Package: VEwaningVariant (via r-universe)

October 27, 2026

Type Package

Title Vaccine Efficacy Over Time - Variant Aware

Version 1.4

Date 2023-12-11

Description Implements methods for inference on potential waning of vaccine efficacy and for estimation of vaccine efficacy at a user-specified time after vaccination based on data from a randomized, double-blind, placebo-controlled vaccine trial in which participants may be unblinded and placebo subjects may be crossed over to the study vaccine. The methods also for variant stratification and allow adjustment for possible confounding via inverse probability weighting through specification of models for the trial entry process, unblinding mechanisms, and the probability an unblinded placebo participant accepts study vaccine.

License GPL-2

Depends R (>= 3.5.0), survival, stats, graphics, ggplot2, Rcpp, RcppArmadillo, methods

NeedsCompilation yes

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.2.3

Collate 'RcppExports.R' 'censorFit.R' 'confirmIntegerLike.R' 'pzero.R' 'survFunc.R' 'entryFit.R' 'esttheta.R' 'fitCensor.R' 'psiFit.R' 'unblindFit.R' 'fitERPsi.R' 'gFunc_v.R' 'plot.VEwaningVariant.R' 'print.VEwaningVariant.R' 'variant.R' 'variantData.R' 've.R' 'verifyPhase.R' 'verifyModels.R' 'verifyInputs.R' 'veWaningVariant.R'

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

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

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

Date/Publication 2023-12-14 02:37:54 UTC

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

RemoteRef HEAD

RemoteSha b1023c5020606da87e171bcb7fb784309f004671

Contents

plot 2
print 3
variantData 4
ve 5
veWaningVariant 6

Index **11**

`plot` *Plot Analysis Results*

Description

Plot the Estimated Vaccine Efficacy

Usage

```
## S3 method for class 'VEwaningVariant'
plot(x, y, ..., times = NULL, xlim = NULL, nTimes = 20L, ylim = NULL)
```

Arguments

`x`	An object of class VEwaningVariant. The object returned by a call to veWaningVariant()
`y`	Ignored
`...`	Ignored
`times`	A numeric vector object or NULL. The times since full efficacy at which the vaccine efficacy is to be estimated. If NULL, the times will be generated internally as a vector of length nTimes spanning the range [0, maxTime], where maxTime is the maximum time since vaccination present in the original analysis. Values provided outside of [0, maxTime] are ignored.
`xlim`	A numeric vector object of length 2 or NULL. The extrema of the times values at which estimates are to be calculated. A vector of length nTimes spanning the range [xlim[1], xlim[2]] is generated. Note that the specified limits must lie in the range [0, maxTime]. If input times is a vector object, this input is ignored.
`nTimes`	An integer object. The number of time values at which estimates are obatined. The default is 20. If input times is a vector object, this input is ignored.
`ylim`	A numeric vector object or NULL. The y-axis limits for the plots. If NULL, the y-axis limits are taken from the estimated values.

Details

When the variant under analysis is present only in the unblinded phase, vaccine efficacy cannot be estimated. In this case, plot() shows the relative infection rate at times t since full efficacy reached, defined as infection rate at time t = time since full efficacy reached divided by the infection rate at the time full efficacy is reached (t=0).

Value

A gg object.

Examples

```
data(variantData)

set.seed(1234)

ind <- sample(1:nrow(x = variantData), 2000, FALSE)

# NOTE: This sample size is chosen for example only -- larger data sets
# should be used.

res <- veWaningVariant(data = variantData[ind,],
                       L = 52,
                       lag = 6,
                       gFunc = 'piece',
                       v = c(15,30))

plot(x = res)
```

---

print *Print Analysis Results*

Description

Print the primary results of the analysis

Usage

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

Arguments

x	An object of class VEwaningVariant. The object returned by a call to veWaningVariant()
...	Ignored

Value

No return value, called to display key results.

Examples

```
data(variantData)

set.seed(1234)

ind <- sample(1:nrow(variantData), 2000)
# NOTE: This sample size is chosen for example only -- larger data sets
# should be used.

res <- veWaningVariant(data = variantData[ind,],
                       L = 52,
                       gFunc = 'piece',
                       v = c(15,30))

print(x = res)
```

---

variantData *Toy Dataset For Illustration*

Description

This data set is provided for the purposes of illustrating the use of the software.

