

Package: changepoint.influence (via r-universe)

September 16, 2024

Type Package

Title Package to Calculate the Influence of the Data on a Changepoint Segmentation

Version 1.0.2

Date 2024-02-19

Maintainer Rebecca Killick <r.killick@lancs.ac.uk>

BugReports <https://github.com/rkillick/changepoint.influence/issues>

URL <https://github.com/rkillick/changepoint.influence/>

Imports data.table, ggplot2, gridExtra, reshape, graphics, methods

Depends R(>= 3.6), changepoint

Suggests testthat, vdiff

Description Allows users to input their data, segmentation and function used for the segmentation (and additional arguments) and the package calculates the influence of the data on the changepoint locations, see Wilms et al. (2022) <[doi:10.1080/10618600.2021.2000873](https://doi.org/10.1080/10618600.2021.2000873)>. Currently this can only be used with the changepoint package functions to identify changes, but we plan to extend this. There are options for different types of graphics to assess the influence.

License GPL

LazyData true

Repository <https://rkillick.r-universe.dev>

RemoteUrl <https://github.com/rkillick/changepoint.influence>

RemoteRef HEAD

RemoteSha 863e1a6fe11ec022dc77a412c4cf6935419a7aeb

Contents

changepoint.influence-package	2
InfluenceMap	4
LocationStability	6
ParameterStability	9
StabilityOverview	10
welldata	12

Index	14
--------------	-----------

changepoint.influence-package

Package to Calculate the Influence of the Data on a Changepoint Segmentation

Description

Allows users to input their data, segmentation and function used for the segmentation (and additional arguments) and the package calculates the influence of the data on the changepoint locations, see Wilms et al. (2022) <doi:10.1080/10618600.2021.2000873>. Currently this can only be used with the changepoint package functions to identify changes, but we plan to extend this. There are options for different types of graphics to assess the influence.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

The package allows users to input their data, segmentation and function used for the segmentation (and additional arguments) and the package calculates the influence of the data on the changepoint locations.

The influence() function is the first port of call to calculate the influence. We provide two methods for influence detection, via "delete" and "outlier" options which respectively consider the effect of deleting a data point or making it an outlier. Currently we provide this method for cpt objects (as generated by the "changepoint" package) but plan to extend this to other objects in the future. Please add requests for objects to include to our github issues.

Users are encouraged to explore the documentation for the StabilityOverview() graphic, followed by the LocationStability() and ParameterStability() graphics for a more granular view, followed by the InfluenceMap() as the highest level of detail.

Author(s)

Rebecca Killick [aut, cre], Ines Wilms [aut]

Maintainer: Rebecca Killick <r.killick@lancs.ac.uk>

References

Wilms I, Killick R, Matteson DS (2022) Graphical Influence Diagnostics for Changepoint Models, *Journal of Computational and Graphical Statistics*, **31:3**, 753–765 DOI: 10.1080/10618600.2021.2000873

See Also

[influence-methods](#), [StabilityOverview](#), [LocationStability](#), [ParameterStability](#), [InfluenceMap](#)

