

# Package ‘aedd0’

November 27, 2023

**Title** Automated and Early Detection of Disease Outbreaks

**Version** 0.1.1

**Description** A powerful tool for automating the early detection of disease outbreaks in time series data. ‘aedd0’ employs advanced statistical methods, including hierarchical models, in an innovative manner to effectively characterize outbreak signals. It is particularly useful for epidemiologists, public health professionals, and researchers seeking to identify and respond to disease outbreaks in a timely fashion. For a detailed reference on hierarchical models, consult Henrik Madsen and Poul Thyregod’s book (2011), ISBN: 9781420091557.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** knitr, MASS, rmarkdown, testthat (>= 3.0.0), zoo

**Config/testthat/edition** 3

**Imports** ggplot2, checkmate, dplyr, lifecycle, magrittr, purrr, tibble, tidyr, utils, rlang

**URL** <https://ssi-dk.github.io/aedd0/>, <https://github.com/ssi-dk/aedd0>

**VignetteBuilder** knitr

**BugReports** <https://github.com/ssi-dk/aedd0/issues>

**NeedsCompilation** no

**Author** Kasper Schou Telkamp [aut] (<<https://orcid.org/0009-0001-5126-0190>>),  
Lasse Engbo Christiansen [aut, cre]  
(<<https://orcid.org/0000-0001-5019-1931>>),  
Statens Serum Institut, SSI [cph, fnd]

**Maintainer** Lasse Engbo Christiansen <lsec@ssi.dk>

**Repository** CRAN

**Date/Publication** 2023-11-27 10:10:08 UTC

## R topics documented:

aeddo	2
autoplot	4
nll_poisson_gamma	5
plot.aeddo	6

<b>Index</b>	<b>8</b>
--------------	----------

---

aeddo	<i>Automated and Early Detection of Disease Outbreaks</i>
-------	---

---

### Description

#### [Stable]

This function performs automated an early detection of disease outbreaks, (aeddo), on a time series data set. It utilizes hierarchical models in an innovative manner to infer one-step ahead random effects. In turn, these random effects are used directly to characterize an outbreak.

### Usage

```
aeddo(
  data = data.frame(),
  formula = formula(),
  k = integer(),
  sig_level = 0.95,
  exclude_past_outbreaks = TRUE,
  init_theta = numeric(),
  lower = numeric(),
  upper = numeric(),
  method = "BFGS"
)
```

### Arguments

data	A tibble containing the time series data, including columns 'y' for observed values, 'n' for population size, and other covariates of interest.
formula	A model formula for the fixed effects in the hierarchical model to fit to the data.
k	An integer specifying the rolling window size employed for parameter estimation.
sig_level	The quantile from the random effects distribution used for defining the for outbreak detection threshold, a numeric value between 0 and 1.
exclude_past_outbreaks	logical value indicating whether past outbreak related observations should be excluded from future parameter estimation.
init_theta	Initial values for model parameters in optimization.

lower	Lower bounds for optimization parameters.
upper	Upper bounds for optimization parameters.
method	The optimization method to use, either "BFGS" (default) or "L-BFGS-B".

### Value

A [tibble](#)-like 'aedseo' object containing:

- 'window\_data': A list of [tibble](#), each representing the data for this windowed parameter estimation.
- 'reference\_data': A list of [tibble](#), each representing the data for the reference time point.
- 'phi': The dispersion parameter.
- 'lambda': The estimated outbreak intensity.
- 'u': The one-step ahead random effect.
- 'u\_probability': The probability of observing the one-step ahead random effect.
- 'outbreak\_alarm': Logical. Indicates if an outbreak is detected.

### Examples

```
# Create an example aedseo_tsd object
aeddo_data <- data.frame(
  time = as.Date(c(
    "2023-01-01",
    "2023-01-02",
    "2023-01-03",
    "2023-01-04",
    "2023-01-05",
    "2023-01-06"
  )),
  y = c(100, 120, 180, 110, 130, 140),
  n = 1
)

# Supply a model formula
fixed_effects_formula <- y ~ 1

# Choose a size for the rolling window
k <- 2
# ... and quantile for the threshold
sig_level <- 0.9

# Employ the algorithm
aeddo_results <- aeddo(
  data = aeddo_data,
  formula = fixed_effects_formula,
  k = k,
  sig_level = sig_level,
  exclude_past_outbreaks = TRUE,
  init_theta = c(1, 0),
```

```
lower = c(-Inf, 1e-6),
upper = c(Inf, 1e2),
method = "L-BFGS-B"
)
# Print the results
print(aeddo_results)
```

---

autoplot

*Create a complete 'ggplot' appropriate to a particular data type*

---

## Description

### [Stable]

This function generates a complete 'ggplot' object suitable for visualizing time series data in an aeddo object. It creates a line plot connecting the observations and adds points at each data point.

