Package 'rprev'

May 4, 2021

Type Package

Title Estimating Disease Prevalence from Registry Data

Version 1.0.5

Date 2021-04-30

Description Estimates disease prevalence for a given index date using existing
registry data extended with Monte Carlo simulations follow-
ing the method of Crouch et al (2014) <doi: 10.1016/j.canep.2014.02.005>.

LazyData true

License GPL-2

Depends R (>= 2.10), survival

Imports data.table, dplyr, ggplot2, lazyeval, lubridate, magrittr,
tidyr

RoxygenNote 7.1.1

Suggests flexsurv, flexsurvcure, knitr, rmarkdown, rms, testthat, covr

URL https://github.com/stulacy/rprev-dev

VignetteBuilder knitr

NeedsCompilation no

Author Stuart Lacy [cre, aut],
Simon Crouch [aut],
Stephanie Lax [aut]

Maintainer Stuart Lacy <stuart.lacy@york.ac.uk>

Repository CRAN

Date/Publication 2021-05-04 16:40:03 UTC

R topics documented:

---

counted_prevalence   *Count prevalence from registry data.*

Description

Counts contribution to prevalence at a specific index from each year of a registry. A person is included as contributing to disease prevalence if they are incident within the specified time-span, and are either alive or censored at the index date. The rationale for including censored cases in prevalence estimation is that such cases have typically been lost to follow-up, and are often more likely to have been alive at the index date than not.

Usage

```
counted_prevalence(formula, index, data, start_date, status_col)
```

Arguments

formula	A formula of the form <event date column> ~ <entry date column>.
index	The date at which to estimate point prevalence as a string in the format YYYY-MM-DD.
data	A data frame with the corresponding column names provided in form.
start_date	The initial date to start counting prevalence from as a Date object. Typically the index date - (Nyears * 365.25). Allows for non-whole year prevalence estimations.
status_col	The name of the column holding a binary indicator variable of whether the individual experienced an event at their event time or was censored.

Value

The number of prevalent cases at the specified index date as a single integer.

---

draw_incident_population

> *Simulates an incident population according to a specific incidence model*

Description

This method defines the main behaviour of an incidence model and must be implemented for any class to be used in `prevalence`.

Usage

```
draw_incident_population(object, data, timeframe, covars)
```

Arguments

object	The incidence model.
data	The original registry data frame passed into `prevalence`. It is supplied in this method so that individual level attributes can be sampled from the prior distribution.
timeframe	The amount of time in days in which to simulate incidence for.
covars	Any patient level covariates that must be included in the new simulated incident population, as a character vector. These will correspond to columns in `data`.

Value

A data frame where each row corresponds to a simulate incident patient. The first column must be incidence time (in days). This will be relative to an unspecified baseline. All covariates specified in `covars` must be present in this data frame.

---

extract_covars

> *Returns the name of the covariates in the registry data set that are required by the survival model.*

Description

Used in `prevalence` to determine which covariates should be sampled when simulating an incident population. This should provide a character vector containing column names of the original registry data set input to `prevalence` that are used by the survival model.

Usage

```
extract_covars(object)
```

Arguments

object   The survival object itself

Value

A character vector holding the column names. Can be NULL if no covariates are used i.e. the survival model is built based on population level data rather than individual level data.

---

fixed_cure   *Builds survival models for diseases with cured fractions using population mortality tables*

Description

Fits a cure model which assumes that if an individual has survived beyond a set time-point then they are considered cured and their mortality reverts to population levels. Please read the detailed description below for how to use this model.

Usage

```
fixed_cure(
  formula = NULL,
  data = NULL,
  cure_time = 10 * 365.25,
  daily_survival = NULL,
  population_covariates = NULL,
  dist = c("exponential", "weibull", "lognormal")
)
```

Arguments

formula   Formula specifying survival function, as used in [prevalence](#) with the surv_formula argument. **Must be in days**.

data   A data frame with the corresponding column names provided in form.

cure_time   Time-limit at which a patient is considered cured. Note that if this is 0 or negative then survival will be based purely off the population rates (anything passed into formula and data will be ignored).

daily_survival A data frame comprising population survival as a daily probability for as long as possible, ideally 100 years (36525 days). Defaults to using UK population survival from the UKmortality data set. It **must contain columns 'age' and 'surv'**, providing the age (in days) and survival probability at that age respectively. It can also be stratified by other variables that are found in the survival formula for this model, such as sex.

population_covariates

> A character vector containing fields to stratify population survival by in addition to age, as described in Details below. These **must** be the names of columns in both `data` and `daily_survival`. If not provided then defaults to the fields that are present in both `data` and `daily_survival`.

dist   The distribution used by the default parametric survival model.

