



Monitoring Convergence in the European Union with the *convergEU* Package for R

Federico M. Stefanini
University of Milan

Nedka D. Nikiforova
University of Florence

Eleonora Peruffo
Eurofound

Martina Bisello
Eurofound

Chiara Litardi
Eurofound

Massimiliano Mascherini
Eurofound

Abstract

Upward convergence, that is an improvement of Members States' economic and social indicators' performance, is a core policy target of the European Union. This concept is embodied in the European Pillar of social rights (EPSR) proclaimed by EU leaders in 2017. In 2018 Eurofound has developed a methodology to measure convergence, which has been implemented in the **convergEU** R package described in this work (rel. 0.5.0). The package extends the original STATA toolbox developed by Eurofound beyond the calculations of the four main measures of convergence as it provides functions to download, filter, impute, smooth indicators. Country and indicator fiches are automatically prepared and compiled in HTML format. Graphical output includes qualitative patterns of change along time to emphasize the key behaviour of an indicator with respect to the average of an aggregation of Member States. Besides EU Member States, the analysis of other collections of regions is supported for general indicators if context information is provided. A user-friendly, no coding required, shiny-based web application provides policy makers with a tool to produce convergence reports on selected indicators and countries.

Keywords: indicators, convergence, Eurofound, R.

1. Introduction

Upward convergence in the economic and social dimensions has always been an European Union political promise: Member States, and their citizens, joined the European Union (EU) and adopted the Euro with the legitimate expectation of improving working and living conditions. Diverging performances among Member States and increasing inequalities within Member States warrant serious concerns as they contradict the expectations of Member States and citizens. Economic divergence undermines the promise of shared economic prosperity while social divergence between Member States and increasing inequalities within Member States undermine cohesion and the ultimate vital goal of the European integration project of improving living and working conditions.

While the economic and social dimension of the EU have been always seen as two separate

entities, as a result of the economic crisis there is now a broad understanding that economic and social convergence should go hand in hand. In order to support and promote this convergence process the European Pillar of Social Rights (EPSR) was proclaimed in 2017 by EU leaders at the Social Summit for Fair Jobs and Growth in Gothenburg, Sweden. More recently, in 2021, the EPSR Action Plan tries to turn the Principle into concrete actions.

The Pillar aims to build a fairer Europe with a strong social dimension by means of 20 principles to support inclusive and well-functioning labour markets and welfare systems and it is structured around three people-centred categories:

1. Equal opportunities and access to the labour market: covering education, gender equality and equal opportunities.
2. Fair working conditions: addressing labour force structure, labour market dynamics and income.
3. Social protection and inclusion, covering fair outcomes through public support and social protection, mainly relating to the provision of services and social safety nets.

The endorsement of the Pillar was a major initiative to strengthen the social dimension of the EU, to give to it the same dignity as the economic dimension and to anchor them both at the heart of the future of the Union.

The concept of upward convergence of Member States performances, namely an increase in performances accompanied with a decrease in Member States differences, is one of the backbones of the European Pillar of Social Rights. In this regard, the Pillar should be seen as an important tool to ensure that social objectives counter-balance objectives of an essentially macro-economic nature in a new social and economic governance tools of the EU.

As part of its work programme, Eurofound has done substantial work in the field of convergence developing a methodology and a monitoring tool to measure it. In particular, Eurofound (Mascherini, Bisello, Dubois, and Eiffe 2018) has provided a set of statistical tools in order to monitor upward convergence for a given indicator through various convergence measures, each with different policy meaning. This Eurofound toolbox has been firstly implemented as set of Stata script and is available at <https://www.eurofound.europa.eu/data/convergence-hub/convergence-methodology>.

In this work, the Eurofound statistical toolbox for monitoring upward convergence has been implemented as an R package (Stefanini, Nikiforova, Litardi, Bisello, Peruffo, and Mascherini 2020a). Besides including the four main measures of convergence, the **convergEU** R package also contains supporting functions to filter, impute, smooth indicators and to prepare country and indicator fiches in HTML format. Qualitative patterns of change along time are defined for indicators with respect to the EU average to highlight the underlying key features that are also summarized by mnemonic and informative labels. To the best of our knowledge, no other software package provides such a detailed analysis of convergence.

This paper is organized as follows. In Section 2, we describe the main measures of convergence. In Section 2.6, qualitative patterns are also defined by studying changes of indicator values over time. Core functions of the R package are illustrated in Section 3 through several examples; actual calculations of patterns are illustrated in Section 3.1.6. Smoothing of raw data and compilation of fiches are respectively presented in Sections 3.2 and 3.3. Discussion and final remarks end the manuscript (Section 4). Appendix I deals with some instructions to download and prepare data, including the deterministic imputation of missing values. Other auxiliary functions developed in the package are presented in Appendix II. Notation is introduced in Section 2.1 and later extended as needed.

2. Quantifying convergence

Eurofound defines upward convergence as the improvement of Member States performances towards a policy target, combined with a reduction of disparities among Member States. Upward convergence, or moving closer together in an upward trajectory, is therefore the union of two concepts: an improvement in performance towards a desirable target and convergence itself, i.e. the reduction of disparities in performances. The concept of performance improvement is ultimately related to a policy target: in other words, the desirable change of the indicator towards, for example, better living and working conditions. In the case of employment rates, upward convergence is observed when the indicator increases and disparities among countries decline. Such a scenario would represent a maximisation of the policy target. For unemployment rates, upward convergence is observed when the indicator falls and disparities among countries decrease - minimisation of the policy target.

Measuring upward convergence inherently involves the measurement of these two concepts: improvement and convergence. Improvements are usually measured through changes in unweighted averages of Member States performances. The use of unweighted averages assigns to each Member States the same level of representativeness and importance in the computation of the overall trend. This would have not been the case, for example, if population based weights were used as larger Member States would have been over represented in the computation of levels. However, as evidenced by Eurofound in 2018 ([Mascherini *et al.* 2018](#)), average levels may hide patterns in individual countries. In this regard, the improvement of an indicator at the EU level does not necessarily imply that levels are improving for all the Member States. In order to take account of these aspects, Eurofound distinguishes between two types of upward convergence. Weak upward convergence occurs only if the EU average records an improvement in its performance, while reducing its disparities. In a weak convergence situation, the EU average is improving, but not for every single Member State, and some Member States could risk to be left behind. Strict upward convergence occurs if all countries (and regions) improve their performance (towards the policy target) while reducing the disparities between them. In this case no country is left behind as they all improve. In addition to these two cases, and following the same logic, Eurofound ([Mascherini *et al.* 2018](#)) defines three other possible situations: upward divergence, downward divergence and downward convergence. Convergence, namely the reduction of disparities among Member States, is then usually measured through the following statistical measures: Beta, Sigma, Delta and Gamma convergence.

