

Package ‘ConvergenceClubs’

December 14, 2018

Title Finding Convergence Clubs

Description Functions for clustering regions that form convergence clubs, according to the definition of Phillips and Sul (2009) <doi:10.1002/jae.1080>.

Version 1.4.3

Date 2018-12-05

URL <https://CRAN.R-project.org/package=ConvergenceClubs>

BugReports <https://github.com/rhobis/ConvergenceClubs/issues>

Depends R (>= 3.0.1)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports lmtest (>= 0.9-35), sandwich (>= 2.3-4)

NeedsCompilation no

Author Roberto Sichera [aut, cre, cph],
Pietro Pizzuto [aut]

Maintainer Roberto Sichera <roberto.sichera@unipa.it>

Repository CRAN

Date/Publication 2018-12-14 14:50:03 UTC

R topics documented:

ConvergenceClubs-package	2
club	3
computeH	4
coreG	5
dim.convergence.clubs	6
estimateMod	7
filteredGDP	8
findClubs	8

GDP	11
mergeClubs	11
mergeDivergent	13
plot.convergence.clubs	15
print.convergence.clubs	17
summary.convergence.clubs	17

Index	18
--------------	-----------

ConvergenceClubs-package

ConvergenceClubs: Finding Convergence Clubs

Description

Functions for clustering regions that form convergence clubs, according to the definition of Phillips and Sul (2009) <doi:10.1002/jae.1080>.

Main functions

The package's main functions are `findClubs` and `mergeClubs`. The former finds clubs of convergence, given a dataset with units in rows and years in columns, returning an object of class `convergence.clubs`. The latter takes as argument an object of class `convergence.clubs` and applies the clustering procedure to the convergence clubs contained in the argument, according to either Phillips and Sul (2009) or von Lyncker and Thoennessen (2016) procedure.

Author(s)

Maintainer: Roberto Sichera <roberto.sichera@unipa.it> [copyright holder]

Authors:

- Pietro Pizzuto <pietro.pizzuto02@unipa.it>

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

von Lyncker, K.; Thoennessen, R., 2017. Regional club convergence in the EU: evidence from a panel data analysis. *Empirical Economics* 52 (2), 525-553

See Also

Useful links:

- <https://CRAN.R-project.org/package=ConvergenceClubs>
- Report bugs at <https://github.com/rhobis/ConvergenceClubs/issues>

club	<i>Find a club</i>
------	--------------------

Description

Add units to core group according to step 3 of the clustering algorithm by Phillips and Sul (2007, 2009), in order to find the enlarged club.

Usage

```
club(X, dataCols, core, time_trim, HACmethod = c("FQSB", "AQSB"),
     cstar = 0)
```

Arguments

X	matrix or dataframe containing data (preferably filtered data in order to remove business cycles)
dataCols	integer vector with the column indices of the data
core	an integer vector containing the id's of units in core group
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running log t regression model. Phillips and Sul (2007, 2009) suggest to discard the first third of the period.
HACmethod	string indicating whether a Fixed Quadratic Spheric Bandwidth (HACmethod="FQSB") or an Adaptive Quadratic Spheric Bandwidth (HACmethod="AQSB") should be used for the truncation of the Quadratic Spectral kernel in estimating the <i>logt</i> regression model with heteroskedasticity and autocorrelation consistent standard errors. The default method is "FQSB".
cstar	numeric scalar, indicating the threshold value of the sieve criterion c^* to include units in the detected core (primary) group (step 3 of Phillips and Sul (2007, 2009) clustering algorithm). The default value is 0.

Value

A list of three objects: *id*, a vector containing the row indices of club units in the original dataframe (input of function `findClubs`); *rows*, a vector of row indices of club units in the current dataset (input of function `club`); *model*, a list containing information about the model used to run the t-test on the units in the club.