Examples

```
#### Load the data in the R package changeoint.influence ####
data("welldata")
welllog = welldata[1001:2000] # Extract the mid section of the data as analyzed in other papers
n = length(welllog)
var = NULL; for (i in 30:1000){var[i]=var(welllog[(i-29):i])}
welllogs = welllog/sqrt(median(var, na.rm = TRUE))
# rescale the data to have unit variance across time,
# note that there may still be changes in variance across the series.

#### Apply PELT to the welllog data ####
out.PELT = cpt.mean(welllogs, method = 'PELT')

#### Calculate the influence measures ####
welllogs.inf = influence(out.PELT)
# the code extracts all the details of the original cpt.mean() function call
# and uses these in the calculation of the influence for the modified data.

#### Stability Dashboards ####
StabilityOverview(welllogs, cpts(out.PELT), welllogs.inf, las = 1, ylab='Nuclear-Magnetic Response',
  legend.args=list(display=TRUE,x="bottomright",y=NULL,cex=1.5,bty="n",horiz=FALSE,xpd=FALSE))
# We can specify where the legend will sit in the graphic via the legend.args
# which are passed to the legend() function. We can also include additional arguments
# to pass to the plotting such as las=1 here.

#### Location Stability plot ####
exp.seg=LocationStability(cpts(out.PELT), welllogs.inf, type = 'Difference', cpt.lwd = 4, las = 1)
# Note that if the expected segmentation is not provided, it will be calculated and then
# returned so that the user can avoid calculating this again in other plot calls.

#### Parameter Stability plot ####
ParameterStability(welllogs.inf, original.mean = rep(param.est(out.PELT)$mean,
  times=diff(c(0,out.PELT@cpts))), las = 1, ylab = 'Nuclear-Magnetic Response')
# Note that the original.mean argument is provided for each timepoint so is a length n vector.

#### Influence Map ####
## Not run:
library(ggplot2)
welllogs.inf = influence(out.PELT, method = "delete")
InfluenceMap(cpts(out.PELT), welllogs.inf, data=welllogs, include.data=TRUE,
  ylab='Nuclear-Magnetic\n Response',
  ggops=theme(axis.text=element_text(size=15),axis.title=element_text(size=20),
```

```

    plot.title=element_text(size=25))
# The InfluenceMap uses ggplot2 functions, thus you can add theme options via the ggops argument.
# Here we change the text sizes to ensure readable titles and labels for a report.

welllogs.inf = influence(out.PELT, method = "outlier")
InfluenceMap(cpts(out.PELT), welllogs.inf, data = welllogs, include.data = TRUE,
  ylab='Nuclear-Magnetic\n Response')

## End(Not run)

```

InfluenceMap

Influence Map Graphic

Description

Plots the highest detail level of the changepoint location stability according to the influence measure.

Usage

```

InfluenceMap(original.cpts, influence, resid=NULL, data=NULL, include.data=FALSE,
  influence.col=c("#0C4479", "white", "#AB9783"), cpt.col=c("#009E73", "#E69F00", "#E41A1C"),
  cpt.lty=c("dashed", "dotdash", "dotted"), ylab='', ggops=NULL)

```

Arguments

<code>original.cpts</code>	An ordered vector of the changepoint locations found by your favourite changepoint method.
<code>influence</code>	The influence as calculated the <code>influence()</code> function provided within this package. This is a list object.
<code>resid</code>	An $n \times n$ matrix containing the difference of the observed class (<code>influence</code>) from the expected class at each datapoint. If this is left as <code>NULL</code> , it will be calculated and returned to the user.
<code>data</code>	A vector containing the data on which you have run your changepoint method.
<code>include.data</code>	Is a plot of the data to be included above the histogram. Default is <code>FALSE</code> .
<code>influence.col</code>	A length 3 vector giving the lower, middle (0) and upper bounds for the influence map colour grading. Note that you should choose these colours to not conflict with the colours used for <code>cpt.col</code> . We advise using "white" for the middle choice as this provides a clean (majority white) heatmap.
<code>cpt.col</code>	Colour of the <code>original.cpts</code> lines when plotted. We need three colours specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. These are plotted on the Influence Map as well as the original data if <code>include.data=TRUE</code> . Any values accepted by the <code>col</code> plotting argument are allowed.
<code>cpt.lty</code>	Line type of the <code>original.cpts</code> lines when plotted. We need three line types specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. Any values accepted by the <code>lty</code> plotting argument are allowed. Only used if <code>include.data=TRUE</code> .

<code>ylab</code>	The label for the y-axis, character vector expected.
<code>ggops</code>	Any other settings to be passed to the <code>ggplot()</code> function. Note that you will need to <code>library(ggplot2)</code> if calling <code>ggplot</code> functions in arguments. See examples.

Details

This function creates the highest detail graphic to display the results of a changepoint influence analysis on the location of the changepoints. The graphic is an $n \times n$ heatmap of the difference between the observed segmentations under the "delete" or "outlier" Influence analysis and the expected segmentation. Note that the expected segmentations take into account the fact that a changepoint at a timepoint, say 100, will move (to 99) when a timepoint prior to it is deleted and that adding an outlier will introduce new changepoints.

Datapoints on the vertical axis without a single coloured co-ordinate on the horizontal axis can be considered as non-influential since they do not trigger any changepoint instability. Rows with coloured pixels correspond to data points which are instability triggers.