Convergence can be monitored in at least four ways using Beta, Sigma, Delta and Gamma convergence measures respectively. Each measure has its own characteristics and limitations, allowing to investigate similar yet different aspects of the convergence process given an indicator as benchmark. Each aspect has its own political meaning when applied to monitoring economic and social indicators. More specifically, beta-convergence aims to detect if least performing countries (laggards) are improving (or not) their performance to the level of best performing countries (leaders). This measure is used, for example, to monitor the catch up of poorer countries, i.e. countries with a low GDP per capita, towards wealthier countries ([Sala-i-Martin 1996](#)). Sigma-convergence, attempts to investigate decreases in the variation of a given performance outcome (i.e. an indicator in our setting) through two statistical measures of variability, namely the standard deviation and the coefficient of variation ([Friedman 1992](#); [Quah 1993](#)). For instance, [Kenworthy \(1999\)](#) defined it as the best quantitative measure to assess homogeneity across indicators. Sigma-convergence represents a straightforward measure of convergence because it directly reveals how countries become similar to each other with respect to a given indicator. From a policy perspective, sigma-convergence identifies convergence towards the mean of the economic and social indicator considered that could be driven by an improvement of the low performers, or of a deterioration of the levels of the best performers. Therefore, additional analysis should be implemented in order to understand the nature of sigma-convergence. It must also be noted that beta and sigma-convergence

are strictly related to each other. That is, under specific assumptions, beta-convergence may imply sigma-convergence (Sala-i-Martin 1996). However, if some country-specific economic events occur, this may imply some kind of dispersion. Thus, it follows that beta-convergence is necessary, but not sufficient condition for sigma-convergence (Barro and Sala-i-Martin 1992; Friedman 1992; Quah 1993; Mascherini *et al.* 2018).

Gamma-convergence allows to measure changes in country rankings in relation to a given indicator through specific statistical measures of association, as for instance the Kendall index of rank concordance (Boyle and McCarthy 1999). It has been developed as a variant of beta-convergence in order to capture the cross-countries' dynamics and movements (Boyle and McCarthy 1999). However, as stated by Heichel, Pape, and Sommerer (2005), country rankings can also change without a decrease of cross-country variation or a movement towards the best performer. To overcome this issue, Heichel *et al.* (2005) developed the concept of delta-convergence, which measures the countries' distance from the best performer, or from a group of best performers or even from a policy target. Even though delta-convergence could be affected by the presence of potential outliers, it provides a useful measure if convergence occurs. While not often used in scientific literature, reducing the distance from a policy target is extremely important in the European policy debate. Therefore, this type of convergence is what emerge from the European debate regarding economic and social convergence of Member States and the need of upward convergence (European Commission 2018).

2.1. Notation

In this section, the general notation is defined to indicate aggregations of Member States where an indicator Y has been quantified, in particular:

- $Y_{EU,i,t}$ is the random variable representing the EU average, indicator i at time $t \in \Omega_{T,EU,i}$;
- $Y_{m,i,t}$ is the random variable for indicator i of Member State m at time $t \in \Omega_{T,m,i}$; in general, other collections of regions than EU Member States may be considered;
- observed values are $y_{EU,i,t}$ and $y_{m,i,t}$, with $m \in \Omega_M$ where M is the selected aggregation, e.g. $M \in \{EU12, EU15, \dots\}$
(see https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:EU_enlargements).

Given a collection of values (x_1, x_2, \dots) the maximum value is indicated as $x^{(M)} = \max(x_1, x_2, \dots)$. If the collection of time points is the same for all Member States then the notation for the sample space is simplified into Ω_T , that is, Ω_T indicates that $\Omega_{T,EU,i}, \Omega_{T,m,i} \forall m \in \Omega_M$ given i .

It is also useful to indicate as n_M the number of Member States (or other collections of regions) considered in a given aggregation Ω_M , for example, $n_M = 12$ if $M = EU12$. Then, the average of indicator values $y_{m,i,t}$ over Ω_M at time t is:

$$\bar{y}_{M,i,t} = n_M^{-1} \sum_{m \in \Omega_M} y_{m,i,t} \quad (1)$$

A given indicator is of type "lowBest" if the best possible situation is represented by the lowest value; for example the minimisation of the Poverty rate is a positive result. Otherwise, a given indicator is of type "highBest", if the best situation is reached when the indicator takes the highest value, for example for the Employment rate. Nevertheless, we will not introduce specific notation to distinguish the two kinds of indicators.

2.2. Beta-convergence

Beta-convergence derives from economic growth theory (Solow 1956) and it refers to a catching-up process in which poorer countries grow faster than the rich ones. In addition to the purpose of testing the classical hypothesis of income convergence, the analysis of beta-convergence can be applied to other variables of interest to assess if poorer countries or regions catch up with better-performing ones; for instance, Signorelli (2005) analyses convergence for employment rates. The computation of beta-convergence is performed according to the following linear regression model:

$$\ln(y_{m,i,t+\tau}) - \ln(y_{m,i,t}) = \beta_0 + \beta_1 \ln(y_{m,i,t}) + \epsilon_{m,i,t} \quad (2)$$

where $\tau \in \{1, 2, \dots\}$ is the length of the time window, which is typically equal to 1 or more years.

Even if all subsequent times $\{t, t+1, \dots, t+\tau-1, t+\tau\}$ could be included into the regression model, only the reference time t and $t+\tau$ with $3 \leq \tau \leq 5$ are typically analyzed in practice. The current implementation of the beta-convergence function in the **convergEU** package can calculate convergence for several target years, but it always maintains the same reference time t . The left hand side of equation 2 is divided by the amount of time elapsed as a default option:

$$\tau^{-1} \{\ln(y_{m,i,t+\tau}) - \ln(y_{m,i,t})\} = \beta_0 + \beta_1 \ln(y_{m,i,t}) + \epsilon_{m,i,t} \quad (3)$$

The computation of beta-convergence, formulas (2) and (3), allows to assess the relationship between the growth of a given indicator and its initial value. More precisely, if the β_1 estimated coefficient is statistically significant and negative (i.e. $\hat{\beta}_1 < 0$), then an evidence of beta-convergence is present; otherwise, if $\hat{\beta}_1 > 0$, then there is an evidence of divergence.

