

# Package: SEQTaRget (via r-universe)

June 8, 2026

**Type** Package

**Title** Sequential Trial Emulation

**Version** 1.4.2.9002

**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

**Suggests** rmarkdown, testthat (>= 3.0.0)

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

**Config/roxygen2/markdown** TRUE

**Config/roxygen2/version** 8.0.0

**Config/testthat/edition** 3

**Depends** R (>= 4.1)

**URL** <https://causalinference.github.io/SEQTaRget/>,  
<https://github.com/CausalInference/SEQTaRget>

**VignetteBuilder** knitr

**Config/pak/sysreqs** libicu-dev

**Repository** <https://remlapmot.r-universe.dev>

**Date/Publication** 2026-06-08 13:16:22 UTC

**RemoteUrl** <https://github.com/CausalInference/SEQTaRget>

**RemoteRef** HEAD

**RemoteSha** ea65b42d44b6bcf146a57d9b0cdfba92e18a9673

**RemoteSubdir** SEQTaRget

## 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
SEQ_data . . . . .	8
SEQdata . . . . .	8
SEQdata.LTFU . . . . .	9
SEQdata.multitreatment . . . . .	9
SEQestimate . . . . .	10
SEQopts . . . . .	11
SEQoutput-class . . . . .	16
SEQquential . . . . .	17
show,SEQoutput-method . . . . .	19
<b>Index</b>	<b>20</b>

---

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

A fastglm object, or a named list of fastglm objects when subgroups are specified

---

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 denominator models

---

diagnostics	<i>Function to return diagnostic tables from a SEquential object</i>
-------------	--

---

**Description**

Function to return diagnostic tables from a SEquential object

**Usage**

```
diagnostics(object)
```

**Arguments**

object            SEQoutput object

**Value**

List of diagnostic tables

---

hazard_ratio	<i>Function to return hazard ratios from a SEquential object</i>
--------------	--

---

**Description**

Function to return hazard ratios from a SEquential object

**Usage**

```
hazard_ratio(object)
```

**Arguments**

object            SEQoutput object

**Value**

A named vector of hazard ratios, or a named list of vectors when subgroups are specified

---

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; one of: "survival" (default), "risk", "inc"
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**

A data frame of survival values, or a named list of data frames when subgroups are specified

---

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      *Function to return risk information from a SEquential object*

---

**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      *Function to return risk information from a SEquential object*

---

**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

---

SEQ_data	<i>Function to return the internal data from a SEquential object</i>
----------	--

---

**Description**

Function to return the internal data from a SEquential object

**Usage**

```
SEQ_data(object)
```

**Arguments**

object            SEQoutput object

**Value**

data.table

---

SEQdata	<i>Simulated observational example data for SEquential</i>
---------	--

---

**Description**

Simulated observational example data for [SEquential\(\)](#)

**Usage**

```
SEQdata
```

**Format**

A data frame with 12,180 rows and 11 columns:

**ID** Integer: Unique ID emulating individual patients

**time** Integer: Time of observation, always begins at 0, max time of 59. Should be continuous

**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 one, always one variable emulating an excuse for treatment switch

---

 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 54,687 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 one, always one 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

**eligible\_cense** Binary: emulates columns which are eligible to entering into censoring models (e.g. if you want to limit columns for the LTFU model)

---

 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 one, always one variable emulating an excuse for treatment switch

---

SEQestimate

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

---

**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 <code>SEQexpand()</code> , 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 <code>SEQopts()</code>
<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),
expand.only = FALSE,
fastglm.method = 2L,
followup.class = FALSE,
followup.include = TRUE,
followup.max = Inf,
followup.min = 0,
followup.spline = FALSE,
followup.spline.df = 4L,
glm.package = "fastglm",
hazard = FALSE,
indicator.baseline = "_bas",
indicator.squared = "_sq",
km.curves = FALSE,
multinomial = FALSE,
ncores = availableCores(omit = 1L),
nthreads = getDTthreads(),
numerator = NA,
parallel = FALSE,
parglm.control = NULL,
plot.colors = c("#F8766D", "#00BFC4", "#555555"),
plot.labels = NA,
plot.subtitle = NA,
plot.title = NA,
plot.type = "survival",
risk.times = NA,
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 = 0,
```

