Package 'PandemicLP'

September 14, 2021

Type Package

Title Long Term Prediction for Epidemic and Pandemic Data

Version 1.1.0

Date 2021-09-03

Contact CovidLP team <covidlp.team@gmail.com>

Maintainer Guido Alberti Moreira <guidoalber@gmail.com>

Description Implementation of the methodology described in <http://est.ufmg.br/covidlp/home/pt/> which can be also found in help(models). Implemented models are currently the Poisson distribution. The mean function can be the basic generalized logistic form, or the seasonal effect which has under- or over-reporting effects in up to three weekdays, or the two curves form. Bayesian inference is made available through the 'stan' software and its diagnostic functions pool can be used. Plot methods are implemented to mimic the graphics from the 'shiny' app in the URL using the 'plotly' library.

License GPL-3

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0), rstan (>= 2.18.1), rstantools (>= 2.1.1), stats

Imports utils, methods, Rcpp (>= 0.12.0), plotly (>= 4.9.2.1), dplyr (>= 1.0.1), curl (>= 4.3), tidyr (>= 1.1.1), covid19br

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

Suggests knitr, rmarkdown, webshot

URL <http://est.ufmg.br/covidlp/home/pt/>

VignetteBuilder knitr

SystemRequirements GNU make

RoxygenNote 7.1.1

NeedsCompilation yes

Author Débora de Freitas Magalhães [aut],
Marta Cristina Colozza Bianchi da Costa [aut],
Guido Alberti Moreira [cre, aut]
(<https://orcid.org/0000-0001-7557-0874>),
Thais Pacheco Menezes [aut],
Marcos Oliveira Prates [ctb] (<https://orcid.org/0000-0001-8077-4898>)

Repository CRAN

Date/Publication 2021-09-14 16:40:14 UTC

R topics documented:

country_list . . . 3
covid19BH . . . 3
density.pandemicEstimated . . . 5
load_covid . . . 6
models . . . 7
pandemicData-objects . . . 9
pandemicEstimated-objects . . . 9
PandemicLP . . . 10
pandemicPredicted-objects . . . 11
pandemicStats-objects . . . 11
pandemic_model . . . 12
pandemic_stats . . . 16
plot.pandemicData . . . 18
plot.pandemicPredicted . . . 19
plottedPandemic-objects . . . 21
plottedPandemicData-objects . . . 21
posterior_predict.pandemicEstimated . . . 22
print.pandemicEstimated . . . 23
print.pandemicPredicted . . . 24
print.pandemicStats . . . 25
print.plottedPandemic . . . 26
print.plottedPandemicData . . . 26
print.summary.pandemicEstimated . . . 27
state_list . . . 29
traceplot,pandemicEstimated-method . . . 29

Index **31**

country_list *List of countries available in the Covid-19 database*

Description

This function provides a list of country names inside the Covid-19 database. The `country_name` argument inside the `load_covid` function should be spelled as they are listed here.

Usage

```
country_list()
```

See Also

`load_covid`

covid19BH *Covid-19 data for Belo Horizonte/MG - Brazil*

Description

The purpose of this page is to show the user how to format the Y input argument in `pandemic_model` function, when epidemiological data was obtained outside of the `load_covid` function.

The Covid-19 data for the city of Belo Horizonte, MG - Brazil will be used to illustrate how to correctly format the epidemiological data required in the `pandemic_model` function. See the **Examples** section.

For complete information on the required data format, check the `Y` input argument description in `?pandemic_model`.

Usage

```
covid19BH
```

Format

This data frame has 103 observations and 6 variables. It contains the number of Covid-19 confirmed cases and deaths for the city of Belo Horizonte, from the date of the first notified case in 2020-03-16 to 2020-06-26.

1. date - dates in the YYYY-MM-DD format
2. new_confirmed - number of new cases
3. new_deaths - number of new deaths
4. last_available_confirmed - cumulative number of cases
5. last_available_deaths - cumulative number of deaths
6. estimated_population_2019 - size of Belo Horizonte's population

Source

https://brasil.io/dataset/covid19

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

load_covid, pandemic_model, posterior_predict.pandemicEstimated, pandemic_stats and plot.pandemicPredicted.

Examples

```
## formating the data frame for pandemic_model function
covid19BH
names(covid19BH)

data = covid19BH
names(data) = c("date", "new_cases", "new_deaths", "cases", "deaths", "population")

class(data$date)
data$date = as.Date(data$date)
class(data$date)

data      #note:  data frame is ordered by date in descending order!
data = data[order(data$date, decreasing=FALSE), ]
data      #data frame ordered by dates in ascending order

## building the Y list required
pop = data$population[1]
Y = list(data = data, name = "Belo Horizonte/MG", population = pop)

## fitted model:
##pandemic_model function may take a few minutes...
## Not run:
outputBH = pandemic_model(Y, control = list(max_treedepth = 50, adapt_delta = 0.999))
outputBH

summary(outputBH)

##convergence diagnostics
traceplot(outputBH)
density(outputBH)
stan_ac(outputBH$fit, pars = c("a","b","c","f"))

## making predictions
predictions = posterior_predict(outputBH)

## calculating prediction intervals and statistics
stats = pandemic_stats(predictions)
```

```
## plotting results
plot(predictions)
## End(Not run)
```

`density.pandemicEstimated` *Draw estimated density of the parameters for the pandemic model*

Description

Uses stan's stan_dens function to draw the marginal posterior for the relevant parameters of the estimated model. Defined as a method for the stats::density generic function.