2.3. Delta-convergence

The concept of delta-convergence has been formulated by Heichel *et al.* (2005). Given a set C of Member States, for a given indicator delta-convergence allows to evaluate how the Member States belonging to C are becoming similar to the best-performing one. Let $y_{i,t}^{(M)}$ be the maximum value over Member States in the reference set Ω_M , that is:

$$y_{i,t}^{(M)} = \max(\{y_{m,i,t} : m \in \Omega_M\}). \quad (4)$$

The distance of a Member State m from the top performer for an indicator i at time t is:

$$y_{i,t}^{(M)} - y_{m,i,t} \quad (5)$$

Thus, the overall distance at time t , called delta, is the sum of distances over the reference set Ω_M of Member States:

$$\delta_{i,t} = \sum_{m \in \Omega_M} (y_{i,t}^{(M)} - y_{m,i,t}) \quad (6)$$

for the considered indicator i . If the overall distances over a period of time decrease, then there is an evidence of delta-convergence for this considered period of time; otherwise, the Member States belonging to the set Ω_M are diverging.

2.4. Gamma-convergence

We now introduce gamma convergence (Boyle and McCarthy 1999), which is an index based on ranks. Let $t \in \Omega_T = \{0, 1, 2, \dots\}$ be the considered time points and $\{\tilde{y}_{m,i,t} : m \in \Omega_M\}$ be the ranks for indicator i over Member States at each time point t .

The sum of ranks within Member State m is:

$$\tilde{y}_{m,i}^{(c)} = \sum_{t \in \Omega_T} \tilde{y}_{m,i,t}. \quad (7)$$

Thus, the variance of the sum of ranks over the given interval, defined as $Var [\{\tilde{y}_{m,i}^{(c)} : m \in \Omega_M\}]$, may be compared to $Var [\{\tilde{y}_{m,i,0} : m \in \Omega_M\}]$, the variance of ranks in the reference time $t = 0$. The Kendall Index of rank concordance of Member States with respect to the aggregation Ω_M for the indicator i at $t^{(M)} = \max(\Omega_T)$ time is defined as:

$$KI(C, i, t^{(M)}) = \frac{Var [\{\tilde{y}_{m,i}^{(c)} : m \in \Omega_M\}]}{(t^{(M)} + 1)^2 Var [\{\tilde{y}_{m,i,0} : m \in \Omega_M\}]} \quad (8)$$

This index assumes values between 0 and 1. A low index value indicates that a high number of changes in the position of Member States have been observed in the time points in Ω_T , while a high index value indicates that few changes have been observed.

2.5. Sigma-convergence

The concept of sigma-convergence refers to a reduction in disparities over time as summarized by a social or economic indicator (Barro and Sala-i-Martin 1992; Friedman 1992; Quah 1993). The decrease in a function of variability over time indicates that a variable is becoming increasingly homogeneous cross-nationally.

The reduction in heterogeneity (possibly disparity) is typically assessed using two statistical summaries, the first is the standard deviation:

$$s_{M,i,t} = \sqrt{(n_M^{-1} \sum_{m \in \Omega_M} (y_{m,i,t} - \bar{y}_{M,i,t})^2)} \quad (9)$$

and the second is the coefficient of variation:

$$CV(M, i, t) = 100 \cdot \frac{s_{M,i,t}}{\bar{y}_{M,i,t}}. \quad (10)$$

Therefore, for each year $t \in \Omega_T$, the above summaries are calculated to quantify if a reduction in heterogeneity took place. If the standard deviation and the coefficient of variation decrease over time, then there is an evidence of sigma-convergence. Otherwise, an increase in the standard deviation and in the coefficient of variation indicate that Member States belonging to the set Ω_M are diverging. Given that the coefficient of variation (equation 10) rescales the standard deviation in formula (9) according to the mean, the same amount of variability becomes more and more negligible as the mean increases. Thus, in situations in which one does not need to perform a comparison among indicators, it is more appropriate to use the standard deviation to assess sigma-convergence.

2.6. Patterns in Member State dynamics

Measures like the Beta convergence summarize what happens between two subsequent time points for all Member States belonging to a selected aggregate, for example EU12. Besides declaring convergence or divergence (weak/not weak, upward/ downward), the attention may be directed towards the qualitative appreciation of patterns of change occurred to a Member State with respect to the unweighted average of the same indicator over the selected countries.

The Eurofound report (Mascherini *et al.* 2018) calculate the gradient as follows:

- Member State m , $\beta_m := \nabla f_m(y) = \frac{y_{m,i,t+\delta} - y_{m,i,t}}{\delta}$;
- EU average (EU): $\beta_{EU} := \nabla \mu_{EU}(y) = \frac{y_{EU,i,t+\delta} - y_{EU,i,t}}{\delta}$

while the value of delta is: $\Delta_{t+\delta,t} \sigma^2 := (y_{m,i,t+\delta} - y_{EU,i,t+\delta})^2 - (y_{m,i,t} - y_{EU,i,t})^2$, with $\delta = 1$. The value of the above three quantities and the relationship between betas determine the qualitative pattern in subsequent years (Mascherini *et al.* 2018). For example, if a "highBest"

indicator is considered, then $(\beta_{EU} > 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$ induces an equivalence class of segments numerically labelled as 1 where EU averages increased and the considered Member State increased more than EU, while both its values remained below the value taken by the EU mean in the corresponding year. The most important patterns are defined in similar ways and their numerical labels are shown in Table 1.