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

computeH

*Compute H values***Description**

Computes H values (cross-sectional variance) according to the clustering algorithm by Phillips and Sul (2007, 2009)

Usage

```
computeH(X, quantity = "H", id)
```

Arguments

X	matrix or dataframe containing data (preferably filtered, in order to remove business cycles)
quantity	string indicating the quantity that should be returned. The options are "H", the default, only the vector of cross-sectional variance is returned; "h", only the matrix of transition path h is returned; "both", a list containing both h and H is returned.
id	optional; row index of regions for which H values are to be computed; if missing, all regions are used

Details

The cross sectional variation H_{it} is computed as the quadratic distance measure for the panel from the common limit and under the hypothesis of the model should converge to zero as t tends towards infinity:

$$H_t = N^{-1} \sum_{i=1}^N (h_{it} - 1)^2 \rightarrow 0, \quad t \rightarrow \infty$$

where

$$h_{it} = \frac{\log y_{it}}{(N^{-1} \sum_{i=1}^N \log y_{it})}$$

Value

A numeric vector, a matrix or a list, depending on the value of quantity

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

Examples

```

data("filteredGDP")

h <- computeH(filteredGDP[,-1], quantity="h")
H <- computeH(filteredGDP[,-1], quantity="H")
b <- computeH(filteredGDP[,-1], quantity="both")

```

coreG	<i>Find core (primary) group</i>
-------	----------------------------------

Description

Find the Core (primary) group according to step 2 of the clustering algorithm by Phillips and Sul (2007, 2009)

Usage

```

coreG(X, dataCols, time_trim, threshold = -1.65, HACmethod = c("FQSB",
  "AQSB"), type = c("max", "all"))

```

Arguments

X	matrix or dataframe containing data (preferably filtered data in order to remove business cycles)
dataCols	integer vector with the column indices of the data
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running log t regression model. Phillips and Sul (2007, 2009) suggest to discard the first third of the period.
threshold	numeric value indicating the threshold to be used to perform the one-tail t test; default is -1.65.
HACmethod	string indicating whether a Fixed Quadratic Spheric Bandwidth (HACmethod="FQSB") or an Adaptive Quadratic Spheric Bandwidth (HACmethod="AQSB") should be used for the truncation of the Quadratic Spectral kernel in estimating the <i>log t</i> regression model with heteroskedasticity and autocorrelation consistent standard errors. The default method is "FQSB".
type	one of "max" or "all"; "max" includes only the region with maximum t-value. The default option is "max"; "all" includes all units that pass the test t in the core formation (step 2).

Details

According to the second step of the Phillips and Sul clustering algorithm (2007, 2009), the *log t* regression should be run for the first k units $2 < k < N$ maximizing k under the condition that $t - value > -1.65$. In other words, the core group size k^* is chosen as follows:

$$k^* = \operatorname{argmax}_k \{t_k\}$$

subject to

$$\min t_k > -1.65$$

Such behavior is obtained with `type="max"`; if `type="all"`, all units that satisfy $t_k > -1.65$ are added to core group.

If the condition $t_k > -1.65$ does not hold for $k = 2$ (the first two units), the algorithm drops the first unit and repeats the same procedure for the next pair of units. If $t_k > -1.65$ does not hold for any couple of units, the whole panel diverges.

Value

A numeric vector containing the row indices of the units included in the core group; if a core group cannot be found, returns FALSE.

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

dim.convergence.clubs dim method for S3 object convergence.clubs

Description

dim method for S3 object convergence.clubs

Usage

```
## S3 method for class 'convergence.clubs'
dim(x, ...)
```

Arguments

`x` an object of class `convergence.clubs`.
`...` other parameters to pass to function `summary()`.

Value

an integer vector with two values: the first one indicates the number of clubs, the second one the number of divergent units

estimateMod	<i>Log-t test for convergence</i>
-------------	-----------------------------------

Description

Estimates the *log-t* regression model proposed by Phillips and Sul (2007, 2009) in order to investigate the presence of convergence by adopting the Andrews estimator of long-run variance (fixed or adaptive bandwidth of the kernel).

Usage

```
estimateMod(H, time_trim = 1/3, HACmethod = c("FQSB", "AQSB"))
```

Arguments

H	vector of H values
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running <i>log t</i> regression model. Phillips and Sul (2007, 2009) suggest to discard the first third of the period.
HACmethod	string indicating whether a Fixed Quadratic Spectral Bandwidth (HACmethod="FQSB") or an Adaptive Quadratic Spectral Bandwidth (HACmethod="AQSB") should be used for the truncation of the Quadratic Spectral kernel in estimating the <i>log t</i> regression model with heteroskedasticity and autocorrelation consistent standard errors. The default method is "FQSB".

