Package ‘swash’

May 24, 2026

Type Package

Title Health Geography Toolbox for Model-Based Analysis of Infections Panel Data

Version 2.0.1

Date 2026-05-22

Author Thomas Wieland [aut, cre] (ORCID: <https://orcid.org/0000-0001-5168-9846>)

Maintainer Thomas Wieland <geowieland@googlemail.com>

Depends R (>= 3.5.0), lubridate, sf, spdep, zoo, strucchange

Description Within epidemic outbreaks, infections grow and decline differently between regions, and the velocity of spatial spread differs between countries. The swash library offers a set of model-based analyses for these topics. Spread velocity may be analysed with the Swash-Backwash Model for the Single Epidemic Wave and corresponding functions for bootstrap confidence intervals, country comparison, and visualization of results. Differences in epidemic growth between regions may be analysed using logistic growth models, exponential growth models, Hawkes processes and breakpoint analyses. All functionalities are accessed by the class ``infpan'' for infections panel data defined in this package, which is built from a data.frame provided by the user.

License GPL (>= 2)

Imports methods

NeedsCompilation no

Repository CRAN

Date/Publication 2026-05-24 08:50:02 UTC

Contents

`swash-package` *swash: Health Geography Toolbox for Model-Based Analysis of Infections Panel Data*

Description

The R library *swash* provides a toolbox for quantitative analyses in health geography with a focus on the spatial spread of infectious diseases. It bundles functions developed by the author between 2020 and 2023 for the quantitative analysis of (panel) infection data during the COVID-19 pandemic. The aim was to consolidate these methods and analytical tools into a unified and coherent framework. The target audience of this R package consists of researchers and practitioners in the fields of health geography, spatial epidemiology, and statistics.

In order to use all functionalities, the user should import her/his infections panel data using the function `load_infections_paneldata()`, which returns an instance of class `infpan`. The panel data is checked whether it is balanced and whether it includes missing values. From an `infpan` object, the user may utilize the following built-in analysis models and visualization functions:

- *Swash-Backwash Model for the Single Epidemic Wave*, including further analysis towards bootstrap-based inference and country comparison as well as visualization
- *Growth Analysis* with *logistic* and *exponential growth models* for cumulative or incremental infections, whereby the former is intended for the entire infection wave, and the latter for the initial phase of the infection wave; including visualization
- *Hawkes process* models for incremental infections; including visualization
- *Breakpoints analysis* using the *Bai-Perron algorithm* implemented in `strucchange::breakpoints`; including visualization
- Calculation of further epidemic indicators from the infections panel data such as the *effective reproduction number*
- *Plots* of infection curves by region

`infpan` objects and objects resulting from the functions mentioned above have summary() and plot() methods. All mentioned functions may be used stand-alone as well.

Details

Based on an `infpan` object, several indicators may be calculated from incremental infections values, such as incidence or effective reprodction number R_t. Infection curves may be plotted by `plot(infpan)`. All built-in model analyses may be conducted based on an instance of class `infpan`.

The Swash-Backwash Model (SBM) for the Single Epidemic Wave is the spatial equivalent of the classic epidemiological SIR (Susceptible-Infected-Recovered) model. It was developed by Cliff and Haggett (2006) to model the velocity of spread of infectious diseases across space. Current applications can be found, for example, in Smallman-Raynor et al. (2022a,b). This package enables the calculation of the Swash-Backwash Model for user-supplied panel data on regional infections. The core of this is the `swash_backwash()` function, which calculates the model and creates a model object of the `sbm` class defined in this package. This class can be used to visualize results (`summary()`, `plot()`) and calculate bootstrap confidence intervals for the model estimates (`confint(sbm)`); the latter returns an object of the built-in class `sbm_ci`. Two `sbm_ci` objects for different countries may be compared with `compare_countries()`, which allows the estimation of mean differences of a user-specified model parameter (e.g., spatial reproduction number R_{OA}) between two countries. This makes it possible to check whether the spatial spread velocity of a communicable disease is significantly different in one country than in another country; the result is an object of class `countries`. To calculate the SBM model based on an `infpan` object, use the corresponding method `swash(infpan)`.

The library allows for estimating growth models based on time series of infections. Logistic and exponential growth models (see, e.g., Chowell et al. 2014, 2015, Pell et al. 2018, Wieland 2020a, 2020b) as well as Hawkes process models (see, e.g. Rizoiu et al. 2018) are provided. Additionally, breakpoints in time series may be detected (see, e.g., Wieland 2020b). A model for a single time series may be estimated with the built-in functions `logistic_growth()`, `exponential_growth()`, `hawkes_growth()`, or `breaks_growth()`, respectively. These function return objects of class `loggrowth`, `expgrowth`, `hawkes`, and `breaksgrowth`, respectively, all of them defined in this package. Plotting is available via the `plot` method. Estimating such a model based on an `infpan` object is provided by the `infpan` methods `growth()`, `growth_initial()`, `growth_hawkes()`, and `growth_breaks()`, respectively, all of them resulting in an object of class `growthmodels`.

The package also contains other functions for spatio-temporal analysis, including spatial statistics (`nbstat()` for neighborhood statistics) and fit metrics (`metrics()`, `binary_metrics()`, `binary_metrics_glm()`). The package includes example data from the COVID-19 pandemic.

Author(s)

Thomas Wieland

References

Chowell G, Simonsen L, Viboud C, Yang K (2014) Is West Africa Approaching a Catastrophic Phase or is the 2014 Ebola Epidemic Slowing Down? Different Models Yield Different Answers for Liberia. *PLoS currents 6*. doi:10.1371/currents.outbreaks.b4690859d91684da963dc40e00f3da81

Chowell G, Viboud C, Hyman JM, Simonsen L (2015) The Western Africa ebola virus disease epidemic exhibits both global exponential and local polynomial growth rates. *PLOS Currents Outbreaks*, ecurrents.outbreaks.8b55f4bad99ac5c5db3663e916803261. doi:10.1371/currents.outbreaks.8b55f4bad99ac5c5db36

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Li, MY (2018) An Introduction to Mathematical Modeling of Infectious Diseases. doi:10.1007/9783319721224

Nishiura H, Chowell G (2009) The effective reproduction number as a prelude to statistical estimation of time-dependent epidemic trends. In Chowell G, Hyman JM, Bettencourt LMA (eds.) Mathematical and statistical estimation approaches in epidemiology, 103–121. doi:10.1007/97890-48123131_5

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L. (2018) SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In: *Proceedings of the 2018 World Wide Web Conference. WWW'18*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 419–428. doi:10.1145/3178876.3186108

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022a) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022b) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

Viboud C, Bjørnstad ON, Smith DL, Simonsen L, Miller MA, Grenfell BT (2006) Synchrony, Waves, and Spatial Hierarchies in the Spread of Influenza. *Science* 312, 447-451. doi:10.1126/science.1125237

Wieland T (2020a) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2020b) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Wieland T (2022) Spatial patterns of excess mortality in the first year of the COVID-19 pandemic in Germany. *European Journal of Geography* 13(4), 18-33. doi:10.48088/ejg.t.wie.13.4.018.033

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave
```

```
infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c(
      "Population" = "pop"
        ),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

is(infpan_CH)
# "infpan"

plot(
  infpan_CH,
  plot_rollmean = TRUE
  )
# Plot cases

infpan_CH <- calculate_Rt(
  infpan_CH,
  verbose = TRUE
  )
# Calculate effective reproduction number

summary(infpan_CH)
# Summary of infpan object

timestamps(infpan_CH)
# Time stamps of infpan object

CH_covidwave1 <-
  swash(
    infpan_CH,
    verbose = TRUE
    )
# Swash-Backwash Model for Swiss COVID19 cases
# Spatial aggregate: NUTS 3 (cantons)

summary(CH_covidwave1)
# Summary of Swash-Backwash Model
```

`as_balanced` *Correction of Non-balanced Panel Dataset with Regional Infection Data*

Description

This function corrects non-balanced input panel data by replacing missing entries with a user-given constant (e.g., 0).

Usage

```
as_balanced(
  data,
  col_cases,
  col_date,
  col_region,
  fill_missing = 0
  )
```

Arguments

`data`	`data.frame` with regional infection data
`col_cases`	Column containing the cases (numeric)
`col_date`	Column containing the time points (e.g., days)
`col_region`	Column containing the unique identifier of the regions (e.g., name, NUTS 3 code)
`fill_missing`	Constant to fill missing values (default and recommended: 0)

Details

The Swash-Backwash Model for the Single Epidemic Wave does not necessarily require balanced panel data in order for the calculations to be carried out. However, for a correct estimation it is implicitly assumed that the input data is balanced. The function corrects non-balanced panel data. It is executed automatically whithin the `swash_backwash()` function and `swash()` method, respectively, but can also be used separately.

Value

`data`	Corrected input dataset (`data.frame`)

Author(s)

Thomas Wieland

See Also

`is_balanced`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
```

```
# Extract first COVID-19 wave

COVID19Cases_geoRegion_balanced <-
  is_balanced(
  data = COVID19Cases_geoRegion,
  col_cases = "entries",
  col_date = "datum",
  col_region = "geoRegion"
)
# Test whether "COVID19Cases_geoRegion" is balanced panel data

COVID19Cases_geoRegion_balanced$data_balanced
# Balanced? TRUE or FALSE

if (COVID19Cases_geoRegion_balanced$data_balanced == FALSE) {
  COVID19Cases_geoRegion <-
    as_balanced(
    COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion"
  )
}
# Correction of dataset "COVID19Cases_geoRegion"
# not necessary as parameter balance of is_balanced is set TRUE by default
```

`binary_metrics` *Fit metrics of observed and expected binary variables*

Description

Calculation of fit metrices for binary variables (Sensitivity, specificity, accuracy)

Usage

```
binary_metrics(
  observed,
  expected,
  no_information_rate = "negative"
  )
```

Arguments

`observed`	Numeric vector: Y observed
`expected`	Numeric vector: Y expected
`no_information_rate`	`character` argument which indicates whether the no-information rate is calculated based on negatives or positives

Details

The function computes model performance metrices for binary outcomes. Observed and expected data must be stated by the user. The function returns sensitivity, specificity, accurracy, and no-information rate.

Value

`list` with two entries:

`fit_metrics`:	`list` with fit metrics (`sens`, `spec`, ...)
`observed_expected`:	`data.frame` with observed, expected and hit (1/0)

Author(s)

Thomas Wieland

References

Altman DG, Bland JM (1994) Diagnostic tests. 1: Sensitivity and specificity. *British Medical Journal* 308, 1552. doi:10.1136/bmj.308.6943.1552.

Boehmke B, Greenwell B (2020) Hands-On Machine Learning with R (1 ed.). Taylor & Francis, New York, NY.

See Also

`metrics`, `binary_metrics_glm`

Examples

```
obs <- c(1,1,0,0,0,0,1,0,1)
exp <- c(0,1,0,0,0,0,1,0,0)

binary_metrics(
  obs,
  exp
)
```

`binary_metrics_glm` *Fit metrics for binary logit model*

Description

Calculation of fit metrices for binary variables (Sensitivity, specificity, accuracy) out of binary logit models (`glm` object)

Usage

```
binary_metrics_glm(
  logit_model,
  threshold = 0.5
  )
```

Arguments

`logit_model`	`glm` object with binary logit model
`threshold`	Threshold for distinction of probability with respect to TRUE or FALSE

Details

The function computes model performance metrices for binary outcomes. A binary logit model (glm) must be stated by the user. The function returns sensitivity, specificity, accurracy, and no-information rate.

Value

`list` with two entries:

`fit_metrics:`	`list` with fit metrics (`sens`, `spec`, ...)
`observed_expected:`	`data.frame` with observed, expected and hit (1/0)

Author(s)

Thomas Wieland

References

Altman DG, Bland JM (1994) Diagnostic tests. 1: Sensitivity and specificity. *British Medical Journal* 308, 1552. doi:10.1136/bmj.308.6943.1552.

Boehmke B, Greenwell B (2020) Hands-On Machine Learning with R (1 ed.). Taylor & Francis, New York, NY.

See Also

`metrics`, `binary_metrics`

Examples

```
dep <- c(1,1,0,0,0,0,1,0,1, 1)
x <- c(2,3,1,1,0,1,3,2,1,3)

testmodel <-
  glm(
    dep~x,
    family=binomial()
  )
```

```
summary(testmodel)

binary_metrics_glm(testmodel)
```

`breaksgrowth-class` *Class* `"breaksgrowth"`

Description

The class `"breaksgrowth"` contains the results of the breaks_growth() function. Use `summary(breaksgrowth)` for results summary.

Objects from the Class

Objects can be created by the function `breaks_growth`.

Slots

`GrowthModel_OLS`**:** Object of class `list` Results of the OLS fit (predicted, parameters)

`t`**:** Object of class `numeric` Input time points data

`y`**:** Object of class `numeric` Input infections data

`config`**:** Object of class `list` Model fit configurations

Methods

summary `signature(object = "breaksgrowth")`: Prints a summary of `breaksgrowth` objects

plot `signature(x = "breaksgrowth")`: Plots the results of the breakpoint analysis

print `signature(x = "breaksgrowth")`: Prints an `breaksgrowth` object; use `summary(breaksgrowth)` for results

Author(s)

Thomas Wieland

References

Bai J, Perron P (2003) Computation and analysis of multiple structural change models. *Journal of Applied Econometrics* 18(1), 1-22. doi:10.1002/jae.659

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Zeileis C, Kleiber W, Krämer K, Hornik, K (2003) Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis* 44(1-2), 109-123. doi:10.1016/S01679473(03)00030-6

Examples

```
showClass("breaksgrowth")
```

`breaks_growth` *Time Series Model with Breakpoints*

Description

Estimation of breakpoints in linear regression models from incremental (e.g., daily) infections data

Usage

```
breaks_growth(
  y,
  t,
  ln = FALSE,
  add_constant = 1,
  alpha = 0.05,
  ...,
  verbose = FALSE
  )
```

Arguments

`y`	`numeric` `vector` with cumulative infections data over time
`t`	`vector` of class `numeric` or `Date` with time points or dates
`ln`	`bool` argument which indicates whether dependent variable should be transformed by natural logarithm
`add_constant`	Numeric constant to be added to y if zero values occur
`alpha`	Significance level α for 1-α*100 confidence intervals
`...`	Other parameters passed to `strucchange::breakpoints` (see the corresponding documentation)
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

This function allows detecting breakpoints in a linear regression time series model. The user must specify the dependent variable (e.g, daily infections) and the time variable (time counter or date values). The estimation is performed using OLS. The function internally uses the function `breakpoints` from the `strucchange` package (Zeileis et al. 2003), where breakpoints are identified using the Bai-Perron algorithm (Bai & Perron 2003).

