Package ‘impactflu’

January 14, 2020

Title Quantification of Population-Level Impact of Vaccination

Version 0.1.0

Description Implements the compartment model from Tokars (2018) <doi:10.1016/j.vaccine.2018.10.026>. This enables quantification of population-wide impact of vaccination against vaccine-preventable diseases such as influenza.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

RoxygenNote 7.0.2

LinkingTo Rcpp

Imports Rcpp, tibble, dplyr, rlang, glue, lubridate, magrittr

Suggests testthat (>= 2.1.0)

NeedsCompilation yes

Author Arseniy Khvorov [aut, cre]

Maintainer Arseniy Khvorov <khvorov45@gmail.com>

Repository CRAN

Date/Publication 2020-01-14 11:00:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>generate_counts</td>
<td>2</td>
</tr>
<tr>
<td>generate_dates</td>
<td>2</td>
</tr>
<tr>
<td>method1</td>
<td>3</td>
</tr>
<tr>
<td>sim_reference</td>
<td>5</td>
</tr>
</tbody>
</table>

Index 7
**generate_counts**

*Generate normal counts*

**Description**

Generates counts from a normal distribution density function.

**Usage**

```r
generate_counts(init_pop_size, n_timepoints, overall_prop, mean, sd)
```

**Arguments**

- `init_pop_size`: Initial population size
- `n_timepoints`: Number of timepoints
- `overall_prop`: Overall proportion of the population to be included in the counts over all the timepoints
- `mean`: Mean of the normal distribution
- `sd`: Standard deviation of the normal distribution

**Value**

An integer vector of counts of length `n_timepoints`

**Examples**

```r
# Tokars (2018) vaccinations
vacs_tok <- generate_counts(1e6, 304, 0.55, 100, 50)
# Tokars (2018) cases
casen_tok <- generate_counts(1e6, 304, 0.12, 190, 35)
```

**generate_dates**

*Generate dates*

**Description**

Generate dates given timepoint indices, start date and step unit

**Usage**

```r
generate_dates(timepoints, start, unit)
```
method1

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timepoints</td>
<td>Integer vector timepoint indices</td>
</tr>
<tr>
<td>start</td>
<td>Date of index 1</td>
</tr>
<tr>
<td>unit</td>
<td>&quot;year&quot; &quot;month&quot; or &quot;day&quot;</td>
</tr>
</tbody>
</table>

Value

A vector of dates the same length as timepoints

Examples

```r
# Dates from Tokars (2018)
timepoints <- 1L:304L
dates <- generate_dates(timepoints, lubridate::ymd("2017-08-01"), "day")
```

---

Description

Method 1 was said to be as current. Method 3 was determined to be the least biased.

Usage

```r
method1(init_pop_size, vaccinations, cases, ve)
method3(init_pop_size, vaccinations, cases, ve)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>init_pop_size</td>
<td>Integer initial population size</td>
</tr>
<tr>
<td>vaccinations</td>
<td>Integer vector counts of vaccinations</td>
</tr>
<tr>
<td>cases</td>
<td>Integer vector counts of cases</td>
</tr>
<tr>
<td>ve</td>
<td>Vector vaccine effectiveness. If length 1, assumed to not vary with time.</td>
</tr>
</tbody>
</table>

Value

A tibble with the following columns (method-dependent):

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cases</td>
<td>Observed cases</td>
</tr>
<tr>
<td>vaccinations</td>
<td>Observed vaccinations</td>
</tr>
<tr>
<td>ve</td>
<td>Assumed vaccine effectiveness</td>
</tr>
<tr>
<td>pvac</td>
<td>Proportion of the starting population vaccinated</td>
</tr>
<tr>
<td>vc_lag</td>
<td>Vaccine coverage lagged</td>
</tr>
<tr>
<td>pops</td>
<td>Susceptible population</td>
</tr>
</tbody>
</table>
Infection risk
Non-cases is absence of vaccination
Cases in absence of vaccination
Expected number of vaccinations

References

Examples

library(dplyr)

# Simulate a population
nsam <- 1e6L
ndays <- 304L
pop_tok <- sim_reference(
  init_pop_size = nsam,
  vaccinations = generate_counts(nsam, ndays, 0.55, mean = 100, sd = 50),
  cases_novac = generate_counts(nsam, ndays, 0.12, mean = 190, sd = 35),
  ve = 0.48,
  lag = 14,
  deterministic = TRUE
)

# Summarise by month
pop_tok_month <- pop_tok %>%
  mutate(
    datestamp = generate_dates(
      timepoint, lubridate::ymd("2017-08-01"), "day" ),
    year = lubridate::year(datestamp),
    month = lubridate::month(datestamp)
  ) %>%
  group_by(year, month) %>%
  summarise(
    vaccinations = sum(vaccinations), cases = sum(cases), ve = mean(ve)
  ) %>%
  ungroup()

# Estimate averted cases using the two different methods
m1 <- method1(
  nsam, pop_tok_month$vaccinations, pop_tok_month$cases, pop_tok_month$ve
)

m3 <- method3(
  nsam, pop_tok_month$vaccinations, pop_tok_month$cases, pop_tok_month$ve
)

sum(m1$avert)
sum(m3$avert)
**sim_reference**

Simulate an ideal population

**Description**

Simulates an ideal population using the reference model from Tokars (2018).

**Usage**

```r
sim_reference(
    init_pop_size, vaccinations, cases_novac, ve, lag, deterministic,
    seed = sample.int(.Machine$integer.max, 1)
)
```

**Arguments**

- `init_pop_size`: Integer initial population size
- `vaccinations`: Integer vector number of vaccinations at every timepoint
- `cases_novac`: Integer vector number of cases at every timepoint
- `ve`: Vaccine effectiveness (proportion)
- `lag`: Integer lag period measured in timepoints
- `deterministic`: Boolean whether to make the simulation deterministic
- `seed`: Integer seed to use

**Value**

A tibble with the following columns:

- `timepoint`: Index of timepoint
- `vaccinations`: Expected number of vaccinations
- `cases_novac`: Expected number of cases in absence of vaccination
- `ve`: Expected vaccine effectiveness
- `pflu`: Flu incidence
- `cases`: Actual number of cases
- `popn`: Non-cases in absence of vaccination
- `pvac`: Proportion of starting population vaccinated
- `b`: Number vaccinated at that time
- `A`: Non-vaccinated non-cases
- `B`: Vaccinated non-cases lagging
- `E`: Non-vaccinated cases
References


Examples

# Population from Tokars (2018)
nsam <- 1e6L
ndays <- 304L
pop_tok <- sim_reference(
  init_pop_size = nsam,
  vaccinations = generate_counts(nsam, ndays, 0.55, mean = 100, sd = 50),
  cases_novac = generate_counts(nsam, ndays, 0.12, mean = 190, sd = 35),
  ve = 0.48,
  lag = 14,
  deterministic = TRUE
)
head(pop_tok)
sum(pop_tok$avert)
Index

generate_counts, 2
generate_dates, 2

method1, 3
method3 (method1), 3

sim_reference, 5

tibble, 3.5