

## Conclusion

This article introduces the R package **BondValuation** and provides guidance on its application for analysis of large data frames of fixed coupon bonds. The theoretical foundation of the package is the generalized valuation methodology developed by Djatschenko (2019). Its seamless implementation in **BondValuation** is framed by a set of routines that assist the user in data quality evaluation and automatically correct corrupted entries.

**BondValuation** is the first R package that properly handles irregular first and final coupon periods of fixed coupon bonds and provides a comprehensive coverage of the day count conventions (DCC) used in the global bond markets. Currently, 16 different DCCs are implemented, which account for the vast majority of the methods used in the global fixed income markets. Within its scope, the R package **BondValuation** performs correctly and efficiently. Nevertheless, the current version of the software remains open for further development and refinement. Essentially, the calculations are performed under the assumption that interest accrual and temporal structure follow the same DCC. The option `CalcMethod` in the functions `BondVal.Price()` and `BondVal.Yield()` can be used to force the temporal structure to follow the *ACT/ACT (ICMA)* method, while the DCC passed to the respective function is used to compute accrued interest. In future versions of the package, I intend to implement an explicit assignment of DCC to both interest accrual and temporal structure, which will increase the flexibility of the package.

A further limitation is that the calendar dates of the temporal structure are currently returned, regardless of whether or not they are business days. Although this is the common approach in theoretical bond valuation, including the possibility of business day adjustments for cash flows would be particularly appealing to practitioners. Along with business day adjustments, future versions of **BondValuation** can be extended by methods for bond portfolio analysis.

The current version of **BondValuation** is designed for processing non-callable, option-free, non-sinkable fixed coupon bonds and zero bonds. With the implemented methods, callable bonds can be analyzed through appropriate adjustment of the maturity date to the next call date, returning the so-called yield-to-worst and the corresponding duration and convexity measures. Based on the implemented functions, the R package **BondValuation** can be extended to incorporate methods for explicit treatment of callable, sinkable and convertible fixed and floating rate bonds.

The R package **BondValuation** provides the computational foundation for the exploration of a variety of interesting research questions related to the analysis of fixed income securities across markets. Even considering the limitations described above, the software is also useful to practitioners. I intend to continuously extend and improve the package, and I highly appreciate feedback from the users.

## Acknowledgments

I would like to thank Ingo Geishecker for our frequent discussions and his profound advice. I am also grateful to Karl Ludwig Keiber and Philipp Otto for their helpful comments and suggestions, and to Inna Keil for her excellent research assistance. All remaining errors are my own responsibility.

## Bibliography

- Banking Federation of the European Union. Master Agreement for Financial Transactions - Supplement to the Derivatives Annex - Interest Rate Transactions, 2004. URL <http://www.ebf.eu/wp-content/uploads/2017/07/10InterestRateTransactions-2004-02699-01-E.pdf>. [p]
- A. Caputo Silva, L. Oliveira de Carvalho, and O. Ladeira de Medeiros. *PUBLIC DEBT: the Brazilian experience*. National Treasury Secretariat and World Bank, Brasilia, BR, 2010. ISBN 978-85-87841-44-5. URL <http://documents.worldbank.org/curated/en/967171469672182286/pdf/700810ESW0P1160Brazilian0Experience.pdf>. [p]
- D. Christie and SWX Swiss Exchange. Accrued Interest & Yield Calculations and Determination of Holiday Calendars, 2003. URL [http://janroman.dhis.org/finance/General/accrued\\_interest\\_en.pdf](http://janroman.dhis.org/finance/General/accrued_interest_en.pdf). [p]
- W. Djatschenko. The Nitty Gritty of Bond Valuation: A Generalized Methodology for Straight Bond Analysis, July 2019. URL <http://dx.doi.org/10.2139/ssrn.3205167>. Discussion Paper. [p]
- D. Eddelbuettel, K. Nguyen, and T. Leitch. *RQuantLib: R Interface to the 'QuantLib' Library*, 2018. URL <https://CRAN.R-project.org/package=RQuantLib>. R package version 0.4.5. [p]

