Package ‘WarnEpi’

August 26, 2025

Type Package

Title A Comprehensive Tool for Early Warning in Infectious Disease

Version 1.0.1

Description Infectious disease surveillance requires early outbreak detection. This package provides statistical tools for analyzing time-series monitoring data through three core methods:
a) EWMA (Exponentially Weighted Moving Average)
b) Modified-CUSUM (Modified Cumulative Sum)
c) Adjusted-Serfling models
Methodologies are based on:
- Wang et al. (2010) <doi:10.1016/j.jbi.2009.08.003>
- Wang et al. (2015) <doi:10.1371/journal.pone.0119923>
Designed for epidemiologists and public health researchers working with disease surveillance systems.

Language en

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5)

URL `https://github.com/pan-mingyue/WarnEpi`

BugReports `https://github.com/pan-mingyue/WarnEpi/issues`

NeedsCompilation no

Author Xiaoli Wang [aut],
Mingyue Pan [aut, cre]

Maintainer Mingyue Pan <`panmyue18@163.com`>

Repository CRAN

Date/Publication 2025-08-26 19:50:07 UTC

Contents

`aSerfling` *Adjusted Serfling*

Description

Adjusted Serfling regression for periodic disease surveillance, automating epidemic baseline estimation through iterative threshold optimization. Enhances traditional Serfling models by objectively determining epidemic periods and improving peak detection accuracy.

Usage

```
aSerfling(data, col_name, cycles)
```

Arguments

`data`	A data frame containing the warning indicator columns, arranged in time-based order.
`col_name`	A column name for the warning indicator (character).
`cycles`	A numeric vector of disease cycles (e.g., c(52,26) for weekly annual + semi-annual patterns)

Details

Implements an iterative periodic regression for time series with at least 2 full cycles. Key features:

1. Dynamic Epidemic Filtering:
   - Automatically excludes outbreak points via iterative prediction-CI comparison
   - Terminates when adjusted R-squared stabilizes (maximized model fit)
2. Flexible Seasonality Modeling:

$$Y = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{k=1}^{K} \left[\gamma_k \sin\left(\frac{2\pi t}{C_k}\right) + \delta_k \cos\left(\frac{2\pi t}{C_k}\right) \right] + \epsilon$$

   - Supports multiple cycles via `cycles` parameter (e.g., c(52,26) for weekly annual + semi-annual patterns)
   - Self-adapts to pathogen seasonality shifts
3. Peak-Centric Alerting:
   - Flags peaks via optimized threshold (final model's 95% CI upper bound)
   - Avoids subjective epidemic-onset definitions

Value

A list containing:

- output: Full dataset with warning flags (1=alert, 0=normal)
- best_fit: Final lm model object
- fit_times: Iteration count for convergence
- cycles: Input cycle parameters

References

Wang X, Wu S, MacIntyre CR, et al. Using an adjusted Serfling regression model to improve the early warning at the arrival of peak timing of influenza in Beijing. PLoS One, 2015,10(3):e0119923.

Examples

```
## modeling
data(sample_ili)
sf <- aSerfling(data = sample_ili, 'case', cycles = c(52, 26))
sf

## visualize alerts
output <- sf$output
plot(output$date, output$case, type = "l")
points(output$date[output$warning == 1],
       output$case[output$warning == 1], col = "red")
```

`aSerfling_predict` *Apply Adjusted Serfling Model to Subsequent Time Periods*

Description

Projects an existing Serfling model onto new temporally contiguous data to detect epidemic signals. Requires test data to immediately follow training data chronologically to maintain periodicity.

Usage

```
aSerfling_predict(sf, df_test)
```

Arguments

`sf`	Model object from `aSerfling` (must contain `best_fit`, `output`, and `cycles` components)
`df_test`	New data frame with identical structure to training data, containing subsequent time points. Must include the response variable column used in original modeling.