Value

object of class `breaksgrowth-class`

Author(s)

Thomas Wieland

References

Bai J, Perron P (2003) Computation and analysis of multiple structural change models. *Journal of Applied Econometrics* 18(1), 1-22. doi:10.1002/jae.659

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Zeileis C, Kleiber W, Krämer K, Hornik K (2003) Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis* 44(1-2), 109-123. doi:10.1016/S01679473(03)00030-6

See Also

`exponential_growth`, `logistic_growth`, hawkes_growth

Examples

```
data(Infections)
# Confirmed SARS-CoV-2 cases in Germany

breakpoints_infections <- breaks_growth(
  y = Infections$infections_daily,
  t = Infections$day,
  ln = TRUE,
  verbose = TRUE
)
# Breakpoints for time series of infections

summary(breakpoints_infections)
# Summary of breakpoints

plot(breakpoints_infections)
# Plot breakpoints
```

`C19dNUTSdata` *Regional cumulative COVID-19 deaths*

Description

Cumulative COVID-19 deaths absolute and per 100,000 pop at NUTS3 level for 31 EU/EFTA countries

Usage

```
data("C19dNUTSdata")
```

Format

A data frame with 1,143 observations (each one represents a spatial NUTS unit).

`NUTS_ID`	NUTS ID of the spatial unit
`CNTR_CODE`	Country code (= NUTS 0 ID) of the given spatial unit
`NUTS_Level`	NUTS level of the given spatial unit (0 = national, 1, 2, 3)
`NUTS2_ID`	NUTS 2 ID of the spatial unit
`NUTS1_ID`	NUTS 1 ID of the spatial unit
`NUTS_Name`	Latin name of the spatial unit
`C19deaths`	Cumulative COVID-19 deaths [persons]
`pop2020`	Population in 2020 [persons]
`C19deaths_per100000`	Cumulative COVID-19 deaths [per 100,000]
`annotation`	Annotation)

Details

Note: This data was originally released in the author's package C19dNUTS in 2022 (https://cran.r-project.org/package=C19dNUTS). Some of the URLs referred to here were moved or deleted.

The dataset contains cumulative COVID-19 deaths at the regional level (mostly NUTS 3, N=1,143) for 31 EU/EFTA countries (AT, BE, BG, CH, CY, CZ, DE, DK, EE, EL, ES, FI, FR, HR, HU, IE, IS, IT, LT, LU, LV, MT, NL, NO, PL, PT, RO, SE, SI, SK, UK). The `C19deaths` variable contains the absolute number of COVID-19 related deaths, and the variable `C19deaths_per100000` equals the death numbers relative to the population (per 100,000).

Unless otherwise noted, data includes all reported COVID-19 related deaths since the beginning of the COVID-19 pandemic through June 2022. Please refer to the source section below for the exact date on which each raw dataset was retrieved. The spatial level is the current NUTS 2021 classification of the European Union (see ‘https://ec.europa.eu/eurostat/web/nuts/background‘), with one slight modification (see "Technical details" below). The variable `NUTS_Level` documents the spatial level for which the numbers apply (mostly `NUTS_Level` = 3 for NUTS3).

Technical details:

This dataset contains cumulative numbers and no time series, as many countries only publish cumulative data on COVID-19 deaths. In cases where countries only publish COVID-19 deaths in the form of daily data, the numbers were summed up over the entire period under consideration at the respective spatial level.

The definition of a COVID-19 death may vary between countries. The respective definition can usually be found on the website of the national health authority. In some countries, data is reported based on different definitions. For example, Lithuania uses three different definitions, namely a) based on the main cause of death in the death certificate, b) based on a mention in the death certificate and c) died within 28 days of a positive SARS-CoV-2 test (`https://open-data-sets-ls-osp-sdg.hub.arcgis.com/datasets/ba35de03e111430f88a86f7d1f351de6_0/about`). In England, for example, a distinction is made between the deceased who tested positive and those who died from COVID-19 based on the death certificate (https://coronavirus.data.gov.uk/details/deaths). In these cases, the definition used has always been the equivalent of the total number of COVID-19 deaths as reported by the national figures from Johns Hopkins University (https://coronavirus.jhu.edu/data/cumulative-cases).

In some cases, countries publish regional COVID-19 data directly at NUTS3 level (e.g., Germany) or NUTS2 level (e.g., Italy). In most cases, the regional level had to be linked manually using the name of the region (e.g., Bulgaria, Norway, Switzerland). Some countries even publish the relevant data on a smaller scale, i.e. below NUTS3 (e.g., Austria, Netherlands, Poland, England). In these cases, where a reference table (subnational spatial unit <-> NUTS3) was available, the lower level was linked to the NUTS3 level (e.g., England). If no reference table but geodata (shapefiles) for the lower spatial level was available (e.g., Austria, Netherlands, Poland), the lower level was linked to the NUTS3 level via a spatial join (Polygon centroids; in cases where the centroid was outside the polygon, it was placed inside the polygon manually). In these cases, the numbers were then summed up at NUTS3 level.

The spatial reference used here is the current EU NUTS Shapefile (https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/nuts; accessed 2022-06-23). The dataset can be linked directly to this shapefile, where the unique id field to which the link can be made is the column `NUTS_ID`. However, there is one exception: To ensure data compatibility, the UK NUTS3 regions UKM61 and UKM63 were aggregated into one region (UKM61).

The data reflects 1,309,326 COVID-19 related deaths in the 31 countries in the investigated time period. The variable `C19deaths_per100000` is non-normally distributed (Shapiro-Wilk test: $W = 0.92284$, $p < 0.01$). The natural log of `C19deaths_per100000` is spatially autocorrelated (Moran's I with queen contiguous spatial weighting: $I = 0.65228$, $p < 0.01$).

Data limitations:

It can be assumed that there are differences between countries and possibly also over time in the definition of a COVID-19 death (see "Technical details" above). Please check the definition on the website of the respective national health authority.

Data on COVID-19 deaths are incomplete for the following EU/EFTA countries: Bulgaria, France, Poland. In Bulgaria, regional COVID-19 deaths were only published for the years 2020 and 2021 (36,142 COVID-19 related deaths in total), i.e. the cases for 2022 are missing. France only publishes the COVID-19 patients which died in a hospital at the regional level, which equals 120,630 COVID-19 related deaths over the period under consideration (as of 2022-06-30). The total number of COVID-19 related deaths in France for the same time is equal to 149,533, which means that there is a lack of 28,903 COVID-19 fatalities (19.3 %, e.g., people which died in nursing homes). Polish deaths are missing COVID-19 deaths from the first pandemic wave. Therefore only the COVID-19 fatalities from the date 2020-11-24 are included, which equals 102,449 deaths. In the previous period, 13,780 COVID-19 deaths were reported, which are not included in the data set, i.e. 11.9% of the deaths are missing.

Of the 31 EU/EFTA countries included, regional data are only available for 24 countries. The following countries have not published sub-national data for COVID-19 deaths: Cyprus, Finland, Island, Hungary, Estonia, Latvia, Malta. The values for Finland, Hungary, Estonia and Latvia refer to the national level (NUTS 0), which is indicated by the variable `NUTS_Level` = 0. In the cases of Cyprus, Malta and Iceland (which are rather small countries), the NUTS 0 level also corresponds to the NUTS 2 level, which is why they are marked here in the dataset with `NUTS_Level` = 2. It is comparatively difficult to compare the data with Belgium because COVID-19 death figures are only published there at NUTS 1 level (3 regions; `NUTS_Level` = 1).

Some countries report separately persons who died of/with COVID-19 who live outside the country or cannot be assigned to a region (e.g., Greece, Norway). These cases are shown separately in the dataset, but cannot be related to population numbers and cannot be linked to the NUTS shapefile.

Norway does not provide COVID-19 data for the NUTS3 regions NO0B1 and NO0B2.

In the UK, each country (England, Wales, Scotland, and Northern Ireland) is independently responsible for publishing COVID-19 data. Therefore the data are not all available at the same spatial aggregation level (e.g. England: NUTS 3, Wales: NUTS 2).

Source

Raw data of COVID-19 deaths:

Note: Some of the URLs have been moved or deleted.

AT: https://covid19-dashboard.ages.at/data/CovidFaelle_Timeline_GKZ.csv (accessed 2022-06-23)

BE: https://epistat.sciensano.be/Data/COVID19BE_MORT.csv (accessed 2022-06-21)

BG: https://www.nsi.bg/sites/default/files/files/data/table/COVID_2020_2021_EN.xls (accessed 2022-06-29)

CH: https://www.covid19.admin.ch/api/data/20220621-t6j901v4/downloads/sources-csv.zip (accessed 2022-06-21)

CY: https://covid19.who.int/region/euro/country/cy (accessed 2022-06-30)

CZ: https://onemocneni-aktualne.mzcr.cz/api/v2/covid-19/umrti.csv (accessed 2022-06-24)

DE: https://npgeo-corona-npgeo-de.hub.arcgis.com/datasets/917fc37a709542548cc3be077a786c17_0/about (accessed 2022-06-23)

DK: https://files.ssi.dk/covid19/overvagning/dashboard/overvaagningsdata-dashboard-covid19-28062022
(accessed 2022-06-29), folder: "Regionalt_DB", file: "07_antal_doede_pr_dag_pr_region"

EE: https://www.terviseamet.ee/en/coronavirus/coronavirus-dataset (accessed 2022-07-11)

EL: https://github.com/Sandbird/covid19-Greece (accessed 2022-07-02), file "regions"

ES: https://cnecovid.isciii.es/covid19/resources/casos_hosp_uci_def_sexo_edad_provres.csv (accesssed 2022-06-28)

FI: https://covid19.who.int/region/euro/country/fi (accessed 2022-07-01)

FR: https://www.data.gouv.fr/fr/datasets/synthese-des-indicateurs-de-suivi-de-lepidemie-covid-19/ (accessed 2022-07-01), file "table-indicateurs-open-data-dep-2022-06-30-19h00"

HR: https://www.koronavirus.hr/zupanije/139 (accessed 2022-06-28)

HU: https://covid19.who.int/region/euro/country/hu (accessed 2022-07-02)

IE: https://epi-covid-19-hpscireland.hub.arcgis.com/ (accessed 2022-06-29)

IS: https://www.covid.is/data (accessed 2022-06-27)

IT: https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni (accessed 2022-06-24), file "dpc-covid19-ita-regioni-latest_raw"

LV: https://covid19.gov.lv/en/node/16387 (accessed 2022-07-27)

LT: https://open-data-ls-osp-sdg.hub.arcgis.com/datasets/ba35de03e111430f88a86f7d1f351de6_0/explore (accessed 2022-06-27)

LU: https://covid19.public.lu/fr/graph.html (accessed 2022-06-27)

MT: https://covid19.who.int/table (accessed 2022-07-01)

NL: https://data.rivm.nl/covid-19/COVID-19_aantallen_gemeente_per_dag.csv (accessed 2022-06-27)

NO: https://www.fhi.no/contentassets/8a971e7b0a3c4a06bdbf381ab52e6157/vedlegg/2022/ukerapport-uke-20-16.05—22.05.22.pdf (accessed 2022-07-07)

PL: https://www.gov.pl/web/koronawirus/wykaz-zarazen-koronawirusem-sars-cov-2 (accessed 2022-06-23)

PT: https://github.com/dssg-pt/covid19pt-data/blob/master/data.csv (accessed 2022-06-29)

RO: https://covid19.geo-spatial.org/?map=decese (accessed 2022-07-01)

SE: https://experience.arcgis.com/experience/19fc7e3f61ec4e86af178fe2275029c5 (accessed 2022-06-23)

SI: https://www.nijz.si/sites/www.nijz.si/files/uploaded/tedenski_prikaz_umrli20220627.xlsx (accessed 2022-06-28)

SK: https://github.com/Institut-Zdravotnych-Analyz/covid19-data (accessed 2022-06-28), folder "Deaths", file "OpenData_Slovakia_Covid_Deaths_AgeGroup_District"

UK - England: https://coronavirus.data.gov.uk/details/deaths (accessed 2022-06-24), file "ltla_2022_06_23_cumDeaths60Da

UK - Northern Ireland: https://www.nisra.gov.uk/system/files/statistics/Weekly_Deaths%20-%20w%20e%2017th%20June%202022.XLSX (accessed 2022-07-01)

UK - Scotland: https://www.nrscotland.gov.uk/files//statistics/covid19/covid-deaths-22-data-week-25.xlsx (data for 2021-2022) and https://www.nrscotland.gov.uk/files//statistics/covid19/covid-deaths-20-data-final.xlsx (data for 2022) (accessed 2022-07-01)

UK - Wales: https://public.tableau.com/app/profile/public.health.wales.health.protection/viz/COVID-19Rapidmortalitydata/Summary (accessed 2022-07-04)

Population data:

https://ec.europa.eu/eurostat/databrowser/view/DEMO_R_PJANGRP3/default/table?lang=en&category=reg.reg_dem.reg_dempoar (accessed 2022-06-22)

Examples

```
data(C19dNUTSdata)

# Summary:
summary(C19dNUTSdata)

# Check for normal distribution:
hist(C19dNUTSdata$C19deaths_per100000)
shapiro.test(C19dNUTSdata$C19deaths_per100000)

# no. of regions for each country:
table(C19dNUTSdata$CNTR_CODE)
# only for countries with data on at least NUTS 2 level:
table(C19dNUTSdata[C19dNUTSdata$NUTS_Level > 1,]$CNTR_CODE)
```

`calculate_cum` *Cumulative Infection Numbers*

Description

Calculation of the cumulative values of infection numbers for infections panel data.

