

Package: vaccineff (via r-universe)

September 16, 2024

Title Estimate Vaccine Effectiveness Based on Different Study Designs

Version 0.0.3

Description R package with tools for estimating vaccine effectiveness and vaccine related metrics.

URL <https://github.com/epiverse-trace/vaccineff>,
<https://epiverse-trace.github.io/vaccineff/>

BugReports <https://github.com/epiverse-trace/vaccineff/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports survival, stats, checkmate, ggplot2, scales, rlang, MatchIt

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0), qtl

Config/testthat/edition 3

Config/Needs/website epiverse-trace/epiversetheme

Language en-US

VignetteBuilder knitr

Repository <https://epiverse-trace.r-universe.dev>

RemoteUrl <https://github.com/epiverse-trace/vaccineff>

RemoteRef HEAD

RemoteSha 9e183c78c7d60c1b5cf41ac62f7b93818674e6d8

Contents

cohortdata	2
coh_coverage	3
effectiveness	4
get_age_group	6
get_dataset	7
get_dataset.match	7
make_immunization	8
match_cohort	9
plot.effectiveness	12
plot_coverage	12
plot_survival	13
summary.effectiveness	15
summary.match	16
Index	17

cohortdata	<i>Cohort data on vaccineff</i>
------------	---------------------------------

Description

Data from

Usage

cohortdata

Format

An object of class "cross"; see [read.cross](#).

Examples

cohortdata

coh_coverage	<i>Calculate Vaccine Coverage</i>
--------------	-----------------------------------

Description

This function returns the vaccination coverage of a dose along the cohort study. The coverage can be calculated grouped by year, day, and month. This must be specified in the parameter `unit`. If there are no records for some dates, the function assigns 0 instead of NA to calculate the cumulative coverage.

Usage

```
coh_coverage(  
  data_set,  
  vacc_date_col,  
  unit = c("day", "month", "year"),  
  date_interval = NULL  
)
```

Arguments

<code>data_set</code>	data.frame with cohort information (see example).
<code>vacc_date_col</code>	Name of the column(s) that contain the vaccine date to calculate the coverage.
<code>unit</code>	Aggregation unit, must be either "year", "month", or "day".
<code>date_interval</code>	If NULL, the function calculates the coverage interval based on the <code>min()</code> and <code>max()</code> of the <code>vacc_date_col</code> . It is also possible to pass a custom date interval to truncate or expand the date interval (see example).

Value

data.frame with the number of vaccine doses per date, cumulative count of doses, and vaccine coverage.

Examples

```
# Load example data  
data("cohortdata")  
  
# Define date intervals for coverage  
start_cohort <- as.Date("2044-01-01")  
end_cohort <- as.Date("2044-12-31")  
date_interval <- c(start_cohort, end_cohort)  
  
# Calculate coverage  
coh_coverage(  
  data_set = cohortdata,  
  vacc_date_col = "vaccine_date_1",
```

```

    unit = "month",
    date_interval = date_interval
  )

```

effectiveness

Estimate Vaccine Effectiveness (VE)

Description

This function provides methods for estimating VE. It relies on the implementation of the Kaplan-Meier estimator and the Cox model for proportional hazards in the package `{survival}`. Currently, the default method of the function is HR (Hazard Ratio). Thus, the $VE = 1 - HR$, where HR is calculated using the Cox model. The proportional hazards hypothesis is tested using the Schoenfeld test, and the resultant p-value is provided in the results. Log-log plots are also generated using the Kaplan-Meier survival estimator to provide a visual test for the proportional hazards hypothesis. The functions uses the default name of columns `outcome_status`, `time_to_event` and `vaccine_status`, and the default status names `v` and `u`. However, custom names can be provided through the parameters `outcome_status_col`, `time_to_event_col`, `vacc_status_col`, `vaccinated_status` and `unvaccinated_status`. The return is a list with the call and the name of the method used for the estimation of VE (CI95%), the result of the performance test, and a suitable plot for the method. The object returned is compatible with the methods `summary` and `plot`.