- International Capital Market Association. Rule 251 Accrued Interest Calculation - Excerpt from ICMA's Rules and Recommendations, 2010. URL <https://www.isda.org/a/NIJEE/ICMA-Rule-Book-Rule-251-reproduced-by-permission-of-ICMA.pdf>. [p]
- International Swaps and Derivatives Association, Inc. EMU and Market Conventions: Recent Developments, 1998. [p]
- International Swaps and Derivatives Association, Inc. 2006 ISDA Definitions. International Swaps and Derivatives Association, Inc., New York, 2006. [p]
- Investment Industry Association of Canada (IIAC). Canadian Conventions in Fixed Income Markets - A Reference Document of Fixed Income Securities Formulas and Practices; Release: 1.3, 2018. URL <https://iiac.ca/wp-content/uploads/Canadian-Conventions-in-FI-Markets-Release-1.3.pdf>. [p]
- Itaú Unibanco. *Brazilian Sovereign Fixed Income and Foreign Exchange Markets*. Itaú Unibanco, 1st edition, 2017. [p]
- D. Krgin. *The Handbook of Global Fixed Income Calculations*. Wiley, New York, 1st edition, 2002. ISBN 978-0-471-21835-7. [p]
- J. Mayle. *Standard Securities Calculation Methods: Fixed Income Securities Formulas for Price, Yield, and Accrued Interest*, volume 1. Securities Industry Association, New York, 3rd edition, 1993. ISBN 1-882936-01-9. [p]
- Municipal Securities Rulemaking Board. *MSRB Rule Book*. Municipal Securities Rulemaking Board, Washington, DC, 2017. URL <http://www.msrb.org/msrb1/pdfs/MSRB-Rule-Book-October-2017.pdf>. [p]
- QuantLib Team. Quantlib: A free/open-source library for quantitative finance, 2018. URL <http://quantlib.org/>. [p]

Wadim Djatschenko  
European University Viadrina  
Frankfurt (Oder)  
Germany  
ORCID: 0000-0003-0653-8779  
[wadim.djatschenko@gmx.de](mailto:wadim.djatschenko@gmx.de)

# ConvergenceClubs: A Package for Performing the Phillips and Sul's Club Convergence Clustering Procedure

by Roberto Sichera and Pietro Pizzuto

**Abstract** This paper introduces package *ConvergenceClubs*, which implements functions to perform the Phillips and Sul (2007, 2009) club convergence clustering procedure in a simple and reproducible manner. The approach proposed by Phillips and Sul to analyse the convergence patterns of groups of economies is formulated as a nonlinear time varying factor model that allows for different time paths as well as individual heterogeneity. Unlike other approaches in which economies are grouped a priori, it also allows the endogenous determination of convergence clubs. The algorithm, usage, and implementation details are discussed.

## Introduction

Economic convergence refers to the idea that per-capita incomes of poorer economies will tend to grow at faster rates than those of richer economies. The issue has been widely investigated in economic literature since the classical contributions on economic growth and development (Solow, 1956; Myrdal, 1957). In addition to the traditional concepts of beta and sigma convergence, an increasing amount of literature has recently emerged on the concept of *club convergence*. This notion was originally introduced by Baumol (1986) to describe convergence among a subset of national economies and it has quickly spread also at the regional level. Several contributions have tried to empirically investigate the topic proposing different methodologies. For example, Quah (1996) developed a Markov chain model with probability transitions to estimate the evolution of income distribution. Le Gallo and Dall'Erba (2005) proposed a spatial approach to detect convergence clubs using the Getis–Ord statistic. Corrado et al. (2005) introduced a multivariate stationarity test in order to endogenously identify regional club clustering.