Usage

```
## S3 method for class 'pandemicEstimated'
density(x, waves = 1:x$n_waves, ...)
```

Arguments

`x`	Output of the `pandemic_model` function
`waves`	If the estimated model has more than 1 wave, this parameter controls which waves parameters are shown. Default are all waves.
`...`	Additional parameters passed on the `stan_dens` function

See Also

`pandemic_model` and `stan_dens`.

Examples

```
## Not run:
dataMG = load_covid("Brazil","MG")
estimMG = pandemic_model(dataMG)
density(estimMG)
## End(Not run)
```

load_covid *Load Covid-19 Data*

Description

This function pulls Covid-19 data up to a certain date, for a specified country (and state, if `country_name = "Brazil"`).The output of this function is in the correct format to be used directly into the model adjustment function `pandemic_model` included in this package.

Usage

```
load_covid(country_name, state_name = NULL, last_date)
```

Arguments

`country_name`	string specifying the country of interest. Check `country_list()` for the list of countries available in the database.
`state_name`	optional string specifying the state of interest - only brazilian states currently available in the database. `state_name` should be either `NULL` or a string of length 2. Check `state_list()` for the state abbreviations that will be used and the corresponding state names.
`last_date`	optional date, character or factor argument specifying the last date in the data. It should be in the YYYY-MM-DD or YYYY/MM/DD format. The default is the most recent date available in the database.

Details

The current version of this function uses the `covid19br` package to retrieve the data. Be aware that the country names might have been altered between different package versions. Check `country_list()` for the updated list of `country_name` options.

Value

An object of S3 class `pandemicData`. It is a list with 3 items:

`data`**:** data frame with the number of cumulative cases, new cases, cumulative deaths and new deaths associated with Covid-19 for each date, up to the `last_date` in the specified region.

`name`**:** string with the country name (and state name, if available).

`population`**:** numeric object that contains the population size of the given region.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

country_list, state_list, pandemic_model, posterior_predict.pandemicEstimated, pandemic_stats and plot.pandemicPredicted.

Examples

```
## Not run:
load_covid("Brazil","MG")
load_covid(country_name = "India", last_date = "2020-06-15")
load_covid("United States of America")
load_covid(country_name = "italy")
## End(Not run)
```

---

models *Models used in the PandemicLP package*

Description

This document explains the models used in the PandemicLP package in detail.

Growth curve for the mean cases

The count data for number of cases or deaths is modeled according to an epidemiological model of growth. In particular, the average counts are $\mu(t)$ modeled with a generalized logistic curve:

$$\mu(t) = acf\frac{e^{-ct}}{(b+e^{-ct})^{f+1}}.$$

All parameters, that is a, b, c and f are positive.

Parameter c is interpreted as the infection rate. Parameter f controls the asymmetry, so if it is equal to 1, then the curve is symmetric. If it is lesser than 1, then the cases grow slower before the peak than they decrease after. The behavior is inverted when f is greater than 1.

The counts for the Covid-19 pandemic typically had a behavior with positive asymmetry, and so the default for the package functions is to use a greater than 1 truncation for f.

It was common in the early stages of the Covid-19 pandemic that the predictions would result in very high and absurd values for the total number of cases (TNC). It is straightforward to show that

$$TNC = \frac{a}{b^f}.$$

Since all locations displayed a total number of cases that never exceeded 5% of that location's population, another truncation is applied, so that $a \leq b^f 0.08Pop$, where Pop is the location's population.

Probabilistic model

The simplest probabilistic model for the counts is the Poisson model. If y_t is the count at time t, then

$$y_t|\theta \sim Poisson(\mu(t)),$$

where θ represents the model parameters.

Advanced modeling

Here we present some other forms for the growth curve in the mean.

Seasonal effects: A weekly seasonal effect can be added. This is done by multiplying $\mu(t)$ by a positive effect d when t is the desired weekday. If $d < 1$ then that weekday represents underreporting. It is overreporting if $d > 1$.

Multiple curves: Additionally, two curves can be fitted, as happened in the Covid-19 pandemic in many locations. In this case the model is slightly different. In this case,

$$\mu(t) = \mu_1(t) + \mu_2(t)$$

$$\mu_j(t) = a_j c_j \frac{e^{-c_j t}}{(b_j + e^{-c_j t})^2} \Phi(\alpha_j(t - \delta_j)), j = 1, 2,$$

where $\Phi(.)$ is the probit function. The probit function induces asymmetry in the curve, similarly to parameter f.

Prior distribution

Apart from the truncation mentioned above, the prior is defined as below. Note that every available model used in the `pandemic_model` function uses only a subset of these parameters. The priors are described here for reference.

$$a \sim Gamma(0.1, 0.1)$$

$$b \sim LogNormal(0, 20)$$

$$c \sim Gamma(2, 9)$$

$$f \sim Gamma(0.01, 0.01).$$

$$d_k \sim Gamma(2, 1), k = 1, ..., 3$$

$$\delta_j \sim Normal(0, 100), j = 1, 2$$

$$\alpha_j \sim Gamma(0.01, 0.01), j = 1, 2$$

Options for the `pandemic_model` function

Two arguments in the function change the fitted model, as described below:

- 'seasonal_effect': By leaving this argument `NULL`, the standard model is fitted. By supplying it with a vector of up to three weekdays, the desired seasonal effects are added to the model.
- 'n_waves': By leaving this argument equal to 1, the standard model is fitted. By changing it to 2 implies a two waves model. Future versions of the package will allow for more waves.