Details

The following linear model is estimated:

$$\log \frac{H_1}{H_t} - 2 \log(\log t) = \alpha + \beta \log t + u_t$$

Heteroskedasticity and autocorrelation consistent (HAC) standard errors are used with Quadratic Spectral kernel (Andrews, 1991), If HACmethod="FQSB", a fixed bandwidth parameter is applied, while with HACmethod="AQSB" an adaptive bandwidth parameter is employed.

Value

A named vector containing information about the model used to run the t-test on the units in the club: beta coefficient, standard deviation, t-statistics and p-value.

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

Andrews, D. W., 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica: Journal of the Econometric Society*, 817-858.

filteredGDP

Filtered per-capita GDP of 152 Countries from 1970 to 2003

Description

A dataset containing the per-capita GDP of 152 Countries over 34 years (Phillips and Sul, 2009). Data were filtered in order to remove business cycles.

Usage

```
data(filteredGDP)
```

Format

A data frame with 152 rows and 35 variables.

Details

ID Country names (character);

Y1970, ..., Y2003 per-capita GDP from year 1970 to 2003 (filtered in order to remove business cycles).

References

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

findClubs

Finds convergence clubs

Description

Find convergence clubs by means of Phillips and Sul clustering procedure.

Usage

```
findClubs(X, dataCols, unit_names = NULL, refCol, time_trim = 1/3,
          cstar = 0, HACmethod = c("FQSB", "AQSB"))
```


Arguments

X	dataframe containing data (preferably filtered data in order to remove business cycles)
dataCols	integer vector with the column indices of the data
unit_names	integer scalar indicating, if present, the index of a column with codes of the units
refCol	integer scalar indicating the index of the column to use for ordering data
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running log t regression model. Phillips and Sul (2007, 2009) suggest to discard the first third of the period.
cstar	numeric scalar, indicating the threshold value of the sieve criterion c^* to include units in the detected core (primary) group (step 3 of Phillips and Sul (2007, 2009) clustering algorithm). The default value is 0.
HACmethod	string indicating whether a Fixed Quadratic Spectral Bandwidth (HACmethod="FQSB") or an Adaptive Quadratic Spectral Bandwidth (HACmethod="AQSB") should be used for the truncation of the Quadratic Spectral kernel in estimating the <i>log-t</i> regression model with heteroskedasticity and autocorrelation consistent standard errors. The default method is "FQSB".

Details

In order to investigate the presence of convergence clubs according to the Phillips and Sul clustering procedure, the following steps are implemented:

1. (Cross section last observation ordering): Sort units in descending order according to the last panel observation of the period;
2. (Core group formation): Run the log t regression for the first k units ($2 < k < N$) maximizing k under the condition that t -value is > -1.65 . In other words, chose the core group size k^* as follows:

$$k^* = \operatorname{argmax}_k \{t_k\}$$

subject to

$$\min\{t_k\} > -1.65$$

If the condition $t_k > -1.65$ does not hold for $k = 2$ (the first two units), drop the first unit and repeat the same procedure. If $t_k > -1.65$ does not hold for any units chosen, the whole panel diverges;

3. (Sieve the data for club membership): After the core group is detected, run the *logt* regression for the core group adding (one by one) each unit that does not belong to the latter. If t_k is greater than a critical value c^* add the new unit in the convergence club. All these units (those included in the core group k^* plus those added) form the first convergence club;
4. (Recursion and stopping rule): If there are units for which the previous condition fails, gather all these units in one group and run the *log-t* test to see if the condition $t_k > -1.65$ holds. If the condition is satisfied, conclude that there are two convergence clubs. Otherwise, step 1 to 3 should be repeated on the same group to determine whether there are other subgroups that constitute convergence clubs. If no further convergence clubs are found (hence, no k in step 2 satisfies the condition $t_k > -1.65$), the remaining units diverge.

Note that Phillips and Sul (2007) suggest to make sure $t_k > -1.65$ for the first club. Otherwise, repeat the procedure by increasing the value of the c^* parameter until the condition $t_k > -1.65$ is satisfied for the first club.

