

Package ‘SEQTaRget’

January 23, 2026

Type Package

Title Sequential Trial Emulation

Version 1.3.4

Description Implementation of sequential trial emulation for the analysis of observational databases.

The 'SEQTaRget' software accommodates time-varying treatments and confounders, as well as binary and failure time outcomes. 'SEQTaRget' allows to compare both static and dynamic strategies, can be used to estimate observational analogs of intention-to-treat and per-protocol effects, and can adjust for potential selection bias induced by losses-to-follow-up. (Paper to come).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Suggests rmarkdown, testthat (>= 3.0.0)

Imports data.table, doFuture, doRNG, fastglm, future, future.apply, ggplot2, knitr, methods, stringr, survival, parallelly

Config/testthat/edition 3

Depends R (>= 4.1)

URL <https://causalinference.github.io/SEQTaRget/>

VignetteBuilder knitr

NeedsCompilation no

Author Ryan O'Dea [aut, cre] (ORCID: <<https://orcid.org/0009-0000-0103-9546>>),
Alejandro Szmulewicz [aut] (ORCID:
<<https://orcid.org/0000-0002-2664-802X>>),
Tom Palmer [aut] (ORCID: <<https://orcid.org/0000-0003-4655-4511>>, ROR:
<<https://ror.org/0524sp257>>),
Miguel Hernán [aut] (ORCID: <<https://orcid.org/0000-0003-1619-8456>>),
The President and Fellows of Harvard College [cph] (ROR:
<<https://ror.org/03vek6s52>>)

Maintainer Ryan O'Dea <ryan.odea@psi.ch>

Repository CRAN
Date/Publication 2026-01-23 19:50:33 UTC

Contents

compevent	2
covariates	3
denominator	3
diagnostics	4
hazard_ratio	4
km_curve	5
km_data	5
numerator	6
outcome	6
risk_comparison	7
risk_data	7
SEQdata.LTFU	8
SEQdata.multitreatment	8
SEQestimate	9
SEQopts	10
SEQoutput-class	14
SEquential	15
SEQ_data	16
show,SEQoutput-method	17
Index	18

compevent	<i>Function to return competing event models from a SEquential object</i>
-----------	---

Description

Function to return competing event models from a SEquential object

Usage

compevent(object)

Arguments

object SEQoutput object

Value

List of fastglm objects

covariates	<i>Retrieves Outcome, Numerator, and Denominator Covariates</i>
------------	---

Description

Retrieves Outcome, Numerator, and Denominator Covariates

Usage

covariates(object)

Arguments

object object of class SEQoutput

Value

List of SEQuential covariates

denominator	<i>Retrieves Denominator Models from SEQuential object</i>
-------------	--

Description

Retrieves Denominator Models from SEQuential object

Usage

denominator(object)

Arguments

object object of class SEQoutput

Value

List of both numerator models

diagnostics*Function to return diagnostic tables from a SEquential object*

Description

Function to return diagnostic tables from a SEquential object

Usage

```
diagnostics(object)
```

Arguments

object SEQoutput object

Value

List of diagnostic tables

hazard_ratio*Function to return hazard ratios from a SEquential object*

Description

Function to return hazard ratios from a SEquential object

Usage

```
hazard_ratio(object)
```

Arguments

object SEQoutput object

Value

List of hazard ratios

km_curve	<i>Function to print Kaplan-Meier curves</i>
----------	--

Description

Function to print Kaplan-Meier curves

Usage

```
km_curve(  
  object,  
  plot.type = "survival",  
  plot.title,  
  plot.subtitle,  
  plot.labels,  
  plot.colors  
)
```

Arguments

object	SEQoutput object to plot
plot.type	character: type of plot to print
plot.title	character: defines the title of the plot
plot.subtitle	character: plot subtitle
plot.labels	length 2 character: plot labels
plot.colors	length 2 character: plot colors

Value

ggplot object of plot plot.type

km_data	<i>Function to return survival data from a SEquential object</i>
---------	--

Description

Function to return survival data from a SEquential object

Usage

```
km_data(object)
```

Arguments

object	SEQoutput object
--------	------------------

Value

List of dataframes of survival values

numerator	<i>Retrieves Numerator Models from SEquential object</i>
-----------	--

