Package 'serosv'

April 9, 2025

Type Package

Title Model Infectious Disease Parameters from Serosurveys

Version 1.1.0

Description An easy-to-use and efficient tool to estimate infectious diseases parameters using serological data. Implemented models include SIR models (basic_sir_model(), static_sir_model(), mseir_model(), sir_subpops_model()), parametric models (polynomial_model(), fp_model()), nonparametric models (lp_model()), semiparametric models (penalized_splines_model()), hierarchical models (hierarchical_bayesian_model()). The package is based on the book ``Modeling Infectious Disease Parameters Based on Serological and Social Contact Data: A Modern Statistical Perspective" (Hens, Niel & Shkedy, Ziv & Aerts, Marc & Faes, Christel & Damme, Pierre & Beutels, Philippe., 2013) <doi:10.1007/978-1-4614-4072-7>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

RoxygenNote 7.3.1

Imports deSolve, dplyr, ggplot2, locfit, purrr, stringr, magrittr, methods, mgcv, mixdist, patchwork, assertthat, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0), boot, stats4

Suggests covr, knitr, rmarkdown, bookdown, testthat (>= 3.0.0)

Collate 'data.R' 'mseir_model.R' 'sir_basic_model.R' 'sir_static_model.R' 'sir_subpops_model.R' 'fractional_polynomial_models.R' 'polynomial_models.R' 'utils.R' 'compare_models.R' 'correct_prevalence.R' 'weibull_model.R' 'nonparametric.R' 'semiparametric_models.R' 'mixture_model.R' 'hierarchical_bayesian_model.R' 'serosv.R' 'stanmodels.R' 'plots.R' 'compute_ci.R'

Config/testthat/edition 3

URL https://oucru-modelling.github.io/serosv/, https://github.com/OUCRU-Modelling/serosv

VignetteBuilder knitr

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

BugReports https://github.com/OUCRU-Modelling/serosv/issues

NeedsCompilation yes

Author Anh Phan Truong Quynh [aut, cre],
Nguyen Pham Nguyen The [aut],
Long Bui Thanh [aut],
Tuyen Huynh [aut],
Thinh Ong [aut] (<https://orcid.org/0000-0001-6772-9291>),
Marc Choisy [aut] (<https://orcid.org/0000-0002-5187-6390>)

Maintainer Anh Phan Truong Quynh <anhptq@oucru.org>

Repository CRAN

Date/Publication 2025-04-09 04:30:02 UTC

Contents

`serosv-package` *serosv: model infectious disease parameters*

Description

An easy-to-use and efficient tool to estimate infectious diseases parameters using serological data. Implemented models include SIR models (basic_sir_model(), static_sir_model(), mseir_model(), sir_subpops_model()), parametric models (polynomial_model(), fp_model()), nonparametric models (lp_model()), semiparametric models (penalized_splines_model()), hierarchical models (hierarchical_bayesian_model()). The package is based on the book "Modeling Infectious Disease Parameters Based on Serological and Social Contact Data: A Modern Statistical Perspective" (Hens,

Niel & Shkedy, Ziv & Aerts, Marc & Faes, Christel & Damme, Pierre & Beutels, Philippe., 2013) doi:10.1007/9781461440727.

Author(s)

Maintainer: Anh Phan Truong Quynh <anhptq@oucru.org>

Authors:

- Nguyen Pham Nguyen The <nguyenpnt@oucru.org>
- Long Bui Thanh
- Tuyen Huynh <tuyenhn@oucru.org>
- Thinh Ong <thinhop@oucru.org> (ORCID)
- Marc Choisy <mchoisy@oucru.org> (ORCID)

See Also

Useful links:

- https://oucru-modelling.github.io/serosv/
- https://github.com/OUCRU-Modelling/serosv
- Report bugs at https://github.com/OUCRU-Modelling/serosv/issues

---

`compare_models` *Compare models*

Description

Compare models

Usage

```
compare_models(...)
```

Arguments

`...`	models to be compared. Must be models created by serosv. If models' names are not provided, indices will be used instead for the 'model' column in the returned data.frame.

Value

a data.frame of 4 columns

`model`	name or index of the model
`type`	model type of the given model (a serosv model name)
`AIC`	AIC value for the model (lower value indicates better fit)
`BIC`	BIC value for the model (lower value indicates better fit)

compute_ci *Compute confidence interval*

Description

Compute confidence interval

Usage

```
compute_ci(x, ci = 0.95, le = 100, ...)
```

Arguments

x	- serosv models
ci	- confidence interval
le	- number of data for computing confidence interval
...	- arbitrary argument

Value

confidence interval dataframe with 4 variables, x and y for the fitted values and ymin and ymax for the confidence interval

compute_ci.fp_model *Compute confidence interval for fractional polynomial model*

Description

Compute confidence interval for fractional polynomial model

Usage

```
compute_ci.fp_model(x, ci = 0.95, le = 100, ...)
```

Arguments

x	- serosv models
ci	- confidence interval
le	- number of data for computing confidence interval
...	- arbitrary argument

Value

confidence interval dataframe with 4 variables, x and y for the fitted values and ymin and ymax for the confidence interval

compute_ci.lp_model *Compute confidence interval for local polynomial model*

Description

Compute confidence interval for local polynomial model

Usage

```
compute_ci.lp_model(x, ci = 0.95, ...)
```

Arguments

x	- serosv models
ci	- confidence interval
...	- arbitrary arguments

Value

confidence interval dataframe with 4 variables, x and y for the fitted values and ymin and ymax for the confidence interval

compute_ci.mixture_model *Compute confidence interval for mixture model*

Description

Compute confidence interval for mixture model

Usage

```
compute_ci.mixture_model(x, ci = 0.95, ...)
```

Arguments

x	- serosv mixture_model object
ci	- confidence interval
...	- arbitrary arguments

Value

list of confidence interval for susceptible and infected. Each confidence interval is a list with 2 items for lower and upper bound of the interval.

