Package 'tLagInterim'

August 17, 2022

Title Interim Monitoring of Clinical Trials with Time-Lagged Outcome

Version 1.0

Author Shannon T. Holloway [aut, cre],
Marie Davidian [aut],
Anastasios Tsiatis [aut]

Description Implements inverse and augmented inverse probability weighted estimators for common treatment effect parameters at an interim analysis with time-lagged outcome that may not be available for all enrolled subjects. Produces estimators, standard errors, and information that can be used to compute stopping boundaries using software that assumes that the estimators/test statistics have independent increments.
Tsiatis, A. A. and Davidian, M., (2022) <arXiv:2204.10739> .

License GPL-2

Encoding UTF-8

Depends R (>= 3.5.0), survival

Imports R.utils

RoxygenNote 7.2.1

Collate 'KM.R' 'miscfunc.R' 'martingale.R' 'augment.R' 'ipw.R' 'ness.R' 'onestep.R' 'print.tLagInterim.R' 'verifyInputs.R' 'tLagInterim.R' 'tLagInterimData.R' 'type_Mean.R' 'type_OddsRatio.R' 'type_RiskDiff.R' 'type_RiskRatio.R'

NeedsCompilation no

Maintainer Shannon T. Holloway <shannon.t.holloway@gmail.com>

Repository CRAN

Date/Publication 2022-08-17 13:00:02 UTC

R topics documented:

`print` *Print results from a tLagInterim object*

Description

Print results from a tLagInterim object

Usage

```
## S3 method for class 'tLagInterimObj'
print(x, ...)
```

Arguments

`x`	A tLagInterimObj object, returned by tLagInterim().
`...`	Ignored.

Value

No return value, called to display key results.

Examples

```
data(tLagIntCat)
# f basis functions#'
f <- function(x.data) {
       return( as.matrix(x = cbind(1.0, x.data)) )
     }

# h basis functions#'
h <- function(b.data, x.data, t.data, times) {

  # Number of basis functions L
  # (note that the number of basis functions does not and cannot depend
  #  on the treatment group; `h` is called internally multiple times -- each
  #  call is for a single treatment group.)
  L <- 2

  # Number of subjects in data
  n_subjects <- nrow(x = b.data)

  # Number of time points
  n_times <- length(x = times)

  # Initialize array of basis functions for this trt
  h.basis <- array(data = 0.0, dim = c(n_subjects, n_times, L))

  # Indicator of still being in hospital at any censoring time
```

```
  lindicator <- outer(X = t.data$lu, Y = times, "<=") * {t.data$ldelta == 2L}
  h.basis[, , 1L] <- lindicator

  # Time from leaving hospital to obstime for those known to
  # leave hospital at each censoring time
  h.basis[, , 2L] <- {times - t.data$lu} * lindicator

  return( h.basis )
}

# fit with only baseline covariates provided, categorical outcome, user-specified f, h
res <- tLagInterim(b.data = b.data.cat,
                   x.data = x.data.cat,
                   t.data = NULL,
                   outcome = "categorical",
                   f = f,
                   h = h)
print(res)
```

`tLagIntBin` *Toy Dataset With a Binary Outcome For Illustration*

Description

These data are provided for the purposes of illustrating the use of the software when the outcome of interest is binary. Though these data were generated to mimic conditions of a clinical trial, they should not be interpreted as representing true clinical trial data.

Usage

```
data(tLagIntBin)
```

Format

Each dataset provides three data.frames: `b.data.bin` containing the basic observed data on 722 enrolled subjects at the time of an interim analysis at time 52 week, with columns with headers

- "subjID" - unique subject identifiers,
- "u" - minimum of time lag or censoring time,
- "delta" - time lag/censoring indicator, and
- "Y" - the outcome if it is available, = 0 if not.
- "a" - treatment indicator;

`x.data.bin` contains the baseline covariates for the 722 subjects.

- "subjID" - unique subject identifiers,
- "X1" - a continuous covariate;

and t.data.bin contains time-dependent covariate information comprising

- "subjID" - unique subject identifiers,
- "lu" - time to leaving hospital, death, or censoring
- "ldelta" (0), death (1), or left hosp (2)

---

tLagIntCat *Toy Dataset With a Categorical Outcome For Illustration*

Description

These data are provided for the purposes of illustrating the use of the software when the outcome of interest is categorical. Though these data were generated to mimic conditions of a clinical trial, they should not be interpreted as representing true clinical trial data.