Table 1: Numerical labels (column T) for the most important patterns obtained by calculating gradients and considering values of a "highBest" indicator

Defining features	T
$(\beta_{EU} > 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	1
$(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	2
$(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	3
$(\beta_{EU} > 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	4
$(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	5
$(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	6
$(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	7
$(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	8
$(\beta_{EU} < 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0)$	9
$(\beta_{EU} < 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	10
$(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	11
$(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0)$	12
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} > 0) \wedge (y_{m,i,t} > y_{EU,i,t})$	13
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} = 0) \wedge (y_{m,i,t} > y_{EU,i,t})$	14
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} < 0) \wedge (y_{m,i,t} > y_{EU,i,t})$	15
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} < 0) \wedge (y_{m,i,t} < y_{EU,i,t})$	16
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} = 0) \wedge (y_{m,i,t} < y_{EU,i,t})$	17
$(\beta_{EU} = \beta_m) \wedge (\beta_{EU} > 0) \wedge (y_{m,i,t} < y_{EU,i,t})$	18
$(y_{m,i,t} > y_{EU,i,t}) \wedge (y_{m,i,t+1} < y_{EU,i,t+1})$	19
$(y_{m,i,t} < y_{EU,i,t}) \wedge (y_{m,i,t+1} > y_{EU,i,t+1})$	20
Other (Inspection required)	21

The association of labels in case of an indicator of type "lowBest" is performed by considering that the highest possible value in the sample space refers to the worst socio-economic feature of a Member State (with respect to that indicator). Instead of creating new labels to tag patterns of this class of indicators, we transform the original indicator after noting that the absolute positioning of values is not relevant while judging for the presence of a given pattern. Thus, the indicators of type "lowBest" are transformed, and the distance from the maximum value for each original observation is calculated. If the original index decreases then the transformed value increases, and the pattern recognition scheme applies in the same way as for indicators of type "highBest".

A self-explanatory name may be assigned to each pattern of a given type of indicator to help communicating key features without resorting to plots. In Table 3, the 20 most important patterns are shown with the assigned names specified given the type of indicator, i.e. "lowBest" or "highBest".

Table 2: 80 measured values of access time

70.0	68.6	67.9	66.3	71.0	64.2	69.6	71.0
...
71.7	70.5+	72.5-	68.2	67.6	68.6	70.5-	65.8

It is worth emphasizing that, despite the usefulness, even self-describing labels may become deceiving in the special case when empirical values leading to a given pattern are "geometrically" very close to those labelled into a different way. If one of these border-like numerical

Table 3: Summary of the 20 most important patterns according to the type of indicator. The following labels are adopted: down: a decrease along time; Down: a decrease along time faster than "down"; up: an increase along time; Up: an increase along time faster than "up"; NC: Not Considered.

Pattern Name	Type of indicator: "highBest"				Type of indicator: "lowBest"			
	Type	EU better	<i>EU</i>	<i>m</i>	Type	EU better	<i>EU</i>	<i>m</i>
Catching up	1	Yes	up	Up	10	Yes	down	Down
Flattening	2	No	Up	up	12	No	Down	down
Inversion	3	No	up	down	11	No	down	up
Outperforming	4	No	up	Up	9	No	down	Down
Slower pace	5	Yes	Up	up	7	Yes	Down	down
Diving	6	Yes	up	down	8	Yes	down	up
Defending better	7	No	Down	down	5	No	Up	up
Escaping	8	No	down	up	6	No	up	down
Falling away	9	Yes	down	Down	4	Yes	up	Up
Underperforming	10	No	down	Down	1	No	up	Up
Recovering	11	Yes	down	up	3	Yes	up	down
Reacting better	12	Yes	Down	down	2	Yes	Up	up
Parallel better over	13	No	up	up	16	No	down	down
Parallel equal over	14	No	flat	flat	17	No	flat	flat
Parallel worse over	15	No	down	down	18	No	up	up
Parallel worse under	16	Yes	down	down	13	Yes	down	down
Parallel equal under	17	Yes	flat	flat	14	Yes	flat	flat
Parallel better under	18	Yes	up	up	15	Yes	up	up
Crossing	19	NC	up	down	20	NC	down	up
Crossing reversed	20	NC	down	up	19	NC	up	down

configurations happens then quantitative values should be also inspected before defining any policy. That is, each pattern has its own mathematical definition, but the discretization involved in the labelling procedure may hide some numerical features at boundaries.

3. Working with the `convergEU` R package

The `convergEU` R package is a collection of functions and of data objects suited for the analysis of economic and social convergence. It has been mainly designed to perform the analysis of Member States within the EU, but other collections of regions may be studied by providing custom labels. In this Section we illustrate through several examples how to calculate convergence through the core functions implemented in the R package.

A detailed introduction dealing with data downloading, formatting and imputation is contained in Appendix I. The final dataset stored within an object of class `tibble` has to be tidy, i.e. a table year by Member State without missing values. Note that, in this package, many functions return a list of three components, `res`, `msg`, `err`, respectively to provide the result (if available), messages for the user and error messages, if the result cannot be calculated.

3.1. Analysis of convergence

In Section 2, we have introduced the main measures of convergence implemented in the `convergEU` R package. In this Subsection we illustrate how to calculate each measure of convergence through the functions implemented in the package. To this end, we consider the dataset `EmpImp`, which is the `lfsa_ergaed` indicator downloaded from Eurostat and processed as described in Appendix I. Here, the "EU12" aggregation of Member States is selected:

```
EU12estr<-convergEU_glb()$EU12$memberStates$codeMS
EmpImpF<- dplyr::select(EmpImp,time,all_of(EU12estr))
head(EmpImpF)
```

```
## # A tibble: 6 x 13
```



```
##   time  BE   DK   FR   DE   EL   IE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1992 72.1 84.3 77.7 78.7 67.8 58.4
## 2 1993 72.4 82.2 77.6 77.6 68.1 59.2
## 3 1994 72.2 81.6 76.9 77.4 69.1 60.8
## 4 1995 72.9 82.6 77.8 77.7 69.3 62.8
## 5 1996 72.9 83.1 78   77.5 70.1 64.5
## 6 1997 73.8 83.4 77.5 77.2 70.3 65.5
## # ... with 6 more variables: IT <dbl>,
## #   LU <dbl>, NL <dbl>, PT <dbl>, ES <dbl>,
## #   UK <dbl>

check_data(EmpImpF)$res

## [1] TRUE
```

The `check_data` function return `TRUE` thus it confirms that the dataset is ready for the analysis of convergence.