Description

Retrieves Numerator Models from SEquential object

Usage

numerator(object)

Arguments

object object of class SEQoutput

Value

List of both numerator models

outcome	<i>Retrieves Outcome Models from SEquential object</i>
---------	--

Description

Retrieves Outcome Models from SEquential object

Usage

outcome(object)

Arguments

object object of class SEQoutput

Value

List of all outcome models

risk_comparison	<i>Function to return risk information from a SEquential object</i>
-----------------	---

Description

Function to return risk information from a SEquential object

Usage

```
risk_comparison(object)
```

Arguments

object	SEQoutput object
--------	------------------

Value

A data frame of risk information at end of followup (risk ratios, risk differences and confidence intervals, if bootstrapped)

risk_data	<i>Function to return risk information from a SEquential object</i>
-----------	---

Description

Function to return risk information from a SEquential object

Usage

```
risk_data(object)
```

Arguments

object	SEQoutput object
--------	------------------

Value

A data table of risk information at the end of followup

 SEQdata.LTFU

Simulated lost-to-followup example data for [SEquential\(\)](#)

Description

Simulated lost-to-followup example data for [SEquential\(\)](#)

Usage

SEQdata.LTFU

Format

A dataframe with 4,139 rows and 13 columns:

ID Integer: Unique ID emulating individual patients

time Integer: Time of observation, always begins at 0, max time of 59; however, if lost-to-followup, time is truncated at a random point

eligible Binary: eligibility criteria for timepoints

outcome Binary: If an outcome is observed at this time point

tx_init Binary: If treatment is observed at this time point

sex Binary: Sex of the emulated patient

N Numeric: Normal random variable from $N(10,5)$

L Numeric: 4% continuously increase from $U(0, 1)$

P Numeric: 2% continuously decrease from $U(9, 10)$

excusedOne Binary: Once one, always one variable emulating an excuse for treatment switch

excusedZero Binary: Once zero, always zero variable emulating an excuse for treatment switch

LTFU Binary: Flag for losing a simulated ID to followup, if 1 there are no more records of the ID afterwards

 SEQdata.multitreatment

Simulated multitreatment example data for [SEquential\(\)](#) multinomial models

Description

Simulated multitreatment example data for [SEquential\(\)](#) multinomial models

Usage

SEQdata.multitreatment

Format

A dataframe with 5,976 rows and 11 columns:

ID Integer: Unique ID emulating individual patients

time Integer: Time of observation, always begins at 0, max time of 59; however, if lost-to-followup, time is truncated at a random point

eligible Binary: eligibility criteria for timepoints

outcome Binary: If an outcome is observed at this time point

tx_init Integer: Which treatment is observed at this time point

sex Binary: Sex of the emulated patient

N Numeric: Normal random variable from $N(10,5)$

L Numeric: 4% continuously increase from $U(0, 1)$

P Numeric: 2% continuously decrease from $U(9, 10)$

excusedOne Binary: Once one, always one variable emulating an excuse for treatment switch

excusedZero Binary: Once zero, always zero variable emulating an excuse for treatment switch

SEQestimate	<i>Estimate the (very rough) time to run SEquential analysis on current machine</i>
-------------	---

Description

Estimate the (very rough) time to run SEquential analysis on current machine

Usage

```
SEQestimate(
  data,
  id.col,
  time.col,
  eligible.col,
  treatment.col,
  outcome.col,
  time_varying.cols = list(),
  fixed.cols = list(),
  method,
  options,
  verbose = TRUE
)
```

Arguments

<code>data</code>	data.frame or data.table, if not already expanded with SEQexpand() , will perform expansion according to arguments passed to either <code>params</code> or <code>...</code>
<code>id.col</code>	String: column name of the id column
<code>time.col</code>	String: column name of the time column
<code>eligible.col</code>	String: column name of the eligibility column
<code>treatment.col</code>	String: column name of the treatment column
<code>outcome.col</code>	String: column name of the outcome column
<code>time_varying.cols</code>	List: column names for time varying columns
<code>fixed.cols</code>	List: column names for fixed columns
<code>method</code>	String: method of analysis to perform
<code>options</code>	List: optional list of parameters from SEQopts()
<code>verbose</code>	Logical: if TRUE, cats progress to console, default is TRUE