How to interpret the Influence Map (please also read the paper in the references for fuller details):

colouring: Colouring above the diagonal indicates that an alteration of the corresponding data point (on the vertical axis) affects earlier data points, colouring below the diagonal indicates that subsequent data points are affected.

horizontal span: A stop in colouring indicates that change-points have moved, while a continuation of colouring to the last data point indicates that, in total, fewer or additional changepoints are detected.

local vs global: Most colouring originates on the diagonal, thereby indicating that a data point's alteration mainly affects neighbouring data points that most often belong to the same segment. By contrast, in some cases a coloured pixel may originate away from the diagonal, thereby exercising global influence.

height: All data points (on the vertical axis) that appear in the coloured area are influential and assert influence over the corresponding data points on the horizontal axis. The height can be seen as the extent to which instability arises in this influential region.

Value

The function returns a plot denoted the Influence Map. If `resid=NULL` then the residuals (observed class - expected class) are also returned.

Author(s)

Rebecca Killick

References

Wilms I, Killick R, Matteson DS (2022) Graphical Influence Diagnostics for Changepoint Models, *Journal of Computational and Graphical Statistics*, **31**:3, 753–765 DOI: [10.1080/10618600.2021.2000873](https://doi.org/10.1080/10618600.2021.2000873)

See Also

[influence-methods](#), [StabilityOverview](#), [ParameterStability](#), [LocationStability](#)

Examples

```
#### Generate Simulated data example ####
set.seed(30)
x=c(rnorm(50),rnorm(50,mean=5),rnorm(1,mean=15),rnorm(49,mean=5),rnorm(50,mean=4))
xcpt = cpt.mean(x,method='PELT') # Get the changepoints via PELT

#### Influence Map ####
## Not run:
library(ggplot2)
x.inf = influence(xcpt, method = "delete")
InfluenceMap(cpts(xcpt), x.inf, data = x, include.data = TRUE,
             ggops = theme(axis.text = element_text(size=15), axis.title = element_text(size=20),
                           plot.title = element_text(size=25)))

x.inf = influence(xcpt, method = "outlier")
InfluenceMap(cpts(xcpt), x.inf, data=x, include.data = TRUE,
             ggops = theme(axis.text = element_text(size=15), axis.title = element_text(size=20),
                           plot.title = element_text(size=25)))

## End(Not run)
```

LocationStability *Location Stability Graphic*

Description

Plots the middle detail level of the changepoint location stability according to the influence measure.

Usage

```
LocationStability(original.cpts, influence, expected.class=NULL,
                 type=c("Difference", "Global", "Local"), data=NULL, include.data=FALSE, cpt.lwd=4,
                 cpt.col=c("#009E73", "#E69F00", "#E41A1C"), cpt.lty=c("dashed", "dotdash", "dotted"),
                 ylab='', xlab='Index', ...)
```

Arguments

<code>original.cpts</code>	An ordered vector of the changepoint locations found by your favourite changepoint method.
<code>influence</code>	The influence as calculated the <code>influence()</code> function provided within this package. This is a list object.
<code>expected.class</code>	Only needed for <code>type="Difference"</code> . An <code>nxn</code> matrix containing the expected class of each datapoint from the <code>original.cpts</code> segmentation under perturbation of each datapoint. If this is left as <code>NULL</code> and <code>type="Difference"</code> , it will be calculated and returned to the user.

<code>type</code>	The type of Location Stability plot, can be "Difference", "Global" or "Local". If all three are listed (as in the default) then three graphs will be plotted. <code>type="Difference"</code> will histogram the difference between the observed and expected segmentations at each of the datapoints. <code>type="Global"</code> will histogram the observed segmentations. <code>type="Local"</code> will histogram the observed segmentations with the original <code>.cpt</code> s locations removed. See details for more description of the graphics.
<code>data</code>	A vector containing the data on which you have run your changepoint method.
<code>include.data</code>	Is a plot of the data to be included above the histogram. Default is FALSE.
<code>cpt.lwd</code>	The line width to be used when plotting the original <code>.cpt</code> s on the data. Standard <code>lwd</code> values allowed. A single value is expected.
<code>cpt.col</code>	Colour of the original <code>.cpt</code> s lines when plotted. We need three colours specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. Any values accepted by the <code>col</code> plotting argument are allowed.
<code>cpt.lty</code>	Line type of the original <code>.cpt</code> s lines when plotted. We need three line types specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. Any values accepted by the <code>lty</code> plotting argument are allowed.
<code>ylab, xlab</code>	The labels for the x- and y-axis, character vector expected.
<code>...</code>	Any other arguments to be passed to the <code>plot()</code> or equivalently <code>hist()</code> function.

