Package: ggchangepoint (via r-universe)

September 12, 2024

Type Package

Title Combines Changepoint Analysis with 'ggplot2'

Version 0.1.0

Description R provides fantastic tools for changepoint analysis, but plots generated by the tools do not have the 'ggplot2' style. This tool, however, combines 'changepoint', 'changepoint.np' and 'ecp' together, and uses 'ggplot2' to visualize changepoints.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports changepoint, changepoint.np, dplyr, ecp, ggplot2, Rdpack, tibble, utils

RdMacros Rdpack

RoxygenNote 7.1.2

Suggests rmarkdown, knitr, gstat, datasets

VignetteBuilder knitr

URL https://pursuitofdatascience.github.io/ggchangepoint/

BugReports https://github.com/PursuitOfDataScience/ggchangepoint/issues

Repository https://pursuitofdatascience.r-universe.dev

RemoteUrl https://github.com/pursuitofdatascience/ggchangepoint

RemoteRef HEAD

RemoteSha 140c71999194db248df188043db9f431e6b1b5a4

Contents

---

cpt_wrapper *Changepoint wrapper*

Description

This function wraps a number of cpt functions from the changepoint package and the cpt.np() function from the changepoint.np package. It is handy that users can use this function to get the same changepoint results as these functions output individually. Moreover, it returns a tibble that inherits the tidyverse sytle. Functions from the changepoint package do require data normality assumption by default, yet changepoint.np is a non-parametric way to detect changepoints and let data speak by itself. If user sets change_in as cpt_np, a seed should be set before using the function for the sake of reproducibility. For more details on the changepoint and changepoint.np packages, please refer to their documentation.

Usage

```
cpt_wrapper(data, change_in = "mean_var", cp_method = "PELT", ...)
```

Arguments

data	A vector.
change_in	Choice of mean_var, mean, var, and cpt_np. Each choice corresponds to cpt.meanvar(), cpt.mean(), cpt.var() and cpt.np() respectively. The default is mean_var.
cp_method	A wide range of choices (i.e., AMOC, PELT, SegNeigh or BinSeg). Please note when change_in is cpt_np, PELT is the only option.
...	Extra arguments for each cpt function mentioned in the change_in section.

Value

A tibble includes which point(s) is/are the changepoint along with raw changepoint value corresponding to that changepoint.

References

Killick R, Eckley I (2014). "changepoint: An R package for changepoint analysis." *Journal of statistical software*, **58**(3), 1–19.

Examples

```
set.seed(2022)
cpt_wrapper(c(rnorm(100,0,1),rnorm(100,0,10)))
cpt_wrapper(c(rnorm(100,0,1),rnorm(100,10,1)))
```

`ecp_wrapper` *ecp wrapper*

Description

The ecp package provides a non-parametric way to detect changepoints. Unlike the changepoint package, it does not assume raw data to have any formal distribution. This wrapper function wraps two functions from the ecp package, i.e., `e.divisive()` and `e.agglo()`. Users can use either function by switching the `algorithm` argument. Before using the wrapper function , seed should be set for the sake of reproducibility.

Usage

```
ecp_wrapper(data, algorithm = "divisive", min_size = 2, ...)
```

Arguments

`data`	A vector.
`algorithm`	Either `divisive` or `agglo`. `divisive` is the default.
`min_size`	Minimum number of observations between change points. By default is 2. This argument is only applied when `algorithm` = `"divisive"`.
`...`	Extra arguments to pass on either from `e.divisive()` or `e.agglo()`.

Value

A tibble includes which point(s) is/are the changepoint along with raw changepoint value corresponding to that changepoint.

References

James NA, Matteson DS (2013). “ecp: An R package for nonparametric multiple change point analysis of multivariate data.” *arXiv preprint arXiv:1309.3295*.

Examples

```
set.seed(2022)
ecp_wrapper(c(rnorm(100,0,1),rnorm(100,0,10)))
ecp_wrapper(c(rnorm(100,0,1),rnorm(100,10,1)))
```

`ggchangepoint` — `ggchangepoint` *package*

Description

Combines Changepoint Anaysis with 'ggplot2'.

Details

ggchangepoint tries to offer several changepoint R packages in a tidy format and output the ggplot2 plots so that the tidyverse users can gain some familiarity to work with the changepoint analysis. For the moment, I only include three changepoint packages ('changepoint', 'changepoint.np' and 'ecp'). More changepoint packages will be included as time progresses.

`ggcptplot` — *Plot for the changepoint package*

Description

The plot for changepoints detected by the changepoint package is a line plot for the raw data and the vertical lines representing each changepoint. The x-axis is the row number of the raw data in the original data vector. The plot inherits ggplot2, meaning users can add ggplot2 functions on top the changepoint plot for customization.

Usage

```
ggcptplot(
  data,
  change_in = "mean_var",
  cp_method = "PELT",
  ...,
  cptline_alpha = 1,
  cptline_color = "blue",
  cptline_type = "solid",
  cptline_size = 0.5
)
```

Arguments

`data`	A vector.
`change_in`	Choice of `mean_var`, `mean`, `var`, and `cpt_np`. Each choice corresponds to `cpt.meanvar()`, `cpt.mean()`, `cpt.var()` and `cpt.np()` respectively. The default is `mean_var`.
`cp_method`	A wide range of choices (i.e., `AMOC`, `PELT`, `SegNeigh` or `BinSeg`). Please note when `change_in` is `cpt_np`, `PELT` is the only option.

`...` Extra arguments for each `cpt` function mentioned in the `change_in` section.

`cptline_alpha` The value of alpha for the vertical changepoint line(s), default is 1, meaning no transparency.

`cptline_color` The color for the vertical changepoint line(s), default is `blue`.

`cptline_type` The linetype for the vertical changepoint line(s), default is `solid`.

`cptline_size` The size for the vertical changepoint line(s), default is `0.5`.

Value

A line plot with data points along with the vertical lines representing changepoints.

Examples

```
ggcptplot(c(rnorm(100,0,1),rnorm(100,0,10)))
ggcptplot(c(rnorm(100,0,1),rnorm(100,10,1)))
```

ggecpplot *Plot for the ecp package*

Description

The plot for changepoints detected by the ecp package is a line plot for the raw data and the vertical lines representing each changepoint. The x-axis is the row number of the raw data in the original data vector. The plot inherits ggplot2, meaning users can add ggplot2 functions on top the changepoint plot for customization.

Usage

```
ggecpplot(
  data,
  algorithm = "divisive",
  min_size = 2,
  ...,
  cptline_alpha = 1,
  cptline_color = "blue",
  cptline_type = "solid",
  cptline_size = 0.5
)
```

Arguments

`data` A vector.

`algorithm` Either `divisive` or `agglo`. `divisive` is the default.

`min_size` Minimum number of observations between change points. By default is 2. This argument is only applied when `algorithm = "divisive"`.

`...` Extra arguments to pass on either from `e.divisive()` or `e.agglo()`.

`cptline_alpha` The value of alpha for the vertical changepoint line(s), default is 1, meaning no transparency.

`cptline_color` The color for the vertical changepoint line(s), default is `blue`.

`cptline_type` The linetype for the vertical changepoint line(s), default is `solid`.

`cptline_size` The size for the vertical changepoint line(s), default is `0.5`.

Value

A line plot with data points along with the vertical lines representing changepoints.

Examples

```
ggecpplot(c(rnorm(100,0,1),rnorm(100,0,10)))
ggecpplot(c(rnorm(100,0,1),rnorm(100,0,10)))
```

Index