Usage

```
data(tLagIntCat)
```

Format

Each dataset provides three data.frames: b.data.cat containing the basic observed data on 477 enrolled subjects at the time of an interim analysis, with columns with headers

- "subjID" - unique subject identifiers,
- "u" - minimum of time lag or censoring time,
- "delta" - time lag/censoring indicator, and
- "Y" - the outcome if it is available, = 0 if not.
- "a" - treatment indicator;

x.data.cat contains the baseline covariates for the 477 subjects.

- "subjID" - unique subject identifiers,
- "X1" - a continuous covariate;

and t.data.cat contains time-dependent covariate information comprising

- "subjID" - unique subject identifiers,
- "lu" - time to leaving hospital, death, or censoring
- "ldelta" (0), death (1), or left hosp (2)

tLagIntCont — *Toy Dataset With a Continuous Outcome For Illustration*

Description

These data are provided for the purposes of illustrating the use of the software when the outcome of interest is continuous. Though these data were generated to mimic conditions of a clinical trial, they should not be interpreted as representing true clinical trial data.

Usage

```
data(tLagIntCont)
```

Format

The dataset provides three data.frames: `b.data.cont` containing the basic observed data on 245 enrolled subjects at the time of an interim analysis, with columns with headers

- "subjID" - unique subject identifiers,
- "u" - minimum of time lag or censoring time,
- "delta" - time lag/censoring indicator, and
- "Y" - the outcome if it is available, = 0 if not.
- "a" - treatment indicator;

`x.data.cont` contains the baseline covariates for the 245 subjects.

- "subjID" - unique subject identifiers,
- "X1" - a continuous covariate;

and `t.data.cont` contains time-dependent covariate information comprising "subjID" and 6 measurements of a single continuous covariate measured at times (t1 = 0, t2 = 4, t3 = 12, t4 = 24, t5 = 52)

tLagInterim — *Group Sequential Methods for Interim Monitoring of Randomized Clinical Trials with Time-lagged Outcome*

Description

Implements methods for estimation of treatment effect parameters to support interim monitoring of clinical trials in which the outcome is ascertained after a time lag, so that not all subjects enrolled at the time of an interim analysis will have the outcome available. The methods take advantage of all available data to increase the precision of the analysis and thus lead to potentially earlier stopping.

Usage

```
tLagInterim(
  b.data,
  x.data = NULL,
  t.data = NULL,
  outcome = c("continuous", "binary", "categorical"),
  trteff = c("risk.diff", "risk.ratio", "odds.ratio"),
  ...,
  f = NULL,
  h = NULL,
  baseTx = 0L,
  baseY = 0L
)
```

Arguments

b.data	A data frame containing the basic observed data on the n enrolled subjects at the time of an interim analysis at time t, with columns with headers • "subjID" (unique subject identifiers), • "a" (treatment indicator), • "u" (minimum of time lag or censoring time), • "delta" (time lag/censoring indicator), and • "Y" (the outcome if it is available, = 0 if not).
x.data	A data frame whose columns are baseline covariates, which is input to the user-specified function f (see example) to create the M+1 baseline basis functions f_0, f_1, ..., f_M, where f_0 = 1 for all subjects; f_0 must be created in the function f. If not provided or NULL, the AIPW2 estimator will be computed if t.data and h are provided; otherwise, only the IPW estimator will be computed. Must contain a column with header "subjID" containing the unique subject identifiers.
t.data	A data frame in whatever form the user specifies containing the time-dependent covariate information, which is input along with x.data to the user-specified function h (see example) to create the time-dependent basis functions h_l, l=1, ..., L. These basis functions can involve both baseline and time-dependent covariates. If not provided or NULL, the IPW and AIPW1 estimators will be computed if x.data and f are provided; otherwise, only the IPW estimator will be computed. Must contain a column with header "subjID" containing the unique subject identifiers.
outcome	Choices are "continuous", "binary", or "categorical". If outcome = "categorical", for each category there must be at least one subject with available outcome. If outcome = "binary", there must be at least one subject for each level. If outcome is not specified as one of "continuous", "binary", or "categorical", an error will be generated.
trteff	If outcome = "binary", must be provided; trteff = "risk.diff" for risk difference, trteff = "risk.ratio" for the logarithm of the risk ratio (log relative risk), and trteff = "odds.ratio" for the log odds ratio. If outcome = "binary" but trteff is not provided, an error will be generated. Ignored if outcome = "continuous" or "categorical."

