Package ‘CohortSurvival’

May 7, 2026

Title Estimate Survival from Common Data Model Cohorts

Version 1.1.0

Description Estimate survival using data mapped to the Observational Medical Outcomes Partnership common data model. Survival can be estimated based on user-defined study cohorts.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports broom, CDMConnector (>= 2.0.0), cli, clock, DBI, dplyr, glue, omopgenerics (>= 1.1.0), PatientProfiles (>= 1.3.1), purrr, rlang, survival (>= 3.7.0), stats, stringr, tibble, tidyr

Suggests testthat (>= 3.0.0), CodelistGenerator, roxygen2, knitr, tictoc, rmarkdown, ggplot2, patchwork, cmprsk, duckdb, gt, flextable, scales, visOmopResults (>= 1.4.0), extrafont

Config/testthat/edition 3

Config/testthat/parallel true

VignetteBuilder knitr

URL `https://darwin-eu-dev.github.io/CohortSurvival/`

NeedsCompilation no

Author Kim López-Güell [aut, cre] (ORCID: <`https://orcid.org/0000-0002-8462-8668`>), Edward Burn [aut] (ORCID: <`https://orcid.org/0000-0002-9286-1128`>), Martí Català [aut] (ORCID: <`https://orcid.org/0000-0003-3308-9905`>), Xintong Li [aut] (ORCID: <`https://orcid.org/0000-0002-6872-5804`>), Danielle Newby [aut] (ORCID: <`https://orcid.org/0000-0002-3001-1478`>), Nuria Mercade-Besora [aut] (ORCID: <`https://orcid.org/0009-0006-7948-3747`>)

Maintainer Kim López-Güell <`kim.lopez@spc.ox.ac.uk`>

Depends R (>= 4.1.0)

Repository CRAN

Date/Publication 2025-12-04 09:40:08 UTC

Contents

`addCohortSurvival` *Add survival information to a cohort table*

Description

Add survival information to a cohort table

Usage

```
addCohortSurvival(
  x,
  cdm,
  outcomeCohortTable,
  outcomeCohortId = 1,
  outcomeDateVariable = "cohort_start_date",
  outcomeWashout = Inf,
  censorOnCohortExit = FALSE,
  censorOnDate = NULL,
  followUpDays = Inf,
  name = NULL
)
```

Arguments

`x`	cohort table to add survival information
`cdm`	CDM reference
`outcomeCohortTable`	The outcome cohort table of interest.
`outcomeCohortId`	ID of event cohorts to include. Only one outcome (and so one ID) can be considered. It can either be a cohort_definition_id value or a cohort_name.

`outcomeDateVariable`	Variable containing date of outcome event
`outcomeWashout`	Washout time in days for the outcome. If an individual has an outcome during the washout period, status and time will be set to NA
`censorOnCohortExit`	If TRUE, an individual's follow up will be censored at their cohort exit
`censorOnDate`	if not NULL, an individual's follow up will be censored at the given date
`followUpDays`	Number of days to follow up individuals (lower bound 1, upper bound Inf)
`name`	Name of the new table, if NULL a temporary table is returned.

Value

Two additional columns will be added to x. The "time" column will contain number of days to censoring. The "status" column will indicate whether the patient had the event (value: 1), or did not have the event (value: 0)

Examples

```
cdm <- mockMGUS2cdm()
cdm$mgus_diagnosis <- cdm$mgus_diagnosis |>
  addCohortSurvival(
    cdm = cdm,
    outcomeCohortTable = "death_cohort",
    outcomeCohortId = 1
  )
```

`asSurvivalResult` *A tidy implementation of the summarised_characteristics object.*

Description

A tidy implementation of the summarised_characteristics object.

Usage

```
asSurvivalResult(result)
```

Arguments

`result`	A summarised_characteristics object.

Value

A tibble with a tidy version of the summarised_characteristics object.