Details

This function creates a more granular graphic to display the results of a changepoint influence analysis on the location of the changepoints. The graphic is a histogram of the observed segmentations under the "delete" or "outlier" Influence analysis. The colour and line type of the bars at the `original.cpts` locations reflect their stability. The first value of their arguments denotes a stable changepoint - which appears at the same location in all influence segmentations. The second argument denotes an unstable changepoint - which doesn't appear at the same location in all influence segmentations, either it moves or is deleted. The third argument denotes changepoint locations which are deemed outliers as two changepoints occur at consecutive locations (surrounding the outlying observation). Please note that the `type="Global"` only uses colour and not line type.

`type="Difference"` gives the difference between the observed and expected changepoint segmentations under the "delete" or "outlier" Influence analysis. A positive value can only occur where a changepoint is contained in the observed segmentations but is not present in the expected (an additional changepoint time). A negative value can only occur at the original changepoint location where the changepoint is not present in atleast one of the observed segmentations. Note that the expected segmentations take into account the fact that a changepoint at a timepoint, say 100, will move (to 99) when a timepoint prior to it is deleted.

`type="Global"` histograms the observed segmentations. Colour is added to the original changepoint locations and a horizontal (light grey) line is added to the plot to denote the maximum count. Any original changepoint bars that do not meet this grey line indicates that the changepoint is unstable as it either moves or is deleted in atleast one of the observed segmentations. For large datasets this can be difficult to view what is going on at any locations that appear as black bars as these are typically small counts. Hence the inclusion of the "Local" option.

type="Local" histograms the observed segmentations with the original changepoint locations removed. This is to allow users to see the smaller counts that can be masked in larger datasets. These are the locations where either original changepoints move to or additional changepoints are added.

Value

The function returns plot(s) and a list containing the labels of the original.cpts as either "stable", "unstable", or "outlier". If type="Difference" and expected.class=NULL then the expected class is also returned as the first element of the list.

Author(s)

Rebecca Killick

References

Wilms I, Killick R, Matteson DS (2022) Graphical Influence Diagnostics for Changepoint Models, *Journal of Computational and Graphical Statistics*, **31:3**, 753–765 DOI: 10.1080/10618600.2021.2000873

See Also

[influence-methods](#), [StabilityOverview](#), [ParameterStability](#), [InfluenceMap](#)

Examples

```
##### Generate Simulated data example #####
set.seed(30)
x = c(rnorm(50), rnorm(50, mean = 5), rnorm(1, mean = 15), rnorm(49, mean = 5), rnorm(50, mean = 4))
xcpt = cpt.mean(x,method='PELT') # Get the changepoints via PELT

##### Get the influence for both delete and outlier options #####
x.inf = influence(xcpt)

##### Location Stability Difference plot #####
exp.class=LocationStability(cpts(xcpt), type = 'Difference', x.inf, cpt.lwd = 4, las = 1)
# note that the expected.class is also returned

##### Location Stability Global plot #####
exp.class=LocationStability(cpts(xcpt), type = 'Global', x.inf, cpt.lwd = 4, las = 1)

##### Location Stability Local plot #####
exp.class=LocationStability(cpts(xcpt), type = 'Local', x.inf, cpt.lwd = 4, las = 1)
```

ParameterStability *Parameter Stability Graphic*

Description

Plots the middle detail level of the changepoint parameter stability according to the influence measure.

Usage

```
ParameterStability(influence,original.mean=NULL,digits=6,ylab='',xlab='Index',
  cpt.col='red',cpt.width=3,...)
```

Arguments

influence	The influence as calculated the <code>influence()</code> function provided within this package. This is a list object.
original.mean	A vector, <i>length n</i> , of the mean under the original segmentation at each time-point.
digits	The number of significant figures to round the mean values to before plotting. (Purely to reduce the number of points plotted to make the graphics smaller for storage and loading)
ylab, xlab	The labels for the x- and y-axis, character vector expected.
cpt.col	Colour of the original parameter vector when plotted. Any values accepted by the <code>col</code> plotting argument are allowed.
cpt.width	Width of the original parameter vector when plotted. Any values accepted by the <code>lwd</code> plotting argument are allowed.
...	Any other arguments to be passed to the <code>plot()</code> function.