`compute_ci.penalized_spline_model` *Compute confidence interval for penalized_spline_model*

Description

Compute confidence interval for penalized_spline_model

Usage

```
compute_ci.penalized_spline_model(x, ci = 0.95, ...)
```

Arguments

`x`	- serosv models
`ci`	- confidence interval
`...`	- arbitrary arguments

Value

list of confidence interval for seroprevalence and foi Each confidence interval dataframe with 4 variables, x and y for the fitted values and ymin and ymax for the confidence interval

`compute_ci.weibull_model` *Compute confidence interval for Weibull model*

Description

Compute confidence interval for Weibull model

Usage

```
compute_ci.weibull_model(x, ci = 0.95, ...)
```

Arguments

`x`	- serosv models
`ci`	- confidence interval
`...`	- arbitrary argument

Value

confidence interval dataframe with 4 variables, x and y for the fitted values and ymin and ymax for the confidence interval

`correct_prevalence` *Estimate the true sero prevalence using Bayesian estimation*

Description

Estimate the true sero prevalence using Bayesian estimation

Usage

```
correct_prevalence(
  data,
  bayesian = TRUE,
  init_se = 0.95,
  init_sp = 0.8,
  study_size_se = 1000,
  study_size_sp = 1000,
  chains = 1,
  warmup = 1000,
  iter = 2000
)
```

Arguments

`data`	the input data frame, must either have 'age', 'pos', 'tot' columns (for aggregated data) OR 'age', 'status' for (linelisting data)
`bayesian`	whether to adjust sero-prevalence using the Bayesian or frequentist approach. If set to 'TRUE', true sero-prevalence is estimated using MCMC.
`init_se`	sensitivity of the serological test
`init_sp`	specificity of the serological test
`study_size_se`	(applicable when 'bayesian=TRUE') study size for sensitivity validation study (i.e., number of confirmed infected patients in the study)
`study_size_sp`	(applicable when 'bayesian=TRUE') study size for specificity validation study (i.e., number of confirmed non-infected patients in the study)
`chains`	(applicable when 'bayesian=TRUE') number of Markov chains
`warmup`	(applicable when 'bayesian=TRUE') number of warm up runs
`iter`	(applicable when 'bayesian=TRUE') number of iterations

Value

a list of 2 items

`info`	estimated parameters
`corrected_sero`	data.frame containing age, the corresponding estimated seroprevalance, adjusted tot and pos

Examples

```
data <- rubella_uk_1986_1987
correct_prevalence(data)
```

`estimate_from_mixture` *Estimate seroprevalence and foi by combining mixture model and regression*

Description

Refers to section 11.2 - 11.4

Usage

```
estimate_from_mixture(
  age,
  antibody_level,
  threshold_status = NULL,
  mixture_model,
  s = "ps",
  sp = 83,
  monotonize = TRUE
)
```

Arguments

`age`	- vector of age
`antibody_level`	- vector of the corresponding raw antibody level
`threshold_status`	- sero status using threshold approach in line listing (optional, for visualization and comparison only)
`mixture_model`	- mixture_model object generated by serosv::mixture_model()
`s`	- smoothing basis used to fit antibody level
`sp`	- smoothing parameter
`monotonize`	- whether to monotonize seroprevalence (default to TRUE)

Value

a list of class estimated_from_mixture with the following items

`df`	the dataframe used for fitting the model
`info`	a fitted "gam" model for mu(a)
`sp`	seroprevalence
`foi`	force of infection
`threshold_status`	serostatus using threshold method only if provided

See Also

[mgcv::gam()] for more information about the fitted gam object

---

`est_foi` *Estimate force of infection*

Description

Estimate force of infection

Usage

```
est_foi(t, sp)
```

Arguments

`t`	- time (in this case age) vector
`sp`	- seroprevalence vector

Value

computed foi vector

`farrington_model` *The Farrington (1990) model.*

Description

Refers to section 6.1.2.

Usage

```
farrington_model(data, start, fixed = list())
```

Arguments

`data`	the input data frame, must either have 'age', 'pos', 'tot' columns (for aggregated data) OR 'age', 'status' for (linelisting data)
`start`	Named list of vectors or single vector. Initial values for optimizer.
`fixed`	Named list of vectors or single vector. Parameter values to keep fixed during optimization.

Value

a list of class farrington_model with 5 items

datatype	type of datatype used for model fitting (aggregated or linelisting)
df	the dataframe used for fitting the model
info	fitted "glm" object
sp	seroprevalence
foi	force of infection

See Also

[stats::glm()] for more information on the fitted glm object

Examples

```
df <- rubella_uk_1986_1987
model <- farrington_model(
  df,
  start=list(alpha=0.07,beta=0.1,gamma=0.03)
  )
plot(model)
```

---

find_best_fp_powers *Returns the powers of the GLM fitted model which has the lowest deviance score.*

Description

Refers to section 6.2.

Usage

```
find_best_fp_powers(data, p, mc, degree, link = "logit")
```

Arguments

data	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ columns (for aggregated data) OR ‘age‘, ‘status‘ for (linelisting data)
p	a powers sequence.
mc	indicates if the returned model should be monotonic.
degree	the degree of the model. Recommended to be <= 2.
link	the link function. Defaulted to "logit".