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(
  cdm = cdm,
  targetCohortTable = "mgus_diagnosis",
  targetCohortId = 1,
  outcomeCohortTable = "death_cohort",
  outcomeCohortId = 1,
  eventGap = 7
) |>
asSurvivalResult()
```

`availableSurvivalGrouping` *Variables that can be used for faceting and colouring survival plots*

Description

Variables that can be used for faceting and colouring survival plots

Usage

```
availableSurvivalGrouping(result, varying = FALSE)
```

Arguments

`result`	Survival results
`varying`	If FALSE (default), only variables with non-unique values will be returned, otherwise all available variables will be returned.

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(cdm,
                                    targetCohortTable = "mgus_diagnosis",
                                    outcomeCohortTable = "death_cohort")
availableSurvivalGrouping(surv)
```

`estimateCompetingRiskSurvival`

Estimate survival for a given event and competing risk of interest using cohorts in the OMOP Common Data Model

---

Description

Estimate survival for a given event and competing risk of interest using cohorts in the OMOP Common Data Model

Usage

```
estimateCompetingRiskSurvival(
  cdm,
  targetCohortTable,
  outcomeCohortTable,
  competingOutcomeCohortTable,
  targetCohortId = NULL,
  outcomeCohortId = NULL,
  outcomeDateVariable = "cohort_start_date",
  outcomeWashout = Inf,
  competingOutcomeCohortId = NULL,
  competingOutcomeDateVariable = "cohort_start_date",
  competingOutcomeWashout = Inf,
  censorOnCohortExit = FALSE,
  censorOnDate = NULL,
  followUpDays = Inf,
  strata = NULL,
  eventGap = 30,
  estimateGap = 1,
  restrictedMeanFollowUp = NULL,
  minimumSurvivalDays = 1
)
```

Arguments

`cdm`	CDM reference
`targetCohortTable`	The target cohort table of interest.
`outcomeCohortTable`	The outcome cohort table of interest.
`competingOutcomeCohortTable`	The competing outcome cohort table of interest.
`targetCohortId`	Target cohorts to include. It can either be a cohort_definition_id value or a cohort_name. Multiple ids are allowed.

`outcomeCohortId`	Outcome cohorts to include. It can either be a cohort_definition_id value or a cohort_name. Multiple ids are allowed.
`outcomeDateVariable`	Variable containing date of outcome event
`outcomeWashout`	Washout time in days for the outcome
`competingOutcomeCohortId`	Competing outcome cohorts to include. It can either be a cohort_definition_id value or a cohort_name. Multiple ids are allowed.
`competingOutcomeDateVariable`	Variable containing date of competing outcome event
`competingOutcomeWashout`	Washout time in days for the competing outcome
`censorOnCohortExit`	If TRUE, an individual's follow up will be censored at their cohort exit
`censorOnDate`	if not NULL, an individual's follow up will be censored at the given date
`followUpDays`	Number of days to follow up individuals (lower bound 1, upper bound Inf)
`strata`	strata
`eventGap`	Days between time points for which to report survival events, which are grouped into the specified intervals.
`estimateGap`	Days between time points for which to report survival estimates. First day will be day zero with risk estimates provided for times up to the end of follow-up, with a gap in days equivalent to eventGap.
`restrictedMeanFollowUp`	number of days of follow-up to take into account when calculating restricted mean for all cohorts
`minimumSurvivalDays`	Minimum number of days required for the main cohort to have survived

Value

tibble with survival information for desired cohort, including: time, people at risk, survival probability, cumulative incidence, 95 CIs, strata and outcome. A tibble with the number of events is outputted as an attribute of the output

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateCompetingRiskSurvival(
  cdm = cdm,
  targetCohortTable = "mgus_diagnosis",
  targetCohortId = 1,
  outcomeCohortTable = "progression",
  outcomeCohortId = 1,
  competingOutcomeCohortTable = "death_cohort",
  competingOutcomeCohortId = 1,
  eventGap = 7
```

```
)
```

`estimateSingleEventSurvival` *Estimate survival for a given event of interest using cohorts in the OMOP Common Data Model*

Description

Estimate survival for a given event of interest using cohorts in the OMOP Common Data Model

Usage

