Package 'freedom'

July 22, 2025

Title Demonstration of Disease Freedom (DDF)

Version 1.0.1

Description Implements the formulae required to calculate freedom from disease according to Cameron and Baldock (1998) <doi:10.1016/S0167-5877(97)00081-0>. These are the methods used at the Swedish national veterinary institute (SVA) to evaluate the performance of our nation animal disease surveillance programmes.

License GPL-3

URL https://github.com/SVA-SE/freedom

BugReports https://github.com/SVA-SE/freedom/issues

Type Package

LazyLoad yes

VignetteBuilder knitr

Suggests knitr, rmarkdown

Depends R (>= 3.6)

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Thomas Rosendal [aut, cre] (ORCID: <https://orcid.org/0000-0002-6576-9668>)

Maintainer Thomas Rosendal <trosendal@gmail.com>

Repository CRAN

Date/Publication 2020-09-08 12:40:08 UTC

Contents

adjusted_risk 2
EffProbInf 3

hse 4
hse_finite 5
hse_infinite 6
post_fr 7
prior_fr 8
rpert 8
sample_data 9
sysse 10
sysse_finite 11
valid_proportions 12

Index **14**

`adjusted_risk` *adjusted_risk*

Description

Adjusted Risk

Usage

```
adjusted_risk(prop, RR)
```

Arguments

`prop`	A vector of proportions of the population that belong to each URG (Unit risk group)
`RR`	A vector of the relative risks of for each URG. The first of these is the referent group and therefore must be equal to 1

Details

Calculate the adjusted risk for each of the unit risk groups (URG). This can be used at both the herd and the animal level. The proportion vector, for herd level, is therefore the proportion herds in the population that are in each of the unit risk groups. The proportion vector for animal level is the proportion of animals within a given herd that are in each URG.

Value

A vector of Adjusted risks

Examples

```
df <- sample_data(nherds = 100,
                  mean_herd_size = 300,
                  n_herd_urg = 2,
                  herd_dist = c(0.9, 0.1),
                  herd_samp_frac = 0.01,
                  herd_samp_dist = c(0.3, 0.7),
                  n_animal_urg = 1,
                  animal_dist = c(1),
                  animal_samp_frac = 0.05,
                  animal_samp_dist = c(1),
                  seed = 1)
## The proportion of herds in each unit risk group
table(df$herd_urg)/nrow(df)
## Calculate the Adjusted risk for each unit risk group based on the
## proportion in each group and the estimated relative risk of being
## in that group:
AR <- freedom::adjusted_risk(as.numeric(table(df$herd_urg)/nrow(df)),
                             c(1, 2.3))
```

`EffProbInf` *EffProbInf*

Description

EffProbInf

Usage

```
EffProbInf(dp, AR)
```

Arguments

`dp`	A vector The design prevalence
`AR`	A vector of the adjusted risks of the unit risk groups

Details

Calculate the effective probability of infection (EPI) for each unit risk group in the population. This could be either at the herd level or within herd level. The dp for herds is therefore the minimum prevalence among herds that you would like to design the surveillance system to be able to detect. The dp for within herds is therefore the minimum prevalence of the disease within a herd among the animals that you would like to design the surveillance system to detect.

Value

A vector of EPI

Examples

```
df <- sample_data(nherds = 100,
                  mean_herd_size = 300,
                  n_herd_urg = 2,
                  herd_dist = c(0.9, 0.1),
                  herd_samp_frac = 0.01,
                  herd_samp_dist = c(0.3, 0.7),
                  n_animal_urg = 1,
                  animal_dist = c(1),
                  animal_samp_frac = 0.05,
                  animal_samp_dist = c(1),
                  seed = 1)
## The proportion of herds in each unit risk group
table(df$herd_urg)/nrow(df)
## Calculate the Adjusted risk for each unit risk group based on the
## proportion in each group and the estimated relative risk of being
## in that group:
AR <- freedom::adjusted_risk(as.numeric(table(df$herd_urg)/nrow(df)),
                             c(1, 2.3))
EPHI <- EffProbInf(0.05, AR)
```

---

hse *hse*

Description

Herd Sensitivity

Usage

```
hse(id, n_tested, N, test_Se, dp, threshold = 0.1, force = FALSE)
```

Arguments

id	The herdid
n_tested	The number tested in each URG
N	The number of units in each of the URG
test_Se	The sensitivity of the test (length = 1). If you have reason to believe that the test sensitivity is different for different URG. Then supply a vector of Sensitivities. This could conceivably be because of using different tests for different samples from different URG.
dp	The is a vector (length 1) of the design prevalence (df) in the case where there is only one unit risk group (URG) in the herd. Or a vector (length n) of EPIn for each of the URG in the herd.
threshold	The breakpoint above which the finite population size calculation will be used. The default is 0.1 which means that if > 10 population will be assumed; less than or equal to 10 infinite population will be assumed.
force	If force = FALSE (default) then the function errors if n>N. If force = TRUE then this is allowed and uses the hse_infinite to calculate HSe.