Value

list of 3 elements:

p	The best power for fp model.
deviance	Deviance of the best fitted model.
model	The best model fitted

Examples

```
df <- hav_be_1993_1994
best_p <- find_best_fp_powers(
df,
p=seq(-2,3,0.1), mc=FALSE, degree=2, link="cloglog"
)
best_p
```

---

fp_model *A fractional polynomial model.*

Description

Refers to section 6.2.

Usage

```
fp_model(data, p, link = "logit")
```

Arguments

data	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ columns (for aggregated data) OR ‘age‘, ‘status‘ for (linelisting data)
p	the powers of the predictor.
link	the link function for model. Defaulted to "logit".

Value

a list of class fp_model with 5 items

datatype	type of data used for fitting model (aggregated or linelisting)
df	the dataframe used for fitting the model
info	a fitted glm model
sp	seroprevalence
foi	force of infection

See Also

[stats::glm()] for more information on glm object

Examples

```
df <- hav_be_1993_1994
model <- fp_model(
  df,
  p=c(1.5, 1.6), link="cloglog")
plot(model)
```

---

hav_be_1993_1994 *Hepatitis A serological data from Belgium in 1993 and 1994 (aggregated)*

Description

A study of the prevalence of HAV antibodies conducted in the Flemish Community of Belgium in 1993 and early 1994

Usage

```
hav_be_1993_1994
```

Format

A data frame with 3 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

Source

Beutels, M., Van Damme, P., Aelvoet, W. et al. Prevalence of hepatitis A, B and C in the Flemish population. Eur J Epidemiol 13, 275-280 (1997). doi:10.1023/A:1007393405966

Examples

```
# Reproduce Fig 4.1 (upper left panel), p. 63
age <- hav_be_1993_1994$age
pos <- hav_be_1993_1994$pos
tot <- hav_be_1993_1994$tot
plot(
  age, pos / tot,
  pty = "s", cex = 0.06 * tot, pch = 16, xlab = "age",
  ylab = "seroprevalence", xlim = c(0, 86), ylim = c(0, 1)
)
```

hav_be_2002 *Hepatitis A serological data from Belgium in 2002 (line listing)*

Description

A subset of the serological dataset of Varicella-Zoster Virus (VZV) and Parvovirus B19 in Belgium where only individuals living in Flanders were selected

Usage

```
hav_be_2002
```

Format

A data frame with 2 variables:

age Age of individual

seropositive If the individual is seropositive or not

Source

Thiry, N., Beutels, P., Shkedy, Z. et al. The seroepidemiology of primary varicella-zoster virus infection in Flanders (Belgium). Eur J Pediatr 161, 588-593 (2002). doi:10.1007/s0043100210532

Examples

```
# Reproduce Fig 4.1 (upper right panel), p. 63
library(dplyr)
df <- hav_be_2002 %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(
  df$age, df$pos / df$tot,
  pty = "s", cex = 0.06 * df$tot, pch = 16, xlab = "age",
  ylab = "seroprevalence", xlim = c(0, 86), ylim = c(0, 1)
)
```

hav_bg_1964 *Hepatitis A serological data from Bulgaria in 1964 (aggregated)*

Description

A cross-sectional survey conducted in 1964 in Bulgaria. Samples were collected from schoolchildren and blood donors.

Usage

```
hav_bg_1964
```

Format

A data frame with 3 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

Source

Keiding, Niels. "Age-Specific Incidence and Prevalence: A Statistical Perspective." Journal of the Royal Statistical Society. Series A (Statistics in Society) 154, no. 3 (1991): 371-412. doi:10.2307/2983150

Examples

```
# Reproduce Fig 4.1 (lower panel), p. 63
age <- hav_bg_1964$age
pos <- hav_bg_1964$pos
tot <- hav_bg_1964$tot
plot(
    age, pos / tot,
    pty = "s", cex = 0.08 * tot, pch = 16, xlab = "age",
    ylab = "seroprevalence", xlim = c(0, 86), ylim = c(0, 1)
)
```

---

hbv_ru_1999 *Hepatitis B serological data from Russia in 1999 (aggregated)*

Description

A seroprevalence study conducted in St. Petersburg (more information in the book)

Usage

```
hbv_ru_1999
```

Format

A data frame with 4 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

gender Gender of cohort (unsure what 1 and 2 means)

Source

Mukomolov, S., L. Shliakhtenko, I. Levakova, and E. Shargorodskaya. Viral hepatitis in Russian federation. An analytical overview. Technical Report 213 (3), 3rd edn. St Petersburg Pasteur Institute, St Petersburg, 2000.

Examples

```
# Reproduce Fig 4.2, p. 65
library(dplyr)
hbv_ru_1999$age <- trunc(hbv_ru_1999$age / 1) * 1
hbv_ru_1999$age[hbv_ru_1999$age > 40] <- trunc(
  hbv_ru_1999$age[hbv_ru_1999$age > 40] / 5
) * 5
df <- hbv_ru_1999 %>%
  group_by(age) %>%
  summarise(pos = sum(pos), tot = sum(tot))
plot(
  df$age, df$pos / df$tot,
  cex = 0.05 * df$tot, pch = 16, xlab = "age",
  ylab = "seroprevalence", xlim = c(0, 72)
)
```

---

hcv_be_2006 *Hepatitis C serological data from Belgium in 2006 (line listing)*

Description

A study of HCV infection among injecting drug users. All injecting drug users were interviewed by means of a standardized face-to-face interview and information on their socio-demographic status, drug use history, drug use, and related risk behavior was recorded

Usage

```
hcv_be_2006
```

Format

A data frame with 3 variables:

dur Duration of injection/Exposure time (years)

seropositive If the individual is seropositive or not

Source

Mathei, C., Shkedy, Z., Denis, B., Kabali, C., Aerts, M., Molenberghs, G., Van Damme, P. and Buntinx, F. (2006), Evidence for a substantial role of sharing of injecting paraphernalia other than syringes/needles to the spread of hepatitis C among injecting drug users. Journal of Viral Hepatitis, 13: 560-570. doi:10.1111/j.13652893.2006.00725.x