Usage

```

effectiveness(
  data_set,
  start_cohort,
  end_cohort,
  method = "HR",
  outcome_status_col = "outcome_status",
  time_to_event_col = "time_to_event",
  vacc_status_col = "vaccine_status",
  vaccinated_status = "v",
  unvaccinated_status = "u"
)

```

Arguments

<code>data_set</code>	data.frame with cohort information (see example).
<code>start_cohort</code>	Start date of the study.
<code>end_cohort</code>	End date of the study.
<code>method</code>	Method to estimate VE. Default is HR.
<code>outcome_status_col</code>	Name of the column containing status of the event (must be a binary column). Default is <code>outcome_status</code> .
<code>time_to_event_col</code>	Name of the column containing the time-to-event. Default is <code>time_to_event</code> .

`vacc_status_col`
Name of the column containing the vaccination. Default is `vaccine_status`.

`vaccinated_status`
Status assigned to the vaccinated population. Default is `v`.

`unvaccinated_status`
Status assigned to the unvaccinated population. Default is `u`.

Value

Object of the class `effectiveness`: list with results from estimation of VE. `call`: call of `{survival}` method, `ve`: `data.frame` with VE(CI95%), `test`: result from test of performance, `plot`: plot of method, `method`: name of the method used for the estimation.

Examples

```
# Define start and end dates of the study
start_cohort <- as.Date("2044-01-01")
end_cohort <- as.Date("2044-12-31")

# Create `data.frame` with information of immunization
cohortdata <- make_immunization(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  immunization_delay = 14,
  vacc_date_col = c("vaccine_date_2"),
  end_cohort = end_cohort,
  take_first = FALSE
)
head(cohortdata)

# Match the data
matching <- match_cohort(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  start_cohort = start_cohort,
  end_cohort = end_cohort,
  method = "static",
  exact = "sex",
  nearest = c(age = 1)
)

# Extract matched data
cohortdata_match <- get_dataset(matching)

# Calculate vaccine effectiveness
ve <- effectiveness(
  data_set = cohortdata_match,
  start_cohort = start_cohort,
  end_cohort = end_cohort
)
```

```
# View summary of VE
summary(ve)

# Generate plot of method
plot(ve)
```

get_age_group	<i>Construct age-group variable from age column</i>
---------------	---

Description

This method splits an age interval from `min_val` to `max_val` into intervals of size `step`. If the method finds ages greater or equal than `max_val` it assigns the string `">max_val"`. By default `min_val` is set to 0, however it can be assigned by convenience. If the method finds ages lower or equal than `min_val` it assigns the string `"<min_val-1"`. The function warns when `(max_val - min_val)` is not an integer multiple of `step`. In that case the last interval is truncated to the upper value closest to `max_val` for which `(closest_upper - min_val)` is multiple of `step`.

Usage

```
get_age_group(data_set, col_age, max_val, min_val = 0, step)
```

Arguments

<code>data_set</code>	data.frame with at least a column containing the age information
<code>col_age</code>	Name of the column containing the age information
<code>max_val</code>	Maximum value of age interval to split
<code>min_val</code>	Minimum value of age interval to split
<code>step</code>	Step used to split the age interval

Value

Column of type factor with the same length as the number of rows in `data_set`, with levels corresponding to age bins between `min_val` and `max_val`. Ages above `max_val` are represented as `>max_val`.

Examples

```
# load data provided with the package
data(cohortdata)

# assign age groups as a column of the `data.frame`
cohortdata$age_group <- get_age_group(
  data_set = cohortdata,
  col_age = "age",
  max_val = 80,
```

```

    step = 10
  )

# view the `data.frame` with new column
head(cohortdata)

```

get_dataset

Generic Function for Extracting Datasets

Description

This function serves as a generic method to extract datasets from objects of various classes. It dispatches to specific methods based on the class of the input object.

Usage

```
get_dataset(object, ...)
```

Arguments

object	An object from which to extract the dataset.
...	Additional arguments passed to other functions.

Value

The get_dataset extracted from the object.

get_dataset.match

Function for Extracting Matched Dataset

Description

This function extracts the matched datasets generated by match_cohort.

Usage

```

## S3 method for class 'match'
get_dataset(object, ...)

```

Arguments

object	Object of the class match.
...	Additional arguments passed to other functions.