`...`	Ignored.
`f`	A user-specified function taking the data frame x.data as input, which returns an (n x M+1) matrix whose first column is all ones and remaining columns are the M user-specified basis functions f_1, ..., f_M for each subject (see example below). If `x.data` is not provided, `f` is ignored.
`h`	A user-specified function taking the data frames b.data, x.data, t.data and a vector of censoring times as input. This function must return an array of dimension n x nt x L, where n = number of rows of the passed input b.data, and nt = number of censoring times passed as input, so that the (i,j,l) element of h.basis is the value of the lth basis function h_l at the jth censoring time for the ith subject (see example below). If t.data is not provided, h is ignored. See Details for further information.
`baseTx`	Type depends on class of treatment data. Treatment will be converted to 0/1 internally, this input specifies the value of b.data$a that is the base (control) value.
`baseY`	Used only for binary outcomes. Type depends on class of outcome data. Outcome will be converted to 0/1 internally, this input specifies the value of b.data$Y that is the base (0) value.

Details

The data at the time of the desired interim analysis at time "t" must be input in one required and two optional data frames. The required data frame contains the basic information on treatment assignment, whether or not the outcome is available and the time lag, and, if available, the outcome itself. The first optional data frame contains baseline covariate information. The second optional data frame must contain information relevant to constructing time-dependent covariates, and its form is specified by the user; an example is provided.

Three types of outcome are supported: (1) continuous, (2) binary, and (3) ordered categorical. For a continuous outcome, the treatment effect parameter is the difference in treatment means. For a categorical outcome, the treatment effect parameter is the log odds ratio under an assumed proportional odds model. For a binary outcome, the treatment effect parameter can be one of (a) the risk difference (equivalent to the difference in treatment means), (b) logarithm of the risk ratio (log relative risk), or (c) log odds ratio.

If the outcome is ordered categorical, the categories must be ordered such that the outcomes are "worse" as one progresses from the base level to the final level.

If the outcome is binary and its levels are not coded as 0, 1, the coding for the base level must be provided as input. The outcome will be recast internally as 0, 1. The underlying models for each type of treatment effect are models for the probability that Y = 1. There must be at least one subject with available outcome equal to each of 0 and 1.

The basic analysis data frame b.data must contain the following variables for each subject:

subjID An identifier unique to each subject.

a The treatment assignment indicator; treatments must be binary.

u The time lag T at which the outcome was ascertained, if it was ascertained, or the censoring time on the scale of subject time.

delta The indicator of T <= C, so that the outcome is observed if delta = 1

Y The outcome if it is available (delta=1); otherwise Y should be set equal to zero (delta=0); thus, Y = delta times outcome

Each column of the baseline covariate data frame x.data should be a baseline covariate. Data must contain a `subjID` column that contains the same subject identifiers as used in `b.data`.

The time-dependent data frame must contain the information used to construct time-dependent covariates in a format that is input into the user-specified function h() that constructs the basis functions. As this data frame is only used to construct the h basis functions, the format and contents are, for the most part, entirely up to the user. The notable exception is that it must contain a `subjID` column that contains the same subject identifiers as used in `b.data`.

The function `h` is called multiple times internally – each call is for a single treatment group. The function is provided only the data for the specific treatment group under consideration, e.g., when estimating the L basis functions for a = 0, the b.data, x.data, and t.data passed to h() contain only the rows for subjects in the a = 0 treatment arm; further, the nt censoring times are only those for this subset of subjects.

The returned object contains the information needed to conduct any desired interim analysis (information-based or fixed-sample-based) for efficacy or futility using standard interim analysis software that assumes the test statistic has independent increments, such as the R package ldbounds.

Value

An S3 object of class tLagInterim containing a list of variable length depending on which estimators can be computed given the inputs. The elements of the list have the following names:

`nt`	The number of subjects enrolled at the time of the interim analysis.
`cens`	The proportion of these subjects for whom the outcome is not available (i.e., the time lag is censored).
`IPW`	A data frame containing the IPW estimate of the treatment effect parameter, its standard error, a 95% Wald confidence interval for the treatment effect, the corresponding Wald test statistic, the effective sample size n_ESS(t) (for fixed-sample-based monitoring), and the information Inf(t) = 1/(standard error)^2 (for information-based monitoring).
`AIPW1`	If x.data and f are provided, a data frame containing the same information as for the IPW estimator for the AIPW1 estimator that incorporates baseline covariate information only.
`AIPW2`	If either (i) x.data and f are not provided and t.data and h are, or (ii) both x.data and f and t.data and h are provided, a data frame containing the same information as for the IPW estimator for the AIPW2 estimator that incorporates time-dependent covariate information (alone or in addition to baseline covariate information).