Examples

```
# Reproduce Fig 4.3, p. 66
library(dplyr)
# snapping age to aggregated age group
# (credit: https://stackoverflow.com/a/12861810)
groups <- c(0.5:24.5)
range <- 0.5
low <- findInterval(hcv_be_2006$dur, groups)
high <- low + 1
low_diff <- hcv_be_2006$dur - groups[ifelse(low == 0, NA, low)]
high_diff <- groups[ifelse(high == 0, NA, high)] - hcv_be_2006$dur
mins <- pmin(low_diff, high_diff, na.rm = TRUE)
pick <- ifelse(!is.na(low_diff) & mins == low_diff, low, high)
hcv_be_2006$dur <- ifelse(
  mins <= range + .Machine$double.eps, groups[pick], hcv_be_2006$dur
)
hcv_be_2006 <- hcv_be_2006 %>%
  group_by(dur) %>%
  summarise(tot = n(), pos = sum(seropositive))

plot(
  hcv_be_2006$dur, hcv_be_2006$pos / hcv_be_2006$tot,
  cex = 0.1 * hcv_be_2006$tot, pch = 16,
  xlab = "duration of injection (years)",
  ylab = "seroprevalence", xlim = c(0, 25), ylim = c(0, 1)
)
```

---

hierarchical_bayesian_model

Hierarchical Bayesian Model

Description

Refers to section 10.3

Usage

```
hierarchical_bayesian_model(
  data,
  type = "far3",
  chains = 1,
  warmup = 1500,
  iter = 5000
)
```

Arguments

data	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ columns (for aggregated data) OR ‘age‘, ‘status‘ for (linelisting data)
type	type of model ("far2", "far3" or "log_logistic")
chains	number of Markov chains
warmup	number of warmup runs
iter	number of iterations

Value

a list of class hierarchical_bayesian_model with 6 items

datatype	type of datatype used for model fitting (aggregated or linelisting)
df	the dataframe used for fitting the model
type	type of bayesian model far2, far3 or log_logistic
info	parameters for the fitted model
sp	seroprevalence
foi	force of infection

Examples

```
df <- mumps_uk_1986_1987
model <- hierarchical_bayesian_model(df, type="far3")
model$info
plot(model)
```

---

lp_model *A local polynomial model.*

Description

Refers to section 7.1. and 7.2.

Usage

```
lp_model(data, kern = "tcub", nn = 0, h = 0, deg = 2)
```

Arguments

data	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ columns (for aggregated data) OR ‘age‘, ‘status‘ for (linelisting data)
kern	Weight function, default = "tcub". Other choices are "rect", "trwt", "tria", "epan", "bisq" and "gauss". Choices may be restricted when derivatives are required; e.g. for confidence bands and some bandwidth selectors.
nn	Nearest neighbor component of the smoothing parameter. Default value is 0.7, unless either h is provided, in which case the default is 0.
h	The constant component of the smoothing parameter. Default: 0.
deg	Degree of polynomial to use. Default: 2.

Value

a list of class lp_model with 6 items

datatype	type of datatype used for model fitting (aggregated or linelisting)
df	the dataframe used for fitting the model
pi	fitted locfit object for pi
eta	fitted locfit object for eta
sp	seroprevalence
foi	force of infection

See Also

[locfit::locfit()] for more information on the fitted locfit object

Examples

```
df <- mumps_uk_1986_1987
model <- lp_model(
  df,
  nn=0.7, kern="tcub"
  )
plot(model)
```

---

mixture_model *Fit a mixture model to classify serostatus*

Description

Refers to section 11.1 - 11.4

Usage

```
mixture_model(
  antibody_level,
  breaks = 40,
  pi = c(0.2, 0.8),
  mu = c(2, 6),
  sigma = c(0.5, 1)
)
```

Arguments

antibody_level	- vector of the corresponding raw antibody level
breaks	- number of intervals which the antibody_level are grouped into
pi	- proportion of susceptible, infected
mu	- a vector of means of component distributions (vector of 2 numbers in ascending order)
sigma	- a vector of standard deviations of component distributions (vector of 2 number)

Value

a list of class mixture_model with the following items

df	the dataframe used for fitting the model
info	list of 3 items parameters, distribution and constraints for the fitted model
susceptible	fitted distribution for susceptible
infected	fitted distribution for infected

Examples

```
df <- vzv_be_2001_2003[vzv_be_2001_2003$age < 40.5,]
data <- df$VZVmIUml[order(df$age)]
model <- mixture_model(antibody_level = data)
model$info
plot(model)
```

---

mseir_model *MSEIR model*

Description

Refers to section 3.4.

Usage

```
mseir_model(a, gamma, lambda, sigma, nu)
```

Arguments

a	age sequence
gamma	time in maternal class.
lambda	time in susceptible class.
sigma	time in latent class.
nu	time in infected class.