```

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 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>expand.only</code>	Logical: if TRUE, <code>SEQential()</code> returns the expanded <code>data.table</code> immediately after expansion and skips weighting, outcome modelling and survival/risk steps. Useful when you only need the expanded dataset (e.g. to inspect or store separately). Default is FALSE
<code>fastglm.method</code>	Integer: decomposition method for fastglm (0L-column-pivoted QR, 1L-unpivoted QR, 2L-LLT Cholesky, 3L-LDLT Cholesky), 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 follow-up time since trial enrollment to include, must be non-negative, default is 0
<code>followup.spline</code>	Logical: treat followup as a natural cubic spline ( <code>splines::ns()</code> ), default is FALSE
<code>followup.spline.df</code>	Integer: degrees of freedom passed to <code>splines::ns()</code> when <code>followup.spline = TRUE</code> . With <code>df = k</code> , <code>ns()</code> places <code>k - 1</code> interior knots at quantiles of followup. Must be $\geq 1$ ; <code>df = 1</code> is equivalent to a linear term and is generally not what you want. Default is 4 (3 interior knots).
<code>glm.package</code>	Character: package to use for fitting GLMs, either "fastglm" (default) or "parglm". When "parglm" is selected the <code>nthreads</code> option controls the number of threads passed to <code>parglm::parglm.fit()</code> . For most realistic SEQTaRget workloads (expanded datasets up to approximately a few million rows) "fastglm" is faster; "parglm" may help only on substantially larger datasets where the parallel chunking outweighs its setup overhead.
<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 <code>parallely::availableCores()</code>
<code>nthreads</code>	Integer: number of threads to use for <code>data.table</code> processing, default is <code>data.table::getDTthreads()</code>
<code>numerator</code>	String: numerator covariates to the right hand side of a formula object
<code>parallel</code>	Logical: define if the SEQential process is run in parallel, default is FALSE

<code>parglm.control</code>	A control object from <code>parglm::parglm.control()</code> to pass to <code>parglm::parglm.fit()</code> . Only used when <code>glm.package = "parglm"</code> . Defaults to <code>parglm::parglm.control(method = "FAST")</code> . If you encounter a <code>chol(): decomposition failed</code> error (e.g. with near-singular model matrices on large datasets), pass <code>parglm.control = parglm::parglm.control(method = "LAPACK")</code> to use the more numerically stable QR decomposition instead, or switch to using the <code>fastglm</code> backend.
<code>plot.colors</code>	Character: Colors for output plot if <code>km.curves = TRUE</code> , defaulted to <code>ggplot2</code> defaults
<code>plot.labels</code>	Character: Color labels for output plot if <code>km.curves = TRUE</code> in order e.g. <code>c("risk.0", "risk.1")</code>
<code>plot.subtitle</code>	Character: Subtitle for output plot if <code>km.curves = TRUE</code>
<code>plot.title</code>	Character: Title for output plot if <code>km.curves = TRUE</code>
<code>plot.type</code>	Character: Type of plot to create if <code>km.curves = TRUE</code> , available options are 'survival' (the default), 'risk', and 'inc' (in the case of censoring)
<code>risk.times</code>	Numeric vector: follow-up times (in the data's follow-up units) at which to report risk difference and risk ratio when <code>km.curves = TRUE</code> . Each requested time is snapped to the latest available follow-up at or before it. The final follow-up time is always included. Default NA reports only the final follow-up time.
<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 <code>selection.random</code> , should be bound <code>[0, 1]</code> , default is <code>0.8</code>
<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 <code>Inf</code> (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: IPCW weights truncated at this lower bound, must be non-negative, default is <code>0</code> . Truncation is applied only to the weights used to fit the outcome model; the weights reported in <code>weight.statistics</code> and in the returned data (when <code>data.return = TRUE</code> ) are the untruncated values.

<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> . The percentiles are taken from the untruncated weight distribution (as reported in <code>weight.statistics</code> ), and as with <code>weight.lower/weight.upper</code> the truncation affects only the weights used to fit the outcome model.
<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> . As with <code>weight.lower</code> , truncation affects only the weights used to fit the outcome model, not those reported in <code>weight.statistics</code> or the returned data.
<code>weighted</code>	Logical: whether or not to perform 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 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 perform 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
<code>fixed.cols</code>	List: column names for fixed columns
<code>method</code>	String: method of analysis to perform; should be one of "ITT", "dose-response", or "censoring"
<code>options</code>	List: optional list of parameters from <a href="#">SEQopts()</a>
<code>verbose</code>	Logical: if TRUE, cats progress to console, default is TRUE



---

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 [SEQquential\(\)](#)

### **Value**

No return value, sends information about SEQoutput to the console

# Index

## \* datasets

- SEQdata, [8](#)
- SEQdata.LTFU, [9](#)
- SEQdata.multitreatment, [9](#)

- compevent, [2](#)
- covariates, [3](#)

- data.table::getDTthreads(), [14](#)
- denominator, [3](#)
- diagnostics, [4](#)

- hazard\_ratio, [4](#)

- km\_curve, [5](#)
- km\_data, [5](#)

- numerator, [6](#)

- outcome, [6](#)

- parallelly::availableCores(), [14](#)
- parglm::parglm.control(), [15](#)

- risk\_comparison, [7](#)
- risk\_data, [7](#)

- SEQ\_data, [8](#)
- SEQdata, [8](#)
- SEQdata.LTFU, [9](#)
- SEQdata.multitreatment, [9](#)
- SEQestimate, [10](#)
- SEQexpand(), [11](#)
- SEQopts, [11](#)
- SEQopts(), [11](#), [17](#)
- SEQoutput-class, [16](#)
- SEQential, [17](#)
- SEQential(), [8](#), [9](#), [14](#), [19](#)
- show, SEQoutput-method, [19](#)