```
estimateSingleEventSurvival(
  cdm,
  targetCohortTable,
  outcomeCohortTable,
  targetCohortId = NULL,
  outcomeCohortId = NULL,
  outcomeDateVariable = "cohort_start_date",
  outcomeWashout = Inf,
  censorOnCohortExit = FALSE,
  censorOnDate = NULL,
  followUpDays = Inf,
  strata = NULL,
  eventGap = 30,
  estimateGap = 1,
  restrictedMeanFollowUp = NULL,
  minimumSurvivalDays = 1
)
```

Arguments

`cdm`	CDM reference
`targetCohortTable`	targetCohortTable
`outcomeCohortTable`	The outcome cohort table of interest.
`targetCohortId`	Target cohorts to include. It can either be a cohort_definition_id value or a cohort_name. Multiple ids are allowed.
`outcomeCohortId`	Outcome cohorts to include. It can either be a cohort_definition_id value or a cohort_name. Multiple ids are allowed.
`outcomeDateVariable`	Variable containing date of outcome event

`outcomeWashout`	Washout time in days for the outcome
`censorOnCohortExit`	If TRUE, an individual's follow up will be censored at their cohort exit
`censorOnDate`	if not NULL, an individual's follow up will be censored at the given date
`followUpDays`	Number of days to follow up individuals (lower bound 1, upper bound Inf)
`strata`	strata
`eventGap`	Days between time points for which to report survival events, which are grouped into the specified intervals.
`estimateGap`	Days between time points for which to report survival estimates. First day will be day zero with risk estimates provided for times up to the end of follow-up, with a gap in days equivalent to eventGap.
`restrictedMeanFollowUp`	number of days of follow-up to take into account when calculating restricted mean for all cohorts
`minimumSurvivalDays`	Minimum number of days required for the main cohort to have survived

Value

tibble with survival information for desired cohort, including: time, people at risk, survival probability, cumulative incidence, 95 CIs, strata and outcome. A tibble with the number of events is outputted as an attribute of the output

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(
  cdm = cdm,
  targetCohortTable = "mgus_diagnosis",
  targetCohortId = 1,
  outcomeCohortTable = "death_cohort",
  outcomeCohortId = 1,
  eventGap = 7
)
```

`mockMGUS2cdm` *Create mock CDM reference with survival::mgus2 dataset*

Description

Create mock CDM reference with survival::mgus2 dataset

Usage

```
mockMGUS2cdm()
```

Value

CDM reference containing data from the survival::mgus2 dataset

Examples

```
cdm <- mockMGUS2cdm()
cdm$person
```

`optionsTableSurvival` *Additional arguments for the function tableSurvival()*

Description

It provides a list of allowed inputs for .option argument in tableSurvival and their given default value.

Usage

```
optionsTableSurvival()
```

Value

The default .options named list.

Examples

```
{
optionsTableSurvival()
}
```

`plotSurvival` *Plot survival results*

Description

Plot survival results

Usage

```
plotSurvival(
  result,
  ribbon = TRUE,
  facet = NULL,
  colour = NULL,
  cumulativeFailure = FALSE,
  riskTable = FALSE,
  riskInterval = 30,
  logLog = FALSE,
  timeScale = "days",
  style = NULL
)
```

Arguments

`result`	Survival results
`ribbon`	If TRUE, the plot will join points using a ribbon
`facet`	Variables to use for facets
`colour`	Variables to use for colours
`cumulativeFailure`	whether to plot the cumulative failure probability instead of the survival probability
`riskTable`	Whether to print risk table below the plot
`riskInterval`	Interval of time to print risk table below the plot
`logLog`	If TRUE, the survival probabilities are transformed using the log-log formula
`timeScale`	The scale of time in the x-axis. Can be "days", "months", or "years"
`style`	A character string defining the visual theme to apply to the plot. You can set this to NULL to apply the standard ggplot2 default style, or provide a name for one of the package's pre-defined styles. Refer to the plotStyle() function for all available style pre-defined themes. For further customization, you can always modify the returned ggplot object directly.

Value

A plot of survival probabilities over time

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(cdm,
                                    targetCohortTable = "mgus_diagnosis",
                                    outcomeCohortTable = "death_cohort")
plotSurvival(surv)
```

`riskTable` *Table with survival events*

Description

Table with survival events

Usage