Usage

```
calculate_cum(
  object,
  col_name = NULL,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`col_name`	`character` value specifying the column name of the computed cumulative values
`overwrite`	`bool` argument which indicates whether the column should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Calculates the cumulative values of the infections panel data for all time points for each region. If `col_name` is `NULL`, the column is defined as "<Column name of cases>_cum". Set `overwrite` to TRUE, if an existing column should be overwritten. The method uses the function `cumsum` from the `base` package (see the corresponding documentation).

Value

`infpan` object including column with cumulative values in the infections panel data

Author(s)

Thomas Wieland

References

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

See Also

`infpan-class`, `load_infections_paneldata`, `calculate_Rt`, `calculate_incidence`, `calculate_rollmean`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

infpan_CH <- calculate_cum(
  infpan_CH,
  col_name = "cumulatives",
  verbose = TRUE
)
# Calculate rolling mean of cases as "cumulatives"

summary(infpan_CH)
# Summary of infpan object
```

`calculate_cum-methods` *Methods for Function* `calculate_cum`

Description

Methods for function `calculate_cum`

Methods

`signature( object = "infpan", col_name = NULL, overwrite = FALSE, verbose = FALSE)` Calculates the cumulative values of the infections panel data for all time points for each region. If `col_name` is `NULL`, the column is defined as "<Column name of cases>_cum". The method uses the function `cumsum` from the base package (see the corresponding documentation).

Author(s)

Thomas Wieland

`calculate_incidence` *Incidence from Infection Numbers*

Description

Calculation of the incidence from infection numbers and population for infections panel data.

Usage

```
calculate_incidence(
  object,
  use_column = NULL,
  col_name = NULL,
  pop_factor = 100000,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`use_column`	`character` value specifying which column should be used for incidence calculation
`col_name`	`character` value specifying the column name of the computed incidence
`pop_factor`	`numeric` value specifying the factor with which the incidence should be multiplied (e.g., cases/pop*100000)
`overwrite`	`bool` argument which indicates whether the column should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Calculates the incidence of the infections panel data for all time points for each region. Use `use_column` to specify which column should be used for the calculation of incidence. The following values are permitted: "Cases" (default, incremental cases), "Cum. cases" (cumulative cases), "Roll. mean" (rolling mean of cases), or "Roll. sum" (rolling sum of cases). If the specified column does not exist in the infections panel data of the `infpan` object, the function raises an error. If in the `infpan` object, no "Population" column is defined, incidence calculation is not possible. If `col_name` is `NULL`, the column is defined as "<Column name of cases>_inc". Set `overwrite` to TRUE, if an existing column should be overwritten.

Value

`infpan` object including column with incidence values in the infections panel data

Author(s)

Thomas Wieland

References

an der Heiden M, Hamouda O (2020) Schätzung der aktuellen Entwicklung der SARS-CoV-2-Epidemie in Deutschland - Nowcasting. *Epidemiologisches Bulletin* 17, 10-15. doi:10.25646/6692

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

See Also

`infpan-class`, `load_infections_paneldata`, `calculate_Rt`, `calculate_cum`, `calculate_rollmean`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

infpan_CH <- calculate_incidence(
  infpan_CH,
  col_name = "incidence",
  verbose = TRUE
)
# Calculate incidence of cases as "incidence"

summary(infpan_CH)
# Summary of infpan object
```

`calculate_incidence-methods` *Methods for Function* `calculate_incidence`

Description

Methods for function `calculate_incidence`

Methods

`signature( object, use_column = "Cases", col_name = NULL, pop_factor = 100000, overwrite = FALSE, verbose =`
: Calculates the incidence of the infections panel data for all time points for each region. Use `use_column` to specify which column should be used for the calculation of incidence. The following values are permitted: "Cases" (default, incremental cases), "Cum. cases" (cumulative cases), "Roll. mean" (rolling mean of cases), or "Roll. sum" (rolling sum of cases). If the specified column does not exist in the infections panel data of the `infpan` object, the function raises an error. If in the `infpan` object, no "Population" column is defined, incidence calculation is not possible. If `col_name` is `NULL`, the column is defined as "<Column name of cases>_inc". Set `overwrite` to TRUE, if an existing column should be overwritten.

Author(s)

Thomas Wieland

`calculate_rollmean` *Rolling Means of Infection Numbers*

Description

Calculation of the rolling means of infection numbers for infections panel data.

Usage

```
calculate_rollmean(
  object,
  k = 7,
  align = "center",
  fill = NA,
  col_name = NULL,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`k`	`integer` width of the rolling window (default: 7)
`align`	`character` specifying whether the rolling mean should be left- or right-aligned or centered (default) compared to the rolling window (default: center)
`fill`	`numeric` value or `NA` for the filling value at the left/within/right end of the data range
`col_name`	`character` value specifying the column name of the computed rolling means
`overwrite`	`bool` argument which indicates whether the column should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Calculates the rolling mean of the infections panel data for all time points for each region. Set the rolling window by the parameter `k` (default: 7). Set the fill value for the observations left/within/right to the data range with parameter `fill` (default: NA). Parameter `align` defines whether the index of the result should be left- or right-aligned or centered (default). If `col_name` is `NULL`, the column is defined as "<Column name of cases>_rm". Set `overwrite` to TRUE, if an existing column should be overwritten. The method uses the function `rollmean` from the `zoo` package (see the corresponding documentation).

Value

`infpan` object including column with rolling means in the infections panel data

Author(s)

Thomas Wieland

References

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

See Also

`infpan-class`, `load_infections_paneldata`, `calculate_rollsum`, `calculate_Rt`, `calculate_cum`, `calculate_incidence`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
```

```
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

infpan_CH <- calculate_rollmean(
  infpan_CH,
  col_name = "RollingMean",
  verbose = TRUE
)
# Calculate rolling mean of cases as "RollingMean"

summary(infpan_CH)
# Summary of infpan object
```

`calculate_rollmean-methods`

Methods for Function `calculate_rollmean`

Description

Methods for function `calculate_rollmean`

Methods

`signature( object = "infpan", k = 7, align = "center", fill = NA, col_name = NULL, verbose = FALSE)`
Calculates the rolling mean of the infections panel data for all time points for each region. Set the rolling window by the parameter `k` (default: 7). Set the fill value for the observations left/within/right to the data range with parameter `fill` (default: NA). Parameter `align` defines whether the index of the result should be left- or right-aligned or centered (default). If `col_name` is `NULL`, the column is defined as "<Column name of cases>_rm". The method uses the function `rollmean` from the `zoo` package (see the corresponding documentation).

Author(s)

Thomas Wieland

`calculate_rollsum` *Rolling Sums of Infection Numbers*

Description

Calculation of the rolling sums of infection numbers for infections panel data.

Usage

```
calculate_rollsum(
  object,
  k = 7,
  align = "center",
  fill = NA,
  col_name = NULL,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`k`	`integer` width of the rolling window (default: 7)
`align`	`character` specifying whether the rolling mean should be left- or right-aligned or centered (default) compared to the rolling window (default: center)
`fill`	`numeric` value or `NA` for the filling value at the left/within/right end of the data range
`col_name`	`character` value specifying the column name of the computed rolling sums
`overwrite`	`bool` argument which indicates whether the column should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Calculates the rolling sum of the infections panel data for all time points for each region. Set the rolling window by the parameter `k` (default: 7). Set the fill value for the observations left/within/right to the data range with parameter `fill` (default: NA). Parameter `align` defines whether the index of the result should be left- or right-aligned or centered (default). If `col_name` is `NULL`, the column is defined as "<Column name of cases>_rs". Set `overwrite` to TRUE, if an existing column should be overwritten. The method uses the function `rollsum` from the `zoo` package (see the corresponding documentation).

Value

`infpan` object including column with rolling sums in the infections panel data

Author(s)

Thomas Wieland

References

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

See Also

`infpan-class`, `load_infections_paneldata`, `calculate_rollmean`, `calculate_Rt`, `calculate_cum`, `calculate_incidence`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

infpan_CH <- calculate_rollsum(
  infpan_CH,
  col_name = "RollingMSum",
  verbose = TRUE
)
# Calculate rolling sum of cases as "RollingSum"

summary(infpan_CH)
# Summary of infpan object
```

`calculate_rollsum-methods` *Methods for Function* `calculate_rollsum`

Description

Methods for function `calculate_rollsum`

Methods

`signature( object = "infpan", k = 7, align = "center", fill = NA, col_name = NULL, verbose = FALSE)`
: Calculates the rolling sum of the infections panel data for all time points for each region. Set the rolling window by the parameter `k` (default: 7). Set the fill value for the observations left/within/right to the data range with parameter `fill` (default: NA). Parameter `align` defines whether the index of the result should be left- or right-aligned or centered (default). If `col_name` is `NULL`, the column is defined as "<Column name of cases>_rs". The method uses the function `rollsum` from the `zoo` package (see the corresponding documentation).

Author(s)

Thomas Wieland

`calculate_Rt` *Effective Reproduction Number*

Description

Calculation of the effective reproduction number for infections panel data.

Usage

```
calculate_Rt(
  object,
  GP = 4,
  correction = FALSE,
  col_name = NULL,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`GP`	Generation period, in time units (typically days)
`correction`	Correction of values equal to zero? (Recommended)
`col_name`	`character` value specifying the column name of the computed rolling means
`overwrite`	`bool` argument which indicates whether the column should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Calculates the effective reproduction number R_t for all time points for each region in the infections panel data. Set the generation period by the parameter `GP` (default: 4). If `correction` is TRUE, values equal to zero are increased by one. The method uses the built-in function `R_t()`.

Value

`infpan` object including R_t column in the infections panel data

Author(s)

Thomas Wieland

References

an der Heiden M, Hamouda O (2020) Schätzung der aktuellen Entwicklung der SARS-CoV-2-Epidemie in Deutschland - Nowcasting. *Epidemiologisches Bulletin* 17, 10-15. doi:10.25646/6692

Bonifazi G, Lista L, Menasce D, Mezzetto M, Pedrini D, Spighi R, Zoccoli A (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

See Also

`infpan-class`, `load_infections_paneldata`, `calculate_incidence`, `calculate_cum`, `calculate_rollmean`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

infpan_CH <- calculate_Rt(
  infpan_CH,
  verbose = TRUE
  )
# Calculate effective reproduction number

summary(infpan_CH)
# Summary of infpan object
```

`calculate_Rt-methods` *Methods for Function* `calculate_Rt`

Description

Methods for function `calculate_Rt`

Methods

`signature( object = "infpan", GP = 4, correction = FALSE, col_name = NULL, verbose = FALSE)`

Calculates the effective reproduction number R_t for all time points for each region in the infections panel data. Set the generation period by the parameter `GP` (default: 4). If `correction` is TRUE, values equal to zero are increased by one. Set `overwrite` to TRUE, if an existing column should be overwritten. The method uses the built-in function `R_t()`.

Author(s)

Thomas Wieland

`compare_countries` *Two-country Comparison of Swash-Backwash Model Parameters*

Description

This function enables bootstrap estimates for the mean difference of Swash-Backwash Model parameters of two countries to be compared.

Usage

```
compare_countries(
  sbm1,
  sbm2,
  country_names = c("Country 1", "Country 2"),
  indicator = "R_0A",
  iterations = 20,
  samples_ratio = 0.8,
  alpha = 0.05,
  replace = TRUE
  )
```

Arguments

`sbm1`	A `sbm` object for country 1
`sbm2`	A `sbm` object for country 2
`country_names`	`list` with user-given country names (two entries)
`indicator`	`character`, indicator to be analyzed (`"S_A"`, `"I_A"`, `"R_A"`, `"t_LE"`, `"t_LE"`, or `"R_0A"` (default and recommended: `"R_0A"`))
`iterations`	Number of iterations for resampling (default: 100)
`samples_ratio`	Proportion of regions included in each sample (default: 0.8)
`alpha`	Significance level α for 1-α*100 confidence intervals (default: 0.05)
`replace`	Resampling with replacement (`TRUE` or `FALSE`, default: `TRUE` = bootstrap resampling)

Details

The combination of the Swash-Backwash Model and bootstrap resampling allows the estimation of mean differences of a user-specified model parameter (e.g., spatial reproduction number R_{OA}) between two countries. This makes it possible to check whether the spatial spread velocity of a communicable disease is significantly different in one country than in another country. Since the initial data in the Swash-Backwash Model should be balanced, *entity-based bootstrap sampling* is carried out in the `compare_countries()` function. This means that not, for example, 80% of all observations are included in each sample at a sample ratio equal to p = 0.8, but rather all observations for 80% of the regions. For both countries, B bootstrap samples (default: 100) are drawn for which the Swash-Backwash Model is calculated. Based on the distribution of indicators, confidence intervals are calculated at the user-specified significance level α. The `compare_countries()` function calculates the differences of the user's desired indicator between the two samples, D, and also calculates α confidence intervals for this.

Value

object of class `countries`, see `countries-class`

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap.

Ramachandran KM, Tsokos CP (2021) Mathematical Statistics with Applications in R (Third Edition). Ch. 13.3.1 (Bootstrap confidence intervals). doi:10.1016/B9780128178157.000130

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

See Also

`swash`, `countries-class`

Examples

```
data(COVID19Cases_geoRegion)
# Get Swiss COVID19 cases at NUTS 3 level

data(Oesterreich_Faelle)
# Get Austrian COVID19 cases at NUTS 3 level
# (first wave, same final date as in Swiss data: 2020-05-31)

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

CH_covidwave1 <-
  swash_backwash(
    data = COVID19Cases_geoRegion,
```

```
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion"
    )
# Swash-Backwash Model for Swiss COVID19 cases
# Spatial aggregate: NUTS 3 (cantons)

AT_covidwave1 <-
  swash_backwash(
    data = Oesterreich_Faelle,
    col_cases = "Faelle",
    col_date = "Datum",
    col_region = "NUTS3"
  )
# Swash-Backwash Model for Austrian COVID19 cases
# Spatial aggregate: NUTS 3

AT_vs_CH <-
  compare_countries(
    CH_covidwave1,
    AT_covidwave1,
    country_names = c("Switzerland", "Austria"))
# Country comparison Switzerland vs. Austria
# default config: 20 iterations, alpha = 0.05, sample ratio = 80%,
# indicator: R_0A

summary(AT_vs_CH)
# Summary of country comparison

plot(AT_vs_CH)
# Plot of country comparison
```

`confint-methods` *Methods for Function* `confint`

Description

Methods for function `confint`

Methods

`signature(object = "sbm", iterations = 100, samples_ratio = 0.8, alpha = 0.05, replace = TRUE)`
: Creates bootstrap confidence intervals for `sbm` objects. The argument `iterations` indicates the number of bootstrap samples which are drawn. Since the initial data in the Swash-Backwash Model should be balanced, *entity-based bootstrap sampling* is carried out. This means that not, for example, 80% of all observations are included in each sample at a sample ratio equal to $p = 0.8$ (`samples_ratio = 0.8`), but rather all observations for 80% of the regions. The significance level for the confidence intervals α is set by the argument `alpha` (default: 0.05, which corresponds to a 95% confidence level).

