Package 'bets.covid19'

May 12, 2020

Title The BETS Model for Early Epidemic Data

Version 1.0.0

Date 2020-05-07

Description

Implements likelihood inference for early epidemic analysis. BETS is short for the four key epidemiological events being modeled: Begin of exposure, End of exposure, time of Transmission, and time of Symptom onset. The package contains a dataset of the trajectory of confirmed cases during the coronavirus disease (COVID-19) early outbreak. More detail of the statistical methods can be found in Zhao et al. (2020) <arXiv:2004.07743>.

Depends R (>= 3.4.0),

Imports stats, rootSolve, parallel

License CC BY 4.0

URL https://github.com/qingyuanzhao/bets.covid19

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

NeedsCompilation no

Author Qingyuan Zhao [aut, cre],
Nianqiao Ju [aut]

Maintainer Qingyuan Zhao <qyzhao@statslab.cam.ac.uk>

Repository CRAN

Date/Publication 2020-05-12 09:50:06 UTC

R topics documented:

1

Index   **10**

---

age.process   *Processing age to print its distribution*

Description

Processing age to print its distribution

Usage

```
age.process(age)
```

Arguments

age   a vector of age, each entry is either a number (like 34) or age by decade (like
                   30s)

Value

each age is either repeated 10 times or expanded to 10 numbers (for example, 30s is expanded to
30, 31, ..., 39).

---

bets.covid19   *A package for analyzing early epidemic data*

Description

The BETSbets.covid19 package provides likelihood inference for early epidemic data with four
key epidemiological events: Beginning of exposure, End of exposure, time of Transmission, and
time of Symptom onset. It jointly estimates the epidemic doubling time and incubation period and
is able to correct for different kinds of sample selection.

References

Qingyuan Zhao, Nianqiao Ju, Sergio Bacallado, and Rajen Shah. "BETS: The dangers of selection
bias in early analyses of the coronavirus disease (COVID-19) pandemic", 2020. arXiv:2004.07743.

---

bets.inference *Likelihood inference*

Description

Likelihood inference

Usage

```
bets.inference(
  data,
  likelihood = c("conditional", "unconditional"),
  ci = c("lrt", "point", "bootstrap"),
  M = Inf,
  r = NULL,
  L = NULL,
  level = 0.95,
  bootstrap = 1000,
  mc.cores = 1
)
```

Arguments

data	A data.frame with three columns: B, E, S.
likelihood	Conditional on B and E?
ci	How to compute the confidence interval?
M	Right truncation for symptom onset (only available for conditional likelihood)
r	Parameter for epidemic growth (overrides {params}, only available for conditional likelihood)
L	Time of travel restriction (required for unconditional likelihood)
level	Level of the confidence interval (default 0.95).
bootstrap	Number of bootstrap resamples.
mc.cores	Number of cores used for computing the bootstrap confidence interval.

Details

The confidence interval is either not computed ("point"), or computed by inverting the likelihood ratio test ("lrt") or basic bootstrap ("bootstrap")

Value

Results of the likelihood inference, including maximum likelihood estimators and individual confidence intervals for the model parameters based on inverting the likelihood ratio test.

Examples

```
data(wuhan_exported)

data <- subset(wuhan_exported, Location == "Hefei")
data$B <- data$B - 0.75
data$E <- data$E - 0.25
data$S <- data$S - 0.5

# Conditional likelihood inference
bets.inference(data, "conditional")
bets.inference(data, "conditional", "bootstrap", bootstrap = 100, level = 0.5)

# Unconditional likelihood inference
bets.inference(data, "unconditional", L = 54)

# Conditional likelihood inference for data with right truncation
bets.inference(subset(data, S <= 60), "conditional", M = 60)

# Conditional likelihood inference with r fixed at 0 (not recommended)
bets.inference(data, "conditional", r = 0)
```

---

bets.likelihood   *(Profile) Likelihood function*

Description

(Profile) Likelihood function

Usage

```
bets.likelihood(
  params,
  data,
  likelihood = c("conditional", "unconditional"),
  M = Inf,
  r = NULL,
  L = NULL,
  params_init = NULL
)
```

Arguments

params   A vector of parameters (with at least one of the following entries: rho, r, ip_q50, ip_q95)

data	A data frame with three columns: B, E, S
likelihood	Use the conditional or unconditional likelihood function
M	Right truncation for symptom onset
r	Parameter for epidemic growth (overrides {params})
L	Day of travel quarantine
params_init	Initial parameters for computing the profile likelihood

Details

Non-default values of M and r are only available for conditional likelihood.

Value

Log-likelihood function if params has all four entries, rho, r, ip_q50, ip_q95 (or three entires—r, ip_q50, ip_q95—if computing the conditional likelihood). Otherwise returns the profile likelihood for the parameters in params.

