strata are specified, a **log-rank test** is performed to compare survival curves across groups within each stratifier (gender and/or age_group). The overall test and pairwise tests are included in the returned object as tibbles.

Value

A list of class singleEventSurvival. See **Returned object**.

Returned object

A list of class singleEventSurvival. Elements include:

- Per-stratum entries named like "gender=F", "gender=M", "age_group=18-44", etc., and an "overall" element.

Each stratum element contains:

- data: a tibble with KM step data: time, n_risk, n_event, n_censor, survival, std_err, optional lower, upper (when confInt > 0), and derived hazard, cum_hazard, cum_event, cum_censor.
- summary: a list with n, events, censored, medianSurvival, q25Survival, q75Survival, meanSurvival, and timeScale.

Additionally, if gender is in strata, a logrank_test_gender element is included; if age_group is in strata, a logrank_test_age_group element is included. Each contains:

- testType: "overall" or "pairwise"
- stratum1, stratum2: labels of compared strata
- chisq: chi-square test statistic
- df: degrees of freedom
- pvalue: p-value for the test

Index

ggsurvfit::survfit(), *3*

singleEventSurvival, 2
survival::coxph(), *3*
survival::survfit(), *3*
survival::survreg(), *2*, *3*