Beta-convergence

The `beta_conv` function is invoked to calculate beta-convergence (Subsection 2.2). It is worthwhile to remember that the dataset should be in the format "years by countries", and that no other variable besides time and countries indicator must be present. Last, the time variable must be sorted in ascending order.

For the `empImpF` dataset related to the Employment rate indicator, we first obtain the beta-convergence from 1992 to 1999:

```
EmpBC <- beta_conv(EmpImpF,
  time_0 = 1992,
  time_t = 1999,
  all_within = FALSE,
  timeName = "time")
```

Note that the option `all_within = FALSE` is the default in order to ensure that just the first and the last years are considered, e.g. 1992 and 1999 in this example. In the implementation of the function `beta_conv`, the same reference time is maintained across different years and the division on the left hand side by the amount of time elapsed may be skipped if the argument `useTau = FALSE` is specified:

```
EmpBC1 <- beta_conv(EmpImpF,
  time_0 = 1992,
  time_t = 1999,
  all_within = FALSE,
  timeName = "time",
  useTau = FALSE)
```

The output of the `beta_conv` function consists of the usual three list components: `res` that contains the results, `msg` that possibly carries messages for the user and `err` which is a string containing an error message, if an error occurs:

```
names(EmpBC1)

## [1] "res" "msg" "err"
```

The `res` component is structured as a list with the following components:

- data entering into the regression model of beta-convergence:

```
EmpBC1$res$workTB
```

```
## # A tibble: 12 x 3
##   deltaIndic indic countries
##   <dbl> <dbl> <chr>
## 1 0.0487 4.28 BE
## 2 0.0130 4.43 DK
## 3 0.00385 4.35 FR
## 4 0.00380 4.37 DE
## 5 0.0433 4.22 EL
## 6 0.195 4.07 IE
## 7 0.00291 4.23 IT
## 8 0.0470 4.29 LU
## 9 0.0963 4.28 NL
## 10 0.0251 4.36 PT
## 11 0.0850 4.11 ES
## 12 0.0293 4.35 UK
```

- a summary of the fitted model:

```
EmpBC1$res$summary

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>      <dbl>    <dbl>    <dbl> <dbl>
## 1 (Interce~ 1.73      0.438     3.94 0.00276
## 2 indic    -0.392    0.102    -3.83 0.00331
```

- the β_1 estimated coefficient:

```
EmpBC1$res$beta1

## [1] -0.3922787
```

- the Adjusted R-Squared:

```
EmpBC1$res$adj.r.squared

## [1] 0.5542097
```

as well as several other components related to Ordinary Least Squares estimates: point estimates, residuals, fitted values, the residual degrees of freedom. A standard two-tails statistical test is also reported (p-value and adjusted R-squared).

In our example, the estimated coefficient β_1 is highly significant (p-value=0.0033) and negative ($\hat{\beta}_1 = -0.392$). This indicates that beta-convergence is achieved in the time window 1992-1999 for the set "EU12" of Member States. Now, the beta-convergence is calculated in the time window 2008-2013, that is during the financial crisis which started in 2008:

```
EmpBC2 <- beta_conv(EmpImpF,
                    time_0 = 2008,
                    time_t = 2013,
                    all_within = FALSE,
                    timeName = "time")

EmpBC2$res$summary

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>      <dbl>    <dbl>    <dbl> <dbl>
## 1 (Interce~ -0.532    0.317     -1.68 0.125
## 2 indic     0.119    0.0723     1.65 0.131
```

The point estimate of β_1 is positive ($\hat{\beta}_1 = 0.119$), and this result indicates divergence in this time period for Member States in the "EU12" aggregation, even though the p-value is not significant. If the beta-convergence is calculated in recent years after the crisis of 2008, say from 2014 to 2018, the results are:

```
EmpBC3 <- beta_conv(EmpImpF,
  time_0 = 2014,
  time_t = 2018,
  all_within = FALSE,
  timeName = "time")

EmpBC3$res$summary

## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>      <dbl>    <dbl>    <dbl> <dbl>
## 1 (Interce~  0.328    0.0853     3.85 0.00322
## 2 indic      -0.0732   0.0196    -3.73 0.00392
```

The estimated β_1 coefficient is highly significant and negative, thus it indicates convergence of Member States for "EU12".

A plot of the transformed data and of the fitted may be also useful. To this end, the function `beta_conv_graph` is specifically implemented for obtaining this graphical representation as follows (Figure 1):

```
mybetaplot<-beta_conv_graph(betaRes=EmpBC3,
  indiName = 'Mean Life Satisfaction',
  time_0 = 2014, time_t = 2018)
figure1<-mybetaplot+theme_classic()+geom_text(ggplot2::aes(label=EmpBC3$res$workTB$countries),
  hjust=0, vjust=0)+
  geom_abline(intercept = as.numeric(EmpBC3$res$summary[1,2]),
  slope = as.numeric(EmpBC3$res$summary[2,2]))
#figure1
```

where in the argument `betaRes`, the user should specify the output of the `beta_conv` function; arguments `time_0` and `time_t` relate to the starting and ending time respectively, while the indicator name should be specified as a string in the argument `indiName`.