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap.

Ramachandran KM, Tsokos CP (2021) Mathematical Statistics with Applications in R (Third Edition). Ch. 13.3.1 (Bootstrap confidence intervals). doi:10.1016/B9780128178157.000130

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

See Also

`sbm_ci-class`

`countries-class` *Class* `"countries"`

Description

The class `countries` contains the results of a two-country comparison analysis using the Swash-Backwash Model, including two `sbm_ci` objects for each country. Use `summary(countries)` and `plot(countries)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by the function `compare_countries(sbm1, sbm2)`.

Slots

`sbm_ci1`**:** Object of class `"sbm_ci"` Results of `"confint(sbm1)"` for country 1

`sbm_ci2`**:** Object of class `"sbm_ci"` Results of `"confint(sbm1)"` for country 2

`D`**:** Object of class `"numeric"` Results: Difference D between the samples with respect to the chosen indicator

`D_ci`**:** Object of class `"numeric"` Results: α confidence intervals of D

`config`**:** Object of class `"list"` Configuration details for bootstrap sampling

`country_names`**:** Object of class `"character"` User-stated country names

`indicator`**:** Object of class `"character"` User-stated indicator to be tested

Methods

- **plot** `signature(x = "countries")`: Plots the results of a two-country comparison with the Swash-Backwash Model
- **show** `signature(object = "countries")`: Prints an `countries` object; use `summary(sbm_ci)` for results
- **print** `signature(object = "countries")`: Prints an `countries` object; use `summary(sbm_ci)` for results
- **summary** `signature(object = "countries")`: Prints a summary of a `countries` object (results of the two-country comparison)

Author(s)

Thomas Wieland

Examples

```
showClass("countries")
```

`COVID19Cases_geoRegion`

Switzerland Daily COVID-19 cases by region

Description

A dataset containing COVID-19 cases by region (NUTS 3 = cantons) and time periods (days) for Switzerland (Source: Federal Office of Public Health FOPH).

Usage

```
data(COVID19Cases_geoRegion)
```

Format

A data.frame with multiple columns:

geoRegion	(character) Region for which the data was collected.
datum	(Date) Date of record.
entries	(integer) Number of reported cases on this date.
sumTotal	(integer) Cumulative case numbers.
timeframe_14d	(logical) Indicates whether the time period covers the last 14 days.
timeframe_all	(logical) Indicates whether the time period covers all previous data.
offset_last7d	(integer) Offset of the last 7 days.
sumTotal_last7d	(integer) Cumulative case numbers of the last 7 days.
offset_last14d	(integer) Offset of the last 14 days.

sumTotal_last14d (integer) Cumulative case numbers of the last 14 days.

offset_last28d (integer) Offset of the last 28 days.

sumTotal_last28d (integer) Cumulative case numbers of the last 28 days.

sum7d (numeric) Sum of the last 7 days.

sum14d (numeric) Sum of the last 14 days.

mean7d (numeric) Average of the last 7 days.

mean14d (numeric) Average of the last 14 days.

entries_diff_last_age (integer) Difference from the last age group.

pop (integer) Population of the region.

inz_entries (numeric) Incidence of the entries.

inzsumTotal (numeric) Incidence of cumulative cases.

inzmean7d (numeric) Incidence of the 7-day average.

inzmean14d (numeric) Incidence of the 14-day average.

inzsumTotal_last7d (numeric) Incidence of cumulative cases in the last 7 days.

inzsumTotal_last14d (numeric) Incidence of cumulative cases in the last 14 days.

inzsumTotal_last28d (numeric) Incidence of cumulative cases in the last 28 days.

inzsum7d (numeric) Incidence of the last 7 days.

inzsum14d (numeric) Incidence of the last 14 days.

sumdelta7d (numeric) Difference in sums of the last 7 days.

inzdelta7d (numeric) Difference in incidence of the last 7 days.

type (character) Type of recorded data (e.g., COVID-19 cases).

type_variant (character) Variant of the data type.

version (character) Version of the data collection.

datum_unit (character) Unit of date specification (e.g., day).

entries_letzter_stand (integer) Last known count of entries.

entries_neu_gemeldet (integer) Newly reported entries.

entries_diff_last (integer) Difference in last entries.

Details

The data is included as it was published in by the Swiss Federal Office of Public Health (Bundesamt fuer Gesundheit, BAG). Note that the reporting date equals the date of SARS-CoV-2 testing.

Source

Federal Office of Public Health - FOPH (2023) COVID-19 Dashboard Source Data. `https://www.covid19.admin.ch/api/data/documentation` (retrieved 2023-06-28)

References

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

COVID19Cases_geoRegion_balanced <-
  is_balanced(
  data = COVID19Cases_geoRegion,
  col_cases = "entries",
  col_date = "datum",
  col_region = "geoRegion"
)
# Test whether "COVID19Cases_geoRegion" is balanced panel data

COVID19Cases_geoRegion_balanced$data_balanced
# Balanced? TRUE or FALSE
```

`did_fatalities_splm_coef` *Results from a Difference-in-Differences Model*

Description

Example data frame with results from a difference-in-differences model

Usage

```
data(did_fatalities_splm_coef)
```

Format

A `data.frame` with multiple columns:

Var Coefficient name

Estimate Coef. estimate

Std_Error_Bonferroni Coef. standard error

t_value_Bonferroni Coef. t value

Pr_t_Bonferroni Coef. p value

CI_lower_Bonferroni Coef. lower confidence interval

CI_upper_Bonferroni Coef. upper confidence interval

Details

Data frame with results from a difference-in-differences model (SPLM model), example data

Source

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

Examples

```
data(did_fatalities_splm_coef)
# Results of a difference-in-differences model

plot_coef_ci(
  point_estimates = did_fatalities_splm_coef$Estimate,
  confint_lower = did_fatalities_splm_coef$CI_lower_Bonferroni,
  confint_upper = did_fatalities_splm_coef$CI_upper_Bonferroni,
  coef_names = did_fatalities_splm_coef$Var,
  skipvars = c(
    "Alpha_share",
    "lambda",
    "rho",
    "log(D_Infections_daily_7dsum_per100000_lag2weeks)",
    "vacc_cum_per100000_lag2weeks"
    ),
  lwd = 13,
  pch = 19,
  auto_color = TRUE
)
# Plot with point estimates and confidence intervals
```

`expgrowth-class` *Class* "expgrowth"

Description

The class "expgrowth" contains the results of the exponential_growth() function. Use `summary(expgrowth)` for results summary.

Objects from the Class

Objects can be created by the function `exponential_growth`.

Slots

- `GrowthModel_OLS`**:** Object of class `list` Results of the OLS fit (predicted, parameters)
- `GrowthModel_NLS`**:** Object of class `list` Results of the NLS fit (predicted, parameters)
- `t`**:** Object of class `numeric` Input time points data
- `y`**:** Object of class `numeric` Input infections data
- `config`**:** Object of class `list` Model fit configurations

Methods

- **summary** `signature(object = "expgrowth")`: Prints a summary of `expgrowth` objects
- **plot** `signature(x = "expgrowth")`: Plots the results of the exponential growth model (observed, predicted)
- **print** `signature(x = "expgrowth")`: Prints an `expgrowth` object; use `summary(expgrowth)` for results

Author(s)

Thomas Wieland

References

Bonifazi G, Lista L, Menasce D, Mezzetto M, Pedrini D, Spighi R, Zoccoli A (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Chowell G, Viboud C, Hyman JM, Simonsen L (2015) The Western Africa ebola virus disease epidemic exhibits both global exponential and local polynomial growth rates. *PLOS Currents Outbreaks*, ecurrents.outbreaks.8b55f4bad99ac5c5db3663e916803261. doi:10.1371/currents.outbreaks.8b55f4bad99ac5c5db36

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Examples

```
showClass("expgrowth")
```

`exponential_growth` *Exponential Growth Model for Epidemic Data*

Description

Estimation of exponential growth models from daily infections data

Usage

```
exponential_growth(
  y,
  t,
  GI = 4,
  nls = TRUE,
  nls_start = list(a = 1, b = 0.1),
  add_constant = 1,
  verbose = FALSE
  )
```

Arguments

`y`	`numeric` `vector` with cumulative infections data over time
`t`	`vector` of class `numeric` or `Date` with time points or dates
`GI`	Generation interval for computing R_0
`nls`	Nonlinear estimation? `TRUE` or `FALSE`
`nls_start`	A `list` with start values for the two parameters to be estimated
`add_constant`	Numeric constant to be added to `y` if zero values occur (only relevant for OLS estimation)
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

This function allows the estimation of an exponential growth model. The user must specify the dependent variable (e.g., daily infections) and the time variable (time counter or date values). The estimation is performed using a linearized model as an OLS estimator, and, if `nls=TRUE`, also by NLS. The results are the exponential growth rate r, basic reproduction number R_0, and the doubling rate.

Value

object of class `expgrowth-class`

Author(s)

Thomas Wieland

References

Bonifazi G, Lista L, Menasce D, Mezzetto M, Pedrini D, Spighi R, Zoccoli A (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Chowell G, Viboud C, Hyman JM, Simonsen L (2015) The Western Africa ebola virus disease epidemic exhibits both global exponential and local polynomial growth rates. *PLOS Currents Outbreaks*, ecurrents.outbreaks.8b55f4bad99ac5c5db3663e916803261. doi:10.1371/currents.outbreaks.8b55f4bad99ac5c5db36

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

See Also

`growth_initial`, `logistic_growth`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_ZH <-
  COVID19Cases_geoRegion[
  (COVID19Cases_geoRegion$geoRegion == "ZH")
  & (COVID19Cases_geoRegion$sumTotal > 0)
  ,]
# COVID cases for Zurich

expgrowth_ZH <- exponential_growth(
  y = COVID19Cases_ZH$sumTotal[1:28],
  t = COVID19Cases_ZH$datum[1:28]
)
# Exponential growth model for the first 4 weeks

summary(expgrowth_ZH)
# Summary of exponential growth model

plot(expgrowth_ZH)
# Plot of exponential growth model

expgrowth_ZH@GrowthModel_OLS$fit_metrics
expgrowth_ZH@GrowthModel_OLS$fit_metrics
# Fit metrics for OLS and NLS models
```

`growth` *Logistic Growth Models for Regional Infections*

Description

Estimates N logistic growth models for N regions.

Usage

```
growth(
  object,
  S_iterations = 10,
  S_start_est_method = "bisect",
  seq_by = 10,
  nls = TRUE,
  add_constant = 1,
  overwrite = FALSE,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`S_iterations`	Number of iterations for saturation value search
`S_start_est_method`	Method for saturation value search, either "bisect" or "trial_and_error"
`seq_by`	No. of segments for the "trial_and_error" estimation of the saturation value
`nls`	Nonlinear estimation? `TRUE` or `FALSE`
`add_constant`	Numeric constant to be added to y if zero values occur (only relevant for OLS estimation)
`overwrite`	`bool` argument which indicates whether the column containing cumulative cases should be overwritten if already existing
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The function estimates logistic growth models for regional infections based on a `infpan` object. See `logistic_growth` for further details.

Value

object of class `growthmodels-class`

Author(s)

Thomas Wieland

References

Chowell G, Simonsen L, Viboud C, Yang K (2014) Is West Africa Approaching a Catastrophic Phase or is the 2014 Ebola Epidemic Slowing Down? Different Models Yield Different Answers for Liberia. *PLoS currents 6*. doi:10.1371/currents.outbreaks.b4690859d91684da963dc40e00f3da81

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

See Also

`logistic_growth`, `exponential_growth`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1_growth <-
  growth(infpan_CH)
summary(CH_covidwave1_growth)
# Logistic growth models for infpan object infpan_CH
```

`growth-methods` *Methods for Function* `growth`

Description

Methods for function `growth`

Methods

`signature(object = "infpan", S_iterations = 10, S_start_est_method = "bisect", seq_by = 10, nls = TRUE, add`
: Estimation of N logistic growth models for N regions. Both OLS and NLS estimation are estimated by default (set `nls = FALSE` to skip NLS estimation). Parameters `S_iterations`, `S_start_est_method`, and `seq_by` are used to control the saturation value estimation (see `logistic_growth`).

Author(s)

Thomas Wieland

`growthmodels-class` *Class* `"growthmodels"`

Description

The class `"growthmodels"` contains the results of growth model analyses and the related input data as well as additional information. The `swash` package includes the following model analyses under the heading "growth models": Exponential growth models, logistic growth models, Hawkes Process models, and time series models with breakpoints. Use `summary(growthmodels)` for results summary. See the corresponding functions for details: exponential_growth, logistic_growth, hawkes_growth, breaks_growth.

Objects from the Class

Objects can be created by the functions `exponential_growth`, `logistic_growth`, `breaks_growth`, or `hawkes_growth`.

Slots

`results`**:** Object of class `"data.frame"` Model results as a table with coefficents, fit metrics, etc.

