

Package ‘LogEpi’

June 3, 2020

Type Package

Title Logistic curves applied to COVID-19

Version 3.0

Author Beatriz Cuyabano and Gabriel Rovere

Maintainer Beatriz Cuyabano <bia.cdc@gmail.com>

Description This packages adjust logistic curves to COVID-19.

Depends R (>= 3.1.0)

Imports plyr,
lubridate,
readxl,
httr,
colorspace

License None

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

R topics documented:

extract_covid19_data	1
load_JH_db	2
logCurve	3
mkEpiComparePlot	3
mkEpiCurves	4
mkEpiTable	6
Index	7

extract_covid19_data	<i>Extract Covid-19 data from a specific location</i>
----------------------	---

Description

This function will extract the Covid-19 data from a specific location, out of the downloaded databases

Usage

```
extract_covid19_data(location, data, folder, filename)
```

Arguments

location	The name of the location of interest. If data="global", location is a country. If data="US", location is a state.
data	The data base where the location is at. Either location="global" or location="US".
folder	The folder in which the downloaded data was saved with load_JH_db().
filename	The standard filename with which the data was saved with load_JH_db().

Value

A list containing the data on confirmed cases, deaths, and recovered individuals. Data for recovered individuals is currently not available for States in the US data.

See Also

[load_JH_db](#)

Examples

```
load_JH_db("Desktop", "covid19JHU")
extract_covid19_data("China", "global", "Desktop", "covid19JHU")
extract_covid19_data("Michigan", "US", "Desktop", "covid19JHU")
```

load_JH_db	<i>Download latest Covid-19 data from JHU database</i>
------------	--

Description

This function will load the latest Covid-19 data for the historical number of confirmed cases, deaths, and recovery. Both global and United States data will be downloaded. The data is provided by Johns Hopkins University (<https://systems.jhu.edu/research/public-health/ncov/>), available at repository <https://github.com/CSSEGISandData/COVID-19>

Usage

```
load_JH_db(folder, filename)
```

Arguments

folder	The folder in which the downloaded data will be saved.
filename	The standard filename to save the data,

Value

countries A list of the countries with records in the database.
US_states A list of the states/provinces in the US with records in the database.

Examples

```
load_JH_db("Desktop", "covid19JHU")
```

logCurve	<i>Fit the logistic curve</i>
----------	-------------------------------

Description

This function will fit the logistic curve for an epidemic event based on the parameters and days count.

Usage

```
logCurve(par, t, X = NULL)
```

Arguments

par	The parameters for the logistic curve.
t	The days counts.
X	Covariates, if considered. Default is X=NULL.

Value

The values of the logistic curve.

See Also

[mkEpiCurves](#)

Examples

```
logCurve("Desktop","covid19JHU")
```

mkEpiComparePlot	<i>Create comparative epidemic plots</i>
------------------	--

Description

This function will make comparative epidemic plots.

Usage

```
mkEpiComparePlot(  
  location,  
  data,  
  folder,  
  filename,  
  cases.count = 100,  
  plot.as.panel = TRUE  
)
```

Arguments

location	The name of the locations of interest.
data	The data base where each location is at. Either "global" or "US".
folder	The folder in which the downloaded data was saved with load_JH_db().
filename	The standard filename with which the data was saved with load_JH_db().
cases.count	The minimum number of cases from which the data should be considered. Default is 100.
plot.as.panel	Logical, defining if plots should be plotted as panel in a single figure, or as individual figures. Default is plot.as.panel=TRUE.

Value

A set of three comparative epidemic plots.

Examples

```
load_JH_db("Desktop", "covid19JHU")
mkEpiComparePlot(location=c("Italy", "Spain", "United Kingdom", "France"), data=rep("global", 4), "Desktop", "covid19JHU")
mkEpiComparePlot(location=c("New York", "New Jersey", "California"), data=rep("US", 3), "Desktop", "covid19JHU")
```

mkEpiCurves

Create epidemic curves

Description

This function will estimate the logistic curves for the Covid-19 epidemic, by default accounting only for the days passed since the start of the epidemic. Covariates can be added to fit the curve.