Details

This function creates a more granular graphic to display the results of a changepoint influence analysis on the estimated segment parameter. The graphic depicts the observed segment parameters under the "delete" or "outlier" Influence analysis. The intensity of the grey denotes how often that parameter values was seen across all segmentations. We overlay this with the original segment parameters.

Value

The function returns a plot (silently).

Author(s)

Rebecca Killick

References

Wilms I, Killick R, Matteson DS (2022) Graphical Influence Diagnostics for Changepoint Models, *Journal of Computational and Graphical Statistics*, **31:3**, 753–765 DOI: 10.1080/10618600.2021.2000873

See Also

[influence-methods](#), [StabilityOverview](#), [LocationStability](#), [InfluenceMap](#)

Examples

```
##### Generate Simulated data example #####
set.seed(30)
x=c(rnorm(50),rnorm(50,mean=5),rnorm(1,mean=15),rnorm(49,mean=5),rnorm(50,mean=4))
xcpt = cpt.mean(x,method='PELT') # Get the changepoints via PELT

##### Get the influence for both delete and outlier options #####
x.inf = influence(xcpt)

##### Parameter Stability plot #####
ParameterStability(x.inf, original.mean = rep(param.est(xcpt)$mean,
  times=diff(c(0,xcpt@cpts))), las = 1)
# note that the original mean is an n length vector and you can use the above code
# to get this from the original changepoint locations.
```

StabilityOverview

Stability Overview Graphic

Description

Plots the overview of the stability according to the influence measure.

Usage

```
StabilityOverview(data, original.cpts, influence, cpt.lwd=2,
  cpt.col=c("#009E73", "#E69F00", "#E41A1C"), cpt.lty=c("dashed", "dotdash", "dotted"),
  ylab=' ', xlab='Index', legend.args=list(display=TRUE, x="left", y=NULL, cex = 1, bty="n",
  horiz=TRUE, xpd=FALSE), ...)
```

Arguments

<code>data</code>	A vector containing the data on which you have run your changepoint method.
<code>original.cpts</code>	An ordered vector of the changepoint locations found by your favourite changepoint method.
<code>influence</code>	The influence as calculated the <code>influence()</code> function provided within this package. This is a list object.
<code>cpt.lwd</code>	The line width to be used when plotting the <code>original.cpts</code> on the data. Standard lwd values allowed. A single value is expected.

<code>cpt.col</code>	Colour of the original <code>.cpts</code> lines when plotted. We need three colours specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. Any values accepted by the <code>col</code> plotting argument are allowed.
<code>cpt.lty</code>	Line type of the original <code>.cpts</code> lines when plotted. We need three line types specified here, corresponding to the "stable", "unstable" and "outlier" categories respectively. Any values accepted by the <code>lty</code> plotting argument are allowed.
<code>ylab, xlab</code>	The labels for the x- and y-axis, character vector expected.
<code>legend.args</code>	These arguments are passed to the <code>legend()</code> function to control the legend position etc..
<code>...</code>	Any other arguments to be passed to the <code>plot()</code> function.

Details

This function creates a first summary graphic to display the results of a changepoint influence analysis. The graphic is a plot of the original data with the changepoints as vertical lines at their respective positions. The colour and line type of the changepoint vertical lines reflect their stability. The first value of their arguments denotes a stable changepoint - which appears at the same location in all influence segmentations. The second argument denotes an unstable changepoint - which doesn't appear at the same location in all influence segmentations, either it moves or is deleted. The third argument denotes changepoint locations which are deemed outliers as two changepoints occur at consecutive locations (surrounding the outlying observation).

Value

The function returns a plot and a list containing the labels of the original `.cpts` as either "stable", "unstable", or "outlier".