Details

This function extends the surveillance capability of an established `aSerfling` model by:

- Automatically generating time indices continuing from the training set
- Preserving all terms from the original model fit
- Calculating prediction intervals using the trained coefficients
- Flagging values exceeding the 95% upper prediction bound as warnings

Critical requirements:

1. Test data must maintain the same time resolution (weekly/monthly) as training data
2. The first test observation must be the immediate next time point after the last training observation
3. Column names and cycle parameters must match the original model specification

Value

A data frame containing warning results. The value of the warning column is 1 for warning and 0 for no warning.

References

Wang X, Wu S, MacIntyre CR, et al. Using an adjusted Serfling regression model to improve the early warning at the arrival of peak timing of influenza in Beijing. PLoS One, 2015,10(3):e0119923.

Examples

```
data(sample_ili)

## Split into sequential training/test sets
df_train <- sample_ili[1:150,]
df_test <- sample_ili[151:200,]

## modeling
sf <- aSerfling(df_train, 'case', cycles = c(52, 26))

## apply the model to test set
pre <- aSerfling_predict(sf, df_test)

## visualize alerts
plot(pre$date, pre$case, type = "l")
points(pre$date[pre$warning == 1],
       pre$case[pre$warning == 1], col = "red")
```

`EWMA` *Exponentially Weighted Moving Average*

Description

Detects anomalies in infectious disease surveillance data using an Exponentially Weighted Moving Average (EWMA) algorithm. Designed for time series data, it flags potential outbreaks by smoothing past observations with decayed weights and comparing against control thresholds.

Usage

```
EWMA(data, column, lambda = 0.5, k = 3, move_t, ignore_t = 2)
```

Arguments

`data`	A data frame containing the warning indicator columns, arranged in time-based order.
`column`	A column name or column number, used to specify the warning indicator.
`lambda`	The weight factor λ, ranging from 0 to 1(higher values prioritize recent observations).
`k`	The standard deviation coefficient k.
`move_t`	The moving period t_{move}.
`ignore_t`	The number of nearest time units to be ignored by the model, t_{ignore}.

Details

Let $\mathbf{X} = (X_1, \ldots, X_T)^\top$ be an observed time series of disease case counts, where X_t represents the aggregated counts at time t (e.g., daily, weekly, or monthly observations). We assume $X_t \sim N(\mu, \sigma^2)$ for the underlying distribution.

The EWMA (Exponentially Weighted Moving Average) model is defined as:

$$Z_1 = X_1$$
$$Z_t = \lambda X_t + (1-\lambda) Z_{t-1}$$
$$UCL_t = \hat{\mu}_t + k\hat{\sigma}_t \sqrt{\frac{\lambda}{2-\lambda}}$$

where:

- Z_t: The EWMA statistic at time t, representing an exponentially weighted average of current and past observations.
- λ: Weight factor ($0 < \lambda < 1$), higher values prioritize recent observations
- k: Standard deviation coefficient (typically 2-3)
- UCL_t: Upper Control Limit at time t, forming a dynamic threshold for anomaly detection.
- $\hat{\mu}_t, \hat{\sigma}_t$: Estimated from moving window $(X_{t-t_{move}-t_{ignore}}, \ldots, X_{t-1-t_{ignore}})$

An alarm is triggered when $Z_t > UCL_t$, with the alarm set defined as:

$$\mathcal{T} = \{t : Z_t > UCL_t\}$$

Value

A data frame containing warning results. The value of the warning column is 1 for warning and 0 for no warning.

References

Wang X, Zeng D, Seale H, et al. Comparing early outbreak detection algorithms based on their optimized parameter values. J Biomed Inform, 2010,43(1):97-103.