```
riskTable(
  x,
  eventGap = NULL,
  header = c("estimate"),
  type = "gt",
  groupColumn = NULL,
  hide = c("result_id", "estimate_type"),
  style = NULL,
  .options = list()
)
```

Arguments

`x`	Result from estimateSingleEventSurvival or estimateCompetingRiskSurvival.
`eventGap`	Event gap defining the times at which to report the risk table information. Must be one of the eventGap inputs used for the estimation function. If NULL, all available are reported.
`header`	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header. Elements in header can be: • Any of the columns returned by `tableColumns(result)` to create a header for these columns. • Any other input to create an overall header.
`type`	Character string specifying the desired output table format. See `tableType()` for supported table types. If `type = NULL`, global options (set via `setGlobalTableOptions()`) will be used if available; otherwise, a default `'gt'` table is created.
`groupColumn`	Columns to use as group labels, to see options use `tableColumns(result)`. By default, the name of the new group will be the tidy* column names separated by ";". To specify a custom group name, use a named list such as: list("newGroupName" = c("variable_name", "variable_level")). *tidy: The tidy format applied to column names replaces "_" with a space and converts to sentence case. Use `rename` to customise specific column names.
`hide`	Columns to drop from the output table. By default, `result_id` and `estimate_type` are always dropped.
`style`	Defines the visual formatting of the table. This argument can be provided in one of the following ways:

1. **Pre-defined style:** Use the name of a built-in style (e.g., `"darwin"`). See `tableStyle()` for available options.
2. **YAML file path:** Provide the path to an existing `.yml` file defining a new style.
3. **List of custome R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If `style = NULL`, the function will use global options (see `setGlobalTableOptions()`) or an existing `_brand.yml` file (if found); otherwise, the default style is applied. For more details, see the *Styles* vignette on the package website.

`.options`	A named list with additional formatting options. `visOmopResults::tableOptions()` shows allowed arguments and their default values.

Value

A tibble containing the risk table information (n_risk, n_events, n_censor) for all times within the event gap specified.

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(cdm,
                                    targetCohortTable = "mgus_diagnosis",
                                    outcomeCohortTable = "death_cohort")
riskTable(surv)
```

`tableSurvival` *Table with survival summary*

Description

Table with survival summary

Usage

```
tableSurvival(
  x,
  times = NULL,
  timeScale = "days",
  header = c("estimate"),
  estimates = c("median_survival", "restricted_mean_survival"),
  type = "gt",
  groupColumn = NULL,
  hide = c("result_id", "estimate_type"),
  style = NULL,
  .options = list()
)
```

Arguments

`x`	Result from estimateSingleEventSurvival or estimateCompetingRiskSurvival
`times`	Times at which to report survival in the summary table
`timeScale`	Time unit to report survival in: days, months or years
`header`	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header. Elements in header can be: • Any of the columns returned by `tableColumns(result)` to create a header for these columns. • Any other input to create an overall header.
`estimates`	Character vector specifying which estimates to include in the table. Options include: "median_survival", "restricted_mean_survival", "q0_survival", "q05_survival", "q25_survival", "q75_survival", "q95_survival", "q100_survival". By default it includes c("median_survival", "restricted_mean_survival").
`type`	Character string specifying the desired output table format. See `tableType()` for supported table types. If `type = NULL`, global options (set via `setGlobalTableOptions()`) will be used if available; otherwise, a default `'gt'` table is created.
`groupColumn`	Columns to use as group labels, to see options use `tableColumns(result)`. By default, the name of the new group will be the tidy* column names separated by ";". To specify a custom group name, use a named list such as: list("newGroupName" = c("variable_name", "variable_level")). *tidy: The tidy format applied to column names replaces "_" with a space and converts to sentence case. Use `rename` to customise specific column names.
`hide`	Columns to drop from the output table. By default, `result_id` and `estimate_type` are always dropped.
`style`	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. **Pre-defined style:** Use the name of a built-in style (e.g., `"darwin"`). See `tableStyle()` for available options. 2. **YAML file path:** Provide the path to an existing `.yml` file defining a new style. 3. **List of custome R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If `style = NULL`, the function will use global options (see `setGlobalTableOptions()`) or an existing `_brand.yml` file (if found); otherwise, the default style is applied. For more details, see the *Styles* vignette on the package website.
`.options`	A named list with additional formatting options. `visOmopResults::tableOptions()` shows allowed arguments and their default values.

Value

A tibble containing a summary of observed survival in the required units

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(cdm,
                                    targetCohortTable = "mgus_diagnosis",
                                    outcomeCohortTable = "death_cohort")
tableSurvival(surv, times = c(50,100,365))
```

`tableSurvivalAttrition` *Display the attrition of a survival result in a visual table*

Description

Display the attrition of a survival result in a visual table

Usage