Value

A list of (very rough) estimates for the time required for SEquential containing:

- `modelTime` estimated time used when running models
- `expansionTime` estimated time used when expanding data
- `totalTime` sum of model and expansion time

SEQopts

Parameter Builder for SEquential Model and Estimates

Description

Parameter Builder for SEquential Model and Estimates

Usage

```
SEQopts(
  bootstrap = FALSE,
  bootstrap.nboot = 100,
  bootstrap.sample = 0.8,
  bootstrap.CI = 0.95,
  bootstrap.CI_method = "se",
  cense = NA,
  cense.denominator = NA,
  cense.eligible = NA,
  cense.numerator = NA,
  compevent = NA,
```

```
covariates = NA,
data.return = FALSE,
denominator = NA,
deviation = FALSE,
deviation.col = NA,
deviation.conditions = c(NA, NA),
deviation.excused = FALSE,
deviation.excused_cols = c(NA, NA),
excused = FALSE,
excused.cols = c(NA, NA),
fastglm.method = 2L,
followup.class = FALSE,
followup.include = TRUE,
followup.max = Inf,
followup.min = -Inf,
followup.spline = FALSE,
hazard = FALSE,
indicator.baseline = "_bas",
indicator.squared = "_sq",
km.curves = FALSE,
multinomial = FALSE,
ncores = availableCores(omit = 1L),
nthreads = getDTthreads(),
numerator = NA,
parallel = FALSE,
plot.colors = c("#F8766D", "#00BFC4", "#555555"),
plot.labels = NA,
plot.subtitle = NA,
plot.title = NA,
plot.type = "survival",
seed = NULL,
selection.first_trial = FALSE,
selection.prob = 0.8,
selection.random = FALSE,
subgroup = NA,
survival.max = Inf,
treat.level = c(0, 1),
trial.include = TRUE,
visit = NA,
visit.denominator = NA,
visit.numerator = NA,
weight.eligible_cols = c(),
weight.lower = -Inf,
weight.lag_condition = TRUE,
weight.p99 = FALSE,
weight.preexpansion = TRUE,
weight.upper = Inf,
weighted = FALSE
```

)

Arguments

<code>bootstrap</code>	Logical: defines if <code>SEQential()</code> should run bootstrapping, default is <code>FALSE</code>
<code>bootstrap.nboot</code>	Integer: number of bootstraps, default is 100
<code>bootstrap.sample</code>	Numeric: percentage of data to use when bootstrapping, should be in [0, 1], default is 0.8
<code>bootstrap.CI</code>	Numeric: defines the confidence interval after bootstrapping, default is 0.95 (95% CI)
<code>bootstrap.CI.method</code>	Character: selects which way to calculate bootstraps confidence intervals ("se", "percentile"), default is "se"
<code>cense</code>	String: column name for additional censoring variable, e.g. loss-to-follow-up
<code>cense.denominator</code>	String: censoring denominator covariates to the right hand side of a formula object
<code>cense.eligible</code>	String: column name for indicator column defining which rows to use for censoring model
<code>cense.numerator</code>	String: censoring numerator covariates to the right hand side of a formula object
<code>compevent</code>	String: column name for competing event indicator
<code>covariates</code>	String: covariates to the right hand side of a formula object
<code>data.return</code>	Logical: whether to return the expanded dataframe with weighting information, default is <code>FALSE</code>
<code>denominator</code>	String: denominator covariates to the right hand side of a to formula object
<code>deviation</code>	Logical: create switch based on deviation from column <code>deviation.col</code> , default is <code>FALSE</code>
<code>deviation.col</code>	Character: column name for deviation
<code>deviation.conditions</code>	Character list: RHS evaluations of the same length as <code>treat.levels</code>
<code>deviation.excused</code>	Logical: whether deviations should be excused by <code>deviation.excused_cols</code> , default is <code>FALSE</code>
<code>deviation.excused_cols</code>	Character list: excused columns for deviation switches
<code>excused</code>	Logical: in the case of censoring, whether there is an excused condition, default is <code>FALSE</code>
<code>excused.cols</code>	List: list of column names for treatment switch excuses - should be the same length, and ordered the same as <code>treat.level</code>
<code>fastglm.method</code>	Integer: decomposition method for <code>fastglm</code> (1-QR, 2-Cholesky, 3-LDLT, 4-QR.FPIV), default is 2L