`growth_models`**:** Object of class `"list"` containing all models

`model_type`**:** Object of class `"character"` describing the type of model

`results_cols`**:** Object of class `"character"` Vector with column names containing results

`results_cols_names`**:** Object of class `"character"` Vector with descriptions of the column names

`data_statistics`**:** Object of class `"numeric"` Diagnostics of input data

`time_format`**:** Object of class `"character"` Format of time points in time column

`timestamp`**:** Object of class `"list"` Time stamps of any update of the instance

Methods

print `signature(x = "growthmodels")`: Prints a `growthmodels` object; use `summary(growthmodels)` for results

show `signature(object = "growthmodels")`: Prints a `growthmodels` object; use `summary(growthmodels)` for results

summary `signature(object = "growthmodels")`: Prints a summary of `growthmodels` objects (model results)

Author(s)

Thomas Wieland

References

Bai J, Perron P (2003) Computation and analysis of multiple structural change models. *Journal of Applied Econometrics* 18(1), 1-22. doi:10.1002/jae.659

Bonifazi G et al. (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Chowell G, Simonsen L, Viboud C, Yang K (2014) Is West Africa Approaching a Catastrophic Phase or is the 2014 Ebola Epidemic Slowing Down? Different Models Yield Different Answers for Liberia. *PLoS currents* 6. doi:10.1371/currents.outbreaks.b4690859d91684da963dc40e00f3da81

Chowell G, Viboud C, Hyman JM, Simonsen L (2015) The Western Africa ebola virus disease epidemic exhibits both global exponential and local polynomial growth rates. *PLOS Currents Outbreaks*, ecurrents.outbreaks.8b55f4bad99ac5c5db3663e916803261. doi:10.1371/currents.outbreaks.8b55f4bad99ac5c5db36

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L. (2018) SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In: *Proceedings of the 2018 World Wide Web Conference. WWW'18*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 419–428. doi:10.1145/3178876.3186108

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Zeileis C, Kleiber W, Krämer K, Hornik, K (2003) Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis* 44(1-2), 109-123. doi:10.1016/S01679473(03)00030-6

Examples

```
showClass("growthmodels")
```

`growth_breaks` *Time Series Model with Breakpoints for Regional Infections*

Description

Conducts N breakpoints analyses for infection time series in N regions.

Usage

```
growth_breaks(
  object,
  ln = FALSE,
  add_constant = 1,
  alpha = 0.05,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`ln`	`bool` argument which indicates whether the dependent variable should be transformed by natural logarithm
`add_constant`	Numeric constant to be added to y if zero values occur
`alpha`	Significance level α for 1-α*100 confidence intervals
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The method detects breakpoints in regional infections time series based on an `infpan` object. The function internally uses the function `breakpoints` from the `strucchange` package (Zeileis et al. 2003), where breakpoints are identified using the Bai-Perron algorithm (Bai & Perron 2003). See `breaks_growth` for further details of the estimation.

Value

object of class `growthmodels-class`

Author(s)

Thomas Wieland

References

Bai J, Perron P (2003) Computation and analysis of multiple structural change models. *Journal of Applied Econometrics* 18(1), 1-22. doi:10.1002/jae.659

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Zeileis C, Kleiber W, Krämer K, Hornik, K (2003) Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis* 44(1-2), 109-123. doi:10.1016/S01679473(03)00030-6

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1_breaks <-
  growth_breaks(infpan_CH)
summary(CH_covidwave1_breaks)
# Breakpoints for infpan object infpan_CH
```

`growth_breaks-methods` *Methods for Function* `growth_breaks`

Description

Methods for function `growth_breaks`

Methods

`signature(object = "infpan", ln = FALSE, add_constant = 1, alpha = 0.05, verbose = FALSE)`
Estimation of N breakpoint analyses for infections panel data for N regions. For details, see `breaks_growth`.

Author(s)

Thomas Wieland

`growth_hawkes` *Hawkes Processs models for Regional Infections*

Description

Estimates N Hawkes process models for N regions.

Usage

```
growth_hawkes(
  object,
  optim_method = "L-BFGS-B",
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`optim_method`	`character` value for the optimization method. Passed to argument `method` in `stats` function `optim()`
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The function estimates Hawkes process models for regional infections based on an `infpan` object. See `hawkes_growth` for further details of the estimation.

Value

object of class `growthmodels-class`

Author(s)

Thomas Wieland

References

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L. (2018) SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In: *Proceedings of the 2018 World Wide Web Conference. WWW'18*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 419–428. doi:10.1145/3178876.3186108

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1_Hawkes <-
  growth_hawkes(infpan_CH)
summary(CH_covidwave1_Hawkes)
# Hawkes process models for infpan object infpan_CH
```

`growth_hawkes-methods` *Methods for Function* `growth_hawkes`

Description

Methods for function `growth_hawkes`

Methods

`signature(object = "infpan", optim_method = "L-BFGS-B", verbose = FALSE)` Estimation of N Hawkes process models for N regions. Set argument `optim_method` for using the optimization method from `stats::optim`.

Author(s)

Thomas Wieland

`growth_initial` *Exponential Growth Models for Regional Infections*

Description

Estimates N exponential growth models for a given time period in N regions.

Usage

```
growth_initial(
  object,
  time_units = 10,
  GI = 4,
  nls = TRUE,
  nls_start = list(a = 1, b = 0.1),
  add_constant = 1,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`time_units`	`numeric` value for the analysis time (time units from start)
`GI`	Generation interval for computing R_0
`nls`	Nonlinear estimation? `TRUE` or `FALSE`
`nls_start`	A `list` with start values for the two parameters to be estimated
`add_constant`	Numeric constant to be added to y if zero values occur (only relevant for OLS estimation)
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The method estimates exponential growth models for regional infections based on an `infpan` object. Such models are design for the analysis of the *initial* phase of an epidemic spread. The user must state how much time units (from start) are included. See `exponential_growth` for further details of the estimation.

Value

object of class `growthmodels-class`

Author(s)

Thomas Wieland

References

Bonifazi G et al. (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Chowell G, Viboud C, Hyman JM, Simonsen L (2015) The Western Africa ebola virus disease epidemic exhibits both global exponential and local polynomial growth rates. *PLOS Currents Outbreaks*, ecurrents.outbreaks.8b55f4bad99ac5c5db3663e916803261. doi:10.1371/currents.outbreaks.8b55f4bad99ac5c5db36

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1_initialgrowth_3weeks <-
  growth_initial(
    infpan_CH,
    time_units = 21
  )
summary(CH_covidwave1_initialgrowth_3weeks)
# Exponential models for infpan object CH_covidwave1
# initial growth in the first 3 weeks
```

`growth_initial-methods`

Methods for Function `growth_initial`

Description

Methods for function `growth_initial`

Methods

`signature(object = "infpan", time_units = 10, GI = 4, nls = TRUE, nls_start = list(a = 1, b = 0.1), add_const`
: Estimation of N exponential growth models for the initial phase of an epidemic spread for N regions. Set argument `GI` for the calculation of the basic reproduction number, and control OLS/NLS estimation with arguments `nls`, `nls_start`, and `add_constant` (see `exponential_growth`).

Author(s)

Thomas Wieland

`hawkes-class` *Class* `"hawkes"`

Description

The class `"hawkes"` contains the results of the hawkes_growth() function. Use `summary(hawkes)` for results summary.

Objects from the Class

Objects can be created by the function `hawkes_growth`.

Slots

`t`: Object of class `numeric` Input time points data

`y`: Object of class `numeric` Input infections data

`mu`: Object of class `numeric` Estimated μ parameter (background rate)

`alpha`: Object of class `numeric` Estimated α parameter (excitation parameter)

`beta`: Object of class `numeric` Estimated β parameter (decay)

`br`: Object of class `numeric` Estimated breaking ratio (α/β)

`y_pred`: Object of class `numeric` Predicted values of y

`fit_metrics`: Object of class `list` Fit metrics for model, output from built-in function `fit_metrics`

`config`: Object of class `list` Model fit configurations

Methods

summary `signature(object = "hawkes")`: Prints a summary of hawkes objects

print `signature(x = "hawkes")`: Prints an hawkes object; use `summary(hawkes)` for results

plot `signature(x = "hawkes")`: Plots the results of the Hawkes model (observed, predicted)

Author(s)

Thomas Wieland

References

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L. (2018) SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In: *Proceedings of the 2018 World Wide Web Conference. WWW'18*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 419–428. doi:10.1145/3178876.3186108

Examples

```
showClass("hawkes")
```

`hawkes_growth` *Hawkes Process Model for Epidemic Data*

Description

Estimation of Hawkes Process models from incremental infections data

Usage

```
hawkes_growth(
  y,
  optim_method = "L-BFGS-B",
  verbose = FALSE
  )
```

Arguments

y	`numeric vector` with incremental infections data over time (e.g., daily infections)
optim_method	`character` specifying the optimization algorithm, passed to `stats::optim`
verbose	`bool` argument which indicates whether progress messages are displayed

Details

This function allows the estimation of a Hawkes Process model, with the time decay being expressed as exponential function, which results in three estimated parameters: μ (background rate), α (excitation parameter), and β (decay). The user must specify the dependent variable (incremental infections). The estimation is performed using nonlinear estimation via `stats::optim`. See the corresponding documentation for available optimization methods (default: "L-BFGS-B").

Value

object of class `hawkes-class`

Author(s)

Thomas Wieland

References

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L. (2018) SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In: *Proceedings of the 2018 World Wide Web Conference. WWW'18*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, p. 419–428. doi:10.1145/3178876.3186108

See Also

logistic_growth, exponential_growth, breaks_growth

Examples

```
data(Infections)
# Confirmed SARS-CoV-2 cases in Germany

hawkes_BS <- hawkes_growth(
  y = Infections$infections_daily
)
# Hawkes Process model

summary(hawkes_BS)
# Summary of Hawkes model estimates

plot(hawkes_BS)
# Plot of Hawkes Process model
```

`hist_ci` *Creating Histograms with Confidence Intervals*

Description

Plot of a histogram of a given vector `x` and the related confidence intervals (`lower`, `upper`).

Usage

```
hist_ci(
  x,
  alpha = 0.05,
  col_bars = "grey",
  col_ci = "red",
  ...
  )
```

Arguments

`x`	A numeric vector
`alpha`	Significance level α for 1-α*100 confidence intervals
`col_bars`	Color of bars in histogram
`col_ci`	Color of lines for confidence interval
`...`	Additional arguments passed to `barplot()`

Details

Helper function for `plot(sbm_ci)`, but may be used separately.

Value

Histogram plot, no returned value

Author(s)

Thomas Wieland

Examples

```
numeric_vector <- c(1,9,5,6,3,10,20,6,9,14,3,5,8,6,11)
# any numeric vector

hist_ci(numeric_vector)
```

`Infections` *Infections*

Description

Daily SARS-CoV-2 infection in Germany spring 2020

Usage

```
data(Infections)
```

Format

A data.frame with multiple columns:

infectedtest_CW2	Calendar week 2020 of conducted test
infection_date	Estimated date of infection
infections_daily	Daily infections
infections_daily_lwr	Daily infections lower confidence interval
infections_daily_upr	Daily infections upper confidence interval

infections_cum Cumulative infections

infections_cum_lwr Cumulative infections lower confidence interval

infections_cum_upr Cumulative infections upper confidence interval

R4 Estimated effective reproduction number R_t with generation interval = 4

R4_lwr Estimated effective reproduction number R_t with generation interval = 4 lower confidence interval

R4_upr Estimated effective reproduction number R_t with generation interval = 4 upper confidence interval

R7 Estimated effective reproduction number R_t with generation interval = 7

R7_lwr Estimated effective reproduction number R_t with generation interval = 7 lower confidence interval

R7_upr Estimated effective reproduction number R_t with generation interval = 7 upper confidence interval

onsets_of_symptoms Daily onsets of symptoms

onsets_of_symptoms_lwr Daily onsets of symptoms lower confidence interval

onsets_of_symptoms_upr Daily onsets of symptoms upper confidence interval

reported_cases Daily reported cases

day Time counter (day)

ln_inf_cum Nat. log. of cumulative infections

ln_inf_daily Nat. log. of daily infections

ln_R4 Nat. log. of estimated effective reproduction number R_t with generation interval = 4

ln_R7 Nat. log. of estimated effective reproduction number R_t with generation interval = 7

infection_date_CW Calendar week of infection data (numeric)

infection_date_CW2 Calendar week of infection data (categorical)

infectedtest_CW Calendar week of conducted test

conducted_tests No. of conducted tests

negative_tests No. of negative tests

positive_tests No. of positive tests

positive_tests_share Share of positive tests (average per day)

conducted_tests_index No. of conducted tests (average per day), index (CW 14 = 100)

conducted_tests_dailyaverage No. of conducted tests, average per day

positive_tests_dailyaverage Positive tests, average per day

infections_daily_testweighted Daily infections weighted by test volume

ln_inf_daily_tw Nat. log. of daily infections weighted by test volume

Details

Example data with daily SARS-CoV-2 infections in Germany. See Wieland (2020) for data sources and method of backdating infections.

Source

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

Examples

```
data(Infections)
```

`infpan-class` *Class* `"infpan"`

Description

The class `"infpan"` contains infections panel data for N regions and T time points as well as additional information. Use `summary(infpan)` and `plot(infpan)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by importing infections panel data using the function `load_infections_paneldata`.

Slots

`input_data`**:** Object of class `"data.frame"` Model result: Input infections panel data

`data_statistics`**:** Object of class `"numeric"` Data statistics (N regions, T time points, test whether data is balanced, etc.)

`index_col_names`**:** Object of class `"character"` Column names of regions and time points

`cases_col_name`**:** Object of class `"character"` Column name of incremental cases

`other_cols`**:** Object of class `"character"` Names of other relevant columns derived from incremental case data, e.g. effective reproduction number R_t

`time_format`**:** Object of class `"character"` Format of time points in time column

`time_unit`**:** Object of class `"character"` Time unit, default: "days"

`timestamp`**:** Object of class `"list"` Time stamps of any update of the instance

Methods

plot `signature(x = "infpan")`: Plots case data by region for N regions and T time points

calculate_Rt `signature(x = "infpan")`: Calculates the effective reproduction number R_t from `infpan` objects. Returns updated `infpan` instance.

calculate_cum `signature(x = "infpan")`: Calculates cumulative cases from `infpan` objects. Returns updated `infpan` instance.

calculate_rollmean `signature(x = "infpan")`: Calculates rolling means of cases from `infpan` objects. Returns updated `infpan` instance.

calculate_rollsum `signature(x = "infpan")`: Calculates rolling sums of cases from `infpan` objects. Returns updated `infpan` instance.

calculate_incidence `signature(x = "infpan")`: Calculates incidences of cases from `infpan` objects. Returns updated `infpan` instance.

print `signature(x = "infpan")`: Prints an `infpan` object; use `summary(infpan)` for results

show `signature(object = "infpan")`: Shows an `infpan` object; use `summary(infpan)` for results

summary `signature(object = "infpan")`: Prints a summary of `infpan` objects

swash `signature(object = "infpan")`: Performs a Swash-Backwash Model analysis from `infpan` objects. Returns `sbm` instance.

growth `signature(object = "infpan")`: Estimates logistic growth models from `infpan` objects. Returns `growthmodels` instance.

growth_initial `signature(object = "infpan")`: Estimates exponential growth models from `infpan` objects for a given time period. Returns `growthmodels` instance.

growth_hawkes `signature(object = "infpan")`: Estimates Hawkes process models from `infpan` objects. Returns `growthmodels` instance.

breaks_growth `signature(object = "infpan")`: Estimates breakpoints in linear regression models from `infpan` objects. Returns `growthmodels` instance.