More recently, Phillips and Sul (2007, 2009) proposed a time-varying factor model that allows for individual and transitional heterogeneity to identify convergence clubs. Due to its positive attributes, this methodology has become predominant in the analysis of the convergence patterns of economies. In fact, it has several advantages. First, it allows for different time paths as well as individual heterogeneity, therefore, different transitional paths are possible<sup>1</sup>. Second, unlike other approaches in which economies are grouped a priori, this methodology enables the endogenous (data-driven) determination of convergence clubs. Third, the test does not impose any particular assumption concerning trend stationarity or stochastic non-stationarity since it is robust to heterogeneity and to the stationarity properties of the series.

As for existing routines, Phillips and Sul (2007, 2009) provided Gauss (Aptech Systems, 2016) code used in their empirical studies. Schnurbus et al. (2017) provided a set of R functions to replicate the key results of Phillips and Sul (2009), while Du (2018) developed a full Stata (StataCorp, 2017) package to perform the club convergence algorithm. A dedicated R package for this methodology has been missing. The *ConvergenceClubs* (Sichera and Pizzuto, 2019) package fills this gap, since it allows to carry out the Phillips and Sul's methodology in a simple and reproducible fashion, allowing for easy definition of the parameters. Moreover, our package also implements the alternative club merging algorithm developed by von Lyncker and Thoennessen (2017).

The remainder of the paper is organised as follows. First, the club convergence methodology is presented. Then, the main features of the package are listed and described. Finally, an example based on Phillips and Sul (2009) data is provided.

## Methodology

### The log-t test

The approach proposed by Phillips and Sul is based on a modification of the conventional panel data decomposition of the variable of interest. In fact, panel data  $X_{it}$  are usually decomposed in the

<sup>1</sup>For example, in the context of income convergence, the approach proposed by Phillips and Sul allows to account for heterogeneity in technology growth rates and in the speed of convergence, unlike the traditional neoclassical model *à la* Solow that assumes homogeneous technological progress.

following way:

$$X_{it} = g_{it} + a_{it}, \quad (1)$$

where  $g_{it}$  is the systematic factor (including the permanent common component) and  $a_{it}$  is the transitory component. In order to account for temporal transitional heterogeneity they modify eq. (1) as follows:

$$X_{it} = \left( \frac{g_{it} + a_{it}}{\mu_t} \right) \mu_t = b_{it} \mu_t, \quad (2)$$

where  $b_{it}$  is the systematic idiosyncratic element that is allowed to evolve over time and to include a random component that absorbs  $a_{it}$ , and  $\mu_t$  is the common factor. In this dynamic factor formulation, that allows to separate common from idiosyncratic components,  $b_{it}$  becomes the transition path of the economy to the common steady-state growth path determined by  $\mu_t$ . Particularly, the common growth component  $\mu_t$ , may follow either a trend-stationary process or a non-stationary stochastic trend with drift, since a specific assumption regarding the behaviour of  $\mu_t$  is not necessary.

In order to test if different economies converge, a key role is played by the estimation of  $b_{it}$ . According to the authors, the estimation of this parameter is not possible without imposing additional structural restrictions and assumptions. However, as a viable way to model this element, they propose the construction of the following relative transition component:

$$h_{it} = \frac{X_{it}}{N^{-1} \sum_{i=1}^N X_{it}} = \frac{b_{it}}{N^{-1} \sum_{i=1}^N b_{it}}, \quad (3)$$

which is called *relative transition path* and can be directly computed from the data. In such a way it is possible to remove the common steady-state trend  $\mu_t$ , tracing an individual trajectory for each economy  $i$  in relation to the panel average. In other words, the relative transition path describes the relative individual behaviour as well as the relative departures of the  $i$ -th economy from the common growth path  $\mu_t$ .