```
tableSurvivalAttrition(
  result,
  type = "gt",
  header = "variable_name",
  groupColumn = c("cdm_name", "target_cohort", "variable_level"),
  hide = c("estimate_name"),
  style = NULL,
  .options = list()
)
```

Arguments

`result`	A summarised_result object obtained either from `estimateSingleEventSurvival()` or `estimateCompetingRiskSurvival()`.
`type`	Character string specifying the desired output table format. See `tableType()` for supported table types. If `type = NULL`, global options (set via `setGlobalTableOptions()`) will be used if available; otherwise, a default `'gt'` table is created.
`header`	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header. Elements in header can be: • Any of the columns returned by `tableColumns(result)` to create a header for these columns. • Any other input to create an overall header.
`groupColumn`	Columns to use as group labels, to see options use `tableColumns(result)`. By default, the name of the new group will be the tidy* column names separated by ";". To specify a custom group name, use a named list such as: list("newGroupName" = c("variable_name", "variable_level")). *tidy: The tidy format applied to column names replaces "_" with a space and converts to sentence case. Use `rename` to customise specific column names.

`hide`	Columns to drop from the output table. By default, `result_id` and `estimate_type` are always dropped.
`style`	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. **Pre-defined style:** Use the name of a built-in style (e.g., `"darwin"`). See `tableStyle()` for available options. 2. **YAML file path:** Provide the path to an existing `.yml` file defining a new style. 3. **List of custome R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If `style = NULL`, the function will use global options (see `setGlobalTableOptions()`) or an existing `_brand.yml` file (if found); otherwise, the default style is applied. For more details, see the *Styles* vignette on the package website.
`.options`	A named list with additional formatting options. `visOmopResults::tableOptions()` shows allowed arguments and their default values.

Value

A visual table

Examples

```
library(CohortSurvival)

cdm <- mockMGUS2cdm()

surv <- estimateSingleEventSurvival(
  cdm = cdm,
  targetCohortTable = "mgus_diagnosis",
  outcomeCohortTable = "death_cohort"
)

tableSurvivalAttrition(surv)
```

`tableSurvivalEvents` *Table with survival events*

Description

Table with survival events

Usage

```
tableSurvivalEvents(
  x,
  eventGap = NULL,
  header = c("estimate"),
  type = "gt",
  groupColumn = NULL,
  hide = c("result_id", "estimate_type"),
  style = NULL,
  .options = list()
)
```

Arguments

`x`	Result from estimateSingleEventSurvival or estimateCompetingRiskSurvival.
`eventGap`	Event gap defining the times at which to report the risk table information. Must be one of the eventGap inputs used for the estimation function. If NULL, all available are reported.
`header`	A vector specifying the elements to include in the header. The order of elements matters, with the first being the topmost header. Elements in header can be: • Any of the columns returned by `tableColumns(result)` to create a header for these columns. • Any other input to create an overall header.
`type`	Character string specifying the desired output table format. See `tableType()` for supported table types. If `type = NULL`, global options (set via `setGlobalTableOptions()`) will be used if available; otherwise, a default `'gt'` table is created.
`groupColumn`	Columns to use as group labels, to see options use `tableColumns(result)`. By default, the name of the new group will be the tidy* column names separated by ";". To specify a custom group name, use a named list such as: list("newGroupName" = c("variable_name", "variable_level")). *tidy: The tidy format applied to column names replaces "_" with a space and converts to sentence case. Use `rename` to customise specific column names.
`hide`	Columns to drop from the output table. By default, `result_id` and `estimate_type` are always dropped.
`style`	Defines the visual formatting of the table. This argument can be provided in one of the following ways: 1. **Pre-defined style:** Use the name of a built-in style (e.g., `"darwin"`). See `tableStyle()` for available options. 2. **YAML file path:** Provide the path to an existing `.yml` file defining a new style. 3. **List of custome R code:** Supply a block of custom R code or a named list describing styles for each table section. This code must be specific to the selected table type. If `style = NULL`, the function will use global options (see `setGlobalTableOptions()`) or an existing `_brand.yml` file (if found); otherwise, the default style is applied. For more details, see the *Styles* vignette on the package website.

`.options`	A named list with additional formatting options. `visOmopResults::tableOptions()` shows allowed arguments and their default values.

Value

A tibble containing the risk table information (n_risk, n_events, n_censor) for all times within the event gap specified.

Examples

```
cdm <- mockMGUS2cdm()
surv <- estimateSingleEventSurvival(cdm,
                                    targetCohortTable = "mgus_diagnosis",
                                    outcomeCohortTable = "death_cohort")
tableSurvivalEvents(surv)
```

Index