Details

To model population survival, population mortality tables are required, as specified by the `daily_survival` argument. If not provided, then the default population mortality is that of the UK population, which goes up to 100 years of age. If a simulated individual has expected lifespan longer than the maximum age in the mortality table then they are estimated to have died at this age limit, which is why it is advantageous to provide as many accurate survival probabilities as possible.

Due to the linking with the registry data and the ability for user-specified mortality tables, there are stricter requirements on the survival models used in cure models than elsewhere. For example, the time-scale of the survival model specified in `formula` **must be in days** so that it matches up with the mortality tables. Likewise, **age in years must be included as a covariate** in the survival model

Value

An object of class `fixedcure` that can be passed into prevalence.

---

plot.incdiag	*Visualise disease incidence.*

Description

Plots a comparison between the smoothed daily incidence function and actual incidence.

Usage

```
## S3 method for class 'incdiag'
plot(x, level = 0.95, ...)
```

Arguments

x   An `incidence` object.

level   The desired confidence interval width.

...   Arguments passed to `plot`.

Details

This function generates a plot from the cumulative incidence object. The incidence rate per year of the registry is shown in red. Mean incidence rate is shown as a solid blue line, with the confidence interval shown in dashed blue lines. The smooth fitted to the cumulative incidence data is shown in green.

Value

An object of class `ggplot`.

Examples

```
data(prevsim)

## Not run:
inc <- test_homogeneity(prevsim$entrydate, population_size=1e6,
                        start = "2004-01-30", num_reg_years = 9)

plot(inc)

## End(Not run)
```

---

plot.survfit.prev   *Plot bootstrapped survival curves.*

Description

This method plots survival curves for a `survfit.prev` object.

Usage

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

Arguments

x   A `survfit.prev` object.

...   Arguments passed to `plot`.

Details

The survival curve for a model formed on all the data is displayed in orange, while the 95 as a grey
ribbon.

Value

An S3 object of class `ggplot`.

Examples

```
data(prevsim)

## Not run:
prev_obj <- prevalence(Surv(time, status) ~ age(age) + sex(sex) +
                       entry(entrydate) + event(eventdate),
                       data=prevsim, num_years_to_estimate = c(5, 10),
                       population_size=1e6, start = "2005-09-01",
                       num_reg_years = 8, cure = 5)

survobj <- survfit(prev_obj, newdata=list(age=65, sex=0))

plot(survobj)

## End(Not run)
```

---

predict_survival_probability

Predicts survival probability for given individuals at specific times.

Description

This generic method is required for any survival object used in the main prevalence function.

Usage

```
predict_survival_probability(object, newdata, times)
```

Arguments

object	The survival object itself
newdata	Simulated incident individuals, with the same attributes specified in extract_covars and found in the supplied registry data to prevalence.
times	The time at which to estimate the survival probability of the individual in the corresponding row of newdata. Must have as many times as newdata has rows.

Value

A vector with the same length as times providing the survival probability estimates.

| prevalence | *Estimate point prevalence at an index date.* |

Description

Point prevalence at a specific index date is estimated using contributions to prevalence from both available registry data, and from Monte Carlo simulations of the incidence and survival process, as outlined by Crouch et al (2004) (see References).