Value

list of class mseir_model with the following parameters

parameters	list of parameters used for fitting the model
output	matrix of proportion for each compartment over time

Examples

```
model <- mseir_model(
  a=seq(from=1,to=20,length=500), # age range from 0 -> 20 yo
  gamma=1/0.5, # 6 months in the maternal antibodies
  lambda=0.2,  # 5 years in the susceptible class
  sigma=26.07, # 14 days in the latent class
  nu=36.5      # 10 days in the infected class
)
model
```

---

mumps_uk_1986_1987 *Mumps serological data from the UK in 1986 and 1987 (aggregated)*

Description

a large survey of prevalence of antibodies to mumps and rubella viruses in the UK. The survey, covering subjects from 1 to over 65 years of age, provides information on the prevalence of antibody by age

Usage

```
mumps_uk_1986_1987
```

Format

A data frame with 3 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

Source

Morgan-Capner P, Wright J, Miller C L, Miller E. Surveillance of antibody to measles, mumps, and rubella by age. British Medical Journal 1988; 297 :770 doi:10.1136/bmj.297.6651.770

Examples

```
# Reproduce Fig 4.4 (left panel), p. 67
age <- mumps_uk_1986_1987$age
pos <- mumps_uk_1986_1987$pos
tot <- mumps_uk_1986_1987$tot
plot(age, pos / tot,
  cex = 0.008 * tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 45), ylim = c(0, 1)
)
```

---

parvob19_be_2001_2003 *Parvo B19 serological data from Belgium from 2001-2003 (line listing)*

Description

A seroprevalence survey testing for parvovirus B19 IgG antibody, performed on large representative national serum banks in Belgium, England and Wales, Finland, Italy, and Poland. The sera were collected between 1995 and 2004 and were obtained from residual sera submitted for routine laboratory testing.

Usage

```
parvob19_be_2001_2003
```

Format

A data frame with 5 variables:

age Age of individual

seropositive If the individual is seropositive or not

year Year surveyed

gender Gender of individual

parvouml Parvo B19 antibody units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.5 (left upper panel), p. 68
library(dplyr)
df <- parvob19_be_2001_2003 %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.02 * df$tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 82), ylim = c(0, 1)
)
```

`parvob19_ew_1996` *Parvo B19 serological data from England and Wales in 1996 (line listing)*

Description

A seroprevalence survey testing for parvovirus B19 IgG antibody, performed on large representative national serum banks in Belgium, England and Wales, Finland, Italy, and Poland. The sera were collected between 1995 and 2004 and were obtained from residual sera submitted for routine laboratory testing.

Usage

```
parvob19_ew_1996
```

Format

A data frame with 5 variables:

age Age of individual

seropositive If the individual is seropositive or not

year Year surveyed

gender Gender of individual

parvouml Parvo B19 antibody units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.5 (right upper panel), p. 68
# NB: This figure will look different to that of in the book, since we
# believe that the original authors has made some errors in specifying
# the sample size of the dots.
library(dplyr)
df <- parvob19_ew_1996 %>%
  mutate(age = round(age)) %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.02 * df$tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 82), ylim = c(0, 1)
)
```

---

parvob19_fi_1997_1998 *Parvo B19 serological data from Finland from 1997-1998 (line listing)*

Description

A seroprevalence survey testing for parvovirus B19 IgG antibody, performed on large representative national serum banks in Belgium, England and Wales, Finland, Italy, and Poland. The sera were collected between 1995 and 2004 and were obtained from residual sera submitted for routine laboratory testing.

Usage

```
parvob19_fi_1997_1998
```

Format

A data frame with 5 variables:

age Age of individual

seropositive If the individual is seropositive or not

year Year surveyed

gender Gender of individual

parvouml Parvo B19 antibody units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.5 (left bottom panel), p. 68
# NB: This figure will look different to that of in the book, since we
# believe that the original authors has made some errors in specifying
# the sample size of the dots.
library(dplyr)
df <- parvob19_fi_1997_1998 %>%
  mutate(age = round(age)) %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.07 * df$tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 82), ylim = c(0, 1)
)
```

---

parvob19_it_2003_2004 *Parvo B19 serological data from Italy from 2003-2004 (line listing)*

Description

A seroprevalence survey testing for parvovirus B19 IgG antibody, performed on large representative national serum banks in Belgium, England and Wales, Finland, Italy, and Poland. The sera were collected between 1995 and 2004 and were obtained from residual sera submitted for routine laboratory testing.

Usage

```
parvob19_it_2003_2004
```

Format

A data frame with 5 variables:

age Age of individual

seropositive If the individual is seropositive or not

year Year surveyed

gender Gender of individual

parvouml Parvo B19 antibody units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.5 (middle bottom panel), p. 68
# NB: This figure will look different to that of in the book, since we
# believe that the original authors has made some errors in specifying
# the sample size of the dots.
library(dplyr)
df <- parvob19_it_2003_2004 %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.07 * df$tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 82), ylim = c(0, 1)
)
```

---

parvob19_pl_1995_2004 *Parvo B19 serological data from Poland from 1995-2004 (line listing)*

Description

A seroprevalence survey testing for parvovirus B19 IgG antibody, performed on large representative national serum banks in Belgium, England and Wales, Finland, Italy, and Poland. The sera were collected between 1995 and 2004 and were obtained from residual sera submitted for routine laboratory testing.

Usage

```
parvob19_pl_1995_2004
```

Format

A data frame with 5 variables:

age Age of individual

seropositive If the individual is seropositive or not

year Year surveyed

gender Gender of individual

parvouml Parvo B19 antibody units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.5 (right bottom panel), p. 68
# NB: This figure will look different to that of in the book, since we
# believe that the original authors has made some errors in specifying
# the sample size of the dots.
library(dplyr)
df <- parvob19_pl_1995_2004 %>%
  mutate(age = round(age)) %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.07 * df$tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 82), ylim = c(0, 1)
)
```

---

pava *Monotonize seroprevalence*

Description

Monotonize seroprevalence

Usage

```
pava(pos = pos, tot = rep(1, length(pos)))
```

Arguments

pos	the positive count vector.
tot	the total count vector.

Value

computed list of 2 items pai1 for original values and pai2 for monotonized value

---

penalized_spline_model *Penalized Spline model*

Description

Penalized Spline model

Usage