Value

An object of class `convergence.clubs`, containing a list of Convergence Clubs, for each club a list is returned with the following objects: `id`, a vector containing the row indices of the units in the club; `model`, a list containing information about the model used to run the t-test on the units in the club; `unit_names`, a vector containing the names of the units of the club (optional, only included if parameter `unit_names` is given)

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

Andrews, D. W., 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica: Journal of the Econometric Society*, 817-858.

See Also

[mergeClubs](#), Merges a list of clubs created by `findClubs`;

[mergeDivergent](#), Merges divergent units according to the algorithm proposed by von Lyncker and Thoennesen (2016)

Examples

```
data("filteredGDP")

## Not run:
# Cluster Countries using GDP from year 1970 to year 2003
clubs <- findClubs(filteredGDP, dataCols=2:35, unit_names = 1, refCol=35,
                  time_trim = 1/3, cstar = 0, HACmethod = "AQSB")

## End(Not run)

clubs <- findClubs(filteredGDP, dataCols=2:35, unit_names = 1, refCol=35,
                  time_trim = 1/3, cstar = 0, HACmethod = "FQSB")
summary(clubs)
```

GDP

Per-capita GDP of 152 Countries from 1970 to 2003

Description

A dataset containing the per-capita GDP of 152 Countries over 34 years (Phillips and Sul, 2009).

Usage

```
data(GDP)
```

Format

A data frame with 152 rows and 35 variables.

Details

ID Country names (character);

Y1970, ..., Y2003 per-capita GDP from year 1970 to 2003.

References

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

mergeClubs

Merge convergence clubs

Description

Merges a list of clubs created with the function `findClubs` by either Phillips and Sul method or von Lyncker and Thoennessen procedure.

Usage

```
mergeClubs(clubs, time_trim, mergeMethod = c("PS", "vLT"),  
           threshold = -1.65, mergeDivergent = FALSE, estar = -1.65)
```

Arguments

clubs	an object of class convergence.clubs (created by findClubs function)
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running log t regression model; if omitted, the same value used for clubs is used.
mergeMethod	character string indicating the merging method to use. Methods available are "PS" for Phillips and Sul (2009) and "vLT" for von Lyncker and Thoennessen (2017).
threshold	a numeric value indicating the threshold to be used with the t-test.
mergeDivergent	logical, if TRUE, indicates that merging of divergent units should be tried.
estar	a numeric value indicating the threshold e^* to test if divergent units may be included in one of the new convergence clubs. To be used only if mergeDivergent=TRUE

Details

Phillips and Sul (2009) suggest a "club merging algorithm" to avoid over determination due to the selection of the parameter c^* . This algorithm suggests to merge for adjacent groups. In particular, it works as follows:

1. Take the first two groups detected in the basic clustering mechanism and run the log-t test. If the t-statistic is larger than -1.65, these groups together form a new convergence club;
2. Repeat the test adding the next group and continue until the basic condition (t-statistic > -1.65) holds;
3. If convergence hypothesis is rejected, conclude that all previous groups converge, except the last one. Hence, start again the test merging algorithm beginning from the group for which the hypothesis of convergence did not hold.

On the other hand, von Lyncker and Thoennessen (2017), propose a modified version of the club merging algorithm that works as follows:

1. Take all the groups detected in the basic clustering mechanism (P) and run the t-test for adjacent groups, obtaining a $(M \times 1)$ vector of convergence test statistics t (where $M = P - 1$ and $m = 1, \dots, M$);
2. Merge for adjacent groups starting from the first, under the conditions $t(m) > -1.65$ and $t(m) > t(m + 1)$. In particular, if both conditions hold, the two clubs determining $t(m)$ are merged and the algorithm starts again from step 1, otherwise it continues for all following pairs;
3. For the last element of vector M (the value of the last two clubs) the only condition required for merging is $t(m = M) > -1.65$.

Value

Ad object of class convergence.clubs, containing a list of Convergence Clubs, for each club a list is return with the following objects: id, a vector containing the row indices of the units in the club; model, a list containing information about the model used to run the t-test on the units in the club; unit_names, a vector containing the names of the units of the club (optional, only included if parameter unit_names is given)

References

Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.

Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.

von Lyncker, K.; Thoennessen, R., 2017. Regional club convergence in the EU: evidence from a panel data analysis. *Empirical Economics* 52 (2), 525-553

See Also

[findClubs](#), finds convergence clubs by means of Phillips and Sul clustering procedure.