Usage

```
data(variantData)
```

Format

variantData is a data.frame containing 30,000 participants The data.frame contains 10 columns,

ID A unique patient identifier.

X1 A dichotomous baseline covariate; generated from B(0.5).

X2 A continuous baseline covariates; generated from N(45,10^2).

E The entry time into the study.

A The indicator of vaccination at entry (1 = vaccinated at entry; 0 = not vaccinated)

U The minimum of: time to infection, time to refusing study vaccine after unblinding (placebo participants only), or censoring (due to loss to followup or administrative censoring).

Delta Infection-variant category indicator, where -1 indicates a placebo participant that was unblinded and refused study vaccine; 0 indicates censoring; and = ve indicates a participant infected with variant ve (ve = 1, ..., nV).

R The minimum of: time to unblinding, time to infection, or time to censoring.

Gamma Indicator of the R dynamic, where 1 indicates that R is the time to unblinding and 0 indicates that R is either the time to infection or the time to censoring.

Psi Indicator of whether a placebo subject received study vaccine, where 1 indicates that a participant that was either assigned to vaccine or was assigned to placebo, was unblinded and decided to get the study vaccine; 0 otherwise.

All time variables are on the scale of weeks.

`ve` *Retrieve the Estimated Vaccine Efficacy*

Description

Uses a prior veWaningVariant() analysis to estimate the vaccine efficacy at the provided times since full efficacy.

Usage

```
ve(x, times = NULL, nTimes = 20L)
```

Arguments

`x`	An object of class VEwaningVariant. The object returned by a call to veWaningVariant()
`times`	A numeric vector object or NULL. The times since full efficacy at which the vaccine efficacy is to be estimated. If NULL, the times will be generated internally as a vector of length nTimes spanning the range [0, maxTime], where maxTime is the maximum time since vaccination present in the original analysis. Values provided outside of [0, maxTime] are ignored.
`nTimes`	An integer object. The number of time values at which estimates are obtained. The default is 20. If input times is a vector object, this input is ignored.

Details

When the variant under analysis is present only in the unblinded phase, vaccine efficacy cannot be estimated. In this case, ve() returns the relative infection rate at times t since full efficacy reached, defined as infection rate at time t = time since full efficacy reached divided by the infection rate at the time full efficacy is reached (t=0).

Value

A matrix object. The first column contains the times since full efficacy at which the estimates are provided; the second column contains estimated vaccine efficacy or relative infection rate (see Details); and the third is the standard error.

Examples

```
data(variantData)

set.seed(1234)

ind <- sample(1:nrow(variantData), 2000)
# NOTE: This sample size is chosen for example only -- larger data sets
# should be used.

res <- veWaningVariant(data = variantData[ind,],
                       L = 52,
                       gFunc = 'piece',
                       v = c(15,30))

ve(x = res, times = c(10,20,30,40,50))
```

`veWaningVariant` *Estimation of Vaccine Efficacy Over Time - Variant Aware*

Description

Implements methods for inference on potential waning of vaccine efficacy and for estimation of vaccine efficacy at a user-specified time after vaccination based on data from a randomized, double-blind, placebo-controlled vaccine trial in which participants may be unblinded and placebo subjects may be crossed over to the study vaccine. The method allows for variant specification and adjustment for possible confounding via inverse probability weighting through specification of models for the trial entry process, unblinding mechanisms, censoring process, and the probability an unblinded placebo participant accepts study vaccine.

Usage

```
veWaningVariant(
  data,
  L,
  ...,
  phase = "ub",
  cutoff = 0.1,
  lag = 6,
  modelEntry = NULL,
  modelUnblind0 = NULL,
  modelUnblind1 = NULL,
  modelPsi = NULL,
  modelCensor0 = NULL,
  modelCensor1 = NULL,
  gFunc = NULL,
  variant = 0L,
  v = NULL,
```

```
  minWgt = NULL,
  maxWgt = NULL,
  txName = "A",
  U = "U",
  entryTime = "E",
  Gamma = "Gam",
  R = "R",
  Psi = "Psi",
  Delta = "Delta"
)
```