Usage

```
mkEpiCurves(
  data,
  X = NULL,
  project = NULL,
  rates = NULL,
  plot.curves = TRUE,
  plot.title = NULL,
  plot.as.panel = TRUE,
  country = NULL
)
```

Arguments

data	The epidemic table, in the format of a data frame, with columns: date, confirmed, deaths, recovered.
X	Covariates to add to the logistic curve. Must be a list in which covariates for confirmed, deaths, and recovery are given. Examples: X=list(confirmed=X1, deaths=X2, recovered=X3), X=list(confirmed=X1).

<code>project</code>	An integer indicating how many days ahead should be projected. Default <code>project=NULL</code> indicates that no projection will be done.
<code>rates</code>	A list of rates to project confirmed cases and deaths. If no rates are given, the average rate of the last five days in the data will be used.
<code>plot.curves</code>	Should the curves be plotted?
<code>plot.title</code>	A title for the plot. Recommended to use the location which the analyses refers to.
<code>plot.as.panel</code>	If <code>plot.curves=TRUE</code> , should plots be output in a panel, or individually?
<code>country</code>	If the data belongs to a single country, adding the country's name here will allow to search if data on the health system is available.

Value

par The estimated parameters for the logistic curves.

fit The fitted values.

res The residuals of the model.

rates The epidemic rates.

projection The values projected.

rates_proj The epidemic rates used for the projection.

See Also

[load_JH_db](#)

[extract_covid19_data](#)

[mkEpiTable](#)

Examples

```
load_JH_db("Desktop","covid19JHU")
dataCovid <- extract_covid19_data("China","global","Desktop","covid19JHU")
dataChina <- mkEpiTable(dataCovid,"China")
mkEpiCurves(dataChina)
#-----#
load_JH_db("Desktop","covid19JHU")
dataCovid <- extract_covid19_data("New Zealand","global","Desktop","covid19JHU")
dataNZ <- mkEpiTable(dataCovid,"New Zealand")
mkEpiCurves(dataNZ,plot.title="New Zealand")
mkEpiCurves(dataNZ,plot.title="New Zealand",project=7)
mkEpiCurves(dataNZ,plot.title="New Zealand",project=7,rates=list(confirmed=1.2))
mkEpiCurves(dataNZ,plot.title="New Zealand",project=7,rates=list(deaths=1.1))
mkEpiCurves(dataNZ,plot.title="New Zealand",project=7,rates=list(confirmed=1.2,deaths=1.1))
```

mkEpiTable	<i>Create epidemic table</i>
------------	------------------------------

Description

This function will structure the extracted Covid-19 data from a location. A specific province (or county, state, administrative region, etc) can be determined.

Usage

```
mkEpiTable(data, specific = NULL, daily = FALSE)
```

Arguments

data	The dataset extracted with <code>extract_covid19_data()</code> .
specific	A specific province (or county, state, administrative region, etc). Default is <code>specific=NULL</code> , meaning that all areas in the location will be considered.
daily	Return daily notifications instead of cumulative? Default <code>daily=FALSE</code> .

Value

A data frame with the structured Covid-19 data at the location of interest.

See Also

[load_JH_db](#)
[extract_covid19_data](#)

Examples

```
load_JH_db("Desktop", "covid19JHU")
dataCovid <- extract_covid19_data("China", "global", "Desktop", "covid19JHU")
mkEpiTable(dataCovid, "China")
mkEpiTable(dataCovid, "China", specific="Hubei")
mkEpiTable(dataCovid, "China", specific=c("Hubei", "Guangdong"))
```

Index

`extract_covid19_data`, [1](#), [5](#), [6](#)

`load_JH_db`, [2](#), [2](#), [5](#), [6](#)

`logCurve`, [3](#)

`mkEpiComparePlot`, [3](#)

`mkEpiCurves`, [3](#), [4](#)

`mkEpiTable`, [5](#), [6](#)