Value

data.frame extracted from match_cohort.

make_immunization	<i>Construct Information of Immunization</i>
-------------------	--

Description

This function returns a `data.frame` with the relevant information on immunization for the study. An individual is considered immunized after the characteristic time for the immune response of the vaccine passes. This time is provided to the function by the parameter `immunization_delay`. By default, it is set to 0. The function searches for vaccine dates based on the end of the follow-up period of each individual defined by the censoring date (if provided), outcome delay (if present), and end of the study.

The function also works with vaccination information spread across several columns. If this is the case, the parameter `vacc_date_col` must be passed as a vector with the names of all the columns to use (see example). The function uses by default the latest date found. However, it can also select the first date by setting `take_first = TRUE`.

The function returns a column with the immunization date (`immunization`) and a vaccine status column (`vaccine_status`) that is constructed based on `immunization`. For several vaccines, the function also returns the name of the column of the vaccine that was selected as immunizing (`immunizing_dose`). If different custom names (e.g., brands) are associated with each vaccine date, the function can return the custom name of the vaccine selected as immunizing. This information must be passed in the parameter `vacc_name_col`, as a vector in the same order as `vacc_date_col`.

Usage

```
make_immunization(
  data_set,
  outcome_date_col,
  censoring_date_col = NULL,
  vacc_date_col,
  vacc_name_col = NULL,
  vaccinated_status = "v",
  unvaccinated_status = "u",
  immunization_delay = 0,
  end_cohort,
  take_first = FALSE
)
```

Arguments

<code>data_set</code>	<code>data.frame</code> with cohort information (see example).
<code>outcome_date_col</code>	Name of the column that contains the outcome dates.
<code>censoring_date_col</code>	Name of the column that contains the censoring date. NULL by default.
<code>vacc_date_col</code>	Name of the column(s) that contain the vaccine dates.

vacc_name_col Name of the column(s) that contain custom vaccine names for the vaccines (e.g. brand name, type of vaccine)

vaccinated_status Status assigned to the vaccinated population. Default is v.

unvaccinated_status Status assigned to the unvaccinated population. Default is u.

immunization_delay Characteristic time in days before the patient is considered immune.

end_cohort End date of the study.

take_first FALSE: takes the latest vaccine date. TRUE: takes the earliest vaccine date.

Value

Original data.frame passed in data_set and additional columns containing information on the immunization.

Examples

```
# Load data
data(cohort_data)
# Define start and end dates of the study
start_cohort <- as.Date("2044-01-01")
end_cohort <- as.Date("2044-12-31")

# Create `data.frame` with information of immunization
cohortdata <- make_immunization(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  immunization_delay = 14,
  vacc_date_col = "vaccine_date_2",
  end_cohort = end_cohort
)
# `immunization_date` and `vaccine_status` are added to `cohortdata`
head(cohortdata)
```

match_cohort

Match Cohort to Reduce Observational Bias

Description

This function builds pairs of vaccinated and unvaccinated individuals with similar characteristics. The function relies on the matching algorithm implemented in the package {MatchIt}, setting, by default, method = "nearest", ratio = 1, and distance = "mahalanobis". Exact and near characteristics are accepted, passed in the parameters exact and nearest, respectively. The parameter nearest must be provided together with the calipers as a named vector (e.g., nearest = c(characteristic1 = n1, characteristic2 = n2), where n1 and n2 are the calipers). The default matching method of the function is static. This means that pairs are matched once, without

taking into account their vaccination, censoring, and outcome dates. After this, the pairs whose exposition time do not coincide are removed to avoid negative time-to-events. The function returns a matched and adjusted by exposition cohort, with information of the beginning of follow-up period of pairs (`t0_follow_up`), corresponding to the vaccination date of the vaccinated individual, the individual time-to-event (`time_to_event`) and the outcome status (`outcome_status`), both taking into account the right-censoring dates. Pairs are censored if the vaccinated or unvaccinated partner was previously censored (i.e., if `censoring_date_col` is informed) and the censor occurs before their outcomes. Rolling calendar matching method will be included in future releases.