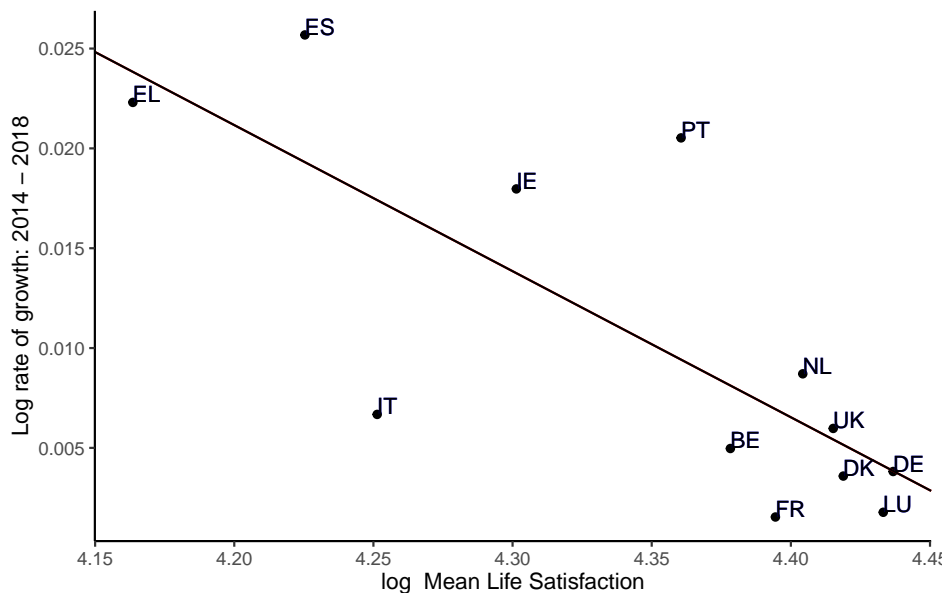


Figure 1: Regression plot for beta convergence, period: 2014-2018

Delta-convergence

Delta-convergence is calculated by invoking the `delta_conv` function, for example with the `EmpImpF` dataset:

```
deltac<-delta_conv(EmpImpF,"time")
deltac$res
```

```
## # A tibble: 27 x 2
##   time delta
##   <dbl> <dbl>
## 1 1992 142.
## 2 1993 122.
## 3 1994 114.
## 4 1995 120.
## 5 1996 119.
## 6 1997 116.
## 7 1998 114.
## 8 1999 114.
## 9 2000 97.7
## 10 2001 91.6
## # ... with 17 more rows
```

where the output is the usual list of three components, `res`, `msg` and `err`. The values of delta-convergence for the example above are plotted using the following R code (Figure 2):

```
timepp<-deltac$res$time
dd<-deltac$res$delta
plot(timepp,dd,xaxt="n", xlab = '',type = "l",ylab='')
axis(side = 1, at = timepp,labels = T,las=2, cex.axis=0.8)
```

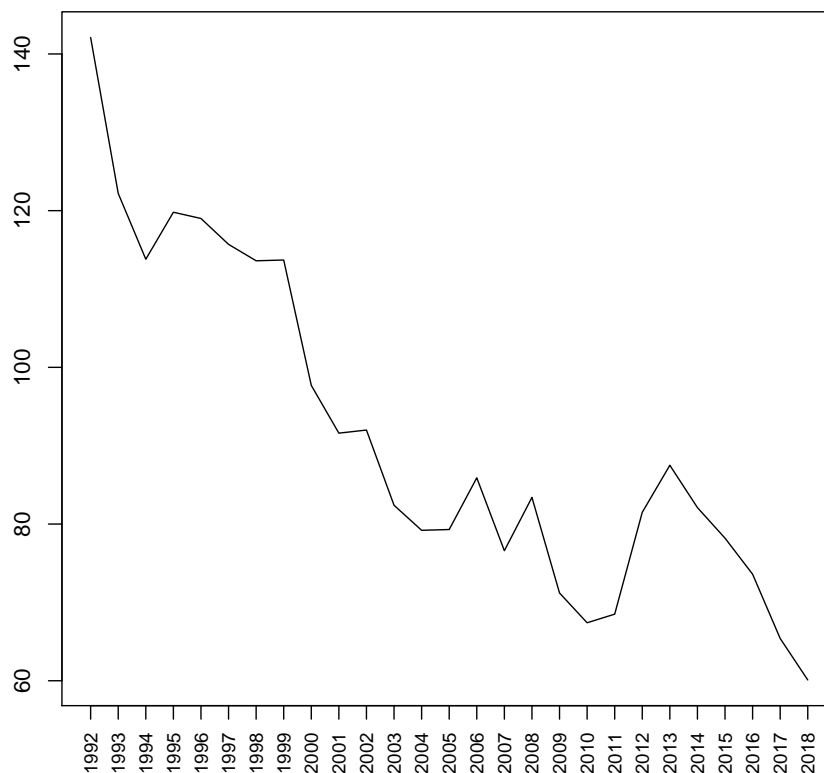


Figure 2: Plot for delta convergence

For example, if we consider the period after the crisis of 2008, there is an evidence of delta-convergence because the overall distances in the period 2014-2018 decrease; divergence is present in the period 2010-2013, that is during the crisis of 2008 because the overall distances in the period 2010-2013 increase.

Deviations of Member States from the EU average may be calculated to deepen the analysis with delta-convergence: the `demea_change` function is implemented in the **convergEU** package

(`help(demea_change)`). It calculates the differences of the indicator values and the EU average at subsequent times within each Member State, then two summary statistics are calculated:

- the total of the positive values

$$y_{m,i}^{(S+)} = \sum_{t \in \Omega_T} (y_{m,i,t} - y_{EU,i,t})^+ \quad (11)$$

where $(y_{m,i,t} - y_{EU,i,t})^+ = (y_{m,i,t} - y_{EU,i,t})$ if the difference is greater than zero, otherwise it is null;

- the total of the negative values

$$y_{m,i}^{(S-)} = \sum_{t \in \Omega_T} (y_{m,i,t} - y_{EU,i,t})^- \quad (12)$$

where $(y_{m,i,t} - y_{EU,i,t})^- = (y_{m,i,t} - y_{EU,i,t})$ if the difference is smaller than zero, otherwise it is null.

To illustrate calculations of $y_{m,i}^{(S+)}$ and $y_{m,i}^{(S-)}$, the previously considered dataset called `EmpImpF` is processed with the selection of Member States Italy, France, Germany and Spain, and a time interval 2003-2016:

```
res1<-demea_change(EmpImpF,
  timeName="time",
  time_0 = 2003,
  time_t = 2016,
  sele_countries= c('IT','FR','DE','ES'),
  doplot=TRUE)
```

where the argument `doplot` is set to `TRUE` in order to also obtain the plot of the calculated differences. It is worth noting that if the argument `sele_countries` is equal to `NA`, then all countries in the dataset are considered. The component `res` of the output contains the following results:

- a list with the differences at each time for each Member State:

```
res1$res$resDiffe

## # A tibble: 14 x 13
##   time    BE    DK    FR    DE    EL    IE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2003 -2.03  6.87  3.37  1.17 -4.23 -3.13
## 2 2004 -1.2   6.60  2.30 -0.2  -4.1  -2.9
## 3 2005 -0.792  6.61  2.41 -0.392 -4.49 -2.09
## 4 2006 -1.44  7.16  2.06  0.158 -4.04 -2.54
## 5 2007 -0.717  6.38  2.38  0.983 -4.32 -2.72
## 6 2008 -0.15  6.95  3.25  1.35 -4.05 -3.85
## 7 2009  0.733  5.93  3.83  2.63 -3.27 -6.77
## 8 2010  1.72  5.32  4.02  3.82 -4.68 -8.28
## 9 2011  1.61  5.51  4.21  5.71 -7.79 -8.29
## 10 2012  2.39  5.79  4.49  6.79 -11.8 -7.61
## 11 2013  2.69  5.99  4.59  7.29 -13.7 -5.31
## 12 2014  2.04  5.34  3.34  6.84 -13.4 -3.86
## 13 2015  1.12  5.02  2.42  6.52 -12.0 -2.88
## 14 2016  1.03  4.53  1.83  6.13 -11.4 -2.77
## # ... with 6 more variables: IT <dbl>,
## # LU <dbl>, NL <dbl>, PT <dbl>, ES <dbl>,
## # UK <dbl>
```

- a list with the differences in absolute value at each time for each Member State:

```
res1$res$diffe_abs_diff

## # A tibble: 13 x 13
##   time      BE      DK      FR      DE      EL
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2004 -0.833 -0.267 -1.07 -0.967 -0.133
## 2 2005 -0.408  0.00833  0.108  0.192  0.392
## 3 2006  0.65   0.55  -0.35 -0.233 -0.45
## 4 2007 -0.725 -0.775  0.325  0.825  0.275
## 5 2008 -0.567  0.567  0.867  0.367 -0.267
## 6 2009  0.583 -1.02   0.583  1.28  -0.783
## 7 2010  0.983 -0.617  0.183  1.18   1.42
## 8 2011 -0.108  0.192  0.192  1.89   3.11
## 9 2012  0.783  0.283  0.283  1.08   4.02
## 10 2013  0.300  0.2    0.1    0.5    1.90
## 11 2014 -0.650 -0.650 -1.25  -0.45  -0.35
## 12 2015 -0.925 -0.325 -0.925 -0.325 -1.38
## 13 2016 -0.0833 -0.483 -0.583 -0.383 -0.617
## # ... with 7 more variables: IE <dbl>,
## #   IT <dbl>, LU <dbl>, NL <dbl>, PT <dbl>,
## #   ES <dbl>, UK <dbl>
```

- a tibble containing negative and positive differences for the selected Member States:

```
res1$res$stats

## # A tibble: 4 x 4
##   MS      negaSum posiSum posi
##   <chr> <dbl> <dbl> <int>
## 1 FR     -4.18  2.64   3
## 2 DE     -2.36  7.33   4
## 3 IT     -1.76  4.09   7
## 4 ES     -6.66  6.49  11
```

- the minimum and the maximum of the calculated differences:

```
res1$res$miniX

## [1] -6.658333

res1$res$maxiX

## [1] 11.10833
```

A graphical representation of the calculated differences is obtained by invoking the `plot` function (Figure 3):

```
fig3<-plot(res1$res$res_graph)
figure3<-fig3+labs(x="",y="")
```

Gamma-convergence

Gamma-convergence is calculated by the `gamma_conv` function, for example with the `EmpImpF` dataset as argument:

```
gammac1<-gamma_conv(EmpImpF,ref=2008,
                    last=2013, timeName="time")
```

where 2008 is the reference time, while 2013 is the last time to be considered. The `res` component of the output contains the value of gamma-convergence:

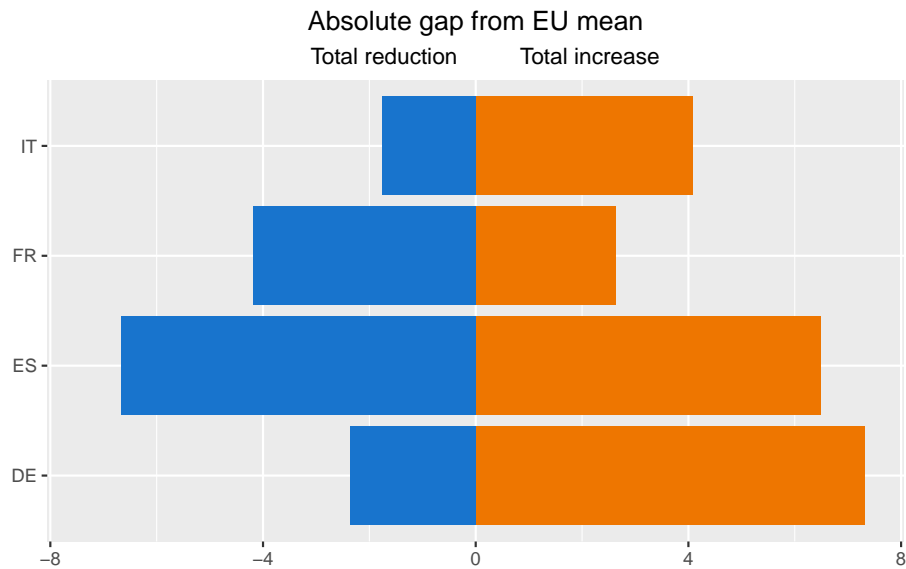


Figure 3: Plot of $y_{m,i}^{(S+)}$ and $y_{m,i}^{(S-)}$ summary statistics

```
gammac1$res
## [1] 0.7712996
```

The high value of the result indicates that few changes have been observed in the Employment rate indicator over the time period 2008-2013 for the aggregation "EU12".

