Package 'anomaly'

September 21, 2018

Type Package

Title Detecting Anomalies in Data

Version 1.1.0

Date 2018-09-19

Description An implementation of CAPA (Collective And Point Anomaly) for the detection of anomalies in time series data. The package also contains Kepler lightcurve data and shows how CAPA can be applied to detect exoplanets.

License GPL

Imports dplyr,rlang,Rdpack,ggplot2

NeedsCompilation yes

RoxygenNote 6.0.1

RdMacros Rdpack

Author Alex Fisch [aut],
Daniel Grose [ctb, cre],
Idris Eckley [ths],
Paul Fearnhead [ths]

Maintainer Daniel Grose <dan.grose@lancaster.ac.uk>

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2018-09-21 07:20:03 UTC

R topics documented:

anomaly_series *A technique for detecting anomalous segments based on CAPA (Collective And Point Anomaly) by Fisch et al.*

Description

CAPA (Collective And Point Anomaly) by A. T. M. Fisch, I. A. Eckley, and P. N. Fearnhead assumes that the data has a certain mean and variance for most timepoints and detects segments in which the mean and/or variance deviates from the typical mean and variance as collective anomalies. It also detects point outliers and returns a measure of strength for the changes in mean and variance. If the number of anomalous windows scales linearly with the number of data points, CAPA scales linearly with the number of data points. At worst, if there are no anomalies at all, the computational cost of CAPA scales quadratically with the number of data points.

Usage

```
anomaly_series(x, penaltywindow = NULL, penaltyanomaly = NULL,
  minimumsegmentlength = 10, warnings = TRUE, method = "meanvar")
```

Arguments

x	A numeric vector containing the data which is to be inspected.
penaltywindow	A numeric constant indicating the penalty for adding an additional epidemic changepoint. It defaults to a BIC style penalty if no argument is provided.
penaltyanomaly	A numeric constant indicating the penalty for adding an additional point anomaly. It defaults to a BIC style penalty if no argument is provided.
minimumsegmentlength	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
warnings	A logical determining whether the warnings should be displayed. It defaults to TRUE.
method	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies caracterised by joint changes in mean and variance (the default) and "mean" for collective anomalies characterised by changes in mean only.

Value

An anomaly_series object.

References

Fisch A, Eckley I and Fearnhead P (2018). "A linear time method for the detection of point and collective anomalies." *ArXiv e-prints*. 1806.01947.

Examples

```
library(anomaly)
set.seed(2018)
# Generate data typically following a normal distribution with mean 0 and variance 1.
# Then introduce 3 anomaly windows and 4 point outliers.
x  = rnorm(5000)
x[401:500]   = rnorm(100,4,1)
x[1601:1800] = rnorm(200,0,0.01)
x[3201:3500] = rnorm(300,0,10)
x[c(1000,2000,3000,4000)] = rnorm(4,0,100)
inferred_anomalies = anomaly_series(x)
plot(inferred_anomalies)
summary(inferred_anomalies)
print(inferred_anomalies)

### Repeat with other method

inferred_anomalies = anomaly_series(x,method="mean")
plot(inferred_anomalies)
summary(inferred_anomalies)
print(inferred_anomalies)

### Real data example:

library(anomaly)
data(Lightcurves)
### Plot the data for Kepler 10965588: No transit apparent
plot(Lightcurves$Kepler10965588$Day,Lightcurves$Kepler10965588$Brightness,xlab = "Day",pch=".")
### Examine a period of 62.9 days for Kepler 10965588
binned_data = period_average(Lightcurves$Kepler10965588,62.9)
inferred_anomalies = anomaly_series(binned_data)
plot(inferred_anomalies,xlab="Bin")
# We found a planet!
```

`Lightcurves` *Kepler Lightcurve data.*

Description

One of the most successful approaches for the detection of exoplanets is the so called transit method: A star's brightness is continuously measured over time by a powerful telescope. If one or multiple planets orbit this star the recorded luminosity of the star will exhibit periodically recurring dips due to the transits of the planet in front of the telescope's lense – an effect comparable to that of an eclipse. Given how small planets are compared to stars the transit signals are known to be very weak.