In presence of convergence, there should be a common limit in the transition path of each economy and the coefficient  $h_{it}$  should converge towards unity ( $h_{it} \rightarrow 1$ ) for all  $i = 1, \dots, N$ , as  $t \rightarrow \infty$ . At the same time, the cross-sectional variation  $H_t$  (computed as the quadratic distance measure for the panel from the common limit) should converge to zero:

$$H_t = N^{-1} \sum_{i=1}^N (h_{it} - 1)^2 \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (4)$$

In order to construct a formal statistical test for convergence, Phillips and Sul (2007, 2009) assume the following semi-parametric specification of  $b_{it}$ :

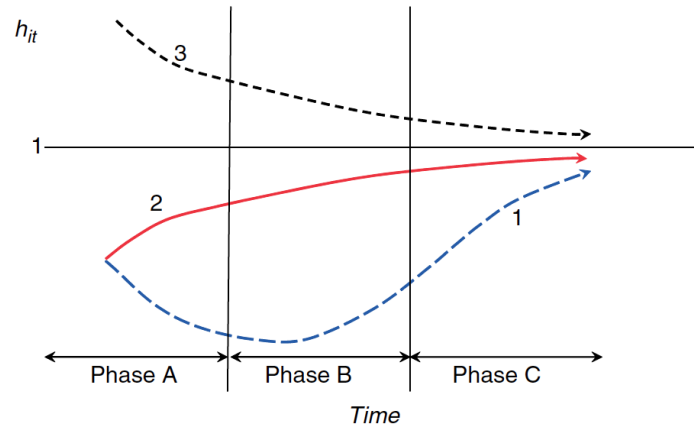
$$b_{it} = b_i + \frac{\sigma_i \zeta_{it}}{L(t) t^\alpha}, \quad (5)$$

where  $b_i$  is fixed (time invariant), the  $\zeta_{it}$  are i.i.d.  $N(0, 1)$  random variables across  $i$ , but weakly dependent over  $t$ ,  $L(t)$  is a slowly varying increasing function (with  $L(t) \rightarrow \infty$  as  $t \rightarrow \infty$ ), and  $\alpha$  is the decay rate, or more specifically in this case, the convergence rate. The null hypothesis of convergence can be written as  $H_0 : b_i = b$  and  $\alpha \geq 0$  versus the alternative  $H_1 : b_i \neq b$  for all  $i$ , or  $\alpha < 0$ . Under  $H_0$ , different transitional paths are possible, including temporary divergence (a stylized way in which economies may converge is shown in fig. 1).

More formally, to test the presence of convergence among different economies, Phillips and Sul (2007, 2009) suggest to estimate the following equation model through the ordinary least squares method:

$$\log \frac{H_1}{H_t} = -2 \log(\log t) = \alpha + \beta \log t + u_t, \quad \text{for } t = [rT], [rT] + 1, \dots, T, \quad (6)$$

where  $H_t = N^{-1} \sum_{i=1}^N (h_{it} - 1)^2$ , and  $H_1/H_t$  is the cross-sectional variance ratio;  $\beta$  is the speed of convergence parameter of  $b_{it}$ ;  $-2 \log(\log t)$  is a penalization function that improves the performance of the test mainly under the alternative;  $r$  assumes a positive value in the interval  $(0, 1]$  in order to discard the first block of observation from the estimation, and  $[rT]$  is the integer part of  $rT$ . To this regard, Phillips and Sul suggest to use  $r \in [0.2, 0.3]$  for small sample size ( $T < 50$ ) as a result of Monte Carlo simulations. The null hypothesis of convergence is tested through a one-sided t-test robust to heteroskedasticity and autocorrelation (HAC) of the inequality  $\alpha > 0$  (using the estimated  $\hat{\beta} = 2\alpha$ ) and specifically it is rejected at the 5% level if  $t_{\hat{\beta}} < -1.65$ . This procedure, generally called *log-t test*,



**Figure 1:** Different transition paths and phases of transition. Source: Phillips and Sul (2009).

has power against cases of club convergence. Hence, if the log-t test is rejected for the whole sample, the authors suggest to repeat the test procedures according to a clustering mechanism consisting of four steps, described below (see next section).