Usage

```
match_cohort(
  data_set,
  outcome_date_col,
  censoring_date_col,
  start_cohort,
  end_cohort,
  method = "static",
  nearest = NULL,
  exact = NULL,
  immunization_date_col = "immunization_date",
  vacc_status_col = "vaccine_status",
  vaccinated_status = "v",
  unvaccinated_status = "u"
)
```

Arguments

<code>data_set</code>	data.frame with cohort information (see example).
<code>outcome_date_col</code>	Name of the column that contains the outcome dates.
<code>censoring_date_col</code>	Name of the column that contains the censoring date. NULL by default.
<code>start_cohort</code>	Start date of the study.
<code>end_cohort</code>	End date of the study.
<code>method</code>	Method to match the cohort. Default is <code>static</code> .
<code>nearest</code>	Named vector with name(s) of column(s) for nearest matching and caliper(s) for each variable (e.g., <code>nearest = c("characteristic1" = n1, "characteristic2" = n2)</code> , where <code>n1</code> and <code>n2</code> are the calipers). Default is NULL.
<code>exact</code>	Name(s) of column(s) for exact matching. Default is NULL.
<code>immunization_date_col</code>	Name of the column that contains the immunization date to set the beginning of the follow-up period (<code>t0_follow_up</code>). Default is <code>immunization_date</code> .
<code>vacc_status_col</code>	Name of the column containing the vaccination. Default is <code>vaccine_status</code> .
<code>vaccinated_status</code>	Status assigned to the vaccinated population. Default is <code>v</code> .

unvaccinated_status

Status assigned to the unvaccinated population. Default is u.

Value

object of the class match. List with results from static match: match: data.frame with adjusted cohort, summary: matching summary, balance_all: balance of the cohort before matching, balance_matched: balance of the cohort after matching.

Four columns are added to the structure provided in data_set: subclass: ID of, matched pair, t0_follow_up: beginning of follow-up period for pair, time_to_event: time to event, and outcome_status: outcome status (1:positive, 0: negative).

Examples

```
# Define start and end dates of the study
start_cohort <- as.Date("2044-01-01")
end_cohort <- as.Date("2044-12-31")

# Create `data.frame` with information on immunization
cohortdata <- make_immunization(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  immunization_delay = 14,
  vacc_date_col = "vaccine_date_2",
  end_cohort = end_cohort
)

# Match the data_set
matching <- match_cohort(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  start_cohort = start_cohort,
  end_cohort = end_cohort,
  method = "static",
  exact = "sex",
  nearest = c(age = 1)
)

# Check match balance and summary
# `warnings_log = TRUE` displays the logs created
# during the iterative process.
summary(matching, warnings_log = TRUE)

# Extract matched data
cohortdata_match <- get_dataset(matching)

# View of mached cohort
head(cohortdata_match)
```

plot.effectiveness *Function for Extracting VE plot*

Description

This function extracts the plot generated by effectiveness.

Usage

```
## S3 method for class 'effectiveness'
plot(x, ...)
```

Arguments

x Object of the class effectiveness.
 ... Additional arguments passed to other functions.

Value

Plot extracted from effectiveness.

plot_coverage *Plot Vaccine Coverage*

Description

This function returns a plot of the vaccine coverage or the cumulative coverage (if cumulative = TRUE). The return is a 2-axis ggplot2 element with the number of vaccines per date on the left axis and the coverage per date on the right axis. Colors can be defined with the parameters of the function doses_count_color and coverage_color. Other graphic elements such as labels and legends can be manipulated using ggplot2.

Usage

```
plot_coverage(
  data_set,
  vacc_date_col,
  unit = c("day", "month", "year"),
  doses_count_color = "steelblue",
  coverage_color = "mediumpurple",
  date_interval = NULL,
  cumulative = FALSE
)
```

Arguments

data_set	data.frame with cohort information (see example).
vacc_date_col	Name of the column(s) that contain the vaccine date to calculate the coverage.
unit	Aggregation unit, must be either "year", "month", or "day".
doses_count_color	Color assigned to the doses count.
coverage_color	Color assigned to the coverage calculation.
date_interval	If NULL, the function calculates the coverage interval based on the min() and max() of the vacc_date_col. It is also possible to pass a custom date interval to truncate or expand the date interval (see example).
cumulative	If TRUE, returns the cumulative number of doses over the time window.

Value

2-axis ggplot2 plot of vaccine coverage and daily doses.