Author(s)

Thomas Wieland

References

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

Examples

```
showClass("infpan")
```

`is_balanced` *Test whether Panel Dataset with Regional Infection Data is Balanced*

Description

The function tests whether the input panel data with regional infections is balanced.

Usage

```
is_balanced(
  data,
  col_cases,
  col_date,
  col_region,
  as_balanced = TRUE,
  fill_missing = 0
  )
```

Arguments

`data`	`data.frame` with regional infection data
`col_cases`	Column containing the cases (numeric)
`col_date`	Column containing the time points (e.g., days)
`col_region`	Column containing the unique identifier of the regions (e.g., name, NUTS 3 code)
`as_balanced`	Boolean argument which indicates whether non-balanced panel data shall be balanced (default: TRUE)
`fill_missing`	Constant to fill missing values (default and recommended: 0)

Details

The Swash-Backwash Model for the Single Epidemic Wave does not necessarily require balanced panel data in order for the calculations to be carried out. However, for a correct estimation it is implicitly assumed that the input data is balanced. The function tests whether the panel data is balanced. It is executed automatically whithin the `swash()` function (using automatic correction with `as_balanced = TRUE`), but can also be used separately.

Value

List with two entries:

`data_balanced`	Result of test (`TRUE` or `FALSE`)
`data`	Input dataset (`data.frame`)

Author(s)

Thomas Wieland

See Also

`as_balanced`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

COVID19Cases_geoRegion_balanced <-
  is_balanced(
  data = COVID19Cases_geoRegion,
  col_cases = "entries",
  col_date = "datum",
  col_region = "geoRegion"
)
# Test whether "COVID19Cases_geoRegion" is balanced panel data

COVID19Cases_geoRegion_balanced$data_balanced
# Balanced? TRUE or FALSE

if (COVID19Cases_geoRegion_balanced$data_balanced == FALSE) {
  COVID19Cases_geoRegion <-
    as_balanced(
    COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion"
  )
}
# Correction of dataset "COVID19Cases_geoRegion"
# not necessary as parameter balance of is_balanced is set TRUE by default
```

`load_infections_paneldata`

Import of infections panel data

Description

Loading infections panel data (data.frame) and creating an object of class infpan

Usage

```
load_infections_paneldata(
  data,
  col_cases,
```

```
col_date,
col_region,
other_cols = NULL,
time_format = "%Y-%m-%d",
time_unit = "days",
verbose = FALSE
)
```

Arguments

`data`	`data.frame` with regional infection data
`col_cases`	`character`, Column containing the cases (numeric)
`col_date`	`character`, Column containing the time points (e.g., days)
`col_region`	`character`, Column containing the unique identifier of the regions (e.g., name, NUTS 3 code)
`other_cols`	`list`, Further columns in the input data
`time_format`	`character`, Time format of the values in `col_date`
`time_unit`	`character`, Time unit of the values in `col_date`, e.g., "days"
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The function imports user-given infections panel data. The input data is checked in several ways (e.g., whether data is balanced or not). Other relevant columns from the input data may be defined in the `character` vector `other_cols`: "R_t" (Effective reproduction number), "Cum. cases" (Cumulative cases), "Incidence" Incidence (per xxx pop), "Population" (Population size of the region), "Roll. mean" (Rolling mean of cases), and "Roll. sum" (Rolling sum of cases).

The output is an object of class `infpan`. The results can be viewed using `summary(infpan)`. From an instance of class `infpan`, all built-in analyses for infections panel data may be conducted, e.g., the Swash-Backwash Model (`swash(infpan)`) or logistic growth models (`growth(infpan)`).

Value

object of class `infpan-class`

Author(s)

Thomas Wieland

References

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

See Also

infpan-class

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1 <-
  swash(
    infpan_CH,
    verbose = TRUE
    )
# Swash-Backwash Model for Swiss COVID19 cases
# Spatial aggregate: NUTS 3 (cantons)

summary(CH_covidwave1)
# Summary of Swash-Backwash Model
```

`loggrowth-class` *Class* `"loggrowth"`

Description

The class `"loggrowth"` contains the results of the logistic_growth() function. Use `summary(loggrowth)` and `plot(loggrowth)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by the function `logistic_growth`.

Slots

`LinModel`**:** Object of class `list` Results of the OLS helper model

`GrowthModel_OLS`**:** Object of class `list` Results of the OLS fit (predicted, parameters, first derivative)

`GrowthModel_NLS`**:** Object of class `list` Results of the NLS fit (predicted, parameters, first derivative)

`t`**:** Object of class `numeric` Input time points data

`y`**:** Object of class `numeric` Input infections data

`config`**:** Object of class `list` Model fit configurations

Methods

plot `signature(x = "loggrowth")`: Plots the results of the logistic growth model (observed, predicted, first derivative)

summary `signature(object = "loggrowth")`: Prints a summary of `loggrowth` objects

print `signature(x = "loggrowth")`: Prints an `loggrowth` object; use `summary(loggrowth)` for results

Author(s)

Thomas Wieland

References

Chowell G, Simonsen L, Viboud C, Yang K (2014) Is West Africa Approaching a Catastrophic Phase or is the 2014 Ebola Epidemic Slowing Down? Different Models Yield Different Answers for Liberia. *PLoS currents 6*. doi:10.1371/currents.outbreaks.b4690859d91684da963dc40e00f3da81

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics 22*, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Examples

```
showClass("loggrowth")
```

`logistic_growth` *Logistic Growth Model for Epidemic Data*

Description

Estimation of logistic growth models from cumulative infections data, linearized OLS and/or NLS

Usage

```
logistic_growth(
  y,
  t,
  S = NULL,
  S_start = NULL,
  S_end = NULL,
  S_iterations = 10,
  S_start_est_method = "bisect",
  seq_by = 10,
  nls = TRUE,
  add_constant = 1,
  verbose = FALSE
  )
```

Arguments

`y`	`numeric` `vector` with cumulative infections data over time
`t`	`vector` of class `numeric` or `Date` with time points or dates
`S`	Saturation value for the model
`S_start`	Start value of the saturation value for estimation
`S_end`	End value of the saturation value for estimation
`S_iterations`	Number of iterations for saturation value search
`S_start_est_method`	Method for saturation value search, either "bisect" or "trial_and_error"
`seq_by`	No of segments for the "trial_and_error" estimation of the saturation value
`nls`	Nonlinear estimation? `TRUE` or `FALSE`
`add_constant`	Numeric constant to be added to `y` if zero values occur (only relevant for OLS estimation)
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

This function allows the estimation of a logistic growth model. The user must specify the dependent variable (cumulative infections) and the time variable (time counter or date values). The estimation is performed using a linearized model with an OLS estimator and nonlinear least squares. For the former, the saturation value can either be specified by the user or found using a search algorithm. The parameters from the OLS fit are used as starting values for the NLS estimation.

Value

object of class `loggrowth-class`

Author(s)

Thomas Wieland

References

Chowell G, Simonsen L, Viboud C, Yang K (2014) Is West Africa Approaching a Catastrophic Phase or is the 2014 Ebola Epidemic Slowing Down? Different Models Yield Different Answers for Liberia. *PLoS currents* 6. doi:10.1371/currents.outbreaks.b4690859d91684da963dc40e00f3da81

Pell B, Kuang Y, Viboud C, Chowell G (2018) Using phenomenological models for forecasting the 2015 ebola challenge. *Epidemics* 22, 62–70. doi:10.1016/j.epidem.2016.11.002

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

See Also

`loggrowth-class`, `growth`, `exponential_growth`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

COVID19Cases_BS <-
  COVID19Cases_geoRegion[(COVID19Cases_geoRegion$geoRegion == "ZH")
                         & (COVID19Cases_geoRegion$sumTotal > 0),]
# COVID cases for Zurich

loggrowth_BS <- logistic_growth (
  y = as.numeric(COVID19Cases_BS$sumTotal),
  t = COVID19Cases_BS$datum,
  S = 5557,
  S_start = NULL,
  S_end = NULL,
  S_iterations = 10,
  S_start_est_method = "bisect",
  seq_by = 10,
  nls = TRUE
)
```

```
# Logistic growth model with stated saturation value

summary(loggrowth_BS)
# Summary of logistic growth model

plot(loggrowth_BS)
# Plot of logistic growth model
```

`metrics` *Fit metrics of observed and expected numeric variables*

Description

Calculation of fit metrics for observed and expected numeric variables (e.g. R^2, $RMSE$, MAE, $MAPE$).

Usage

```
metrics(
  observed,
  expected,
  plot = TRUE,
  plot.main = "Observed vs. expected",
  xlab = "Observed",
  ylab = "Expected",
  point.col = "blue",
  point.pch = 19,
  line.col = "red",
  plot_residuals.main = "Residuals",
  legend.cex = 0.7
)
```

Arguments

`observed`	Numeric vector of observed values.
`expected`	Numeric vector of expected or predicted values.
`plot`	Logical. If `TRUE`, diagnostic plots for observed vs. expected values and relative residual distributions are created.
`plot.main`	Character string. Title of the observed vs. expected plot.
`xlab`	Character string. Label of the x-axis.
`ylab`	Character string. Label of the y-axis.
`point.col`	Color of points in the observed vs. expected plot.
`point.pch`	Plotting character used for points.
`line.col`	Color of the identity line ($y = x$).
`plot_residuals.main`	Character string. Title of the residuals bar plot.
`legend.cex`	Numeric. Character expansion factor for legends.

Details

The function computes several goodness-of-fit metrics comparing observed and expected numeric values. In addition to classical error measures such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), the coefficient of determination (R^2) is calculated.

If `plot = TRUE`, the function produces:

- a scatter plot of observed versus expected values including the identity line,
- a bar plot of relative residual frequencies.

Value

A `list` with two elements:

`fit_metrics`	A `list` containing the computed fit metrics: `SQR`, `SAR`, `SQT`, `R2`, `MSE`, `RMSE`, `MAE`, and `MAPE`.
`observed_expected`	A `data.frame` containing observed values, expected values, residuals, and derived residual measures.

Author(s)

Thomas Wieland

References

Boehmke B, Greenwell B (2020). *Hands-On Machine Learning with R* (1st ed.). Taylor & Francis, New York, NY.

See Also

`binary_metrics`

Examples

```
obs <- c(10, 12, 15, 18, 20)
exp <- c(11, 13, 14, 17, 21)

metrics(
  observed = obs,
  expected = exp
)
```

`nbmatrix` *Construct Neighbourhood Matrix from Polygons*

Description

Building a neighbourhood matrix based on regions (polygons) with contiguous boundaries and resulting a data frame

Usage

```
nbmatrix(
  polygon_sf,
  ID_col,
  row.names = NULL
  )
```

Arguments

`polygon_sf`	`sf` object with polygons
`ID_col`	Column of `polygon_sf` with unique ID of each polygon
`row.names`	row.names for the `sf` object

Details

The function is based on `spdep::poly2nb`, which creates neighbours lists. The input is a `sf` object (spatial data frame) and the results are 1) a `nb` list (`poly2nb` result) and 2) a `data.frame`.

Value

`list` with two entries:

`nb:`	Object of class `"sb"` Neighbours list; see the `spdep::poly2nb` documentation
`nbmat:`	Object of class `"data.frame"` Dataset neighbouring regions

Author(s)

Thomas Wieland

References

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2022) Spatial patterns of excess mortality in the first year of the COVID-19 pandemic in Germany. *European Journal of Geography* 13(4), 18-33. doi:10.48088/ejg.t.wie.13.4.018.033

See Also

`nbstat`

Examples

```
data(RKI_Corona_counties)
# German counties (Source: Robert Koch Institute)

Corona_nbmat <-
  nbmatrix (
    RKI_Corona_counties,
    ID_col="AGS"
  )
# Creating neighborhood matrix
```

`nbstat` *Calculate Neighbourhood Statistics from Polygons*

Description

Calculating descriptive neighbourhood statistics based on regions (polygons) with contiguous boundaries and resulting a data frame

Usage

```
nbstat(
  polygon_sf,
    ID_col,
    link_data,
    data_ID_col,
    data_col,
    func = "sum",
    row.names = NULL
  )
```

Arguments

`polygon_sf`	`sf` object with polygons
`ID_col`	Column of `polygon_sf` with unique ID of each polygon
`link_data`	`data.frame` to merge with
`data_ID_col`	Column with unique ID of each polygon in `data.frame`
`data_col`	Column with regarded numeric values in `data.frame`
`func`	Descriptive statistic (`FUN`) to be computed for `data_col` of the neighbouring regions
`row.names`	row.names for the `sf` object

Details

The function is based on `spdep::poly2nb`, which creates neighbours lists. The input is a `sf` object (spatial data frame) and the results are 1) a `nb` list (`poly2nb` result) and 2) a `data.frame`.

Value

`list` with three entries:

`nbmat:`	Object of class `"data.frame"` Dataset neighbouring regions
`nbmat_data:`	Object of class `"data.frame"` Dataset neighbouring regions and linked data
`nbmat_data_aggreagte:`	Object of class `"data.frame"` Dataset with statistic by region

Author(s)

Thomas Wieland

References

Wieland T (2020) Flatten the Curve! Modeling SARS-CoV-2/COVID-19 Growth in Germany at the County Level. *REGION* 7(2), 43–83. doi:10.18335/region.v7i2.324

Wieland T (2022) Spatial patterns of excess mortality in the first year of the COVID-19 pandemic in Germany. *European Journal of Geography* 13(4), 18-33. doi:10.48088/ejg.t.wie.13.4.018.033

See Also

`nbmatrix`

Examples