```
penalized_spline_model(
  data,
  s = "bs",
  link = "logit",
  framework = "pl",
  sp = NULL
)
```

Arguments

data	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ column for aggregated data OR ‘age‘, ‘status‘ for linelisting data
s	smoothing basis to use
link	link function to use
framework	which approach to fit the model ("pl" for penalized likelihood framework, "glmm" for generalized linear mixed model framework)
sp	smoothing parameter

Value

a list of class penalized_spline_model with 6 attributes

datatype	type of datatype used for model fitting (aggregated or linelisting)
df	the dataframe used for fitting the model
framework	either pl or glmm
info	fitted "gam" model when framework is pl or "gamm" model when framework is glmm
sp	seroprevalence
foi	force of infection

See Also

[mgcv::gam()], [mgcv::gamm()] for more information the fitted gam and gamm model

Examples

```
data <- parvob19_be_2001_2003
data$status <- data$seropositive
model <- penalized_spline_model(data, framework="glmm")
model$info$gam
plot(model)
```

plot.estimate_from_mixture *plot() overloading for result of estimate_from_mixture*

Description

plot() overloading for result of estimate_from_mixture

Usage

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

Arguments

x	the mixture_model
...	arbitrary params.

Value

ggplot object

plot.farrington_model *plot() overloading for Farrington model*

Description

plot() overloading for Farrington model

Usage

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

Arguments

x	the Farrington model object.
...	arbitrary params.

Value

ggplot object

plot.fp_model *plot() overloading for fractional polynomial model*

Description

plot() overloading for fractional polynomial model

Usage

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

Arguments

x	the fractional polynomial model object.
...	arbitrary params.

Value

ggplot object

plot.hierarchical_bayesian_model *plot() overloading for hierarchical_bayesian_model*

Description

plot() overloading for hierarchical_bayesian_model

Usage

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

Arguments

x	hierarchical_bayesian_model object created by serosv.
...	arbitrary params.

Value

ggplot object

plot.lp_model *plot() overloading for local polynomial model*

Description

plot() overloading for local polynomial model

Usage

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

Arguments

x	the local polynomial model object.
...	arbitrary params.

Value

ggplot object

plot.mixture_model *plot() overloading for mixture model*

Description

plot() overloading for mixture model

Usage

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

Arguments

x	the mixture_model
...	arbitrary params.

Value

ggplot object

plot.mseir_model *plot() overloading for MSEIR model*

Description

plot() overloading for MSEIR model

Usage

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

Arguments

x	the mseir_model object.
...	arbitrary params.

Value

ggplot object

plot.penalized_spline_model *plot() overloading for penalized spline*

Description

plot() overloading for penalized spline

Usage

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

Arguments

x	the penalized_spline_model object
...	arbitrary params.

Value

ggplot object

plot.polynomial_model *plot() overloading for polynomial model*

Description

plot() overloading for polynomial model

Usage

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

Arguments

x	the polynomial model object
...	arbitrary params.

Value

ggplot object

plot.sir_basic_model *plot() overloading for SIR model*

Description

plot() overloading for SIR model

Usage

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

Arguments

x	the sir_basic_model object.
...	arbitrary params.

Value

ggplot object

plot.sir_static_model *plot() overloading for SIR static model*

Description

plot() overloading for SIR static model

Usage

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

Arguments

x	the sir_static_model object.
...	arbitrary params.

Value

ggplot object

plot.sir_subpops_model *plot() overloading for SIR sub populations model*

Description

plot() overloading for SIR sub populations model

Usage

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

Arguments

x	the sir_subpops_models object.
...	arbitrary params.

Value

list of ggplot objects, each object is the plot for the corresponding subpopulation

plot.weibull_model *plot() overloading for Weibull model*

Description

plot() overloading for Weibull model

Usage

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

Arguments

x	the Weibull model object.
...	arbitrary params.

Value

ggplot object

plot_gcv *Plotting GCV values with respect to different nn-s and h-s parameters.*

Description

Refers to section 7.2.

Usage

```
plot_gcv(age, pos, tot, nn_seq, h_seq, kern = "tcub", deg = 2)
```

Arguments

age	the age vector.
pos	the pos vector.
tot	the tot vector.#'
nn_seq	Nearest neighbor sequence.
h_seq	Smoothing parameter sequence.
kern	Weight function, default = "tcub". Other choices are "rect", "trwt", "tria", "epan", "bisq" and "gauss". Choices may be restricted when derivatives are required; e.g. for confidence bands and some bandwidth selectors.
deg	Degree of polynomial to use. Default: 2.

Value

plot of gcv value

Examples

```
df <- mumps_uk_1986_1987
plot_gcv(
  df$age, df$pos, df$tot,
  nn_seq = seq(0.2, 0.8, by=0.1),
  h_seq = seq(5, 25, by=1)
)
```

`polynomial_model` *Polynomial models*

Description

Refers to section 6.1.1

Usage

```
polynomial_model(data, k, type, link = "log")
```

Arguments

`data`	the input data frame, must either have ‘age‘, ‘pos‘, ‘tot‘ columns (for aggregated data) OR ‘age‘, ‘status‘ for (linelisting data)
`k`	degree of the model.
`type`	name of method (Muench, Giffith, Grenfell).
`link`	link function.

Value

a list of class polynomial_model with 5 items

`datatype`	type of datatype used for model fitting (aggregated or linelisting)
`df`	the dataframe used for fitting the model
`info`	fitted "glm" object
`sp`	seroprevalence
`foi`	force of infection

Examples