<code>followup.class</code>	Logical: treat followup as a class, e.g. expands every time to it's own indicator column, default is FALSE
<code>followup.include</code>	Logical: whether or not to include 'followup' and 'followup_squared' in the outcome model, default is TRUE
<code>followup.max</code>	Numeric: maximum time to expand about, default is Inf (no maximum)
<code>followup.min</code>	Numeric: minimum time to expand about, default is -Inf (no minimum)
<code>followup.spline</code>	Logical: treat followup as a cubic spline, default is FALSE
<code>hazard</code>	Logical: hazard error calculation instead of survival estimation, default is FALSE
<code>indicator.baseline</code>	String: identifier for baseline variables in covariates, numerator, denominator - intended as an override
<code>indicator.squared</code>	String: identifier for squared variables in covariates, numerator, denominator - intended as an override
<code>km.curves</code>	Logical: Kaplan-Meier survival curve creation and data return, default is FALSE
<code>multinomial</code>	Logical: whether to expect multilevel treatment values, default is FALSE
<code>ncores</code>	Integer: number of cores to use in parallel processing, default is one less than system max, see parallelly::availableCores()
<code>nthreads</code>	Integer: number of threads to use for data.table processing, default is data.table::getDTthreads()
<code>numerator</code>	String: numerator covariates to the right hand side of a formula object
<code>parallel</code>	Logical: define if the SEquential process is run in parallel, default is FALSE
<code>plot.colors</code>	Character: Colors for output plot if km.curves = TRUE, defaulted to ggplot2 defaults
<code>plot.labels</code>	Character: Color labels for output plot if km.curves = TRUE in order e.g. <code>c("risk.0", "risk.1")</code>
<code>plot.subtitle</code>	Character: Subtitle for output plot if km.curves = TRUE
<code>plot.title</code>	Character: Title for output plot if km.curves = TRUE
<code>plot.type</code>	Character: Type of plot to create if km.curves = TRUE, available options are 'survival' (the default), 'risk', and 'inc' (in the case of censoring)
<code>seed</code>	Integer: starting seed
<code>selection.first_trial</code>	Logical: selects only the first eligible trial in the expanded dataset, default FALSE
<code>selection.prob</code>	Numeric: percent of total IDs to select for selection.random, should be bound [0, 1], default is 0.8
<code>selection.random</code>	Logical: randomly selects IDs with replacement to run analysis, default FALSE
<code>subgroup</code>	Character: Column name to stratify outcome models on
<code>survival.max</code>	Numeric: maximum time for survival curves, default is Inf (no maximum)
<code>treat.level</code>	List: treatment levels to compare, default is <code>c(0, 1)</code>

<code>trial.include</code>	Logical: whether or not to include 'trial' and 'trial_squared' in the outcome model, default is TRUE
<code>visit</code>	String: column name for visit indicator variable, e.g. "visit"
<code>visit.denominator</code>	String: visit denominator covariates to the right hand side of a formula object
<code>visit.numerator</code>	String: visit numerator covariates to the right hand side of a formula object
<code>weight.eligible_cols</code>	List: list of column names for indicator columns defining which weights are eligible for weight models - in order of <code>treat.level</code>
<code>weight.lower</code>	Numeric: weights truncated at lower end at this weight, default is <code>-Inf</code>
<code>weight.lag_condition</code>	Logical: whether weights should be conditioned on treatment lag value, default TRUE
<code>weight.p99</code>	Logical: forces weight truncation at 1st and 99th percentile weights, will override provided <code>weight.upper</code> and <code>weight.lower</code>
<code>weight.preexpansion</code>	Logical: whether weighting should be done on pre-expanded data, default TRUE
<code>weight.upper</code>	Numeric: weights truncated at upper end at this weight, default is <code>Inf</code>
<code>weighted</code>	Logical: whether or not to preform weighted analysis, default is FALSE

Value

An object of class 'SEQopts'