[mergeDivergent](#), merges divergent units according to the algorithm proposed by von Lyncker and Thoennessen (2017).

Examples

```
data("filteredGDP")

# Cluster Countries using GDP from year 1970 to year 2003
clubs <- findClubs(filteredGDP, dataCols=2:35, unit_names = 1, refCol=35,
                  time_trim = 1/3, cstar = 0, HACmethod = "FQSB")
summary(clubs)

# Merge clusters
mclubs <- mergeClubs(clubs, mergeMethod='PS', mergeDivergent=FALSE)
summary(mclubs)

mclubs <- mergeClubs(clubs, mergeMethod='vLT', mergeDivergent=FALSE)
summary(mclubs)
```

mergeDivergent

Merge divergent units

Description

Merges divergent units according the algorithm proposed by von Lyncker and Thoennessen (2017)

Usage

```
mergeDivergent(clubs, time_trim, estar = -1.65)
```

Arguments

clubs	an object of class <code>convergence.clubs</code> (created by <code>findClub</code> or <code>mergeClubs</code> function)
time_trim	a numeric value between 0 and 1, representing the portion of time periods to trim when running log t regression model; if omitted, the same value used for <code>clubs</code> is used.
estar	a numeric value indicating the threshold e^* to test if divergent units may be included in one of the new convergence clubs. To be used only if <code>mergeDivergent=TRUE</code> .

Details

von Lyncker and Thoennesen (2017) claim that units identified as divergent by the basic clustering procedure by Phillips and Sul might not necessarily still diverge in the case of new convergence clubs detected with the club merging algorithm. To test if divergent units may be included in one of the new convergence clubs, they propose the following algorithm:

1. Run a log t-test for all diverging units, and if $t_k > -1.65$ all these units form a convergence club (This step is implicitly included in Phillips and Sul basic algorithm);
2. Run a log t-test for each diverging units and each club, creating a matrix of t-values with dimensions $d \times p$, where each row d represents a divergent region and each column p a convergence club;
3. Take the highest $t > e^*$ and add the respective region to the respective club and restart from the step 1. the authors suggest to use $e^* = t = -1.65$;
4. The algorithm stops when no t-value $> e^*$ is found in step 3, and as a consequence all remaining units are considered divergent.

Value

A list of Convergence Clubs, for each club a list is return with the following objects: `id`, a vector containing the row indices of the units in the club; `model`, a list containing information about the model used to run the t-test on the units in the club; `unit_names`, a vector containing the names of the units of the club (optional, only included if it is present in the `clubs` object given in input).

References

- Phillips, P. C.; Sul, D., 2007. Transition modeling and econometric convergence tests. *Econometrica* 75 (6), 1771-1855.
- Phillips, P. C.; Sul, D., 2009. Economic transition and growth. *Journal of Applied Econometrics* 24 (7), 1153-1185.
- von Lyncker, K.; Thoennesen, R., 2017. Regional club convergence in the EU: evidence from a panel data analysis. *Empirical Economics* 52 (2), 525-553

See Also

[mergeClubs](#), Merges a list of clubs created by `findClubs`;

Examples

```

data("filteredGDP")

#Cluster Countries using GDP from year 1970 to year 2003
clubs <- findClubs(filteredGDP, dataCols=2:35, unit_names = 1, refCol=35,
                  time_trim = 1/3, cstar = 0, HACmethod = "FQSB")
summary(clubs)

# Merge clusters and divergent units
mclubs <- mergeClubs(clubs, mergeDivergent=TRUE)
summary(mclubs)

```

plot.convergence.clubs

Plot method for S3 class convergence.clubs

Description

Plot the transition paths of units in the convergence clubs and the average transition paths of those clubs.

Usage

```

## S3 method for class 'convergence.clubs'
plot(x, y = NULL, nrows = NULL,
     ncols = NULL, clubs, avgTP = TRUE, avgTP_clubs, y_fixed = FALSE,
     legend = FALSE, save = FALSE, filename, path, width = 7,
     height = 7, device = c("pdf", "png", "jpeg"), res, ...)