```
data <- parvob19_fi_1997_1998[order(parvob19_fi_1997_1998$age), ]
data$status <- data$seropositive
aggregated <- transform_data(data$age, data$seropositive, heterogeneity_col = "age")

# fit with aggregated data
model <- polynomial_model(aggregated, type = "Muench")
# fit with linelisting data
model <- polynomial_model(data, type = "Muench")
plot(model)
```

---

rubella_mumps_uk *Rubella - Mumps data from the UK (aggregated)*

Description

Rubella - Mumps data from the UK (aggregated)

Usage

```
rubella_mumps_uk
```

Format

A data frame with 5 variables:

age Age group

NN Number of individuals negative to rubella and mumps

NP Number of individuals negative to rubella and positive to mumps

PN Number of individuals positive to rubella and negative to mumps

PP Number of individuals positive to rubella and mumps

Source

Morgan-Capner P, Wright J, Miller C L, Miller E. Surveillance of antibody to measles, mumps, and rubella by age. British Medical Journal 1988; 297 :770 doi:10.1136/bmj.297.6651.770

rubella_uk_1986_1987 *Rubella serological data from the UK in 1986 and 1987 (aggregated)*

Description

Prevalence of rubella in the UK, obtained from a large survey of prevalence of antibodies to both mumps and rubella viruses.

Usage

```
rubella_uk_1986_1987
```

Format

A data frame with 3 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

Source

Morgan-Capner P, Wright J, Miller C L, Miller E. Surveillance of antibody to measles, mumps, and rubella by age. British Medical Journal 1988; 297 :770 doi:10.1136/bmj.297.6651.770

Examples

```
# Reproduce Fig 4.4 (middle panel), p. 67
age <- rubella_uk_1986_1987$age
pos <- rubella_uk_1986_1987$pos
tot <- rubella_uk_1986_1987$tot
plot(age, pos / tot,
  cex = 0.008 * tot, pch = 16, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 45), ylim = c(0, 1)
)
```

set_plot_style *Helper to adjust styling of a plot*

Description

Helper to adjust styling of a plot

Usage

```
set_plot_style(
  sero = "blueviolet",
  ci = "royalblue1",
  foi = "#fc0328",
  sero_line = "solid",
  foi_line = "dashed",
  xlabel = "Age"
)
```

Arguments

sero	- color for seroprevalence line
ci	- color for confidence interval
foi	- color for force of infection line
sero_line	- linetype for seroprevalence line
foi_line	- linetype for force of infection line
xlabel	- x label

Value

list of updated aesthetic values

---

sir_basic_model *Basic SIR model*

Description

Refers to section 3.1.3.

Usage

```
sir_basic_model(times, state, parameters)
```

Arguments

times	time sequence.
state	the initial state of the model.
parameters	the parameters of the model.

Details

In `state`:

- `S`: number of susceptible

- `I`: number of infected

- `R`: number of recovered

In `parameters`:

- `alpha`: disease-related death rate

- `mu`: natural death rate (= 1/life expectancy)

- `beta`: transmission rate

- `nu`: recovery rate

- `p`: percent of population vaccinated at birth

Value

list of class sir_basic_model with the following items

`parameters`	list of parameters used for fitting the model
`output`	matrix of population for each compartment over time

Examples

```
state <- c(S=4999, I=1, R=0)
parameters <- c(
  mu=1/75, # 1 divided by life expectancy (75 years old)
  alpha=0, # no disease-related death
  beta=0.0005, # transmission rate
  nu=1, # 1 year for infected to recover
  p=0 # no vaccination at birth
)
times <- seq(0, 250, by=0.1)
model <- sir_basic_model(times, state, parameters)
model
```

`sir_static_model` *SIR static model (age-heterogeneous, endemic equilibrium)*

Description

Refers to section 3.2.2.

Usage

```
sir_static_model(a, state, parameters)
```

Arguments

a	age sequence.
state	the initial state of the system.
parameters	the model's parameter.

Details

In state:

- s: proportion susceptible

- i: proportion infected

- r: proportion recovered

In parameters:

- lambda: natural death rate

- nu: recovery rate

Value

list of class sir_static_model with the following items

parameters	list of parameters used for fitting the model
output	matrix of proportion for each compartment over time

Examples

```
state <- c(s=0.99,i=0.01,r=0)
parameters <- c(
  lambda = 0.05,
  nu=1/(14/365) # 2 weeks to recover
)
ages<-seq(0, 90, by=0.01)
model = sir_static_model(ages, state, parameters)
model
```

---

sir_subpops_model *SIR Model with Interacting Subpopulations*

Description

Refers to section 3.5.1.

Usage

```
sir_subpops_model(times, state, parameters)
```

Arguments

times	time sequence.
state	the initial state of the model.
parameters	the parameters of the model.

Details

In state:

- s: Percent susceptible
- i: Percent infected
- r: Percent recovered

In parameters:

- mu: natural death rate (1/L).
- beta: transmission rate w.r.t population (beta tilde)
- nu: recovery rate
- k: number of subpopulations

Value

list of class sir_subpops_model with the following items

parameters	list of parameters used for fitting the model
output	matrix of proportion for each compartment over time

Examples

```
k <- 2
state <- c(
  s = c(0.8, 0.8),
  i = c(0.2, 0.2),
  r = c(  0,   0)
)
beta_matrix <- c(
  c(0.05, 0.00),
  c(0.00, 0.05)
)
parameters <- list(
  beta = matrix(beta_matrix, nrow=k, ncol=k, byrow=TRUE),
  nu = c(1/30, 1/30),
  mu = 0.001,
  k = k
)
times<-seq(0,10000,by=0.5)
model <- sir_subpops_model(times, state, parameters)
model
```

`tb_nl_1966_1973` *Tuberculosis serological data from the Netherlands 1966-1973 (aggregated)*

Description

A study of tuberculosis conducted in the Netherlands. Schoolchildren, aged between 6 and 18 years, were tested using the tuberculin skin test.

Usage