Seasonal and multiple waves cannot currently be used in the same model at the same time. Future versions of the package will allow such mixtures and also other distributions for the data.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

pandemic_model and posterior_predict.pandemicEstimated.

---

pandemicData-objects *pandemicData objects: Covid-19 Pandemic Data*

Description

The load_covid function returns an object of S3 class pandemicData in a list format containing the components described below.

Elements for pandemicData object

data data frame with the number of cumulative cases, new cases, cumulative deaths and new deaths associated with Covid-19 for each date, up to the last_date in the specified region.

name string with the country name (and state name, if available).

population numeric object that contains the population size of the given region.

---

pandemicEstimated-objects *pandemicEstimated objects: Fitted PandemicLP model*

Description

The **PandemicLP** model-fitting functions return an object of S3 class pandemicEstimated, which is a list containing the components described below.

Elements for pandemicEstimated objects

model_name the model name used.

family string indicating data distribution.

multiwaves number of waves.

seasonal_effect string vector of the weekdays' name with sazonal effect.

cases.type the type of cases of interest used in modeling the epidemic.

config.inputs a list with the main input arguments used in pandemic_model.

priors a list with information on the prior distributions used and model restrictions (if there are any).

fit an object of S4 Class stanfit representing the fitted results via sampling. For additional information about this element fit, see stanfit.

Y a list with the data.

PandemicLP *PandemicLP: Modeling Pandemic Data*

Description

Details

The **PandemicLP** package provides five main functions that enable the user to make short and long-term predictions of epidemic and pandemic count data.
This package is a result of the joint work by professors and graduate students from the Statistics Department at Universidade Federal de Minas Gerais (UFMG). It originated as a challenge in a graduate course after the suspension of classes due to Covid-19.
Theoretical foundation and more information about the project can be found in est.ufmg.br/covidlp/home/en

Author(s)

Authors:

- Debora de Freitas Magalhaes
- Marta Cristina Colozza Bianchi da Costa
- Guido Alberti Moreira
- Thais Pacheco Menezes
- Marcos Oliveira Prates

Examples

```
## Not run:
data = load_covid("Brazil")
plot(data)
estim = pandemic_model(data)
pred = posterior_predict(estim)
stats = pandemic_stats(pred)
plot(pred)
## End(Not run)
```

pandemicPredicted-objects

pandemicPredicted objects: Predictions made from a fitted PandemicLP model

Description

The **PandemicLP** prediction function returns an object of S3 class `pandemicPredicted`, which is a list containing the components described below.

Elements for `pandemicPredicted` objects

`predictive_Long` The full sample of the predictive distribution for the long-term prediction. The prediction is for daily new cases.

`predictive_Short` The full sample of the predictive distribution for the short-term prediction. The prediction is for daily cumulative cases.

`data` The data passed on from the `pandemicEstimated-objects` under the element `Y$data`.

`location` A string with the name of the location.

`cases_type` A string with either "confirmed" or "deaths" to represent the type of data that has been fitted and predicted.

`pastMu` The fitted means of the data for the observed data points.

`futMu` The predicted means of the data for the predicted data points.

pandemicStats-objects

pandemicStats objects: 95% Credible Interval predictions from PandemicLP model

Description

The `pandemic_stats` function returns an object of S3 class `pandemicStats` in a list format containing the components described below.

Elements for `pandemicStats` objects

`data` A list containing a data frame with the pandemic data observed, a string with the location name, and a string indicating whether the cases predicted are confirmed cases or deaths.

`ST_predict` A data frame with a 95% credible interval for the short-term prediction of cumulative cases.

`LT_predict` A data frame with a 95% credible interval for the long-term prediction of new cases.

`LT_summary` A list with a 95% credible interval for the predicted total number of cases, peak and end dates of the pandemic.

`mu` A data frame with the median value of the mean number of new cases for each day. Starting from the first day of the pandemic to the last day of the long-term horizon.

pandemic_model *Bayesian growth curve models for epidemiological data via Stan*

Description

Bayesian inference for modeling epidemiological data or Covid-19 pandemic data using growth curve models. This function draws the posterior samples of the parameters of the growth curve models available in the PandemicLP package. The sampling algorithm is "NUTS", which is the No-U-Turn sampler variant of Hamiltonian Monte Carlo (Hoffman and Gelman 2011, Betancourt 2017).

See which models are available in the PandemicLP package in models.

See posterior_predict.pandemicEstimated to make predictions, pandemic_stats to provide a few useful statistics based on the predictions and plot.pandemicPredicted to plot the predicted values.

Usage

```
pandemic_model(
  Y,
  case_type = "confirmed",
  family = "poisson",
  seasonal_effect = NULL,
  n_waves = 1,
  p = 0.08,
  phiTrunc = 0,
  fTrunc = 1,
  chains = 1,
  warmup = 2000,
  thin = 3,
  sample_size = 1000,
  init = "random",
  ...,
  covidLPconfig = FALSE
)
```

Arguments

Y — an object of class pandemicData-objects created by function load_covid or a list providing the epidemiological data for the model. The elements of this Y list are:

data: a data frame with at least the 2 columns date and new_cases (or new_deaths). It can also contain all following columns:

date: a date vector. It should be of class 'Date' and format 'YYYY-MM-DD'.

cases: a numeric vector with the time series values of the cumulative number of cases.