```
data(RKI_Corona_counties)
# German counties (Source: Robert Koch Institute)

Corona_nbstat <-
  nbstat (
    RKI_Corona_counties,
    ID_col="AGS",
    link_data = RKI_Corona_counties,
    data_ID_col = "AGS",
    data_col = "EWZ",
    func = "sum"
  )
Corona_nbstat$nbmat_data_aggregate
# Sum of population (EWZ) of neighbouring counties
```

`Oesterreich_Faelle` *Austria Daily COVID-19 cases by region 2020-02-26 to 2020-05-31*

Description

A dataset containing COVID-19 cases by region (NUTS 3) and time periods (days) for Austria (Source: BMSGPK).

Usage

```
data(Oesterreich_Faelle)
```

Format

A data.frame with multiple columns:

NUTS3 (character) Region for which the data was collected.

Datum (Date) Date of record.

Faelle (integer) Number of reported cases on this date.

Details

The data was originally published by BMSGPK at a smaller spatial scale level (political districts, "Politische Bezirke"). The data was linked to a corresponding shapefile from Statistik Austria (2022), joined to the NUTS3 level via a spatial join, and summed over the Austrian NUTS3 regions. The spatial join is based on polygon centroids of the political districts level; in cases where the centroid was outside the polygon, it was placed inside the polygon manually.

Source

BMSGPK, Oesterreichisches COVID-19 Open Data Informationsportal (2022) COVID-19: Zeitliche Darstellung von Daten zu Covid19-Faellen je Bezirk. `https://www.data.gv.at/katalog/dataset/4b71eb3d-7d55-4967-b80d-91a3f220b60c` (retrieved 2022-06-23)

Statistik Austria (2022) Politische Bezirke. https://www.data.gv.at/katalog/dataset/stat_gliederung-osterreichs-in-politische-bezirke131e2 (retrieved 2022-06-27, URL does not exist anymore)

Wieland T (2022) C19dNUTS: Dataset of Regional COVID-19 Deaths per 100,000 Pop (NUTS). R package v1.0.1. doi:10.32614/CRAN.package.C19dNUTS

Examples

```
data(Oesterreich_Faelle)
# Get Austrian COVID19 cases at NUTS 3 level
# (first wave, same final date as in Swiss data: 2020-05-31)

AT_covidwave1 <-
  swash_backwash(
    data = Oesterreich_Faelle,
    col_cases = "Faelle",
    col_date = "Datum",
    col_region = "NUTS3"
  )
# Swash-Backwash Model for Austrian COVID19 cases
# Spatial aggregate: NUTS 3

summary(AT_covidwave1)
# Summary of model results
```

`plot-methods` *Methods for Function* `plot`

Description

Methods for function `plot` for different S4 classes: `infpan`, `sbm`, `sbm_ci`, `loggrowth`, `expgrowth`, `hawkes`, and `breaksgrowth`.

Methods

`signature(x = "infpan") plot.infpan(x, y, ...)`: Plots regional infections against time

Arguments:

- `x`: An object of class `infpan` including infections panel data.
- `y`: Optional argument for additional customization, such as plot style or axis labels.
- `...`: Additional graphical parameters that can be passed to control plot appearance.

Details: This method is used to visualize case data by region for N regions and T time points.

`signature(x = "sbm") plot.sbm(x, y = NULL, col_edges = "blue", xlab_edges = "Time", ylab_edges = "Regions", main_edges = "Edges", col_SIR = c("blue", "red", "green"), lty_SIR = c("solid", "solid", "solid"), lwd_SIR = c(1,1,1), xlab_SIR = "Time", ylab_SIR = "Regions", main_SIR = "SIR integrals", col_cases = "red", lty_cases = "solid", lwd_cases = 1, xlab_cases = "Time", ylab_cases = "Infections", main_cases = "Daily infections", xlab_cum = "Cases", ylab_cum = "Regions", main_cum = "Cumulative infections per region", horiz_cum = TRUE, separate_plots = FALSE)`:

Plots the results of the Swash-Backwash Model. This generates four plots:

- Edges over time.
- SIR integrals.
- All-over infections.
- Total infections per time unit.

Arguments:

- `x`: An object of class `sbm` representing the results of the Swash-Backwash Model.
- `y`: Optional argument for additional customization, such as plot style or axis labels.
- `...`: Additional graphical parameters that can be passed to control plot appearance.

Details: This method is used to visualize the output of the Swash-Backwash Model, providing insight into the dynamics of the modeled epidemic.

`signature(x = "sbm_ci") plot.sbm_ci(x, y, ...)`: Plots the results of bootstrap confidence intervals for the Swash-Backwash Model. This generates a single figure with six subplots:

- S_A (susceptible population),
- I_A (infected population),
- R_A (recovered population),
- t_{FE} (final epidemic time),
- t_{LE} (last epidemic time),
- R_{0A} (basic reproduction number).

Arguments:

- `x`: An object of class `sbm_ci` containing the bootstrap confidence intervals for the Swash-Backwash Model.
- `y`: Optional argument for additional customization, such as plot style or axis labels.
- `...`: Additional graphical parameters for fine-tuning the plots.

Details: This method is used to visualize the bootstrap confidence intervals for various parameters of the Swash-Backwash Model.

`signature(x = "countries") plot.sbm(x, y = NULL, col_bars = "grey", col_ci = "red")`: Plots the results of the between-countries analysis via Swash-Backwash Model. This generates four plots:

- Indicator for country 1
- Indicator for country 2
- Boxplots of the distribution of the indicator in country 1 and 2
- Distribution of the difference between the indicators of country 1 and 2

Arguments:

- `x`: An object of class `countries` representing the results of the Swash-Backwash Model country analysis.
- `y`: Not relevant
- `col_bars`: Color of bars
- `col_ci`: Color of confidence intervals

Details: This method is used to visualize the output of the Swash-Backwash Model, providing insight into the dynamics of the modeled epidemic.

`signature(x = "loggrowth") plot.loggrowth(x, y, ...)`: Plots the results of the logistic growth model, including:

- Observed values
- Predicted values
- First derivative

Arguments:

- `x`: An object of class `loggrowth` containing the data for the logistic growth model.
- `y`: Optional argument for additional customization of the plot (e.g., color, labels).
- `...`: Additional arguments for graphical parameters.

Details: This method is useful for visualizing the observed and predicted growth patterns in an epidemic or similar phenomena modeled by logistic growth.

`signature(x = "expgrowth") plot.expgrowth(x, y, ...)`: Plots the results of the exponentai growth model, including:

- Observed values
- Predicted values

Arguments:

- `x`: An object of class `expgrowth` containing the data for the exponential growth model.
- `y`: Optional argument for additional customization of the plot (e.g., color, labels).
- `...`: Additional arguments for graphical parameters.

Details: This method is useful for visualizing the observed and predicted growth patterns in the initial phase of an epidemic or similar phenomena modeled by exponential growth.

`signature(x = "hawkes") plot.hawkes(x, y, ...)`: Plots the results of the Hawkes process model, including:

- Observed values
- Predicted values

Arguments:

- `x`: An object of class `hawkes` containing the data for the Hawkes model.
- `y`: Optional argument for additional customization of the plot (e.g., color, labels).
- `...`: Additional arguments for graphical parameters.

Details: This method is useful for visualizing the observed and predicted growth patterns of an epidemic or similar phenomena modeled as Hawkes processes.

`signature(x = "breaksgrowth") plot.breaksgrowth(x, y, ...)`: Plots the results of a breakpoint analysis, including:

- Time series data
- Breakpoints

Arguments:

- `x`: An object of class `breaksgrowth` containing the data for the breakspoints model.
- `y`: Optional argument for additional customization of the plot (e.g., color, labels).
- `...`: Additional arguments for graphical parameters.

Details: This method is useful for visualizing the derived breakpoints.

Author(s)

Thomas Wieland

`plot_coef_ci` *Plot Point Estimates With Confidence Intervals*

Description

Plotting point estimates with confidence intervals from regression results

Usage

```
plot_coef_ci(
  point_estimates,
  confint_lower,
  confint_upper,
  coef_names,
  p = NULL,
  estimate_colors = NULL,
  confint_colors = NULL,
```

```
  auto_color = FALSE,
  alpha = 0.05,
  set_estimate_colors = c("red", "grey", "green"),
  set_confint_colors = c("#ffcccb", "lightgray", "#CCFFCC"),
  skipvars = NULL,
  plot.xlab = "Independent variables",
  plot.main = "Point estimates with CI",
  axis.at = seq(-30, 40, by = 5),
  pch = 15,
  cex = 2,
  lwd = 5,
  y.cex = 0.8
  )
```

Arguments

`point_estimates`	`numeric` vector containing point estimates
`confint_lower`	`numeric` vector containing lower confidence intervals
`confint_upper`	`numeric` vector containing upper confidence intervals
`coef_names`	`character` vector containing coefficient names
`p`	`numeric` vector containing p values of the coefficients (optional)
`estimate_colors`	vector containing colors for the point estimates (optional)
`confint_colors`	vector containing colors for the confidence intervals (optional)
`auto_color`	`bool` value which indicates whether the colors are found automatically based on coef and CI values
`alpha`	Significance level α for 1-α*100 confidence intervals
`set_estimate_colors`	Colors for point estimates (significant negative, not significant, significant positive)
`set_confint_colors`	Colors for confidence intervals (significant negative, not significant, significant positive)
`skipvars`	List with coefficients to be dropped
`plot.xlab`	Label of x axis
`plot.main`	Plot title
`axis.at`	Position of y axis
`pch`	Point type
`cex`	Point size
`lwd`	Line width (confidence intervals)
`y.cex`	Font size of y axis

Details

The function checks whether the input vectors have the same length. If `auto_color` is `TRUE`, the colors from `set_estimate_colors` and `set_confint_colors` are used, and the significance level is determined based on the coefficient and confidence interval values (all three below 0 = significant negative, all three above 0 = significant positive).

Value

Coefficients plot, no returned value

Author(s)

Thomas Wieland

References

Wieland T (2025) Assessing the effectiveness of non-pharmaceutical interventions in the SARS-CoV-2 pandemic: results of a natural experiment regarding Baden-Württemberg (Germany) and Switzerland in the second infection wave. *Journal of Public Health* 33(11), 2497-2511. doi:10.1007/s1038902402218x

Examples

```
data(did_fatalities_splm_coef)
# Results of a difference-in-differences model

plot_coef_ci(
  point_estimates = did_fatalities_splm_coef$Estimate,
  confint_lower = did_fatalities_splm_coef$CI_lower_Bonferroni,
  confint_upper = did_fatalities_splm_coef$CI_upper_Bonferroni,
  coef_names = did_fatalities_splm_coef$Var,
  skipvars = c(
    "Alpha_share",
    "lambda",
    "rho",
    "log(D_Infections_daily_7dsum_per100000_lag2weeks)",
    "vacc_cum_per100000_lag2weeks"
    ),
  lwd = 13,
  pch = 19,
  auto_color = TRUE
)
# Plot with point estimates and confidence intervals
```

`print-methods` *Methods for Function* `print`

Description

Methods for function `print`

Methods

`signature(object = "infpan")` Prints an `infpan` object; use `summary(infpan)` for results

`signature(x = "sbm")` Prints a `sbm` object; use `summary(sbm)` for results

`signature(x = "sbm_ci")` Prints a `sbm_ci` object; use `summary(sbm_ci)` for results

`signature(object = "countries")` Prints an `countries` object; use `summary(countries)` for results

`signature(x = "loggrowth")` Prints a `loggrowth` object; use `summary(loggrowth)` for results

`signature(x = "expgrowth")` Prints an `expgrowth` object; use `summary(expgrowth)` for results

`signature(x = "hawkes")` Prints a `hawkes` object; use `summary(hawkes)` for results

`signature(x = "breaksgrowth")` Prints a `breaksgrowth` object; use `summary(breaksgrowth)` for results

`quantile_ci` *Computing Quantiles for a given Numeric Vector*

Description

Computes quantiles for a given vector `x` and the related confidence intervals (`lower`, `upper`).

Usage

```
quantile_ci(
  x,
  alpha = 0.05
  )
```

Arguments

`x`	A numeric vector
`alpha`	Significance level α for 1-α*100 confidence intervals

Details

Helper function for `plot(sbm_ci)`, but may be used separately.

Value

A numeric vector with lower and upper quantile

Author(s)

Thomas Wieland

Examples

```
numeric_vector <- c(1,9,5,6,3,10,20,6,9,14,3,5,8,6,11)
# any numeric vector

quantile_ci(numeric_vector)
```

`RKI_Corona_counties` *German Counties with COVID-19 Cases*

Description

A dataset containing German counties (NUTS 3) with COVID-19 cases (Source: Robert Koch Institute).

Usage

```
data(RKI_Corona_counties)
```

Format

A data.frame with multiple columns:

OBJECTID unknown/not necessary

ADE (unknown/not necessary

GF unknown/not necessary

BSG unknown/not necessary

RS (character) County code 1

AGS (character) County code 2

SDV_RS (character) County code 3

GEN (character) County name

BEZ (character) County type

IBZ unknown/not necessary

BEM unknown/not necessary

NBD unknown/not necessary

SN_L unknown/not necessary

SN_R unknown/not necessary

SN_K unknown/not necessary

SN_V1 unknown/not necessary

SN_V2 unknown/not necessary

SN_G unknown/not necessary

FK_S3 unknown/not necessary

NUTS (character) NUTS 3 code

RS_0 unknown/not necessary

AGS_0 unknown/not necessary

WSK unknown/not necessary

EWZ (numeric) Population

KFL (numeric) Area in sq. km

DEBKG_ID unknown/not necessary

Shape__Are unknown/not necessary

Shape__Len unknown/not necessary

death_rate

cases (numeric) COVID-19 cases

deaths (numeric) COVID-19 associated deaths

cases_per_ (numeric) COVID-19 cases per 100,000 inhabitants

cases_pe_1 unknown/not necessary

BL (character) Federal state

BL_ID (integer) Federal state ID

county (character) County name

last_updat Date of last update

geometry Geometry

Details

The data is included as it was published in by the Robert Koch Institute (Robert Koch-Institut, RKI) but extended by the geometry column (Original data: shapefile).

Source

RKI (2020) RKI Corona Landkreise. Robert Koch-Institut (RKI), dl-de/by-2-0. Attribution: Robert Koch-Institut, Bundesamt für Kartographie und Geodäsie. `https://npgeo-corona-npgeo-de.hub.arcgis.com/datasets/917fc37a7095` (retrieved 2020-03-30)

Examples