```
tb_nl_1966_1973
```

Format

A data frame with 5 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

gender Gender of cohort (unsure what 0 and 1 means)

birthyr Birth year of cohort

Source

Nagelkerke, N., Heisterkamp, S., Borgdorff, M., Broekmans, J. and Van Houwelingen, H. (1999), Semi-parametric estimation of age-time specific infection incidence from serial prevalence data. Statist. Med., 18: 307-320. doi:10.1002/(SICI)10970258(19990215)18:3<307::AIDSIM15>3.0.CO;2-Z

Examples

```
# Reproduce Fig 4.6, p.70
age <- tb_nl_1966_1973$age
birthyr <- tb_nl_1966_1973$birthyr
pos <- tb_nl_1966_1973$pos
tot <- tb_nl_1966_1973$tot
# left panel
plot(age, pos / tot,
  pch = 16, cex = 0.00005 * tot, xlab = "age",
  ylab = "prevalence", xlim = c(6, 18)
)
# right panel
plot(birthyr, pos / tot,
  pch = 16, cex = 0.00005 * tot, xlab = "year", ylab = "prevalence"
)
```

`transform_data` *Generate a dataframe with 't', 'pos' and 'tot' columns from 't' and 'seropositive' vectors.*

Description

Generate a dataframe with 't', 'pos' and 'tot' columns from 't' and 'seropositive' vectors.

Usage

```
transform_data(t, spos, heterogeneity_col = "t")
```

Arguments

`t`	the time vector.
`spos`	the seropositive vector.
`heterogeneity_col`	new name for the time vector (default to "t")

Value

dataframe in aggregated format

Examples

```
df <- hcv_be_2006
hcv_df <- transform_data(df$dur, df$seropositive)
hcv_df
```

`vzv_be_1999_2000` *VZV serological data from Belgium (Flanders) from 1999-2000 (aggregated)*

Description

Age-specific seroprevalence of VZV antibodies, assessed in Flanders (Belgium) between October 1999 and April 2000. This population was stratified by age in order to obtain approximately 100 observations per age group.

Usage

```
vzv_be_1999_2000
```

Format

A data frame with 3 variables:

age Age group

pos Number of seropositive individuals

tot Total number of individuals surveyed

Source

Thiry, N., Beutels, P., Shkedy, Z. et al. The seroepidemiology of primary varicella-zoster virus infection in Flanders (Belgium). Eur J Pediatr 161, 588-593 (2002). doi:10.1007/s0043100210532

Examples

```
# Reproduce Fig 4.7 (left panel), p.71
age <- vzv_be_1999_2000$age
pos <- vzv_be_1999_2000$pos
tot <- vzv_be_1999_2000$tot
plot(age, pos / tot,
  cex = 0.036 * tot, pch = 19, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 45), ylim = c(0, 1)
)
```

---

vzv_be_2001_2003 *VZV serological data from Belgium from 2001-2003 (line listing)*

Description

The survey is the same as the one used to study the seroprevalence of parvovirus B19 in Belgium, as described above.

Usage

```
vzv_be_2001_2003
```

Format

A data frame with 4 variables:

age Age of individual

seropositive If the individual is seropositive or not

gender Gender of individual

VZVmIUml VZV milli international units per ml

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

Examples

```
# Reproduce Fig 4.7 (right panel), p.71
library(dplyr)
df <- vzv_be_2001_2003 %>%
  mutate(age = round(age)) %>%
  group_by(age) %>%
  summarise(pos = sum(seropositive), tot = n())
plot(df$age, df$pos / df$tot,
  cex = 0.036 * df$tot, pch = 19, xlab = "age", ylab = "seroprevalence",
  xlim = c(0, 45), ylim = c(0, 1)
)
```

---

vzv_parvo_be *VZV and Parvovirus B19 serological data in Belgium (line listing)*

Description

VZV and Parvovirus B19 serological data in Belgium (line listing)

Usage

```
vzv_parvo_be
```

Format

A data frame with 7 variables:

id ID of individual

age Age of individual

gender Gender of individual

parvouml Parvo B19 antibody units per ml

parvo_res If an individual is positive for parvovirus B19

VZVmUIml VZV milli international units per ml

vzv_res If an individual is positive for VZV

Source

MOSSONG, J., N. HENS, V. FRIEDERICHS, I. DAVIDKIN, M. BROMAN, B. LITWINSKA, J. SIENNICKA, et al. "Parvovirus B19 Infection in Five European Countries: Seroepidemiology, Force of Infection and Maternal Risk of Infection." Epidemiology and Infection 136, no. 8 (2008): 1059-68. doi:10.1017/S0950268807009661

`weibull_model` *The Weibull model.*

Description

Refers to section 6.1.2.

Usage

```
weibull_model(data)
```

Arguments

`data`	the input data frame, must either have ‘t‘, ‘pos‘, ‘tot‘ column for aggregated data OR ‘t‘, ‘status‘ for linelisting data

Value

list of class weibull_model with the following items

`datatype`	type of datatype used for model fitting (aggregated or linelisting)
`df`	the dataframe used for fitting the model
`info`	fitted "glm" object
`sp`	seroprevalence
`foi`	force of infection

See Also

[stats::glm()] for more information on the fitted "glm" object

Examples

```
df <- hcv_be_2006[order(hcv_be_2006$dur), ]
df$t <- df$dur
df$status <- df$seropositive
model <- weibull_model(df)
plot(model)
```

Index