Examples

```
# Load example data
data("cohortdata")

# Define date intervals for coverage
start_cohort <- as.Date("2044-01-01")
end_cohort <- as.Date("2044-12-31")
date_interval <- c(start_cohort, end_cohort)

# Plot coverage
plot_coverage(
  data_set = cohortdata,
  vacc_date_col = "vaccine_date_1",
  unit = "month",
  doses_count_color = "steelblue",
  coverage_color = "mediumpurple",
  date_interval = date_interval,
  cumulative = FALSE
)
```

plot_survival

Plot the Survival Probability Based on the Kaplan-Meier Model

Description

This function relies on the implementation of the Kaplan-Meier model from the package `{survival}`. It returns a plot of the Survival Probability or the Cumulative Hazard (if `cumulative = TRUE`). The return is a ggplot2 element of the curves with 95% C.I. It is possible to manipulate the colors, labels, legend, and most of the graphic elements.

Usage

```
plot_survival(  
  data_set,  
  start_cohort,  
  end_cohort,  
  outcome_status_col = "outcome_status",  
  time_to_event_col = "time_to_event",  
  vacc_status_col = "vaccine_status",  
  vaccinated_status = "v",  
  unvaccinated_status = "u",  
  vaccinated_color = "steelblue",  
  unvaccinated_color = "darkred",  
  percentage = TRUE,  
  cumulative = FALSE  
)
```

Arguments

<code>data_set</code>	data.frame with cohort information (see example).
<code>start_cohort</code>	Start date of the study.
<code>end_cohort</code>	End date of the study.
<code>outcome_status_col</code>	Name of the column containing status of the event (must be a binary column). Default is <code>outcome_status</code> .
<code>time_to_event_col</code>	Name of the column containing the time-to-event. Default is <code>time_to_event</code> .
<code>vacc_status_col</code>	Name of the column containing the vaccination. Default is <code>vaccine_status</code> .
<code>vaccinated_status</code>	Status assigned to the vaccinated population. Default is <code>v</code> .
<code>unvaccinated_status</code>	Status assigned to the unvaccinated population. Default is <code>u</code> .
<code>vaccinated_color</code>	Color assigned to the vaccinated population.
<code>unvaccinated_color</code>	Color assigned to the unvaccinated population.
<code>percentage</code>	If TRUE, returns probability in percentage.
<code>cumulative</code>	If TRUE, returns cumulative hazards.

Value

{ggplot2} object with plot of survival or cumulative incidence.

Examples

```
# Define start and end dates of the study
start_cohort <- as.Date("2044-01-01")
end_cohort <- as.Date("2044-12-31")

# Create `data.frame` with information on immunization
cohortdata <- make_immunization(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  immunization_delay = 14,
  vacc_date_col = "vaccine_date_2",
  end_cohort = end_cohort
)

# Match the data
matching <- match_cohort(
  data_set = cohortdata,
  outcome_date_col = "death_date",
  censoring_date_col = "death_other_causes",
  start_cohort = start_cohort,
  end_cohort = end_cohort,
  method = "static",
  exact = "sex",
  nearest = c(age = 1)
)

# Extract matched data
cohortdata_match <- get_dataset(matching)

# Plot survival curve
plot_survival(
  data_set = cohortdata_match,
  start_cohort = start_cohort,
  end_cohort = end_cohort
)
```

summary.effectiveness *Summarize VE Results*

Description

Summarizes the results of effectiveness.

Usage

```
## S3 method for class 'effectiveness'
summary(object, ...)
```

Arguments

object Object of the class effectiveness.
... Additional arguments passed to other functions.

Value

Summary of the results from effectiveness.

summary.match	<i>Summarize Matching Results</i>
---------------	-----------------------------------

Description

Summarizes the results of match_cohort.

Usage

```
## S3 method for class 'match'  
summary(object, warnings_log = FALSE, ...)
```

Arguments

object Object of the class match.
warnings_log If TRUE, prints the warnings log.
... Additional arguments passed to other functions.

Value

Summary of the results from matching.

Index

* datasets

- cohortdata, [2](#)

- coh_coverage, [3](#)
- cohortdata, [2](#)

- effectiveness, [4](#)

- get_age_group, [6](#)
- get_dataset, [7](#)
- get_dataset.match, [7](#)

- make_immunization, [8](#)
- match_cohort, [9](#)

- plot.effectiveness, [12](#)
- plot_coverage, [12](#)
- plot_survival, [13](#)

- read.cross, [2](#)

- summary.effectiveness, [15](#)
- summary.match, [16](#)