Usage

```
prevalence(
  index,
  num_years_to_estimate,
  data,
  inc_formula = NULL,
  inc_model = NULL,
  surv_formula = NULL,
  surv_model = NULL,
  registry_start_date = NULL,
  death_column = NULL,
  incident_column = NULL,
  age_column = "age",
  age_dead = 100,
  status_column = "status",
  N_boot = 1000,
  population_size = NULL,
  proportion = 1e+05,
  level = 0.95,
  dist = c("exponential", "weibull", "lognormal"),
  precision = 2
)
```

Arguments

index	The date at which to estimate point prevalence as a string in the format YYYY-MM-DD.
num_years_to_estimate	
	Number of years of data to consider when estimating point prevalence; multiple values can be specified in a vector. If any values are greater than the number of years of registry data available before `index_date`, incident cases for the difference will be simulated.
data	A data frame with the corresponding column names provided in `form`.
inc_formula	A formula specifying the columns used in the incidence process. The LHS should be the name of the column holding the incident dates, with the RHS specifying any variables that should be stratified by, or 1 if no stratification. For example, with the supplied `prevsim` data set, it could be used as follows:

entrydate ~ 1 for a non-stratified process. entrydate ~ sex for a process that will stratify incidence by sex.

inc_model	An object that has a draw_incident_population method. See the vignette for further guidance.
surv_formula	A formula used to specify a survival model, where the LHS a Surv object, as used by flexsurvreg.
surv_model	An object that has a predict_survival_probability method. See the vignette for further guidance.
registry_start_date	
	The starting date of the registry. If not supplied then defaults to the earliest incidence date in the supplied data set.
death_column	A string providing the name of the column which holds the death date information. If not provided then prevalence cannot be counted and estimates will be solely derived from simulation.
incident_column	
	A string providing the name of the column which holds the diagnosis date. If not provided either in this argument or in inc_formula, then prevalence cannot be counted and estimates will be solely derived from simulation.
age_column	A string providing the name of the column that holds patient age. If provided then patients alive at age_dead are set to die. This helps combat 'immortal' patients.
age_dead	The age at which patients are set to be dead if they are still alive, to prevent 'immortal' patients. Used in conjunction with age_column.
status_column	A string providing the name of the column that holds patient event status at the event time. If not provided in surv_formula or in this argument then prevalence cannot be counted.
N_boot	Number of bootstrapped calculations to perform.
population_size	
	Integer corresponding to the size of the population at risk.
proportion	The population ratio to estimate prevalence for.
level	Double representing the desired confidence interval width.
dist	The distribution used by the default parametric survival model.
precision	Integer representing the number of decimal places required.

Details

The most important parameter is num_years_to_estimate, which governs the number of previous years of data to use when estimating the prevalence at the index date. If this parameter is greater than the number of years of known incident cases available in the supplied registry data (specified with argument num_registry_years), then the remaining num_years_to_estimate -num_registry_years years of incident data will be simulated using Monte Carlo simulation.

The larger num_years_to_estimate, the more accurate the prevalence estimate will be, provided an adequate survival model can be fitted to the registry data. It is therefore important to provide as much clean registry data as possible.

Prevalence arises from two stochastic processes: incidence and survival. This is reflected in the function arguments by multiple options for each of these processes.

The incidence process is specified by an object that has an associated `draw_incident_population` method, which produces the new incident population. The default implementation is a homogeneous Poisson process, whereby interarrival times are distributed according to an exponential distribution. The `inc_formula` argument specifies the nature of this process, see the description for more details. See the vignette for guidance on providing a custom incidence object.

The survival process is characterised by a method `predict_survival_probability`, that estimates the probability of a given individual being alive at the index date. The default object is a parametric distribution with the functional form being specified in `surv_formula` and distribution given in `dist`. See the vignette for guidance on providing a custom survival model.

Value

A `prevalence` object containing the following attributes:

`estimates`	Prevalence estimates at the specified years as both absolute and rates.
`simulated`	A `data.table` containing simulated incident cases from each bootstrap iteration Each row corresponds to a simulated incident case with their simulated attributes and survival status. Binary flags are provided beginning `prev_`, which indicate whether that person contributed to the prevalence for the specified time-period. The `prev_registry` flag indicates whether that person was incident during the registry time-span and alive at the index. These cases are used to assess the model fit, as the numbers can be simply compared to the known registry prevalence.
`counted`	The number of incident cases present in the registry data set.
`full_surv_model`	
	The survival model built on the complete registry data set.
`full_inc_model`	The incidence model built on the complete registry data set.
`surv_models`	A list of the survival models fitted to each bootstrap iteration.
`inc_models`	A list of the incidence models fitted to each bootstrap iteration.
`index_date`	The index date.
`est_years`	The years at which prevalence is estimated at.
`counted_incidence_rate`	
	The overall incidence rate in the registry data set.
`registry_start`	The date the registry was identified at starting at.
`proportion`	The denominator to use for estimating prevalence rates.
`status_col`	The column in the registry data containing the survival status.
`N_boot`	The number of bootstrap iterations that were run.
`means`	Covariate means, used when plotting Kaplan-Meier estimators using `survfit`.
`max_event_time`	The maximum time-to-event in the registry data. Again, used in `survfit` to scale the time-axis.
`pval`	The p-value resulting from a hypothesis test on the difference between the simulated and counted prevalence on the time-span covered by the registry. Tests the prevalence fit; if a significant result is found then further diagnostics are required.