Arguments

data	A data.frame object containing all relevant data. Records with missing data will be removed prior to initiating the analysis.
L	A numeric object. The analysis time.
...	Ignored. Used only to require named inputs.
phase	A character object. The phase(s) to include in the analysis. Default is ="ub", indicating that both blinded and unblinded phases will be used to estimate theta values if possible. If ="b", only the blinded phase will be used. If ="u", only the unblinded phase will be used.
cutoff	A numeric object. The minimum proportion of infections that must occur during a phase to be included in the analysis. The default is 0.1 (10 blinded (unblinded) phase, only the unblinded (blinded) phase will be included in the analysis.
lag	A scalar numeric, numeric vector, or character object. The lag time(s) between the initial vaccine dose and full efficacy. If a scalar, the provided lag time applies to all participants. If a numeric vector, the vector contains the individual specific lag time for each participant (see details for further information). If character, the column header of the data containing the lag times. The default value is a scalar value of 6 weeks (42 days) – NOTE this assumes that the data are on the scale of weeks.
modelEntry	A formula object or NULL. The coxph model for entry times. The LHS is set as the appropriate Surv() object internally. If a LHS is provided, it is ignored. If NULL, inputs modelPsi, modelUnblind0, and modelUnblind1 must also be NULL. See Details for further information.
modelUnblind0	A formula object or NULL. The coxph model for unblinding/crossover for placebo (A=0) participants. If NULL, inputs modelEntry, modelPsi, and modelUnblind1 must also be NULL. See Details for further information.
modelUnblind1	A formula object or NULL. The coxph model for unblinding for vaccinated (A=1) participants. If NULL, inputs modelEntry, modelPsi, and modelUnblind0 must also be NULL. See Details for further information.
modelPsi	A formula object or NULL. The logistic model for the probability that a placebo participant (A = 0) is unblinded at R (Gamma = 1) and agrees to take the study vaccine (Psi = 1). If a LHS is provided, it is ignored. If NULL, inputs modelEntry, modelUnblind0, and modelUnblind1 must also be NULL. See Details for further information.

`modelCensor0` A formula object or NULL. The coxph model for censoring for placebo (A=0) participants. The LHS is set as the appropriate Surv() object internally. If a LHS is provided, it is ignored. If NULL, all other models must be NULL. See Details for further information.

`modelCensor1` A formula object or NULL. The coxph model for censoring for vaccinated (A=1) participants. The LHS is set as the appropriate Surv() object internally. If a LHS is provided, it is ignored. If NULL, all other models must be NULL. See Details for further information.

`gFunc` A character vector object. The model of infection rates. Must be one of {'lin', 'piece', 'splin', 'spcub'} for the linear, piecewise constant, linear spline, and cubic spline models, respectively.

`variant` An integer object. The variant for the analysis. If 0, all variants are included in the analysis.

`v` A numeric vector object. The knots or cut-offs to be used for input gFunc. If gFunc = 'lin', this input is ignored. For 'splin' and 'spcub', the knots of the spline on (0,L). For 'piece', the cut-offs on (0,L). Note that this input should not include the extremes 0 and L.

`minWgt` A numeric object. If not NULL, the minimum non-zero value a weight can have, i.e., weight = max(minWgt, weight). If NULL, no lower truncation of weights is performed.

`maxWgt` A numeric object. If not NULL, the maximum value a weight can have, i.e., weight = min(maxWgt, weight). If NULL, no upper truncation of weights is performed.

`txName` A character object. The header of the column of data containing the treatment variable. Default value is 'A'. Treatment must be coded as 0/1, where 1 indicates that participant was vaccinated; 0 otherwise.

`U` A character object. The header of the column of data containing the minimum of time to infection, time to refusing study vaccine after unblinding (placebo participants only), or time to censoring (due to loss to follow up or administrative censoring).

`entryTime` A character object. The header of the column of data containing the time of entry into the study on the scale of the calendar time. Default value is 'E'.

`Gamma` A character object. The header of the column of data containing the category for the R dynamic. Default value is 'Gam'. Data must be 0/1, where 1 indicates that R is the time to unblinding; 0 indicates that R is the infection time or the censoring time.

`R` A character object. The header of the column of data containing the minimum of: time to unblinding, time to infection, or time to censoring.