Examples

```
data(wuhan_exported)

data <- wuhan_exported
data$B <- data$B - 0.75
data$E <- data$E - 0.25
data$S <- data$S - 0.5

params <- c(r = 0.2,
            ip_q50 = 5,
            ip_q95 = 12)

# Conditional likelihood
bets.likelihood(params, data)

# Conditional likelihood with right truncation
bets.likelihood(params, subset(data, S <= 60), M = 60)

# Conditional likelihood with fixed r (not recommended)
bets.likelihood(params, data, r = 0)

# Unconditional likelihood
params["rho"] <- 1
bets.likelihood(params, data, likelihood = "unconditional", L = 54)

# Profile conditional likelihood
bets.likelihood(c(r = 0.2), data, params_init = params)
```

covid19_data   *Confirmed cases of COVID-19*

Description

A dataset containing the trajectory of cases of COVID-19.

Usage

    covid19_data

Format

A data frame with 1091 rows and 20 variables:

Case Label of the case, in the format of Country-Case number.

Nationality/Residence Nationality or residence of the patient.

Gender Male (M) or Female (F).

Age Age of the patient, either an integer or age by decade (for example, 40s).

Cluster Other confirmed cases that this patient had contacts with.

Known Contact Whether the case has contact with earlier confirmed cases or visited Hubei province.

Outside Was the patient infected outside Wuhan? Yes (Y), Likely (L), or No (empty string and the default).

Begin_Wuhan Begin of stay in Wuhan.

End_Wuhan End of Stay in Wuhan.

Infected When was the patient infected? Can be an interval or multiple dates.

Arrived When did the patient arrive in the country where he/she was confirmed a 2019-nCoV case?

Symptom When did the patient first show symptoms of 2019-nCoV (cough, fever, fatigue, etc.)?

Initial After developing symptoms, when was the patient first went to (or taken to) a medical institution?

Hospital If the patient was not admitted to or isolated in a hospital after the initial medical visit, when was the patient finally admitted or isolated?

Confirmed When was the patient confirmed as a case of 2019-nCoV?

Discharged When was the patient discharged from hospital?

Death When did the patient die?

Verified Has this information been verified by another data collector?

Source URLs to the information recorded (usually government websites or news reports).

---

date.process   *Transform date to numeric*

Description

Transform date to numeric

Usage

```
date.process(date)
```

Arguments

date   a vector of dates of the form "DD-MMM" (for example, 23-Jan).

Value

a vector of days since December 1st, 2019 (or example, 23-Jan is converted to 23+31 = 52).

---

preprocess.data   *Prepare data frame for analysis*

Description

Prepare data frame for analysis

Usage

```
preprocess.data(
  data,
  infected_in = c("Wuhan", "Outside"),
  symptom_impute = FALSE
)
```

Arguments

data   A data frame

infected_in   Either "Wuhan" or "Outside"

symptom_impute   Whether to use initial medical visit and confirmation to impute missing symptom onset.

Details

A summary of the procedures:

1. Convert all dates to number of days since 1-Dec-2019.
2. Separates data into those returned from Wuhan and those infected outside of wuhan.
3. Restrict to cases with a known symptom onset date.
4. Parse column 'Infected' into two columns: Infected_first and Infected_last.
5. For all cases, set Infected_first to 1 if it is missing.
6. For outside cases, set Infected_last to be no later than symptom onset.
7. For Wuhan-exported cases, set Infected_last to no later than symptom onset and end of Wuhan stay.

Value

A data frame

Author(s)

Nianqiao Ju <nju@g.harvard.edu>, Qingyuan Zhao <qyzhao@statslab.cam.ac.uk>

Examples

```
data(covid19_data)
head(data <- preprocess.data(covid19_data))

 ## This is how the wuhan_exported data frame is created
data <- subset(data, Symptom < Inf)
data <- subset(data, Arrived <= 54)
data$Location <- do.call(rbind, strsplit(as.character(data$Case), "-"))[, 1]
wuhan_exported <- data.frame(Location = data$Location,
                             B = data$Begin_Wuhan,
                             E = data$End_Wuhan,
                             S = data$Symptom)
## devtools::use_data(wuhan_exported)
```

---

wuhan_exported   *COVID-19 exported from Wuhan*

Description

Constructed from `covid19_data`, see `example(preprocess.data)`.

Usage

```
wuhan_exported
```

Format

A data frame with 378 rows and 4 variables:

Location Where the case is confirmed.

Gender Gender of the patient.

Age Age of the patient.

B Beginning of stay in Wuhan.

E End of stay in Wuhan.

S Symptom onset.

Index