The S3 object has an additional attribute, "estimators", giving a description of which estimators are computed.

References

Tsiatis AA and Davidian M, Group sequential methods for interim monitoring of randomized clinical trials with time-lagged outcome. https://arxiv.org/abs/2204.10739.

Examples

```
# Baseline and time-dependent covariates provided, categorical outcome
data(tLagIntCat)

# f (basis functions for main effects when x contains continuous and
# binary (0/1) covariates); a user-specified function could also
# include dummies for categorical covariates, interaction terms,
# functions of covariates, etc.

f <- function(x.data) {
   f.basis <- cbind(1.0, data.matrix(frame = x.data))
   return( f.basis )
}

# h as for the first two simulation scenarios in the paper
# (categorical outcome), where t.data has columns "lu" = time to
# leaving hospital, death, or censoring, which ever first, and
# "ldelta" = 0 (censored), 1 (death), 2 (left hospital).  The basis
# functions could also include baseline covariates, although that
# is not the case here.

h <- function(b.data, x.data, t.data, times) {

  # Number of basis functions L
  # (note that the number of basis functions does not and cannot depend
  #  on the treatment group; `h` is called internally multiple times -- each
  #  call is for a single treatment group.)
  L <- 2

  # Number of subjects in the provided data
  n_data <- nrow(x = b.data)

  # Number of censoring times provided
  n_times <- length(x = times)

  # Initialize array of basis functions
  h.basis <- array(data = 0.0, dim = c(n_data, n_times, L))

  # Indicator of still being in hospital at any censoring time
  lindicator <- outer(X = t.data$lu, Y = times, "<=") * {t.data$ldelta == 2L}
  h.basis[, , 1L] <- lindicator

  obstime <- max(b.data$u)

  # Time from leaving hospital to obstime for those known to
  # leave hospital at each censoring time
  h.basis[, , 2L] <- {obstime - t.data$lu} * lindicator

  #  Return the basis functions
  return( h.basis )
}
```

```
# Compute all of IPW, AIPW1, AIPW2
tLagInterim(b.data = b.data.cat,
            x.data = x.data.cat,
            t.data = t.data.cat,
            outcome = "categorical",
            f = f,
            h = h)

# Compute IPW, AIPW1 only (no time-dependent covariates)
tLagInterim(b.data = b.data.cat,
            x.data = x.data.cat,
            t.data = NULL,
            outcome = "categorical",
            f = f,
            h = NULL)

# Baseline and time-dependent covariates provided, binary outcome, risk ratio
data(tLagIntBin)

# Compute all of IPW, AIPW1, AIPW2
tLagInterim(b.data = b.data.bin,
            x.data = x.data.bin,
            t.data = t.data.bin,
            outcome = "binary",
            trteff = "risk.ratio",
            f = f,
            h = h)

# Compute IPW, AIPW2 only (no baseline covariates)
tLagInterim(b.data = b.data.bin,
            x.data = NULL,
            t.data = t.data.bin,
            outcome = "binary",
            trteff = "risk.ratio",
            f = NULL,
            h = h)


# Baseline and time-dependent covariates provided, continuous outcome
data(tLagIntCont)

# h as for the third simulation scenario in the paper (continuous
# outcome), where t.data has 5 columns corresponding to the 5
# intended times at which longitudinal measures of the outcome are
# ascertained, and the last observed measure is carried forward to
# all future times if it is not available

h <- function(b.data, x.data, t.data, times) {

  # Number of basis functions L
  # (note that the number of basis functions does not and cannot depend
  #  on the treatment group; `h` is called internally multiple times -- each
```

```
  #  call is for a single treatment group.)
  L <- 1L

  # Number of subjects in provided data
  n_data <- nrow(x = b.data)

  # Number of censoring times provided
  n_times <- length(x = times)

  ti <- c(0,4,12,24,52)

  # Initialize array of basis functions
  h.basis <- array(data = 0.0, dim = c(n_data, n_times, L))

  # last value at each censoring time
  # dropping 1st column as it contains subject ids.
  h.basis[, , 1L] <- t(apply(X = t.data[,-1L],
                             MARGIN = 1L,
                             FUN = function(u) {
                                     u[findInterval(x = times, vec = ti)]
                                   }))

  #  Return the basis functions
  return( h.basis )
}

# Compute all of IPW, AIPW1, AIPW2
tLagInterim(b.data = b.data.cont,
            x.data = x.data.cont,
            t.data = t.data.cont,
            outcome = "continuous",
            f = f,
            h = h)
```

Index