References

Crouch, Simon, et al. "Determining disease prevalence from incidence and survival using simulation techniques." Cancer epidemiology 38.2 (2014): 193-199.

See Also

Other prevalence functions: [test_prevalence_fit](#)()

Examples

```
data(prevsim)

## Not run:
data(prevsim)

prevalence(index='2013-01-30',
           num_years_to_estimate=c(3, 5, 10, 20),
           data=prevsim,
           inc_formula = entrydate ~ sex,
           surv_formula = Surv(time, status) ~ age + sex,
           dist='weibull',
           population_size = 1e6,
           death_column = 'eventdate')

## End(Not run)
```

---

prevsim   *Simulated patient dataset.*

Description

A dataset in the format of a disease registry, where the outcome being modelled is death due to the simulated disease. The registry began in January 2003, with 1000 incident cases being recorded over a period of nearly exactly ten years. The patients are followed up for a further two years until 17.03.2015, at which point any subjects alive are marked as right censored.

Usage

```
prevsim
```

Format

A data frame with 1000 rows and 6 columns:

time time between date of diagnosis and death or censorship in days

status event marker; 1 if patient is deceased and 0 if alive or censored

age age in years at point of entry into the registry

sex string with values 'M' and 'F'

entrydate date of entry into the registry in YYYY-MM-DD format

eventdate date of death or censorship in YYYY-MM-DD format

Details

Demographic and disease-specific data required for prevalence estimations are included, such as sex, age, and dates of entry and event. eventdate marks the date of the last known follow-up with the patient, corresponding to death (status = 1) or censorship (status = 0).

---

rprev   *rprev: Estimate disease point prevalence using a combination of registry data and Monte Carlo simulations.*

Description

The rprev package uses available registry data to estimate point prevalence at a specified index date. This is done by fitting two models to the registry data: an incidence and a survival model. The first model is used to generate an incident population with the survival model determining whether an individual is alive at the index date and therefore contributing to prevalence.

Details

Prevalence is estimated using incident cases from a set number of years, where the larger this values the more accurate the prevalence estimates are. However, if the user asks to use more years of incident cases than are available in the registry data set, then the remaining years of incidence are simulated.

The primary function in this package is thereby [prevalence](#), which performs the combination of counted incidence from the registry data, and the simulated cases, along with the calculation of their survival probabilities at the index date.

[test_homogeneity](#) provides a summary of the incident cases in the registry data set, allowing for inspection of whether the default homogeneous Poisson process assumption holds for the disease in question.

---

sim_prevalence   *Estimate prevalence using Monte Carlo simulation.*

Description

Estimates prevalent cases at a specific index date by use of Monte Carlo simulation. Simulated cases are marked with age and sex to enable agreement with population survival data where a cure model is used, and calculation of the posterior distributions of each.

Usage

```
sim_prevalence(
  data,
  index,
  starting_date,
  inc_model,
  surv_model,
  age_column = "age",
  N_boot = 1000,
  age_dead = 100
)
```

Arguments

data	A data frame with the corresponding column names provided in form.
index	The date at which to estimate point prevalence as a string in the format YYYY-MM-DD.
starting_date	The initial date to start simulating prevalence from as a Date object. Typically the index date - (Nyears * 365.25). Allows for non-whole year prevalence estimations.
inc_model	An object that has a draw_incident_population method. See the vignette for further guidance.
surv_model	An object that has a predict_survival_probability method. See the vignette for further guidance.
age_column	A string providing the name of the column that holds patient age. If provided then patients alive at age_dead are set to die. This helps combat 'immortal' patients.
N_boot	Number of bootstrapped calculations to perform.
age_dead	The age at which patients are set to be dead if they are still alive, to prevent 'immortal' patients. Used in conjunction with age_column.