### The clustering algorithm

When the log t-test is rejected for the whole sample, the test procedure should be repeated according to the following clustering mechanism:

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  $t_k > -1.65$ . In other words, choose the core group size  $k^*$  as follows:

$$k^* = \arg \max_k \{t_k\} \quad \text{subject to } \min \{t_k\} > -1.65. \quad (7)$$

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  $k^*$  is detected, run the log-t 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.

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

### The merging algorithms

Due to the fact that the number of identified clubs strongly depends on the core group formation, a key role is played by the critical value  $c^*$ . The choice of this parameter is related to the desired degree of conservativeness, where a higher level of  $c^*$  corresponds to a more conservative selection. In other words, the higher is  $c^*$  the less likely we add wrong members to the convergence clubs. Related to this, for small samples ( $T < 50$ ) Phillips and Sul suggest to set  $c^* = 0$ .

However, as the same authors suggest, a high value of  $c^*$  can lead to more groups than those really existing. For these reasons Phillips and Sul (2009) suggest a *club merging algorithm* to avoid this

over-determination. 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 the convergence hypothesis is rejected, conclude that all previous groups converge, except the last one. Hence, start again the merging algorithm beginning from the group for which the hypothesis of convergence did not hold.

In our package we also provide the implementation in R of an alternative club merging algorithm developed by [von Lyncker and Thoennessen \(2017\)](#). They introduce two innovations in the club merging algorithm by Phillips and Sul. First, they add a further condition to the club clustering algorithm to avoid mistakes in merging procedures in the case of transition across clubs. Second, they propose an algorithm for diverging units. The first algorithm works as follows:

1. Take all the  $P$  groups detected in the basic clustering mechanism 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  $t(m = M) > -1.65$ .

For the second algorithm, [von Lyncker and Thoennessen \(2017\)](#) claim that units identified as divergent by the original 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; 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-statistic values with dimension  $d \times p$ , where each row  $d$  represents a divergent region and each column  $p$  represents a convergence club;
3. Take the highest t-value greater than a critical parameter  $e^*$  and add the respective region to the corresponding club, then start again from 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.

## The ConvergenceClubs package

**ConvergenceClubs** aims to make the clustering procedure described above easy to perform and simply reproducible.

The log-t test is performed by function `estimateMod()`. It takes as main input the vector of cross-sectional variances  $H$  for the units to be tested for convergence, which can be obtained through function `computeH()`:

```
# Compute cross-sectional variances
computeH(X, quantity = "H", id)

# Perform the log-t test
estimateMod(H, time_trim=1/3, HACmethod = c("FQSB", "AQSB"))
```

The former takes a matrix or data.frame object containing time series data and returns either the vector of cross-sectional variances  $H$  or the matrix of transition paths  $h$ , or both, depending on the value of argument `quantity`. These quantities can also be computed on a subset of units by selecting the unit IDs through argument `id`. Function `estimateMod()` takes two additional arguments, `time_trim` and `HACmethod`, described later. These two functions are available for the user who wants to test the convergence hypothesis on a set of units. This is especially useful to assess the opportunity of carrying out the clustering procedure during the initial phase of a study.

Nonetheless, the log-t test over the whole sample is automatically performed before starting the clustering procedure by function `findClubs()`. This is the main function of the package, as it carries out Phillips and Sul's clustering algorithm:



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

where  $X$  is a data frame containing the data, `dataCols` is an integer vector indicating the column indices of the time series data, and `unit_names` is an integer scalar, indicating the index of the column of  $X$  that includes id codes for the units (e.g. the name of the countries/regions). The parameters of the clustering procedure are regulated by the following arguments.