## Usage

```
autoplot(object, ...)

## S3 method for class 'aeddo'
autoplot(object, ...)
```

## Arguments

object	An aeddo object
...	Additional arguments (not used).

## Value

A 'ggplot' object for visualizing the time series data.

## Examples

```
# Create an example aeddo object
aeddo_data <- data.frame(
  time = as.Date(c(
    "2023-01-01",
    "2023-01-02",
    "2023-01-03",
    "2023-01-04",
    "2023-01-05",
    "2023-01-06"
  )),
  y = c(100, 120, 180, 110, 130, 140),
  n = 1
)

# Supply a model formula
```

```

fixed_effects_formula <- y ~ 1

# Choose a size for the rolling window
k <- 2
# ... and quantile for the threshold
sig_level <- 0.9

# Employ the algorithm
aeddo_results <- aeddo(
  data = aeddo_data,
  formula = fixed_effects_formula,
  k = k,
  sig_level = sig_level,
  exclude_past_outbreaks = TRUE,
  init_theta = c(1, 0),
  lower = c(-Inf, 1e-6),
  upper = c(Inf, 1e2),
  method = "L-BFGS-B"
)

# Create a ggplot visualization for the aeddo object
autoplot(aeddo_results)

```

---

nll\_poisson\_gamma

*Negative Log-Likelihood for Poisson Gamma Model*


---

## Description

**[Stable]**

Calculate the negative log-likelihood for the Poisson Gamma modeling framework.

## Usage

```
nll_poisson_gamma(theta, data, formula)
```

## Arguments

theta	A numeric vector containing model parameters. The first part of the vector represents fixed effects, and the remaining part represents model parameters.
data	A tibble containing the time series data, including columns 'y' for observed values, 'n' for population size, and other covariates of interest.
formula	A formula specifying the model structure.

## Value

The negative log-likelihood value.

## Examples

```
# Initial parameters
theta <- c(0.5, 0.1)

# Sample data
data <- data.frame(
  y = c(10, 15, 20, 30, 50, 100, 200, 40, 20, 10),
  n = c(100, 150, 200, 300, 500, 1000, 2000, 400, 200, 100)
)

# Fixed effects model formula
fixed_effects_formula <- y ~ 1

# Calculate negative log likelihood
nll_poisson_gamma(
  theta = theta,
  data = data,
  formula = fixed_effects_formula
)
```

---

plot.aeddo

*Create a complete 'ggplot' appropriate to a particular data type*

---

## Description

### [Stable]

This function generates a complete 'ggplot' object suitable for visualizing time series data in an aeddo object. It creates a line plot connecting the observations and adds points at each data point.

## Usage

```
## S3 method for class 'aeddo'
plot(x, ...)
```

## Arguments

x	An aeddo object
...	Additional arguments (not used).

## Value

A 'ggplot' object for visualizing the time series data.

## See Also

[autoplot\(\)](#)

## Examples

```
# Create an example aeddo object
aeddo_data <- data.frame(
  time = as.Date(c(
    "2023-01-01",
    "2023-01-02",
    "2023-01-03",
    "2023-01-04",
    "2023-01-05",
    "2023-01-06"
  )),
  y = c(100, 120, 180, 110, 130, 140),
  n = 1
)

# Supply a model formula
fixed_effects_formula <- y ~ 1

# Choose a size for the rolling window
k <- 2
# ... and quantile for the threshold
sig_level <- 0.9

# Employ the algorithm
aeddo_results <- aeddo(
  data = aeddo_data,
  formula = fixed_effects_formula,
  k = k,
  sig_level = sig_level,
  exclude_past_outbreaks = TRUE,
  init_theta = c(1, 0),
  lower = c(-Inf, 1e-6),
  upper = c(Inf, 1e2),
  method = "L-BFGS-B"
)

# Create a ggplot visualization for the aeddo object
plot(aeddo_results)
```

# Index

aeddo, [2](#)

autoplot, [4](#)

autoplot(), [6](#)

nll\_poisson\_gamma, [5](#)

plot (plot.aeddo), [6](#)

plot.aeddo, [6](#)

tibble, [3](#)