```
data(RKI_Corona_counties)
# German counties (Source: Robert Koch Institute)

Corona_nbstat <-
  nbstat (
    RKI_Corona_counties,
    ID_col="AGS",
    link_data = RKI_Corona_counties,
    data_ID_col = "AGS",
    data_col = "EWZ",
    func = "sum"
  )
Corona_nbstat$nbmat_data_aggregate
# Sum of population (EWZ) of neighbouring counties
```

`R_t` *Effective Reproduction Number for Epidemic Data*

Description

Calculation of the effective reproduction number for infection/surveillance data

Usage

```
R_t(
  infections,
  GP = 4,
  correction = FALSE
  )
```

Arguments

`infections`	`numeric` `vector` with infection data
`GP`	Generation period, in time units (typically days)
`correction`	Correction of values equal to zero? (Recommended)

Details

The function calculates the effective reproduction number, R_t, of an infections time series. Set the generation period by the parameter `GP` (default: 4). If `correction` is TRUE, values equal to zero are increased by one.

Value

`list` with two entries:

`R_t`:	Object of class `"numeric"` R_t values
`infections_data`:	Object of class `"data.frame"` Dataset with infections data and R_t

Author(s)

Thomas Wieland

References

an der Heiden M, Hamouda O (2020) Schätzung der aktuellen Entwicklung der SARS-CoV-2-Epidemie in Deutschland - Nowcasting. *Epidemiologisches Bulletin* 17, 10-15. doi:10.25646/6692

Bonifazi G, Lista L, Menasce D, Mezzetto M, Pedrini D, Spighi R, Zoccoli A (2021) A simplified estimate of the effective reproduction number Rt using its relation with the doubling time and application to Italian COVID-19 data. *The European Physical Journal Plus* 136, 386. doi:10.1140/epjp/s13360021013396

Wieland T (2020) A phenomenological approach to assessing the effectiveness of COVID-19 related nonpharmaceutical interventions in Germany. *Safety Science* 131, 104924. doi:10.1016/j.ssci.2020.104924

See Also

`logistic_growth`, `exponential_growth`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

COVID19Cases_BS <-
  COVID19Cases_geoRegion[(COVID19Cases_geoRegion$geoRegion == "ZH")
                         & (COVID19Cases_geoRegion$sumTotal > 0),]
# COVID cases for Zurich

Rt_BS <- R_t(infections = COVID19Cases_BS$entries)
# Effective reproduction number

Rt_BS
```

`sbm-class` *Class* `"sbm"`

Description

The class `"sbm"` contains the results of the Swash-Backwash Model and the related input data as well as additional information. Use `summary(sbm)` and `plot(sbm)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by the function `swash_backwash` and the `infpan` method `swash`, respectively.

Slots

`R_0A`**:** Object of class `"numeric"` Model result: spatial reproduction number R_{0A}

`integrals`**:** Object of class `"numeric"` Model result: integrals S_A, I_A, and R_A

`velocity`**:** Object of class `"numeric"` Model result: velocity measures t_{FE} and t_{LE}

`occ_regions`**:** Object of class `"data.frame"` Model result: Occurence at regional level

`SIR_regions`**:** Object of class `"data.frame"` Model result: Susceptible, infected and recovered regions over time

`cases_by_date`**:** Object of class `"data.frame"` Total cases by date

`cases_by_region`**:** Object of class `"data.frame"` Cumulative cases by region

`input_data`**:** Object of class `"data.frame"` Input data

`data_statistics`**:** Object of class `"numeric"` Diagnostics of input data

`col_names`**:** Object of class `"character"` Original column names in input data

`timestamp`**:** Object of class `"list"` Time stamps of any update of the instance

Methods

confint `signature(object = "sbm")`: Creates bootstrap confidence intervals for `sbm` objects.

plot `signature(x = "sbm")`: Plots the results of the Swash-Backwash Model; two plots: edges over time, total infections per time unit

print `signature(x = "sbm")`: Prints an `sbm` object; use `summary(sbm)` for results

show `signature(object = "sbm")`: Prints an `sbm` object; use `summary(sbm)` for results

summary `signature(object = "sbm")`: Prints a summary of `sbm` objects (results of the Swash-Backwash Model)

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

Examples

```
showClass("sbm")
```

`sbm_ci-class` *Class* `"sbm_ci"`

Description

The class `"sbm_ci"` contains the results of the Swash-Backwash Model, confidence intervals for the model estimates, and the related input data as well as additional information. Use `summary(sbm_ci)` and `plot(sbm_ci)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by the function `confint(sbm)`.

Slots

`R_0A`: Object of class `"numeric"` Model result: spatial reproduction number R_{0A}

`integrals`: Object of class `"numeric"` Model result: integrals S_A, I_A, and R_A

`velocity`: Object of class `"numeric"` Model result: velocity measures t_{FE} and t_{LE}

`occ_regions`: Object of class `"data.frame"` Model result: Occurence at regional level

`cases_by_date`: Object of class `"data.frame"` Total cases by date

`cases_by_region`: Object of class `"data.frame"` Cumulative cases by region

`input_data`: Object of class `"data.frame"` Input data

`data_statistics`: Object of class `"numeric"` Diagnostics of input data

`col_names`: Object of class `"character"` Column names in input data

`integrals_ci`: Object of class `"list"` Confidence intervals for integrals S_A, I_A, and R_A

`velocity_ci`: Object of class `"list"` Confidence intervals for velocity measures t_{FE} and t_{LE}

`R_0A_ci`: Object of class `"numeric"` Confidence intervals for spatial reproduction number R_{0A}

`iterations`: Object of class `"data.frame"` Results of bootstrap sampling iterations

`ci`: Object of class `"numeric"` Lower and upper confidence intervals based on user input

`config`: Object of class `"list"` Configuration details for bootstrap sampling

Methods

- **plot** `signature(x = "sbm_ci")`: Plots the results of bootstrap confidence intervals for the Swash-Backwash Model; one figure with six plots: S_A, I_A, R_A, t_{FE}, t_{LE}, and R_{0A}
- **print** `signature(x = "sbm_ci")`: Prints an `sbm_ci` object; use `summary(sbm_ci)` for results
- **show** `signature(object = "sbm_ci")`: Prints an `sbm_ci` object; use `summary(sbm_ci)` for results
- **summary** `signature(object = "sbm_ci")`: Prints a summary of `sbm_ci` objects (bootstrap confidence intervals for Swash-Backwash Model estimates)

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap.

Ramachandran KM, Tsokos CP (2021) Mathematical Statistics with Applications in R (Third Edition). Ch. 13.3.1 (Bootstrap confidence intervals). doi:10.1016/B9780128178157.000130

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

Examples

```
showClass("sbm_ci")
```

`show-methods` *Methods for Function* `show`

Description

Methods for function `show`

Methods

- `signature(object = "infpan")` Prints an `infpan` object; use `summary(infpan)` for results
- `signature(object = "sbm")` Prints an `sbm` object; use `summary(sbm)` for results
- `signature(object = "sbm_ci")` Prints an `sbm_ci` object; use `summary(sbm_ci)` for results

- `signature(object = "countries")` Prints an `countries` object; use `summary(countries)` for results
- `signature(object = "loggrowth")` Prints an `loggrowth` object; use `summary(loggrowth)` for results
- `signature(object = "expgrowth")` Prints an `expgrowth` object; use `summary(expgrowth)` for results

`summary-methods` *Methods for Function* `summary`

Description

Methods for function `summary`

Methods

- `signature(object = "sbm")` Prints a summary of `sbm` objects (results of the Swash-Backwash Model)
- `signature(object = "sbm_ci")` Prints a summary of `sbm_ci` objects (bootstrap confidence intervals for Swash-Backwash Model estimates)
- `signature(object = "countries")` Prints a summary of a `countries` object built with the function `compare_countries`
- `signature(object = "loggrowth")` Prints a summary of a `loggrowth` object built with the function `logistic_growth`
- `signature(object = "expgrowth")` Prints a summary of a `expgrowth` object built with the function `exponential_growth`
- `signature(object = "hawkes")` Prints a summary of a `hawkes` object built with the function `hawkes_growth`
- `signature(object = "breaksgrowth")` Prints a summary of a `breaksgrowth` object built with the function `breaks_growth`

`swash` *Swash-Backwash Model for the Single Epidemic Wave*

Description

Analysis of regional infection/surveillance data stored in `infpan` object using the Swash-Backwash Model for the single epidemic wave by Cliff and Haggett (2006).

Usage

```
swash(
  object,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `infpan`
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The method performs the analysis of the input panel data with N regions and T time points using the Swash-Backwash Model based on an `infpan` object. The output is an object of class `sbm`. The results can be viewed using `summary(sbm)`. The built-in function `swash_backwash` is used for the analysis. See `swash_backwash` for further details.

Value

object of class `sbm-class`

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

See Also

`sbm-class`, `swash_backwash`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

infpan_CH <- load_infections_paneldata(
    data = COVID19Cases_geoRegion,
```

```
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion",
    other_cols = c("Population" = "pop"),
    verbose = TRUE
  )
# Import as infections panel data set (class infpan)

CH_covidwave1 <-
  swash(
    infpan_CH,
    verbose = TRUE
    )
# Swash-Backwash Model for Swiss COVID19 cases
# Spatial aggregate: NUTS 3 (cantons)

summary(CH_covidwave1)
# Summary of Swash-Backwash Model
```

`swash-methods` *Methods for Function* `swash`

Description

Methods for function `swash`

Methods

`signature( object = "infpan", verbose = FALSE)` Performs the analysis of the input panel data with N regions and T time points using the Swash-Backwash Model based on an `infpan` object. The output is an object of class `"sbm"`. The results can be viewed using `summary(sbm)`. See `swash_backwash` for further details of the model analysis.

Author(s)

Thomas Wieland

`swash_backwash` *Swash-Backwash Model for the Single Epidemic Wave*

Description

Analysis of regional infection/surveillance data using the Swash-Backwash Model for the Single Epidemic Wave by Cliff and Haggett (2006).

Usage

```
swash_backwash(
  infpan = NULL,
  data = NULL,
  col_cases = NULL,
  col_date = NULL,
  col_region = NULL,
  time_format = "%Y-%m-%d",
  verbose = FALSE
  )
```

Arguments

`infpan`	`infpan` object containing regional infection data
`data`	`data.frame` with regional infection data
`col_cases`	Column containing the cases (numeric)
`col_date`	Column containing the time points (e.g., days)
`col_region`	Column containing the unique identifier of the regions (e.g., name, NUTS 3 code)
`time_format`	`character`, Time format of the values in `col_date`
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The function performs the analysis of the input panel data with N regions and T time points using the Swash-Backwash Model. The user must state panel data with daily infections.

The Swash-Backwash Model (SBM) for the Single Epidemic Wave is the spatial equivalent of the classic epidemiological SIR (Susceptible-Infected-Recovered) model. It was developed by Cliff and Haggett (2006) to model the velocity of spread of infectious diseases across space. Current applications can be found, for example, in Smallman-Raynor et al. (2022a,b). The function `swash_backwash()` enables the calculation of the Swash-Backwash Model for user-supplied panel data on regional infections. It calculates the model and creates a model object of the `sbm` class defined in this package. This class can be used to visualize results (`summary()`, `plot()`) and calculate bootstrap confidence intervals for the model estimates (`confint(sbm)`); the latter returns an object of class `sbm_ci` as defined in this package. Two `sbm_ci` objects for different countries may be compared with `compare_countries()`, which allows the estimation of mean differences of a user-specified model parameter (e.g., spatial reproduction number R_{OA}) between two countries. This makes it possible to check whether the spatial spread velocity of a communicable disease is significantly different in one country than in another country; the result is an object of class `countries`.

To calculate the SBM model based on an `infpan` object, use the corresponding method `swash(infpan)`.

Value

object of class `sbm-class`

Author(s)

Thomas Wieland

References

Cliff AD, Haggett P (2006) A swash-backwash model of the single epidemic wave. *Journal of Geographical Systems* 8(3), 227-252. doi:10.1007/s1010900600278

Smallman-Raynor MR, Cliff AD, Stickler PJ (2022) Meningococcal Meningitis and Coal Mining in Provincial England: Geographical Perspectives on a Major Epidemic, 1929–33. *Geographical Analysis* 54, 197–216. doi:10.1111/gean.12272

Smallman-Raynor MR, Cliff AD, The COVID-19 Genomics UK (COG-UK) Consortium (2022) Spatial growth rate of emerging SARS-CoV-2 lineages in England, September 2020–December 2021. *Epidemiology and Infection* 150, e145. doi:10.1017/S0950268822001285.

See Also

`sbm-class`, `swash`

Examples

```
data(COVID19Cases_geoRegion)
# Get SWISS COVID19 cases at NUTS 3 level

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[!COVID19Cases_geoRegion$geoRegion %in% c("CH", "CHFL"),]
# Exclude CH = Switzerland total and CHFL = Switzerland and Liechtenstein total

COVID19Cases_geoRegion <-
  COVID19Cases_geoRegion[COVID19Cases_geoRegion$datum <= "2020-05-31",]
# Extract first COVID-19 wave

CH_covidwave1 <-
  swash_backwash(
    data = COVID19Cases_geoRegion,
    col_cases = "entries",
    col_date = "datum",
    col_region = "geoRegion"
    )
# Swash-Backwash Model for Swiss COVID19 cases
# Spatial aggregate: NUTS 3 (cantons)

summary(CH_covidwave1)
# Summary of Swash-Backwash Model

plot(CH_covidwave1)
# Plot of Swash-Backwash Model edges and total epidemic curve
```

`timestamps` *Show timestamps*

Description

Print timestamps stored in an object.

Usage

```
timestamps(object)
```

Arguments

`object`	An object with a `timestamp` slot.

Value

Prints formatted timestamps to the console.

Index