generate_counts  Generate normal counts

Description
Generates counts from a normal distribution density function.

Usage

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

# 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

generate_dates(timepoints, start, unit)
**Arguments**

- **timepoints**: Integer vector timepoint indices
- **start**: Date of index 1
- **unit**: "year", "month", or "day"

**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**

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

**Arguments**

- **init_pop_size**: Integer initial population size
- **vaccinations**: Integer vector counts of vaccinations
- **cases**: Integer vector counts of cases
- **ve**: Vector vaccine effectiveness. If length 1, assumed to not vary with time.

**Value**

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

- **cases**: Observed cases
- **vaccinations**: Observed vaccinations
- **ve**: Assumed vaccine effectiveness
- **pvac**: Proportion of the starting population vaccinated
- **vc_lag**: Vaccine coverage lagged
- **pops**: Susceptible population
method1

<table>
<thead>
<tr>
<th>variable</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pflu</td>
<td>Infection risk</td>
</tr>
<tr>
<td>popn</td>
<td>Non-cases is absence of vaccination</td>
</tr>
<tr>
<td>cases_novac</td>
<td>Cases in absence of vaccination</td>
</tr>
<tr>
<td>avert</td>
<td>Expected number of vaccinations</td>
</tr>
</tbody>
</table>

**References**


**Examples**

```r
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)
```
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