Examples

```
## simulate reported cases
set.seed(123)
cases <- c(round(rnorm(10, 10, 1)), seq(12,21,3), seq(15,5,-5))
dates <- seq(as.Date("2025-01-01"), by = "7 days", length.out = length(cases))
data_frame <- data.frame(date = dates, case = cases)

## modeling
output <- EWMA(data_frame,'case',lambda = 0.5, k = 3, move_t = 4, ignore_t = 2)
output

## visualize alerts
plot(output$date, output$case, type = "l")
points(output$date[output$warning == 1],
       output$case[output$warning == 1], col = "red")
```

`mCUSUM` *Modified Cumulative Sum*

Description

Modified CUSUM method for outbreak detection in infectious disease surveillance data. Implements three variants (C1', C2', C3') with dynamic thresholds for time series analysis.

Usage

```
mCUSUM(data, column, k = 1, h = 2, move_t)
```

Arguments

`data`	A data frame containing the warning indicator columns, arranged in time-based order.
`column`	A column name or column number, used to specify the warning indicator.
`k`	The standard deviation coefficient k.
`h`	The threshold coefficient h.
`move_t`	The moving period t_{move}.

Details

Let $\mathbf{X} = (X_1, \ldots, X_T)^\top$ be an observed time series of disease case counts, where X_t represents the aggregated counts at time t (e.g., daily, weekly, or monthly observations). We assume $X_t \sim N(\mu, \sigma^2)$ for the underlying distribution.

The modified CUSUM models accumulate excess cases beyond control limits:

$$C1'_0 = C2'_0 = 0$$
$$C1'_t = \max\left(0, X_t - (\hat{\mu}_t + k\hat{\sigma}_t) + C1'_{t-1}\right)$$
$$C2'_t = \max\left(0, X_t - (\hat{\mu}_t + k\hat{\sigma}_t) + C2'_{t-1}\right)$$
$$C3'_t = C2'_t + C2'_{t-1} + C2'_{t-2}$$
$$H_t = h\hat{\sigma}_t$$

where:

- k: Standard deviation coefficient (typical range 0.5–1.5), adjusts sensitivity to deviations
- h: Threshold coefficient (typical range 2–5), controls alarm stringency
- H: Threshold

Model specifications:

- **C1'**: Baseline $\hat{\mu}_t, \hat{\sigma}_t$ estimated from $(X_{t-t_{move}}, ..., X_{t-1})$
- **C2'**: Baseline $\hat{\mu}_t, \hat{\sigma}_t$ estimated from $(X_{t-2-t_{move}}, ..., X_{t-3})$ to avoid recent outbreaks
- **C3'**: 3-day cumulative sum of C2' values
- Alarms trigger when $Cx'_t > H_t$ for each model (x = 1,2,3)

Value

A data frame containing C1', C2' and C3' warning results. The value of the warning column is 1 for warning and 0 for no warning.

References

Wang X, Zeng D, Seale H, et al. Comparing early outbreak detection algorithms based on their optimized parameter values. J Biomed Inform, 2010,43(1):97-103.

Examples

```
## simulate reported cases
set.seed(123)
cases <- c(round(rnorm(10, 10, 1)), seq(12,21,3), seq(15,5,-5))
dates <- seq(as.Date("2025-01-01"), by = "7 days", length.out = length(cases))
data_frame <- data.frame(date = dates, case = cases)

## modeling
output <- mCUSUM(data_frame, 'case', k = 1, h = 2.5, move_t = 4)
output
```

```
## visualize alerts
### C1'
plot(output$date, output$case, type = "l")
points(output$date[output$C1_prime_warning == 1],
       output$case[output$C1_prime_warning == 1], col = "red")

### C2'
plot(output$date, output$case, type = "l")
points(output$date[output$C2_prime_warning == 1],
       output$case[output$C2_prime_warning == 1], col = "red")

### C3'
plot(output$date, output$case, type = "l")
points(output$date[output$C3_prime_warning == 1],
       output$case[output$C3_prime_warning == 1], col = "red")
```

`sample_ili` *Simulated ILI Surveillance Data*

Description

A dataset containing 200 weeks of simulated influenza-like illness case counts.

Usage

`data(sample_ili)`

Format

A data frame with 200 rows and 2 variables:

- date: Date of observation (weekly)
- case: Integer count of reported cases

Index