Value

A list with the following attributes:

results	A data.table containing the simulated incident populations from each simulation along with their covariates and survival status at the index.
full_surv_model	The survival model built on the full registry data set.
full_inc_model	The incidence model built on the full registry data set.
surv_models	A list containing survival models built on each bootstrap sample.
inc_models	A list containing incidence models built on each bootstrap sample.

---

summary.survfit.prev   *Obtain N-year survival probability estimates.*

Description

Summarises survival information at pre-specified years of interest on a `survfit.prev` object.

Usage

```
## S3 method for class 'survfit.prev'
summary(object, years = c(1, 3, 5), ...)
```

Arguments

`object`	A `survfit.prev` object.
`years`	A vector of years for which to estimate survival probability from the boot-strapped survival curves.
`...`	Arguments passed to main `summary` function.

Details

Survival probability is estimated as the mean of the bootstrapped survival curves at a specific time-point, with 2.5 confidence intervals. Survival probability can only be estimated at time points less than the maximum survival time in the original dataset that the `prevalence` object was fitted to.

Value

None, displays the survival probabilities to screen as a side-effect.

Examples

```
data(prevsim)

## Not run:
prev_obj <- prevalence(Surv(time, status) ~ age(age) + sex(sex) +
                       entry(entrydate) + event(eventdate),
                       data=prevsim, num_years_to_estimate = c(5, 10),
                       population_size=1e6, start = "2005-09-01",
                       num_reg_years = 8, cure = 5)

survobj <- survfit(prev_obj, newdata=list(age=65, sex=0))

summary(survobj)

summary(survobj, years=c(1, 3, 5, 7))

## End(Not run)
```

---

survfit.prevalence   *Form bootstrapped survival curves.*

Description

Calculates bootstrapped survival probabilities from the Weibull models fitted to the prevalence object.

Usage

```
## S3 method for class 'prevalence'
survfit(formula, newdata = NULL, ...)
```

Arguments

formula   A prevalence object.

newdata   A list or dataframe with the covariate values to calculate survival probabilities for. Defaults to using the mean values from the the original dataset that the model was fit to.

...   Other arguments to survfit.

Value

An S3 object of class survfit.prev with the following attributes:

time   A vector of time points at which survival probability has been calculated.

surv   A matrix of survival probabilities, where the rows represent a different bootstrapped Weibull model, and the columns are each timepoint.

fullsurv   A vector of survival probabilities for the predictors provided in newdata.

---

test_homogeneity   *Inspects disease incidence for its compatibility with a homogeneous Poisson process.*

Description

Calculates incidence by year of the registry data, along with mean incidence with confidence intervals. A smoothed cumulative incidence function is fit to the data for inspecting deviations in the registry data from a homogeneous Poisson process.

Usage

```
test_homogeneity(
  entry,
  year_start = "01-01",
  truncate_start = FALSE,
  truncate_end = FALSE,
  population_size = NULL,
  df = 4,
  proportion = 1e+05,
  level = 0.95,
  precision = 2
)
```

Arguments

entry	Vector of diagnosis dates for each patient in the registry in the format YYYY-MM-DD.
year_start	Date which to use to delimit years in the format MM-DD. See details for how this is used.
truncate_start	See details.
truncate_end	See details.
population_size	
	The population of the area covered by the registry. If not provided then only absolute incidence can be calculated.
df	The desired degrees of freedom of the smooth.
proportion	The denominator of the incidence rate.
level	The desired confidence interval width.
precision	The number of decimal places required.

Details

Annual incidence rates are calculated for every year that is present in `entry`, with years being delimited by the date specified in `year_start` that include every incident case. For example, under the default values, if the earliest incident date in `entry` is 1981-04-28, and the latest is 2016-12-16, then annual incidence rates will be calculated with the boundaries [1981-01-01, 1982-01-01), ..., [2016-01-01, 2017-01-01).

If `year_start` was specified as '09-01' then the boundaries would be [1980-09-01, 1981-09-01), ..., [2016-09-01, 2017-09-01).