`new_cases`: a numeric vector with the time series values of the number of new confirmed cases.

`deaths`: a numeric vector with the time series values of the cumulative number of deaths.

`new_deaths`: a numeric vector with the time series values of the number of new deaths.

The data frame should be ordered by date in ascending order.

`name`: a string providing the name of Country/State/Location of the epidemiological data.

`population`: a positive integer specifying the population size of the Country/State/Location selected.

For formatting epidemiological data (not provided by the `load_covid` function) in the specified Y list format, see the ??? function, the vignette ???? or **Examples** section in `covid19BH`.

`case_type` a string providing the type of cases of interest in modelling the epidemic. Current options are `"confirmed"` for confirmed cases or `"deaths"` for deaths. The default is `"confirmed"`. This argument is not required when data frame `Y$data` (on the input argument `Y`) contains only information from one of the data series `new_cases` or `new_deaths`.

`family` "poisson" or "negbin". This argument indicates the data distribution. The default is `family="poisson"`.

`seasonal_effect`

string vector indicating the days of the week in which seasonal effect was observed. The vector can contain the full weekday name (sunday to saturday) or the first 3 letters, up to a maximum of three weekdays. For details go to `models`.

`n_waves` a integer positive. This argument indicates the number of waves to be adjusted by mean curve. The default is 1. For details go to `models`.

`p` a numerical value greater than 0 and less than or equal to 1. It is the percentage of the maximum cumulative total number of cases until the end of the epidemic in relation to the population of the location. The default is `p = 0.08`. This is a model restriction. See more on the `models`.

`phiTrunc` a positive real number (or zero). This argument indicates a truncation on the priori of the 'phi' parameter of the Negative Binomial models. This input argument is required only when `family="negbin"`. The default is `phiTrunc=0`. See more on the `models`.

`fTrunc` a positive real number (or zero). This argument indicates a truncation on the priori of the 'f' parameter of the Negative Binomial model with single wave. This input argument is required only when `family="negbin"`. The default is `fTrunc=1`. See more on the `models`.

`chains` a positive integer specifying the number of Markov chains. The default is `1`, which is default value used by the CovidLP app (http://est.ufmg.br/covidlp/home/en/).

`warmup` a positive integer specifying the number of warmup (aka burnin) iterations per chain. These warmup samples are not used for inference. The default is `2000`, if `family="negbin"` the value default becomes `warmup=5000`.

thin — a positive integer specifying the period for saving samples. The default is 3, which is the default value used by the CovidLP app (http://est.ufmg.br/covidlp/home/en/). .

sample_size — a positive integer specifying the posterior sample's size per chain that will be used for inference. The total number of iterations per chain is:

warmup + thin * sample_size

The default is 1000, which is the default value used by CovidLP app (http://est.ufmg.br/covidlp/home/en/).

init — specification of the initial values of the parameters per chain. The default is "random". Go to models for more info about model parameters. Any parameters whose values are not specified will receive initial values generated as described in init = "random". Specification of the initial values for pandemic_model can only be via list. See the detailed documentation for the init argument via list in stan. Alternatively it can be an output of the pandemic_model() function, which uses the last stored iteration from that object as the initial values. If the models are different, an analogy is made.

... — other arguments passed to the function. These are optional arguments for the sampling (**rstan** package). Additional arguments can be control, cores, etc...

covidLPconfig — TRUE or FALSE: flag indicating whether to use default values of the CovidLP app as input arguments. This argument is disabled when family="negbin".

If covidLPconfig = TRUE, the sampling uses the following configuration: chains = 1, warmup = 5000, thin = 3, sample_size = 1000,

control = list(max_treedepth = 50,adapt_delta = 0.999), p = 0.08 for

case_type = "confirmed" or p = 0.02 for case_type = "deaths", init a list with default initial values for the parameters of each model available.

When using covidLPconfig = TRUE the convergence of the chains is not guaranteed. It only replicates the results of the fitted model with the contemplated data in the CovidLP app (http://est.ufmg.br/covidlp/home/en/). For covidLPconfig = FALSE: each argument will be set to its default value, unless the user specifies otherwise.

Value

An object of S3 Class pandemicEstimated-objects representing the fitted results. The fit component of the pandemicEstimated class is an object of S4 Class stanfit.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

load_covid, posterior_predict.pandemicEstimated, pandemic_stats and plot.pandemicPredicted; summary.pandemicEstimated. See which models are available in the PandemicLP package in models.