- `refCol`: takes an integer value representing the index of the column to use for ordering data;
- `time_trim`: accepts numeric scalars between 0 and 1, and indicates the portion of time periods to trim when running the log- $t$  regression model. By default, `time_trim=1/3`, which means that the first third of the time series period is discarded, as suggested by [Phillips and Sul \(2007, 2009\)](#);
- `cstar`: takes a 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`: accepts a character 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 log- $t$  regression model with heteroskedasticity and autocorrelation consistent standard errors. The default method is FQSB.

The clustering procedure is performed by iteratively calling two internal functions: `coreG()` and `club()`, which implement steps 2 and 3 of Phillips and Sul clustering algorithm, respectively.

Function `findClubs()` returns an object belonging to the S3 class "convergence.clubs". Objects belonging to this class are lists that include results about clubs and divergent units that have been detected by the clustering procedure. Their structure can be analysed through function `str()`, and their elements can be accessed as commonly done with list elements.

Information about clubs and divergent units can be easily displayed by means of functions `print()` and `summary()`, for which the package provides specific methods for class "convergence.clubs". A `plot()` method is available for class "convergence.clubs", which provides a way to visualise the transition paths of the units included in convergence clubs, and also the average transition paths for each club:

```
plot(x, y = NULL, nrow = NULL, ncol = 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, ...)
```

Plot customisation (i.e. clubs to be displayed or the number of rows and columns of the graphical layout) and options to export it to a file are discussed in more details in the package manual ([Sichera and Pizzuto, 2019](#)). Finally, the merging algorithms described in the previous section are implemented in function `mergeClubs()`:

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

Merging is performed on argument `clubs`, an object of class "convergence.clubs", by means of either the [Phillips and Sul \(2009\)](#) or the [von Lyncker and Thoennessen \(2017\)](#) algorithm, selected through argument `mergeMethod`. Through argument `threshold` it is possible to change the significance level of the log- $t$  test for club merging. Moreover, argument `mergeDivergent` determines whether the test for diverging units according to [von Lyncker and Thoennessen \(2017\)](#) should be performed, while argument `eststar` is used to set the value of the critical parameter  $e^*$ . Function `mergeClubs()` returns an object of class "convergence.clubs" as well, thus information about the new clubs can be accessed and summarised as previously discussed.

A detailed example of all functionalities of the package is presented in the next section.

## Application to the country GDP dataset

In this section we provide an example that replicates the results of Phillips and Sul (2009). The dataset GDP, available in package **ConvergenceClubs**, covers a panel of 152 countries for the period 1970-2003.

First, we filter the data using the Hodrick-Prescott filter methodology by means of function `hpfilter` in package **mFilter** ([Balcilar, 2018](#)). Filtered data are also available in the package through dataset `filteredGDP`.

```

### Load ConvergenceClubs package
library(ConvergenceClubs)

### Load GDP data
data("GDP")

### Filter data
logGDP <- log(GDP[,-1])
filteredGDP <- apply(logGDP, 1,
                     function(x){mFilter::hpfilter(x, freq=400, type="lambda")$trend})
filteredGDP <- data.frame(Countries = GDP[,1], t(filteredGDP), stringsAsFactors=FALSE)
colnames(filteredGDP) <- colnames(GDP)

## Filtered data are available in the package
data(filteredGDP)

```

By using the `estimateMod()` function we perform the log-t test over the whole sample in order to verify whether all units converge.

```

### log-t test over all units
H <- computeH(filteredGDP[,-1], quantity = "H")
round(estimateMod(H, time_trim=1/3, HACmethod = "FQSB"), 3)
#   beta std.err   tvalue   pvalue
# -0.875   0.005 -159.555   0.000

```

The null hypothesis of convergence is rejected at 5% level since the t-value is smaller than  $-1.65$ . Therefore, we proceed with the identification of convergence clubs, which is performed using the `findClubs()` function. As for the arguments, we set:

- `unit_names=1` indicates that Countries' IDs are represented in the first column of the dataset;
- `dataCols=2:35` indicates the columns (years) for which the test should be performed;
- `refCol=35` represents the final period according to which data should be ordered (see step 1 of the clustering algorithm).
- `time_trim=1/3` represents the portion of time periods to trim when running the log-t regression model;
- `cstar= 0` is the threshold value of the sieve criterion  $c^*$  (see step 3 of the clustering algorithm);
- `HACmethod = 'FQSB'` indicates that the Fixed Quadratic Spectral Bandwidth is used for the truncation of the Quadratic Spectral kernel in estimating the log-t regression model.

```

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

```

```

class(clubs)
# [1] "convergence.clubs" "list"

```

As we can see, `clubs` is an object of class "convergence.clubs", that is a common list, whose structure can be displayed through function `str()`, and whose elements can be accessed as usual:

```
str(clubs, give.attr=FALSE)
```

A method for function `summary()` is provided for class "convergence.clubs". It produces a summary table with the key results of the clustering procedure:

```
summary(clubs)
```

```

# Number of convergence clubs: 7
# Number of divergent units: 0
#
#   | # of units | beta   | std.err | tvalue
# ---|---|---|---|---
# club1 | 50      | 0.382  | 0.041   | 9.282
# club2 | 30      | 0.24   | 0.035   | 6.904
# club3 | 21      | 0.11   | 0.032   | 3.402
# club4 | 24      | 0.131  | 0.064   | 2.055

```



# club5	14	0.19	0.111	1.701
# club6	11	1.003	0.166	6.024
# club7	2	-0.47	0.842	-0.559

The summary shows that there are 7 clubs and no divergent units. For each club, the summary also reports how many units are included, the beta coefficient of the log-t test, its standard error, and the value of the t-statistics. This exercise exactly replicates the results obtained by [Phillips and Sul \(2009\)](#). A minor difference concerns the last two clubs (6 and 7). In the original paper, Phillips and Sul showed a divergence group of 13 countries. However, another iteration of the algorithm using these 13 countries suggests the presence of two clubs consisting of 11 and 2 countries, respectively (on this point see also [Schnurbus et al. \(2017\)](#) and [Du \(2018\)](#)).

As shown in the following example, information about the club composition can be obtained using the `print()` function. For brevity, only the output for the first club is shown.

```
## Print results
print(clubs)

# or just
clubs

# =====
# club 1
# -----
# United.States, Norway, Bermuda, United.Arab.Emirates, Qatar, Luxembourg, Singapore,
# Switzerland, Hong.Kong, Denmark, Ireland, Austria, Australia, Canada, Macao,
# Netherlands, Kuwait, Iceland, United.Kingdom, Germany, France, Sweden, Belgium, Japan,
# Brunei, Finland, Italy, Cyprus, Puerto.Rico, Israel, New.Zealand, Taiwan, Spain, Malta,
# Korea..Republic.of, Portugal, Oman, Mauritius, Antigua, St..Kitts...Nevis, Chile,
# Malaysia, Equatorial.Guinea, Dominica, St.Vincent...Grenadines, Botswana, Thailand,
# Cape.Verde, China, Maldives
#
# beta:      0.3816
# std.err:   0.0411
# tvalue     9.2823
# pvalue:    1
#
# [...]
```

Transition path plots can be generated through the function `plot()`. Here we show some examples.

```
# Plot Transition Paths for all units in each club and average Transition Paths
# for all clubs
plot(clubs)

# Plot Transition Paths
plot(clubs, avgTP = FALSE, nrows = 4, ncols = 2, plot_args = list(type='l'))

# Plot only average Transition Paths of each club
plot(clubs, clubs=NULL, avgTP = TRUE, legend=TRUE, plot_args = list(type='o'))
```

The second and third commands produce [figs. 2 and 3](#), respectively. In the first case, we can see how economies approach the steady-state of each club. Conversely, in the second case, the comparison among the average transitional behaviour of each club is shown.