`Psi` A character object. The header of the column of data containing the indicator of whether a participant received study vaccine, where 1 indicates that a participant was assigned to placebo is unblinded and decides to get the study vaccine or that a participant was assigned to vaccine; 0 otherwise. Default value is 'Psi'.

`Delta` A character object. The header of the column of data containing the infection-variant category indicator.

Details

Analysis data must include the following information:

E The study entry time.

A Binary treatment assignment, where A=0 indicates placebo; A = 1 otherwise.

U The minimum of: time to infection, time to refusing study vaccine after unblinding (placebo participants only), or time to censoring (loss to followup or administrative censoring due to analysis at time L).

Delta Infection-variant category indicator, where: -1 indicates a placebo participant was unblinded and refused study vaccine; 0 indicates censoring; and = ve indicates infection of variant ve (ve = 1, ..., nV).

R The minimum of: time to unblinding, time to infection, or time to censoring.

Gamma Indicator of R dynamic, where 1 indicates that R is the time to unblinding and 0 indicates that R is either the time to infection or the time to censoring.

Psi Indicator of whether a subject received study vaccine, where 1 indicates that a participant was assigned to vaccine or was assigned to placebo is unblinded and decides to get the study vaccine; 0 otherwise.

Data can include baseline covariates. Methods for time-dependent covariates are not currently available.

If input lag is provided as a numeric vector or as a column of the data, lag values should be set to Inf or NA for participants that do not reach full efficacy.

There are 3 possible weighting options, the selection of which is determined by the combination of model inputs.

No weighting: No models are provided as input, i.e., inputs modelEntry, modelUnblind0, modelUnblind1, modelPsi, modelCensor0, and modelCensor1 are not provided or are NULL.

Weighting depends only on the censoring dynamic: Models modelCensor0 and modelCensor1 must be provided and models modelEntry, modelUnblind0, modelUnblind1, modelPsi must be NULL.

Weighting depends on the entry, unblinding, and censoring dynamics: All models must be provided.

The returned S3 object has 6 attributes needed for post-processing tools ve() and plot(). Specifically, "gFunc" is an integer object specifying the model selected for the infection rate (based on input gFunc); "v", the knots or cut-offs to be used by gFunc (input v); "maxTime", the maximum time since full efficacy included in the analysis; "phaseType", =1 only unblinded phase, =2, only blinded phase, =3 both phases, and "wgtType", =0 no weighting, =1 censor weighting, =2 full weighting.

Value

A an S3 object of class "VEwaningVariant", which comprises a list object containing

`theta`	A vector object containing the estimated theta parameters.
`cov`	The covariance estimated using the sandwich estimator.
`SE`	The standard error estimated using the sandwich estimator.

and attributes "gFunc", "maxTime", "v", "phaseType", and "wgtType", which store details of the original analysis that are required for post-processing convenience functions ve() and plot(). See details for further information.

Examples

```
data(variantData)

set.seed(1234)

ind <- sample(1:nrow(variantData), 2000)
# NOTE: This sample size is chosen for example only -- larger data sets
# should be used.

# no weighting -- variant 1 infection only

res_noWgt <- veWaningVariant(data = variantData[ind,],
                             L = 52.0,
                             variant = 1L,
                             gFunc = 'piece',
                             v = c(5.0,10.0))

# censoring only weighting -- variant 1 infection only

res_cens <- veWaningVariant(data = variantData[ind,],
                            L = 52.0,
                            variant = 1L,
                            modelCensor0 = ~ X1+X2,
                            modelCensor1 = ~ X1+X2,
                            gFunc = 'piece',
                            v = c(5.0,10.0))

# full weighting -- variant 1 infection only

## Not run: res_full <- veWaningVariant(data = variantData[ind,],
                            L = 52.0,
                            variant = 1L,
                            modelEntry = ~ X1,
                            modelUnblind0 = ~X1+X2,
                            modelUnblind1 = ~X2,
                            modelPsi = ~X1*X2,
                            modelCensor0 = ~ X1+X2,
                            modelCensor1 = ~ X1+X2,
                            gFunc = 'piece',
                            v = c(5.0,10.0))
## End(Not run)
```

Index

∗ **datasets**
variantData, 4

plot, 2
print, 3

variantData, 4
ve, 5
veWaningVariant, 6