Examples

```
##result of the pandemic_model function may take a few minutes

### generalized logistic poisson model: ################
## Not run:
Y0=load_covid(country_name="Brazil",state_name="SP",last_date='2020-04-25')
plot(Y0,cases="new")
output0=pandemic_model(Y0)
print(output0)
#convergence diagnostics
traceplot(output0)
density(output0)
stan_ac(output0$fit,pars=c("a","b","c","f"))

Y1=load_covid(country_name="Brazil",state_name="SP",last_date='2020-06-18')
plot(Y1,cases="new")
output1=pandemic_model(Y1,case_type="deaths",covidLPconfig=TRUE)
print(output1)
#convergence diagnostics
traceplot(output1)
density(output1)
stan_ac(output1$fit,pars=c("a","b","c","f"))


Y2=load_covid(country_name="Argentina",last_date='2020-05-07')
plot(Y2,cases="new")
output2=pandemic_model(Y2,covidLPconfig=TRUE)
print(output2)
#convergence diagnostics
traceplot(output2)
density(output2)
stan_ac(output2$fit,pars=c("a","b","c","f"))


#including initial values for parameters:
inits3=list(
 list(a=95,b=0.8,c=0.3,f=1.1)
)
output3=pandemic_model(Y2,init=inits3,chains=1,warmup=3000)
print(output3)
#convergence diagnostics
traceplot(output3)
density(output3)
stan_ac(output3$fit,pars=c("a","b","c","f"))

#initival values for 2 chains:
inits4=list(
 list(a=95,b=0.8,c=0.3,f=1.1), list(f=1.01)
)
output4=pandemic_model(Y1,init=inits4,chains=2,warmup=3000)
print(output4)
# show all initival values input by user:
```

```
output4$config.inputs$use_inputs$init
#convergence diagnostics
traceplot(output4)
density(output4)
stan_ac(output4$fit,pars=c("a","b","c","f"))

### seasonal model: ###############
output5=pandemic_model(Y0,seasonal_effect=c("sunday","monday"))
print(output5)
#convergence diagnostics
traceplot(output5)
density(output5)
stan_ac(output5$fit,pars=c("a","b","c","f","d_1","d_2"))

## or, for 'seasonal_effect': strings vector with the 3 initial letters of the weekday(s)
Y3=load_covid(country_name="Brazil",state_name="MG",last_date='2020-09-05')
plot(Y3,cases="new")
#weekdays effect : sunday and monday:
output6=pandemic_model(Y3,seasonal_effect=c("sun","mon"),covidLPconfig=TRUE)
print(output6)
#convergence diagnostics
traceplot(output6)
density(output6)
stan_ac(output6$fit,pars=c("a","b","c","f","d_1","d_2"))

### multi_waves(2) model: ######################
Y4=load_covid(country_name="United States of America",last_date='2020-09-27')
plot(Y4,cases="new")
output7=pandemic_model(Y4,n_waves=2,covidLPconfig=TRUE)
print(output7)
#convergence diagnostics
traceplot(output7)
density(output7)
stan_ac(output7$fit,pars=c("a1","b1","c1","alpha1","delta1","a2","b2","c2","alpha2","delta2"))

## End(Not run)
```

---

pandemic_stats *Relevant Statistics of the Pandemic Model*

Description

This function provides short and long-term predictions for the pandemic. 95% credible intervals are assigned to the number of cases for every future date predicted, as well as for the total number of cases, and dates for the peak and end of the pandemic.

Short-term predictions are made on the cumulative counts and long-term predictions are based on the new case counts.

Usage

```
pandemic_stats(object)
```

Arguments

object	an object of S3 class pandemicPredicted created by method posterior_predict.pandemicEstimated of function posterior_predict.

Details

Total Number of Cases: The total number of cases is obtained by adding the cumulative total cases observed in the data to the predicted new cases for the next 1,000 days.

Estimated Peak Dates: The peak of the pandemic curve represents the highest number of daily cases reported.

The median, 2.5% and 97.5% percentiles are calculated on the mean number of new cases for the pandemic curve (starting from the first observed data point until 1,000 days after the last date observed in the data). The 95% credible interval for the peak of cases is selected such that the two limiting dates of the 97.5% percentile curve coincides with the highest value of the 2.5% percentile curve. This guarantees that all possible curves belonging to the confidence band will peak within the defined interval.

End of the Pandemic Dates: Represents the 99% percentile of the total number of cases in the pandemic.

Value

An object of S3 class pandemicStats. This object is a list containing the following elements:

- data: A list with a data frame containing the observed pandemic data, a string with the location name and a string indicating the type of cases predicted.
- ST_predict: A data frame with the short-term predictions for the number of cumulative cases. For each future date predicted, the mean, median, 2.5% and 97.5% percentiles are provided. The short-term horizon is determined by the horizonShort argument in the posterior_predict function.
- LT_predict: A data frame with the long-term predictions for the number of new cases. For each future date predicted, the mean, median, 2.5% and 97.5% percentiles are provided. The long-term horizon is determined by the horizonLong argument in the posterior_predict function.
- LT_summary: A list with the estimated total number of cases and the dates for the peak and end of the pandemic. In each metric, the median, 2.5% and 97.5% percentiles are provided. For more information, see the **Details** section.
- mu: A data frame with the median values of the mean number of new cases for each date (starting from the first observed data point until the last date in the long-term horizon).

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

load_covid, pandemic_model, posterior_predict.pandemicEstimated and plot.pandemicPredicted.

Examples

```
## Not run:
italy = load_covid("italy")
estim = pandemic_model(italy, case_type = "confirmed", covidLPconfig = TRUE)
pred = posterior_predict(estim)
stats = pandemic_stats(pred)
stats

## End(Not run)
```

---

plot.pandemicData *Plot pandemic data*

Description

S3 method that plots the predicted data into an interactive graphic.