Author(s)

Rebecca Killick

References

Wilms I, Killick R, Matteson DS (2022) Graphical Influence Diagnostics for Changepoint Models, *Journal of Computational and Graphical Statistics*, **31:3**, 753–765 DOI: 10.1080/10618600.2021.2000873

See Also

[influence-methods](#), [LocationStability](#), [ParameterStability](#), [InfluenceMap](#)

Examples

```
#### Generate Simulated data example ####
set.seed(30)
x=c(rnorm(50),rnorm(50,mean=5),rnorm(1,mean=15),rnorm(49,mean=5),rnorm(50,mean=4))
xcpt = cpt.mean(x,method='PELT') # Get the changepoints via PELT

#### Get the influence for both delete and outlier options ####
x.inf = influence(xcpt)
```

```
#### Stability Dashboard ####
StabilityOverview(x,cpts(xcpt),x.inf,las=1,
  legend.args=list(display=TRUE,x="topright",y=NULL,cex=1.5,bty="n",horiz=FALSE,xpd=FALSE))
```

welldata

Welllog data

Description

This data has been used in previous changepoint papers and is described and provided in "On-line inference for hidden Markov models via particle filters" by Fearnhead and Clifford in 2003. The data consists of measurements of the nuclear magnetic response of underground rocks.

Please note that this is the original data. The data analyzed in the majority of publications has been standardized and/or had the outliers removed. Papers typically only analyze a portion of the 4050 vector too.

Usage

```
welldata
```

Format

A vector of length 4050.

Source

<https://doi.org/10.1111/1467-9868.00421>

Examples

```
#### Load the data in the R package changepoint.influence ####
data("welldata")
welllog = welldata[1001:2000]
# Extract the mid section of the data as analyzed in other papers
n = length(welllog)
var = NULL; for (i in 30:1000){var[i]=var(welllog[(i-29):i])}
welllogs = welllog/sqrt(median(var, na.rm = TRUE))
# rescale the data to have unit variance across time,
# note that there may still be changes in variance across the series.

#### Apply PELT to the welllog data ####
out.PELT = cpt.mean(welllogs, method = 'PELT')

#### Calculate the influence measures ####
welllogs.inf = influence(out.PELT)
```

```
# the code extracts all the details of the original cpt.mean() function call
# and uses these in the calculation of the influence for the modified data.

#### Stability Dashboards ####
StabilityOverview(welllogs,cpts(out.PELT),welllogs.inf,las=1,ylab='Nuclear-Magnetic Response',
  legend.args=list(display=TRUE,x="bottomright",y=NULL,cex=1.5,bty="n",horiz=FALSE,xpd=FALSE))
# We can specify where the legend will sit in the graphic via the legend.args
# which are passed to the legend() function. We can also include additional
# arguments to pass to the plotting such as las=1 here.

#### Location Stability plot ####
exp.seg=LocationStability(cpts(out.PELT), welllogs.inf, type = 'Difference', cpt.lwd = 4, las = 1)
# Note that if the expected segmentation is not provided, it will be calculated
# and then returned so that the user can avoid calculating this again in other plot calls.

#### Parameter Stability plot ####
ParameterStability(welllogs.inf, original.mean = rep(param.est(out.PELT)$mean,
  times=diff(c(0,out.PELT@cpts))), las = 1, ylab = 'Nuclear-Magnetic Response')
# Note that the original.mean argument is provided for each timepoint so is a length n vector.

#### Influence Map ####
welllogs.inf = influence(out.PELT, method = "delete")
inf.resid.del=InfluenceMap(cpts(out.PELT), welllogs.inf, data = welllogs, include.data = TRUE,
  ylab = 'Nuclear-Magnetic\n Response')

welllogs.inf = influence(out.PELT, method = "outlier")
inf.resid.out=InfluenceMap(cpts(out.PELT), welllogs.inf, data = welllogs, include.data = TRUE,
  ylab='Nuclear-Magnetic\n Response')
```

Index

- * **changepoint**
 - changepoint.influence-package, 2
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- * **datasets**
 - welldata, 12
- * **influence**
 - changepoint.influence-package, 2
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- * **methods**
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- * **models**
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- * **package**
 - changepoint.influence-package, 2
- * **plot**
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- * **segmentation**
 - InfluenceMap, 4
 - LocationStability, 6
 - ParameterStability, 9
 - StabilityOverview, 10
- changepoint.influence
 - (changepoint.influence-package), 2
- changepoint.influence-package, 2
- InfluenceMap, 3, 4, 8, 10, 11
- LocationStability, 3, 5, 6, 10, 11
- ParameterStability, 3, 5, 8, 9, 11
- StabilityOverview, 3, 5, 8, 10, 10
- welldata, 12
- welllog (welldata), 12