The stars included in this file all have known exoplanets with the following periods:

Kepler 1871056: 2 planets with orbital periods of 40.8 and 140.1 days
Kepler 2307415: 2 planets with orbital periods of 4.61 and 12.12 days

Kepler 3102384: 2 planets with orbital periods of 10.57 and 523.9 days
Kepler 3231341: 4 planets with orbital periods of 4.24, 8.15, 12.33, and 19.00 days
Kepler 3447722: 3 planets with orbital periods of 10.30, 16.09, and 35.68 days
Kepler 4139816: 4 planets with orbital periods of 3.34, 7.82, 20.06, and 46.18 days
Kepler 10965588: 1 planet with orbital period of 62.89 days

More information about the exoplanets above and more data can be found at https://exoplanetarchive.ipac.caltech.edu/index.html

This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

Usage

```
data(Lightcurves)
```

Format

A list of seven dataframes named "Kepler1871056", "Kepler2307415", "Kepler3102384", "Kepler3231341", "Kepler3447722", "Kepler4139816", and "Kepler10965588". Each dataframe consists of two columns called "Brightness" and "Day", containing measurements of a star's brightness and the measurement's timestamp respectively.

---

period_average *A function to search the Kepler data for periodically recurring dips in luminosity*

Description

The signal of transiting planets is very weak, especially if the planet is small. This function amplifies it by exploiting the periodicity of the signal. All observations times are taken modulo the period and then binned. An average is then taken within each bin, those averages then stored as a vector and returned. If the orbital period of an exoplanet (or an integer fraction thereof) is used as argument for "period", the signal to noise ratio of the transit is improved, which can allow for the planet's detection.

Usage

```
period_average(data,period)
```

Arguments

data	A dataframe with one column named "Day" and the other "Brightness", such as Kepler10965588 (included in the package)
period	A numeric which is larger than 0 representing the period (in days) which is to be examined

Value

A vector of numerics

Examples

```
library(anomaly)
data(Lightcurves)
### Plot the data for Kepler 10965588: No transit apparent
plot(Lightcurves$Kepler10965588$Day,Lightcurves$Kepler10965588$Brightness,xlab = "Day",pch=".")
### Examine a period of 62.9 days for Kepler 10965588
binned_data = period_average(Lightcurves$Kepler10965588,62.9)
inferred_anomalies = anomaly_series(binned_data)
plot(inferred_anomalies,xlab="Bin")
# We found a planet!
```

---

plot.anomaly_series *Plots an anomaly_series object.*

Description

Plot method for anomaly_series objects as returned by the anomaly_series method.

Usage

```
## S3 method for class 'anomaly_series'
plot(x, xlab = "", ylab = "", ...)
```

Arguments

x	anomaly_series object
xlab	Character string containing label for the x-axis.
ylab	Character string containing label for the y-axis.
...	Other parameters to be passed to plotting methods.

Value

A ggplot object

Examples

```
library(anomaly)
set.seed(2018)
# Generate data typically following a normal distribution with mean 0 and variance 1.
# Then introduce 3 anomaly windows and 4 point outliers.
x  = rnorm(5000)
x[401:500]   = rnorm(100,4,1)
x[1601:1800] = rnorm(200,0,0.01)
x[3201:3500] = rnorm(300,0,10)
```

```
x[c(1000,2000,3000,4000)] = rnorm(4,0,100)
inferred_anomalies = anomaly_series(x)
plot(inferred_anomalies)
```

`print.anomaly_series` *Print function for anomaly_series objects.*

Description

Prints the contents of an anomaly_series object as returned by the `anomaly_series` method.

Usage

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

Arguments

`x`	An anomaly_series object
`...`	Other parameters to be passed to print methods.

Examples

```
library(anomaly)
set.seed(2018)
# Generate data typically following a normal distribution with mean 0 and variance 1.
# Then introduce 3 anomaly windows and 4 point outliers.
x  = rnorm(5000)
x[401:500]   = rnorm(100,4,1)
x[1601:1800] = rnorm(200,0,0.01)
x[3201:3500] = rnorm(300,0,10)
x[c(1000,2000,3000,4000)] = rnorm(4,0,100)
inferred_anomalies = anomaly_series(x)
print(inferred_anomalies)
```

`summary.anomaly_series`
Summary method for anomaly_series objects.

Description

A function summarising the content of an anomaly_object as produced by the `anomaly_series` method. It prints the number of detected anomalous windows and point anomalies.

Usage

```
## S3 method for class 'anomaly_series'
summary(object, ...)
```

Arguments

object	anomaly_series object
...	Other parameters to be passed to summary methods.

Examples

```
library(anomaly)
set.seed(2018)
# Generate data typically following a normal distribution with mean 0 and variance 1.
# Then introduce 3 anomaly windows and 4 point outliers.
x  = rnorm(5000)
x[401:500]   = rnorm(100,4,1)
x[1601:1800] = rnorm(200,0,0.01)
x[3201:3500] = rnorm(300,0,10)
x[c(1000,2000,3000,4000)] = rnorm(4,0,100)
inferred_anomalies = anomaly_series(x)
summary(inferred_anomalies)
```

Index