Usage

```
## S3 method for class 'pandemicData'
plot(x, y, cases = "new", color = TRUE, ...)
```

Arguments

x	Output of function load_covid.
y	This parameter does nothing
cases	A string which indicates whether new cases, cumulative cases or both plots should be generated. The argument must be a string being either 'new', 'cumulative' or 'both'.
color	A string which indicates whether the plot should be colorful or in gray scales. The argument must be a string being either TRUE or FALSE.
...	Currently unused.

Value

A list containing two objects, new and cumulative "new" shows the plot for the new cases/deaths and "cumulative" shows the plot for the cumulative cases/deaths. If any of them did not get plotted due to lack of prediction or due to the `cases` argument, its value will return NULL.

By default, only the plot with new cases is generated. This is changed with the `cases` argument.

`new`	The plotted new cases/deaths. The plots are for daily new confirmed cases and daily new deaths.
`cumulative`	The plotted cumulative cases/deaths. The plots are for daily cumulative cases or daily cumulative deaths.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

load_covid.

Examples

```
## Not run:
dataMG = load_covid(country_name="Brazil",state_name = "MG",last_date="2020-10-01")
plot(dataMG)
dataJapan = load_covid(country_name="Japan",last_date="2020-10-01")
plot(dataJapan)

## End(Not run)
```

`plot.pandemicPredicted` *Plot pandemic predictions*

Description

S3 method that plots the predicted data into an interactive graphic.

Usage

```
## S3 method for class 'pandemicPredicted'
plot(x, y, term = "long", color = TRUE, summary = TRUE, ...)
```

Arguments

x	Output of function posterior_predict.pandemicEstimated.
y	This parameter does nothing
term	A string which indicates whether long term, short term or both plots should be generated. The argument must be a string being either 'long', 'short' or 'both'.
color	A logical variable indicating whether the plot should be colorful or in gray scales. If TRUE then the plot is colorful otherwise the plot will be gray.
summary	A logical variable indicating whether the plot should contains the summary statistics about the pandemic or not. If TRUE then the plot shows the statistics otherwise no.
...	Currently unused.

Value

A list containing two objects, short and long. "long" shows the long-term predictions and "short" the short-term predictions. If any of them did not get plotted due to lack of prediction or due to the term argument, its value will return NULL.

By default only the long-term plot is generated, which means that the short term plot returns NULL. This is changed with the term argument.

long	The plotted long-term prediction. The predictions are made on daily new confirmed cases or daily new deaths.
short	The plotted short-term prediction. The predictions are made on daily cumulative cases or daily cumulative deaths.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

posterior_predict.pandemicEstimated, pandemic_stats and plottedPandemic-objects.

Examples

```
## Not run:
dataMG = load_covid("Brazil","MG")
estimMG = pandemic_model(dataMG)
predMG = posterior_predict(estimMG)
statsMG = pandemic_stats(predMG)
plot(predMG)
## End(Not run)
```

plottedPandemic-objects

plottedPandemic objects: Plots for pandemic data using the plotly library

Description

The **PandemicLP** plotting function returns an object of S3 class `plottedPandemic`, which is a list containing the components described below.

Elements for `plottedPandemic` objects

`long` The plotted long-term prediction.

`short` The plotted short-term prediction.

plottedPandemicData-objects

plottedPandemicData objects: Plots for pandemic data using the plotly library

Description

The **PandemicLP** plotting function returns an object of S3 class `plottedPandemicData`, which is a list containing the components described below.

Elements for `plottedPandemicData` objects

`new` The plotted new cases/deaths.

`cumulated.` The plotted cumulated cases/deaths.

`posterior_predict.pandemicEstimated` *Draw from the posterior predictive distribution for pandemic data*

Description

The posterior predictive distribution is the distribution of the outcome implied by the model after using the observed data to update our beliefs about the unknown parameters in the model. Simulating data from the posterior predictive distribution using the observed predictors is useful for checking the fit of the model. Drawing from the posterior predictive distribution at interesting values of the predictors also lets us visualize how a manipulation of a predictor affects (a function of) the outcome(s). With new observations of predictor variables we can use the posterior predictive distribution to generate predicted outcomes.

Usage

```
## S3 method for class 'pandemicEstimated'
posterior_predict(object, horizonLong = 500, horizonShort = 14, ...)
```

Arguments

`object`	An object of class `pandemicEstimated` created by function `pandemic_model`.
`horizonLong`	How far into the future the long-term prediction is desired.
`horizonShort`	How far into the future the short-term prediction is desired.
`...`	Currently unused.

Value

An object of class `pandemicPredicted`. It includes the sampled predictive distribution the model used to predict, which is the same as the one used to estimate the data. This object can be used directly into the plot function and contains the following elements:

`predictive_Long`
: A `M` x `horizonLong` matrix with the full sample of the predictive distribution for the long-term prediction, where M is the sample size. The prediction is for daily new cases.

`predictive_Short`
: A `M` x `horizonShort` matrix with the full sample of the predictive distribution for the short-term prediction, where M is the sample size. The prediction is for daily cumulative cases.

`data`
: The data passed on from the `pandemicEstimated-objects` under the element `Y$data`.

`location`
: A string with the name of the location.

`cases_type`
: A string with either "confirmed" or "deaths" to represent the type of data that has been fitted and predicted.

pastMu	The fitted means of the data for the observed data points.
futMu	The predicted means of the data for the predicted data points.

Function pandemic_stats provides a few useful statistics based on the predictions.