Details

Calculate the Herd sensitivity when multiple samples from individual units within the herd. The function uses the assumption of finite population when greater than 10 otherwise the assumption of infinite population.

Value

A vector (length 1)

Examples

```
df <- data.frame(id = seq(1:20),
                 n_tested = rpois(20, 6),
                 N = rpois(20, 50),
                 test_Se = 0.3,
                 dp = 0.05)
## Calculate the herd level sensitivity for each of these herds. If
## the ratio of the number tested to number of animals in the herd
## exceeds the threshold then the finite method is used, otherwise the
## infinite method is used.
hse(df$id,
    df$n_tested,
    df$N,
    df$test_Se,
    df$dp,
    threshold = 0.1)
```

`hse_finite` *hse_finite*

Description

Herd Sensitivity calculated with the assumption of a finite population

Usage

```
hse_finite(id, n_tested, N, test_Se, dp)
```

Arguments

`id`	The herdid.
`n_tested`	The number tested in each URG
`N`	The number of units in each of the URG
`test_Se`	The sensitivity of the test. This may have length == 1 if all URG and all herds have the same test_Se. It may also have length(test_Se) == length(n_tested).
`dp`	The design prevalence (dp) could be length(dp) == 1 if all URG and herds have the same dp. It could alternatively be length(dp) == length(n_tested) if different design prevalences are to be applied to each URG.

Details

Calculate the Herd sensitivity when multiple samples from individual units within the herd. The function uses the total population size to adjust the estimates consistent with a finite population. This method for calculation of HSe is typically used when greater than 10

Value

A data.frame. A dataframe is returned with 2 columns: "id" and HSe

Examples

```
df <- data.frame(id = seq(1:20),
                 n_tested = rpois(20, 5),
                 N = 100,
                 test_Se = 0.3,
                 dp = 0.05)
## Calculate the herd level sensitivity for each of these herds
hse_finite(df$id,
           df$n_tested,
           df$N,
           df$test_Se,
           df$dp)
```

`hse_infinite` *hse_infinite*

Description

Herd Sensitivity calculated with the assumption of an infinite population

Usage

```
hse_infinite(id, n_tested, test_Se, dp)
```

Arguments

`id`	The herdid
`n_tested`	The number tested in each URG
`test_Se`	The sensitivity of the test. This may have length == 1 if all URG and all herds have the same test_Se. It may also have length(test_Se) == length(n_tested).
`dp`	The design prevalence (dp) could be length(dp) == 1 if all URG and herds have the same dp. It could alternatively be length(dp) == length(n_tested) if diff

Details

Calculate the Herd sensitivity when multiple samples from individual units within the herd. The function does not use the population size to adjust the estimate. This is consistent with the assumption of an infinite population size and is generally used when less than 10

Value

A data.frame. A dataframe is returned with 2 columns: "id" and HSe

Examples

```
df <- data.frame(id = seq(1:20),
                 n_tested = rpois(20, 5),
                 test_Se = 0.3,
                 dp = 0.05)

## Calculate the herd level sensitivity for each of these herds given
## the assumption that the herds have an infinite size.
hse_infinite(df$id,
             df$n_tested,
             df$test_Se,
             df$dp)
```

`post_fr` *post_fr*

Description

Calculate the posterior probability of freedom from the prior and the sensitivity of the system

Usage

```
post_fr(prior_fr, Se)
```

Arguments

`prior_fr`	The prior probability of freedom
`Se`	The sensitivity of the surveillance system

Details

The prior probability of freedom at the beginning of the surveillance initiative is a value that is based on some external evidence. Often 0.5 is used as a conservative estimate of the probability that the population is free from the disease. For subsequent time intervals in the surveillance system, the prior year's posterior probability of freedom is used (plus a risk of introduction) as the prior probability in this calculation.

Value

A vector

Examples

```
## Calculate the posterior probability of freedom after applying a
#sensitivity to a prior probability of freedom:
post_pf <- post_fr(0.5, 0.4)
```

prior_fr *prior_fr*

Description

Calculate the prior probability of freedom (year = k)

Usage

```
prior_fr(post_fr, intro)
```

Arguments

post_fr	The posterior probability of freedom (year = k-1)
intro	The annual probability of introduction

Details

In order to calculate the posterior probability of freedom (year = k) , the prior probability of freedom (year = k) is first calculated from the posterior probability of freedom (year = k-1) from the previous year and the annual probability that the disease is introduced into the population.

Value

A vector. The prior probability of freedom (year = k)

Examples

```
## Calculate the posterior probability of freedom after applying a
## sensitivity to a prior probability of freedom:
post_pf <- post_fr(0.5, 0.4)
## Then discount the probability of introduction (0.05) from the
## posterior probability of freedom to calculate the subsequent
## prior probability of freedom for the next time step:
prior_pf <- prior_fr(post_pf, 0.05)
```

rpert *rpert*

Description

Sample a pert distribution

Usage

```
rpert(n, x.min, x.max, x.mode, lambda = 4)
```

Arguments

n	number of samples
x.min	The minimum value in the sample
x.max	The maximum value in the sample
x.mode	The mode of the sample
lambda	lambda

Details

Returns samples from a pert distribution

Value

a numeric vector of length n

Examples

```
## Generate 10000 samples from a pert distribution with a minimum
## of 2, a max of 5, and a mode of 4.
samples <- rpert(10000, 2, 5, 4)
hist(samples)

## Generate a
```

---

sample_data *sample_data*

Description

A function to generate some synthetic data based on a few parameters.