SEQoutput-class	<i>An S4 class used to hold the outputs for the SEquential process</i>
-----------------	--

Description

An S4 class used to hold the outputs for the SEquential process

Slots

`params` SEQparams object
`outcome` outcome covariates
`numerator` numerator covariates
`denominator` denominator covariates
`outcome.model` list of length `bootstrap.nboot` containing outcome coefficients
`hazard` hazard ratio
`survival.curve` ggplot object for the survival curves
`survival.data` data.table of survival data

`risk.difference` risk difference calculated from survival data
`risk.ratio` risk ratio calculated from survival data
`time` time in minutes used for the SEquential process
`weight.statistics` information from the weighting process, containing weight coefficients and weight statistics
`info` list of outcome and switch information (if applicable)
`ce.model` list of competing event models if `compevent` is specified, NA otherwise

SEquential

SEquential trial emulation

Description

SEquential is an all-in-one API to SEquential analysis, returning a `SEQoutput` object of results. More specific examples can be found on pages at <https://causalinference.github.io/SEQTaRget/>

Usage

```

SEquential(
  data,
  id.col,
  time.col,
  eligible.col,
  treatment.col,
  outcome.col,
  time_varying.cols = list(),
  fixed.cols = list(),
  method,
  options,
  verbose = TRUE
)

```

Arguments

<code>data</code>	data.frame or data.table, will preform expansion according to arguments passed through the options argument
<code>id.col</code>	String: column name of the id column
<code>time.col</code>	String: column name of the time column
<code>eligible.col</code>	String: column name of the eligibility column
<code>treatment.col</code>	String: column name of the treatment column
<code>outcome.col</code>	String: column name of the outcome column
<code>time_varying.cols</code>	List: column names for time varying columns

fixed.cols	List: column names for fixed columns
method	String: method of analysis to preform; should be one of "ITT", "dose-response", or "censoring"
options	List: optional list of parameters from SEQopts()
verbose	Logical: if TRUE, cats progress to console, default is TRUE

Details

Implementation of sequential trial emulation for the analysis of observational databases. The SE-Quential software accommodates time-varying treatments and confounders, as well as binary and failure time outcomes. SEQ allows to compare both static and dynamic strategies, can be used to estimate observational analogs of intention-to-treat and per-protocol effects, and can adjust for potential selection bias induced by losses-to-follow-up.

Value

An S4 object of class SEQoutput

Examples

```
data <- SEQdata
model <- SEquential(data, id.col = "ID",
                    time.col = "time",
                    eligible.col = "eligible",
                    treatment.col = "tx_init",
                    outcome.col = "outcome",
                    time_varying.cols = c("N", "L", "P"),
                    fixed.cols = "sex",
                    method = "ITT",
                    options = SEQopts())
```

SEQ_data

Function to return the internal data from a SEquential object

Description

Function to return the internal data from a SEquential object

Usage

```
SEQ_data(object)
```

Arguments

object SEQoutput object

Value

data.table

show,SEQoutput-method *Show method for S4 object - SEQoutput.*

Description

Show method for S4 object - SEQoutput.

Usage

```
## S4 method for signature 'SEQoutput'  
show(object)
```

Arguments

object A SEQoutput object - usually generated from [SEquential\(\)](#)

Value

No return value, sends information about SEQoutput to the console

Index

* datasets

SEQdata.LTFU, [8](#)

SEQdata.multitreatment, [8](#)

compevent, [2](#)

covariates, [3](#)

data.table::getDTthreads(), [13](#)

denominator, [3](#)

diagnostics, [4](#)

hazard_ratio, [4](#)

km_curve, [5](#)

km_data, [5](#)

numerator, [6](#)

outcome, [6](#)

parallelly::availableCores(), [13](#)

risk_comparison, [7](#)

risk_data, [7](#)

SEQ_data, [16](#)

SEQdata.LTFU, [8](#)

SEQdata.multitreatment, [8](#)

SEQestimate, [9](#)

SEQexpand(), [10](#)

SEQopts, [10](#)

SEQopts(), [10](#), [16](#)

SEQoutput-class, [14](#)

SEquential, [15](#)

SEquential(), [8](#), [17](#)

show, SEQoutput-method, [17](#)