References

CovidLP Team, 2020. CovidLP: Short and Long-term Prediction for COVID-19. Departamento de Estatistica. UFMG, Brazil. URL: http://est.ufmg.br/covidlp/home/en/

See Also

pandemic_model, pandemic_stats and plot.pandemicPredicted. Details about the models behind the calculations can be seen in models.

Examples

```
## Not run:
dataMG = load_covid("Brazil","MG")
estimMG = pandemic_model(dataMG)
predMG = posterior_predict(estimMG)
predMG
## End(Not run)
```

---

print.pandemicEstimated

Print method for pandemicEstimated *objects*

Description

The print method for pandemicEstimated object of class S3 displays a compact summary of the fitted model. See the **Details** section below for descriptions of the different components of the printed output. For additional summary statistics and diagnostics use summary.pandemicEstimated.

Usage

```
## S3 method for class 'pandemicEstimated'
print(x, digits = 3, probs = c(0.025, 0.5, 0.975), info = TRUE, ...)
```

Arguments

x	an object of S3 class pandemicEstimated-objects.
digits	Number of digits to use for formatting numbers.
probs	a numeric vector of quantiles of interest. The default is c(0.025,0.5,0.975).
info	TRUE or FALSE: more details for output interpretation. The Default is TRUE.
...	currently unused.

Details

Point estimates: Regardless of the estimation algorithm, point estimates are mean and (or) quantiles computed from simulations. For models fit using MCMC ("sampling", this is default algorithim of `pandemic_model` function), the posterior sample is used. For others estimation algorithm see `sampling` (**rstan** package).

Convergence and efficiency diagnostics for Markov Chains:

Included in the print are: split effective sample sizes (n_eff) and split Rhats.

The R-hat convergence diagnostic compares the between- and within-chain estimates for model parameters and other univariate quantities of interest. If chains have not mixed well (ie, the between- and within-chain estimates don't agree), R-hat is larger than 1. We recommend running at least four chains by default and only using the sample if R-hat is less than 1.05.

Priors:

A list with information about the prior distributions used and model restrictions (if there are any). For more information go to `models`.

Value

Returns `x`, invisibly.

See Also

`summary.pandemicEstimated`.

`print.pandemicPredicted`

Prints prediction summary of pandemic model

Description

S3 method designed to summarize the prediction obtained by the `posterior_predict.pandemicEstimated` function. It is not necessary to call function directly, unless it is desired to change the default arguments.

Usage

```
## S3 method for class 'pandemicPredicted'
print(x, summaryFun = stats::median, printPred = "Long", truncView = 3, ...)
```

Arguments

`x`	An object of S3 class `pandemicPredicted-objects`.
`summaryFun`	Use this function to summarize the predictions. Default argument is median, but can be any function on a vector.
`printPred`	Valid values are 'Long' or 'Short'. Note that 'Short' will show short-term cumulative predictions, while 'Long' will return the long-term daily cases predicted.

truncView	How many predictions to print. Default is 3, which means that the method prints the first 3 days predicted and the last 3 days of the prediction horizon.
...	Currently unused.

See Also

posterior_predict.pandemicEstimated and plot.pandemicPredicted.

Examples

```
## Not run:
dataMG = load_covid("Brazil","MG")
estimMG = pandemic_model(dataMG)
predMG = posterior_predict(estimMG)
print(predMG, summaryFun = mean, truncView = 5)
## End(Not run)
```

---

print.pandemicStats *Print method for* pandemicStats *objects*

Description

S3 method designed for pandemicStats objects. It displays a compact summary of the 95% credible intervals for the predictions calculated by the pandemic model.

Usage

```
## S3 method for class 'pandemicStats'
print(x, ...)
```

Arguments

x	An object of S3 class pandemicStats-objects.
...	Currently unused.

See Also

pandemic_stats

print.plottedPandemic *Print method for* plottedPandemic *objects*

Description

The print method for a plottedPandemic S3 class object displays the single plot or both separated by a prompt for the user.

Usage

```
## S3 method for class 'plottedPandemic'
print(x, ...)
```

Arguments

x	An object of S3 class plottedPandemic-objects.
...	Currently unused.

Details

The plots generated by plot.pandemicPredicted are displayed by this function If only one of the plots has been generated via the argument term, then it is displayed in the Viewer windows as an html object, produced by the plot_ly function. If both long term and short term plots are requested, then the long term plot is displayed first and a message prompts the user to display the next plot, namely the short term one.

Value

Returns x, invisibly.

See Also

posterior_predict.pandemicEstimated and plot.pandemicPredicted

print.plottedPandemicData *Print method for* plottedPandemicData *objects*

Description

The print method for a plottedPandemic S3 class object displays the single plot or both separated by a prompt for the user.

Usage

```
## S3 method for class 'plottedPandemicData'
print(x, ...)
```

Arguments

x	An object of S3 class plottedPandemicData-objects.
...	Currently unused.

Details

The plots generated by plot.pandemicData are displayed by this function If only one of the plots has been generated via the argument cases, then it is displayed in the Viewer windows as an html object, produced by the plot_ly function. If both new and cumulative plots are requested, then the new plot is displayed first and a message prompts the user to display the next plot, namely the cumulative one. The plot shows the confirmed cases and the deaths information, but if the user desire to see only one, he can double click on legend to isolate one trace. Consequently, the axis will be automatically configured to the scale of the desired trace. To see both again, the user just need to double click again.