The `truncate_start` and `truncate_end` arguments remove incident cases in the first and last years before and after the yearly boundaries respectively.

So if they were both `TRUE`, with `year_start` as '09-01' as before, then the boundaries would be [1981-09-01, 1982-09-01), ..., [2015-09-01, 2016-09-01), i.e. the incident cases in [1981-04-28, 1981-09-01) are discarded by `truncate_start` and those in [2016-09-01, 2016-12-16] removed by `truncate_end`.

This helps to ensure that annual incidence is measured on a time-scale appropriate for your registry.

Value

An S3 object of class `incidence` with the following attributes:

`yearly_incidence`
> Vector of absolute incidence values for each included year of the registry

`ordered_diagnoses`
> Vector of times (days) between diagnosis date and the earliest date of inclusion in the registry, ordered shortest to longest.

`smooth`   Smooth fitted to the cumulative incidence data.

`index_dates`   Dates delimiting the years in which incidence is calculated.

`mean`   List containing absolute yearly incidence as well as relative rates.

`pvals`   p-values resulting to a test of over and under dispersion on the incidence data respectively. Used to test the suitability of the homogeneous Poission process assumption.

`dof`   Degrees of freedom of the smooth.

Examples

```
data(prevsim)

## Not run:
test_homogeneity(prevsim$entrydate)

## End(Not run)
```

---

test_prevalence_fit   *Test simulated prevalence fit.*

Description

Calculates a Chi squared test between predicted yearly contributions to prevalence, and the observed values obtained from the registry, indicating whether the simulated prevalence values are accurate.

Usage

```
test_prevalence_fit(object)
```

Arguments

`object`   A `prevalence` object.

Value

P-value from a chi-squared test of difference between prevalence prediction and counted prevalence at the index date.

See Also

Other prevalence functions: [prevalence](#)()

Examples

```
data(prevsim)

## Not run:

obj <- prevalence(Surv(time, status) ~ age(age) + sex(sex) + entry(entrydate) + event(eventdate),
                   data=prevsim, num_years_to_estimate = c(5, 10), population_size=1e6,
                   start = "2005-09-01",
                   num_reg_years = 8, cure = 5)

test_prevalence_fit(obj)

## End(Not run)
```

UKmortality   *General population survival data.*

Description

A dataset containing daily population survival rates for individuals up to 100 years old, from the UK population, derived from the 2009 mortality rates found at: https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/lifeexpectancies/datasets/nationallifetablesunitedkingdomreferencetables, Adapted from public sector information licensed under the Open Government Licence v3.0. Data were relabelled according to the mean year of the three-year birth window. It is stored as a data.table for efficient access.

Usage

```
UKmortality
```

Format

A data frame with 109575 rows and 3 columns:

age age in days

sex string, either 'M' or 'F'

surv survival probability , estimated as the cumulative product of (1 - mortality rate)

---

validate_incidence_model

Tests custom incidence models

Description

Runs checks to assess whether a custom incidence model is suitable for use in `prevalence`. Provides useful diagnostic messages if any issues are encountered.

Usage

```
validate_incidence_model(object, data, timeframe = 3652)
```

Arguments

object	An incidence model to be tested
data	Registry data in the form of a data frame. Ideally will be the same source that will be used for the prevalence estimation later on.
timeframe	How long to generate incident cases for in days. This is disease-specific, but the default of ten years should work well for most diseases.

Value

The dummy incident population that has been generated to allow for further diagnostics to be run.

---

validate_survival_model

*Tests that a custom survival object has the required attributes for use
in the* prevalence *function.*

Description

Runs checks to assess whether a custom survival model is suitable for use in `prevalence`. Provides useful diagnostic messages if any issues are encountered.

Usage

```
validate_survival_model(object, data, timeframe = 3652, sample_size = 10)
```

Arguments

`object`	The custom survival object.
`data`	Registry data in the form of a data frame. Ideally will be the same source that will be used for the prevalence estimation later on.
`timeframe`	Maximum time at which to test survival probability in days. If not supplied then chooses random values over a period of 10 years, which should be suitable for many diseases.
`sample_size`	The number of randomly drawn individuals to predict sample size for.

Value

None. Instead, messages get displayed to the console.

Index