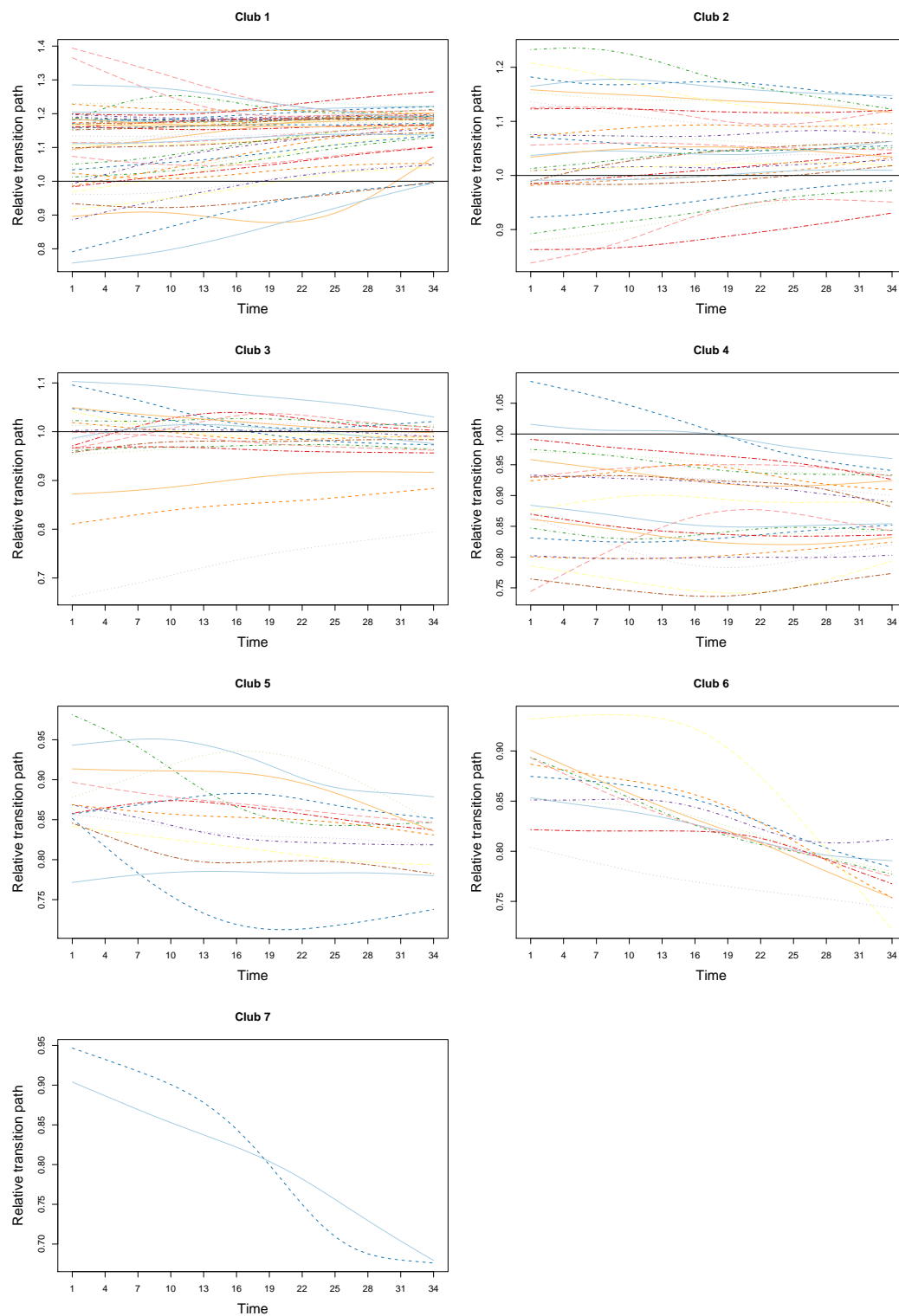


Figure 2: Transition paths within each convergence club.