```

Arguments

x	an x of class convergence.clubs.
y	unused, added for compatibility with function plot
nrows	number of rows of the graphical layout, if NULL, it is automatically defined
ncols	number of columns of the graphical layout, if NULL, it is automatically defined
clubs	numeric scalar or vector, indicating for which clubs the transition path plot should be generated. Optional, if omitted, plots for all clubs are produced. If clubs=NULL, transition path are not plotted for any club.
avgTP	logical, indicates if a plot of the average transition paths of the convergence clubs should be produced, default to TRUE
avgTP_clubs	numeric scalar or vector, indicating for which clubs the average transition path should be displayed. Optional, if omitted, average transition paths for all clubs are plotted
y_fixed	logical, should the scale of the y axis be the same for all plots?

legend	logical, should a legend be displayed?
save	logical, should the plot be saved as a file?
filename	optional, a string indicating the name of the file where the plot should be saved; must include the extension (e.g. "plot.pdf")
path	optional, a string representing the path of the directory where the plot should be saved; the path should not contain the a final slash symbol ("/")
width	the width of the plot, in inches.
height	the height of the plot, in inches.
device	string indicating the format to be used to save the plot; one of "pdf", "png" or "jpeg".
res	the resolution of the image, in ppi; only used with device="png" and device="jpeg"
...	other parameters to pass to function plot().

Details

nrows and ncols are optional parameters used to define the row and column number for the plot layout. Both or just one of them may be specified. If none of them is specified, the layout dimension is chosen automatically.

Graphical parameters of the horizontal line plotted at $y=1$ may be modified by using the following regular plot parameters:

- lty defines the type of line, default is "solid"
- lwd defines the width of the line, default is 2
- col defines the color of the line, default is "black"; set it to "white" to remove the horizontal line.

If legend=TRUE and a column with units' names is available in the x object, those names are truncated to fit the plot's legend. The graphical parameter cex may be used to modify the size of the legend's labels, default is 0.8

Note that, when using RStudio, one may incur in an error if the plot window is too small. Enlarging the plot window usually solves the problem.

Examples

```
data("filteredGDP")

clubs <- findClubs(filteredGDP, dataCols=2:35, unit_names = 1, refCol=35, time_trim = 1/3,
                  cstar = 0, HACmethod = "FQSB")

#plot transition paths for all clubs
plot(clubs)
plot(clubs, y_fixed=TRUE)
plot(clubs, nrows=2,ncols=4)

plot(clubs, ncols=3, lty='dotted', lwd=3, col="blue")
plot(clubs, ncols=3, y_fixed=TRUE, lty='dotted', lwd=3, col="blue")
```



```

#Plot transition paths only for some clubs
plot(clubs, clubs=c(2,4,5))
plot(clubs, nrows=1, ncols=3, clubs=c(2,4,5), avgTP = FALSE)
plot(clubs, nrows=1, ncols=3, clubs=c(2,4,5), avgTP = FALSE, legend=TRUE)
plot(clubs, clubs=c(2,4,5), avgTP_clubs = c(1,3))
plot(clubs, clubs=c(2,4,5), avgTP_clubs = c(1,3), legend=TRUE)

#Only plot average transition paths
plot(clubs, clubs=NULL, avgTP = TRUE, legend=TRUE)

```

```
print.convergence.clubs
```

Print method for S3 object convergence.clubs

Description

Print method for S3 object convergence.clubs

Usage

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

Arguments

x	an object of class convergence.clubs.
...	other parameters to pass to function summary().

```
summary.convergence.clubs
```

Summary method for S3 object convergence.clubs

Description

Summary method for S3 object convergence.clubs

Usage

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

Arguments

object	an object of class convergence.clubs.
...	other parameters to pass to function summary().

Index

*Topic **datasets**

filteredGDP, [8](#)
GDP, [11](#)

club, [3](#)

computeH, [4](#)

ConvergenceClubs
(ConvergenceClubs-package), [2](#)

ConvergenceClubs-package, [2](#)

coreG, [5](#)

dim.convergence.clubs, [6](#)

estimateMod, [7](#)

filteredGDP, [8](#)

findClubs, [8](#), [13](#)

GDP, [11](#)

mergeClubs, [10](#), [11](#), [14](#)

mergeDivergent, [10](#), [13](#), [13](#)

plot.convergence.clubs, [15](#)

print.convergence.clubs, [17](#)

summary.convergence.clubs, [17](#)