Package ‘swash’

February 23, 2026

Type Package

Title Swash-Backwash Model for the Single Epidemic Wave

Version 1.3.3

Date 2026-02-20

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 The Swash-Backwash Model for the Single Epidemic Wave was developed by Cliff and Haggett (2006) <doi:10.1007/s10109-006-0027-8> to model the velocity of spread of infectious diseases across space. This package enables the calculation of the Swash-Backwash Model for user-supplied panel data on regional infections. The package provides additional functions for bootstrap confidence intervals, country comparison, visualization of results, and data management. Furthermore, it contains several functions for analysis and visualization of (spatial) infection data.

License GPL (>= 2)

Imports methods

NeedsCompilation no

Repository CRAN

Date/Publication 2026-02-23 08:10:02 UTC

Contents

`swash-package` *Implementation of the Swash-Backwash Model for the Single Epidemic Wave and additional functions in R*

Description

Swash-Backwash Model for the single epidemic wave (Cliff and Haggett 2006) with additional functions for bootstrap confidence intervals and data management; other functions for spatio-temporal analysis and modeling of infectious diseases

Details

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()` 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 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`.

The package also contains other functions for spatio-temporal analysis and modeling of infectious diseases, such as fitting logistic growth models and exponential growth models for the initial phase of a spread, all of which can be used both on `sbm` objects (`growth(sbm)`, `growth_initial(sbm)`) and stand-alone (`logistic_growth()`, `exponential_growth()`). These function return objects of class `loggrowth` and `expgrowth`, respectively, both defined in this package. Other functions include spatial statistics (`nbstat()` for neighborhood statistics), breakpoints analysis for time series models (`plot_breakpoints()`), and fit metrics (`metrics()`, `binary_metrics()`, `binary_metrics_glm()`). The package includes example data from the SARS-CoV-2/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

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 (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

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

CH_covidwave1 <-
  swash (
    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
```

`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()` function (when using the function `is_balanced()`), 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`	`bool` 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 destinction 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)
```

`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)
```

`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 the 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 (
    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 (
    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"` classes for each country. Use `summary(countries)` and `plot(countries)` for results summary and plotting, respectively.

Objects from the Class

Objects can be created by calls of the form `new("countries", ...)`. 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
- **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)

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)
```

`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

`exp_gr`**:** Object of class `numeric` Exponential growth rate

`y_0`**:** Object of class `numeric` Baseline

`R0`**:** Object of class `numeric` Basic reproduction number, derived from exponential growth model

`doubling`**:** Object of class `numeric` Doubling rate, derived from exponential growth model

`model_data`**:** Object of class `lm` OLS model

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

Methods

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

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

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

```
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,
  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
`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 (daily infections) and the time variable (time counter or date values). The estimation is performed using a linearized model as an OLS estimator. 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 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

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

summary(expgrowth_BS)
# Summary of exponential growth model

expgrowth_BS@doubling
# Doubling rate
```

`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,
  verbose = FALSE
  )
```

Arguments

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

Details

Please note: The method growth() for class 'sbm' is deprecated. In swash version >=2.0.0 it will be replaced by the method growth() for the new class 'infpan'.

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

Value

`list` with two entries:

`results:`	Object of class `"data.frame"` Results of the logistic growth models (coefficients and derivates)
`logistic_growth_models:`	Object of class `"list"` List with N entries for N growth models resp. `loggrowth` objects

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

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

CH_covidwave1_growth <- growth(CH_covidwave1)
CH_covidwave1_growth
# Logistic growth models for sbm object CH_covidwave1
```

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

Description

Methods for function `growth`

Methods

`signature(object = "sbm", S_iterations = 10, S_start_est_method = "bisect", seq_by = 10, nls = TRUE, verbos`
: 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

`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,
  verbose = FALSE
  )
```

Arguments

`object`	object of class `sbm`
`time_units`	`numeric` value for the analysis time (time units from start)
`GI`	Generation interval for computing R_0
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

Please note: The method growth_initial() for class ’sbm’ is deprecated. In swash version >=2.0.0 it will be replaced by the method growth_initial() for the new class ’infpan’.

The function estimates exponential growth models for regional infections based on a `sbm` 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

`list` with two entries:

`results:`	Object of class `"data.frame"` Results of the exponential growth models (growth rate, basic reproduction number, doubling rate) for each region
`exponential_growth_models:`	Object of class `"list"` List with N entries for N exponential models

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

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

CH_covidwave1_initialgrowth_3weeks <-
  growth_initial(
    CH_covidwave1,
    time_units = 21
    )
CH_covidwave1_initialgrowth_3weeks$results
# Exponential models for sbm 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 = "sbm", time_units = 10, GI = 4, verbose = FALSE)` 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(see `exponential_growth`).

Author(s)

Thomas Wieland

`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)
```

`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
```

`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,
  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`
`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 as an OLS estimator and as an NLS estimator. 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

`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 original 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)

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 (
    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: `sbm`, `sbm_ci`, and `loggrowth`.

Methods

`signature(x = "sbm") plot.sbm(x, y, ...)`: Plots the results of the Swash-Backwash Model. This generates two plots:

- Edges over time.
- 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 (growth rate).

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.

Author(s)

Thomas Wieland

`plot_breakpoints` *Calculating and plotting breakpoints in time series*

Description

The function calculates breakpoints for time series and provides plots of the analysis results

Usage

```
plot_breakpoints(
  dataset,
    formula,
    alpha = 0.05,
    line.col = "chocolate",
    ci.col = c(
      rgb(0, 0, 255, maxColorValue = 255, alpha = 0.5),
      rgb(0, 255, 0, maxColorValue = 255, alpha = 0.5),
      rgb(255, 0, 0, maxColorValue = 255, alpha = 0.5)
    ),
    legend.show = TRUE,
```