Usage

```
sample_data(
  nherds = 500,
  mean_herd_size = 50,
  n_herd_urg = 2,
  herd_dist = c(0.8, 0.2),
  herd_samp_frac = 0.5,
  herd_samp_dist = c(0.5, 0.5),
  n_animal_urg = 2,
  animal_dist = c(0.5, 0.5),
  animal_samp_frac = 0.15,
  animal_samp_dist = c(0.5, 0.5),
  seed = NULL
)
```

Arguments

nherds	The total number of herds
mean_herd_size	The mean herd size in the population
n_herd_urg	The number of different herd risk groups
herd_dist	The fraction of herds in each risk group
herd_samp_frac	The total sampling fraction at the herd level
herd_samp_dist	The fraction of samples to be collected from each herd risk group
n_animal_urg	The number of animal level risk groups
animal_dist	The fraction of animals within herds that are part of each risk group
animal_samp_frac	The total sampling fraction of animals within herds
animal_samp_dist	The fraction of samples that are collected from each animal risk group
seed	The seed for the random number generator. Default is a random seed

Value

A data.frame

Examples

```
## Generate the default example data. This will generate a
## data.frame with a herd identifier (ppn), a herd level unit risk
## group identifier (herd_urg), a animal level unit risk group
## identifier (animal_urg), the total number of animals in the unit
## risk group (N_animal_urg) and the number of animals tested in the
## unit risk group (n_animals_urg).

df <- sample_data()
```

---

sysse *sysse*

Description

Calculate the surveillance system sensitivity

Usage

```
sysse(dp, hse)
```

Arguments

dp	The vector of EPIH for all herds tested in the surveillance system
hse	The calculated hse for all the herds tested in the surveillance system

Details

Takes a vector of the sensitivity of herds tested in the surveillance system and a vector of the effective probability of infection in the herds (EPIH) to calculate the total surveillance system sensitivity for the entire program.

Value

A vector (length 1)

Examples

```
df <- data.frame(id = seq(1:20),
                 n_tested = rpois(20, 6),
                 N = rpois(20, 50),
                 test_Se = 0.3,
                 dp = 0.05)
## Calculate the herd level sensitivity for each of these herds. If
## the ratio of the number tested to number of animals in the herd
## exceeds the threshold then the finite method is used, otherwise the
## infinite method is used.
herd_Se <- hse(df$id,
               df$n_tested,
               df$N,
               df$test_Se,
               df$dp,
               threshold = 0.1)
## Calculate the system sensitivity given the testing and sensitivity
## in these herds:
sysse(dp = rep(0.10, nrow(herd_Se)),
      hse = herd_Se$HSe)
```

---

sysse_finite *sysse*

Description

Calculate the surveillance system sensitivity for a finite population of herds

Usage

```
sysse_finite(dp, hse, N)
```

Arguments

dp	The vector of EPIH for all herds tested in the surveillance system.
hse	The calculated hse for all the herds tested in the surveillance system.
N	The total number of herds in the population.

Details

Takes a vector of the sensitivity of herds tested in the surveillance system and a vector of the effective probability of infection in the herds (EPIH) to calculate the total surveillance system sensitivity for the entire program. This is adjusted for the total number of herds in the population.

Value

A vector (length 1)

Examples

```
df <- data.frame(id = seq(1:20),
                 n_tested = rpois(20, 6),
                 N = rpois(20, 50),
                 test_Se = 0.3,
                 dp = 0.05)
 ## Calculate the herd level sensitivity for each of these herds. If
 ## the ratio of the number tested to number of animals in the herd
 ## exceeds the threshold then the finite method is used, otherwise the
 ## infinite method is used.
 herd_Se <- hse(df$id,
                df$n_tested,
                df$N,
                df$test_Se,
                df$dp,
                threshold = 0.1)
 ## Calculate the system sensitivity given the testing and sensitivity
 ## in these herds adjusted for the total number of herds in the population:
 sysse_finite(dp = rep(0.10, nrow(herd_Se)),
              hse = herd_Se$HSe,
              N = 100)
```

`valid_proportions` *valid_proportions*

Description

A function used to check if a vector of proportions is valid

Usage

```
valid_proportions(x, tolerance = 1e-07)
```

Arguments

`x`	numeric
`tolerance`	a tolerance value

Value

logical

Index

adjusted_risk, 2

EffProbInf, 3

hse, 4
hse_finite, 5
hse_infinite, 6

post_fr, 7
prior_fr, 8

rpert, 8

sample_data, 9
sysse, 10
sysse_finite, 11

valid_proportions, 12