Documentation

excode: Excess Count Detection in Epidemiological Time Series

Benedikt Zacher[1], & **Ann Christin Vietor**[1]

[1] Robert Koch-Institut | Unit 32

Cite
Zacher, B., & Vietor, A. (2025). excode: Excess Count Detection in Epidemiological Time Series. Zenodo. https://doi.org/10.5281/zenodo.15322133

This repository contains the R package *excode* with a variety of functions for **ex**cess **co**unt **de**tection in epidemiological time series. Excess count detection is an important part of public health surveillance.

Installation

This is an R package. You can use `install_github()` from devtools to install this package.

```
library(devtools)
install_github("robert-koch-institut/Excess_count_detection_for_epidemiological_time_series")
```

Overview

A variety of algorithms has been developed to identify events such as disease outbreaks or excess mortality. To this end, time series are analysed to detect unusually large (case) counts, that exceed what is normally expected in a certain time period and geographical region. The normal expectancy of cases in a current time period is usually calculated based on historic data. Depending on the time series of interest, the following features need to be taken into account by a model:

- **Seasonal patterns:** Many epidemiological times series that are of public health interest show periodic changes in cases depending on seasons or other calendar periods.
- **Long-term time trends:** The time series may show a long-term increase or decrease in case counts.
- **Historic events:** Events such as disease outbreaks may have caused an excess of case counts in historic data that is used for model estimation. This needs to be considered to avoid overestimation of the normal expectancy for the current time period.

The *excode* package provides a flexible framework that implements well established approaches to control for seasonality, long-term trends and historic events, but also allows the use of customized models. The user can choose between the Poisson and the Negative Binomial distribution, which are the most commonly used probability distributions for modeling count data. By combining hidden Markov models and generalized linear models, *excode* explicitly models normally expected case counts *and* expected excess case counts, i.e. each time point in a time series is labeled either as a normal state or as an excess state. A short overview of the statistical model used can be found at the end of this document.

Available models in *excode*

There are five different model classes implemented in *excode*:

- **Mean:** This model does not account for seasonal or cyclical patterns. The expectancy of normal and excess case counts is modeled as the mean of the historic data.
- **FarringtonNoufaily:** This is a popular algorithm, which models seasonality by segmenting a year into a predefined number of shorter time periods, where each has a different expectancy of observed cases. It also allows integration of a long-term time trend.
- **Harmonic:** The 'Harmonic' model uses sine and cosine function to account for seasonal or cyclical patterns in the data. It also allows integration of long-term time trends.
- **Custom:** The 'Custom' model allows the user define their own model by providing a data frame containing covariate data.

All of the above mentioned models use two states - a 'normal' and an 'excess' state - to detect excess counts. However in some cases it might be necessary to allow multiple states, such as 'normal', 'low excess', 'medium excess', 'high excess', to account for a wide range of excess counts and detect all periods showing an excess in a time series.

- **MultiState:** The 'MultiState' model allows the use of multiple states. The number of states can be specified by the user. Like in the 'Custom' model, the user must provide covariate data to control for possible seasonal or long-term time trends. The use of this model is more advanced than the other models and requires some additional steps before fitting.

Further descriptions and code examples can be found in the `vignette("excode")`.

Running an excode model

The package's core function, `run_excode()` performs the excess count detection. The output of `run_excode()` is a fitted `excodeModel` object.
Before running the excess count detection, the user needs to decide which probability distribution and which model should be used. This is done using the `excodeFamily` and `excodeFormula` objects of the *excode* package. Both are then combined into an `excodeModel` object, which stores all necessary parameters and specifications for model fitting. In this example, the 'Poisson' distribution and the 'Harmonic' model is chosen.

```
# Use a 'Poisson' distribution in excodeFamily
excode_family_pois <- excodeFamily("Poisson")
# Define the 'Harmonic' model in excodeFormula
excode_formula_har = excodeFormula("Harmonic")
# Combine both in the final excodeModel object
excode_har_pois = excodeModel(excode_family_pois,
                             excode_formula_har)

# Run excode on time points 209-295 on Salmonella hadar shadar_df data.frame
result_shadar_har <- run_excode(shadar_df,
                               excode_har_pois,
                               209:295)

result_shadar_har
```

Results can be extracted using the `summary()` function:

```
summary_shadar_har <- summary(result_shadar_har)
kable(as_tibble(summary_shadar_har[82:87,])) %>%
  kable_styling(font_size = 11)
```

Further descriptions, other models and code examples are available in the `vignette("excode")`.

Data

The package includes example datasets which can be used to apply the different algorithms.

The following datasets are provided with this package:

mort_df_germany
sarscov2_df
shadar_df

snewport_df.

German all-cause mortality data

The dataset **mort_df_germany** contains the number of weekly deaths in Germany reported to the German Federal Statistical Office.

SARS-CoV-2 infections in Berlin-Neukölln (Germany)

The dataset **sarscov2_df** contains the daily number of reported **SARS-CoV-2** cases from March to July 2020 in **Berlin-Neukölln**. The dataset was downloaded on **2024-10-31** from this source.

Salmonella Hadar cases in Germany (2001–2006)

The dataset **shadar_df** contains the weekly number of reported **Salmonella Hadar** cases from January 2001 to August 2006 in Germany.

Salmonella Newport cases in Germany (2004–2014)

The dataset **snewport_df** contains the weekly number of reported **Salmonella Newport** cases from January 2004 to February 2014 in Germany.

Administrative and organizational information

This R package was developed by Benedikt Zacher with contributions from Ann Christin Vietor Unit 32 | Surveillance.
The publication of the code as well as the quality management of the metadata is done by department MF 4 | Domain Specific Data and Research Data Management.
Questions regarding the publication infrastructure can be directed to the Open Data Team of the Department MF4 at OpenData@rki.de.

Funding

Benedikt Zacher and Ann Christin Vietor were supported by BMBF (Medical Informatics Initiative: HIGHmed) and the collaborative management platform for detection and analyses of (re-)emerging and foodborne outbreaks in Europe (COMPARE: European Union’s Horizon research and innovation programme, grant agreement No. 643476).

Collaborate

If you want to contribute, feel free to fork this repo and send us pull requests.

Publication platforms

This software publication is available on Zenodo.org, GitHub.com and OpenCoDE:

- https://zenodo.org/communities/robertkochinstitut
- https://github.com/robert-koch-institut
- https://gitlab.opencode.de/robert-koch-institut

License