Value

Returns x, invisibly.

See Also

load_covid and plot.pandemicData

---

print.summary.pandemicEstimated

Summary method for pandemicEstimated *objects*

Description

The summary method for pandemicEstimated object of class S3 displays a compact summary of the fitted model. See the **Details** section below for descriptions of the different components of the printed output.

Usage

```
## S3 method for class 'summary.pandemicEstimated'
print(x, ...)

## S3 method for class 'pandemicEstimated'
summary(object, probs = c(0.025, 0.5, 0.975), digits = 3, info = TRUE, ...)
```

Arguments

x	an object of class "summary.pandemicEstimated".
...	Currently unused.
object	An object of S3 class pandemicEstimated-objects.
probs	A numeric vector of quantiles of interest. The default is c(0.025,0.5,0.975).
digits	Number of digits to use for formatting numbers.
info	TRUE or FALSE: more details for output interpretation. The Default is TRUE.

Details

Point estimates: Mean and Quantiles computed from simulations. For models fit using MCMC ("sampling", this is default algorithim of pandemic_model function), the posterior sample is used. For others estimation algorithm see sampling (**rstan** package).

Convergence and efficiency diagnostics for Markov Chains:

Included in the summary are: split effective sample sizes (n_eff), Monte Carlo standard errors (se_mean) and split Rhats.

The Monte Carlo standard error provides relevant information for a posterior sample with more than one chain.

The R-hat convergence diagnostic compares the between- and within-chain estimates for model parameters and other univariate quantities of interest. If chains have not mixed well (ie, the between- and within-chain estimates don't agree), R-hat is larger than 1. We recommend running at least four chains by default and only using the sample if R-hat is less than 1.05.

covidLPconfig: This subsection shows the main input settings used by the fitted model, and indicates whether default settings of the CovidLP app (http://est.ufmg.br/covidlp/home/en/) were used (covidLPconfig = TRUE or FALSE). Check the default settings of the CovidLP app in pandemic_model.

Priors:

A list with information about the prior distributions used and model restrictions (if there are any). For more information go to models.

Value

The summary method returns an object of class "summary.pandemicEstimated", which is a list with two arrays of summary statistics and diagnostics and others informations for use by the print method. The print method for summary.pandemicEstimated objects is called for its side effect and just returns its input.

Examples

```
## Not run:
Y0=load_covid(country_name="Brazil",state_name="SP",last_date='2020-04-25')
output0=pandemic_model(Y0)
s=summary(output0)
s          #print method for summary.pandemicEstimated
```

```
names(s)  #see output list elements of summary

## End(Not run)
```

`state_list` *List of states available in the Covid-19 database*

Description

This function provides the state abbreviations to be used in the `load_covid` function and the corresponding state names. Only brazilian states are currently available.

Usage

```
state_list()
```

See Also

`load_covid` and `country_list`.

`traceplot,pandemicEstimated-method`
Draw traceplot of the parameters for the pandemic model

Description

Uses stan's traceplot function to draw the traceplots for the relevant parameters of the estimated model.

Usage

```
## S4 method for signature 'pandemicEstimated'
traceplot(object, waves = 1:object$n_waves, ...)
```

Arguments

`object`	Output of the `pandemic_model` function
`waves`	If the estimated model has more than 1 wave, this parameter controls which waves parameters are shown. Default are all waves.
`...`	Aditional parameters passed on to the `traceplot` function

See Also

`pandemic_model` and `traceplot`

Examples

```
## Not run:
dataMG = load_covid("Brazil","MG")
estimMG = pandemic_model(dataMG)
traceplot(estimMG)
## End(Not run)
```

Index

∗ **datasets**
covid19BH, 3

country_list, 3, *7*, *29*
covid19BH, 3, *13*

density.pandemicEstimated, 5

load_covid, *3, 4*, 6, *9*, *12*, *14*, *18, 19*, *27*, *29*

models, 7, *12–14*, *23, 24*, *28*

pandemic_model, *3–9*, 12, *14*, *18*, *22, 23*, *28, 29*
pandemic_stats, *4*, *7*, *11, 12*, *14*, 16, *20*, *23*, *25*
pandemicData-objects, 9
pandemicEstimated-objects, 9
PandemicLP, 10
pandemicPredicted-objects, 11
pandemicStats-objects, 11
plot.pandemicData, 18, *27*
plot.pandemicPredicted, *4*, *7*, *12*, *14*, *18*, 19, *23*, *25, 26*
plot_ly, *26, 27*
plottedPandemic-objects, 21
plottedPandemicData-objects, 21
posterior_predict.pandemicEstimated, *4*, *7*, *9*, *12*, *14*, *17, 18*, *20*, 22, *24–26*
print.pandemicEstimated, 23
print.pandemicPredicted, 24
print.pandemicStats, 25
print.plottedPandemic, 26
print.plottedPandemicData, 26
print.summary.pandemicEstimated, 27

sampling, *9*, *14*, *24*, *28*
stan, *14*
stan_dens, *5*
stanfit, *9*, *14*
state_list, *7*, 29
summary.pandemicEstimated, *14*, *23, 24*
summary.pandemicEstimated *(*print.summary.pandemicEstimated*)*, *27*

traceplot, *29*
traceplot,pandemicEstimated-method, 29