```
    legend.pos = c(
      "topright",
      "topright",
      "topright",
      "topright"
      ),
    xlab = "Time",
    ylab = "Y",
    ylim = NULL,
    plot.main = c(
      "Breakpoints",
      "Segment model fits",
      "BIC and Residual Sum of Squares",
      "Model without breaks (one segment)"
    ),
    output.full = TRUE,
    separate_plots = FALSE,
    ...
  )
```

Arguments

`dataset`	`data.frame` object with (infections) time series
`formula`	`formula` object for linear regression formula
`alpha`	Significance level α for 1-α*100 confidence intervals
`line.col`	Line color in time series plot
`ci.col`	Colors for confidence intervals in 3 plots
`legend.show`	`bool`: Show legend?
`legend.pos`	Position of legend
`xlab`	Label of x axis
`ylab`	Label of y axis
`ylim`	Limits of y axis
`plot.main`	Titles for 4 plots
`output.full`	`bool`: All 4 plots or just one (breakpoints)?
`separate_plots`	`bool`: All 4 plots in one or separate?
`...`	Further arguments passed to breakpoints()

Details

The function uses the breakpoints() function from the *strucchange* package (Zeileis et al. 2003) to estimate breaks in time series given by the user. These breakpoints are visualized, confidence intervals and model diagnostics are calculated, and for M breakpoints, $M+1$ models are estimated and returned.

Value

`data.frame` with one row for each model segment with coefficients, R^2, etc.

Author(s)

Thomas Wieland

References

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 A, Kleiber C, Krämer W, 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

`logistic_growth`, `exponential_growth`

Examples

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

plot_breakpoints(
  Infections,
  log(infections_daily) ~ day,
  output.full = TRUE
  )
# Breakpoints for time series
```

`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
```

`plot_regions` *Plots of Regional Infections Over Time*

Description

Plots regional infection curves in N plots for N regions.

Usage

```
plot_regions(
  object,
  col = "red",
  lty = "solid",
  scale = FALSE,
  normalize_by_col = NULL,
  normalize_factor = 1,
  plot_rollmean = FALSE,
  rollmean_col = "blue",
  rollmean_lty = "solid",
  rollmean_k = 7,
  rollmean_align = "center",
  rollmean_fill = NA,
  growth_col = "orange",
  growth_lty = "solid",
  growth_per_time_unit = 1
  )
```

Arguments

`object`	object of class `sbm`
`col`	Color of line plot
`lty`	Type of line plot
`scale`	Set y axis of the plots uniformly based on the maximum value across all regions? (boolean, default: `FALSE`)
`normalize_by_col`	Normalize infection numbers by stating a column in the input data frame (e.g., regional population)
`normalize_factor`	Multiply density/incidence with a factor (say, 100,000 inhabitants)
`plot_rollmean`	Plot rolling mean?
`rollmean_col`	Color of rolling mean line
`rollmean_lty`	Type of rolling mean line
`rollmean_k`	integer width of the rolling window (see `zoo::rollmean`)

`rollmean_align`	specifying whether the index of the result should be left- or right-aligned or centered (default) compared to the rolling window of observations (see `zoo::rollmean`)
`rollmean_fill`	filling values at the left/within/to the right of the data range (see `zoo::rollmean`)
`growth_col`	Color of growth line
`growth_lty`	Line type of growth line
`growth_per_time_unit`	Time unit for growth rates

Details

Please note: The method plot_regions() for class 'sbm' is deprecated. In swash version >=2.0.0 it will be replaced by the method plot() for the new class 'infpan'.

Plots regional infection curves in N plots for N regions, with the number of columns equals 4 and the number of rows is calculated based the size of N.

Value

Plot only, no returned value

Author(s)

Thomas Wieland

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

plot_regions(CH_covidwave1)
# Plot of regional infections
```

`plot_regions-methods` *Methods for Function* `plot_regions`

Description

Methods for function `plot_regions`

Methods

Please note: The method plot_regions() for class 'sbm' is deprecated. In swash version >=2.0.0 it will be replaced by the method plot() for the new class 'infpan'.

Plots regional infection curves in N plots for N regions, with the number of columns equals 4 and the number of rows is calculated based the size of N. Set the color by the argument `col`. If `scale` is `TRUE`, the y axis of the plots is set uniformly based on the maximum value across all regions. If the input data contains a column to normalize the infection numbers (such as regional population), the user may use this data to normalize the infection numbers by setting `normalize_by_col`. If this density value should be multiplied by a factor (e.g. regional infections per 100,000 inhabitants), this can be set with the argument `normalize_by_col` (default: 1).

Author(s)

`signature( object = "sbm", col = "red", lty = "solid", scale = FALSE, normalize_by_col = NULL, normalize_factor`

Thomas Wieland

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

Description

Methods for function `print`

Methods

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

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

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

`signature(x = "expgrowth")` Prints an `expgrowth` object; use `summary(expgrowth)` 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 days
`correction`	Correction of values equal to zero? (Recommended)

Details

The function calculates the effective reproduction number, R_t, of an infections time series.

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

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

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`

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`.

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

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)

growth `signature(object = "sbm")`: Estimates logistic growth models from `sbm` objects

growth_initial `signature(object = "sbm")`: Estimates exponential growth models from `sbm` objects for a given time period

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.

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("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 = "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 = "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`

`swash` *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(
  data,
  col_cases,
  col_date,
  col_region,
  verbose = FALSE
  )
```

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)
`verbose`	`bool` argument which indicates whether progress messages are displayed

Details

The function performs the analysis of the input panel data using the Swash-Backwash Model. The output is an object of class `"sbm"`. The results can be viewed using `summary(sbm)`. The user must state panel data with daily infections.

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`

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 (
    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
```

Index