Further output is obtained by printing ranks of Member States at each time point by specifying the option `printRanks=T` as follows (first 5 rows shown):

```
gammac1<-gamma_conv(EmpImpF,ref=2008,last=2013,
                    timeName="time",
                    printRanks = T)

## Ranks:
##      BE DK  FR  DE  EL  IE IT  LU  NL  PT  ES  UK
## [1,] 5.0 12  9.0 11.0 3.0 1.0 4  7.0  6.0 10.0  2.0  8.0
## [2,] 6.5 12  9.5  9.5 4.0 1.0 3  5.0  6.5 11.0  2.0  8.0
## [3,] 5.0 12  8.0 10.0 4.0 2.0 3  6.0  7.0 11.0  1.0  9.0
## [4,] 6.0 12 10.0  8.5 4.0 2.0 3  5.0  7.0 11.0  1.0  8.5
## [5,] 6.0 12  9.5  8.0 4.0 2.0 3  5.0  7.0 11.0  1.0  9.5
```

Finally, the gamma-convergence can be also calculated between pairs of subsequent years through the `gamma_conv_msteps` function, as illustrated below using the `EmpImpF` dataset:

```
gammac2<-gamma_conv_msteps(EmpImpF,startTime=2008,
                          endTime=2018, timeName = "time")
gammac2$res

## # A tibble: 11 x 2
##   time gammaConv
##   <dbl>   <dbl>
## 1 2008     NA
## 2 2009   0.426
## 3 2010   0.529
## 4 2011   0.594
## 5 2012   0.629
## 6 2013   0.653
## 7 2014   0.672
## 8 2015   0.691
## 9 2016   0.708
## 10 2017  0.716
## 11 2018  0.714
```

The results indicate that a high number of changes in the position of Member States occurred in the period 2008-2009, while few changes took place in the time period 2017-2018. By definition the first considered year will always be not available (NA).

Sigma-convergence

The sigma-convergence for the Employment rate indicator is calculated after invoking the `sigma_conv` function. For example, the sigma-convergence in the period 2008-2018 is computed as follows:

```
empSC <- sigma_conv(EmpImpF,timeName="time",
  time_0 = 2008,time_t = 2018)
empSC$res

## # A tibble: 11 x 5
##   time stdDev    CV mean devianceT
##   <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 2008  3.81 0.0473  80.6    175.
## 2 2009  4.51 0.0570  79.2    244.
## 3 2010  5.00 0.0635  78.8    300.
## 4 2011  5.51 0.0705  78.2    364.
## 5 2012  6.57 0.0847  77.5    518.
## 6 2013  6.90 0.0895  77.1    571.
## 7 2014  6.53 0.0840  77.7    511.
## 8 2015  5.82 0.0744  78.2    406.
## 9 2016  5.44 0.0689  78.9    355.
## 10 2017  5.18 0.0649  79.8    322.
## 11 2018  5.12 0.0634  80.8    315.
```

which is the `res` component of the output list. If the arguments `time_0` and `time_t` are not specified, then all years are considered, otherwise it is possible to select a time window for which the sigma-convergence is calculated. The resulting object is a tibble containing the values of sigma-convergence, that is the statistics `stdDev`, `CV` and the deviance along time.

A useful graphical representation of the standard deviation and of the coefficient of variation over time is obtained by invoking the `sigma_conv_graph` function (Figure 4):

```
mysigmaplot<-sigma_conv_graph(sigmaconvOut=empSC,
  time_0 = 2008,
  time_t = 2018, aggregation='EU12')
# mysigmaplot
```

where the `sigmaconvOut` argument requires an object obtained as output of the `sigma_conv` function, and where `time_0` and `time_t` arguments are the starting and ending time respectively. The desired aggregation of Member States (i.e. EU12), is specified in the argument `aggregation` as a string.

Both the standard deviation and the coefficient of variation decrease in the period 2013-2018. Thus, there is an evidence of sigma-convergence for the EU12 Member States in this time period. On the other hand, in the period 2008-2012, that is during the crisis of 2008, both the standard deviation and the coefficient of variation increase, thus indicating divergence.

Declaring convergence or divergence

Convergence and divergence may be strict or weak, upward or downward. The function `upDo_CoDi` returns a declaration of convergence according to the indicator type (`lowBest` or `HighBest`), considered in a given time interval for a specific cluster of Member States.

Let's consider a first example for the `emp_20_64_MS` dataset in which the indicator Employment rate is of type "highBest". Suppose that our aim is to determine the type of convergence or divergence occurring in the time interval from 2008 to 2010 using the variance, then we run the following code:

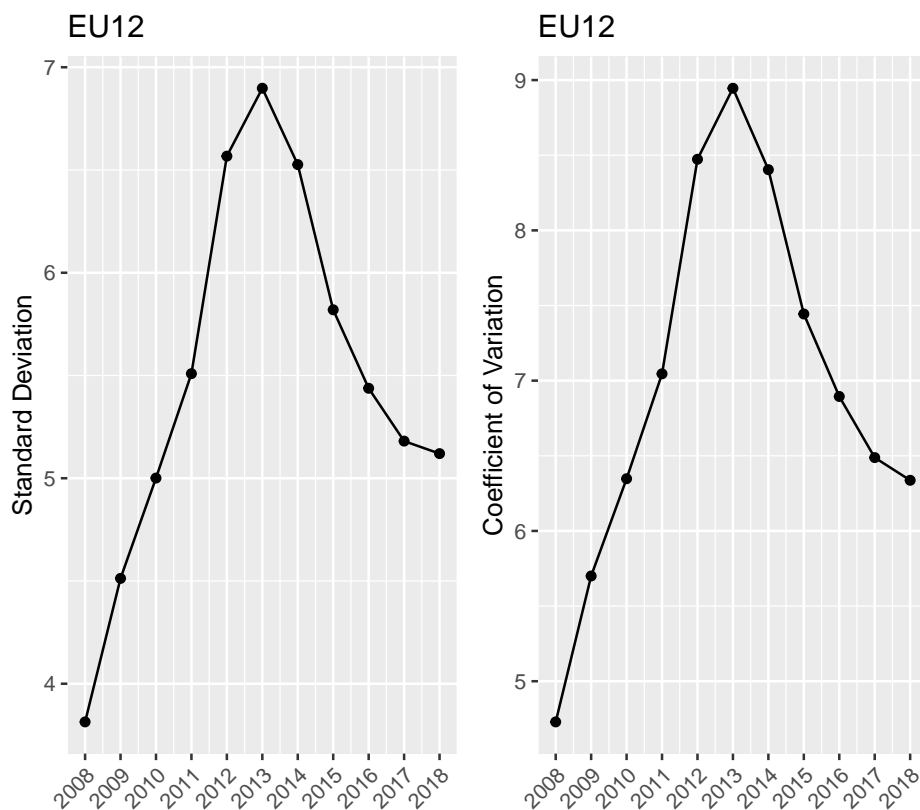


Figure 4: Standard deviation and coefficient of variation by year

```
Empconv<-upDo_CoDi(emp_20_64_MS,
  timeName = "time",
  indiType = "highBest",
  time_0 = 2008,
  time_t = 2010,
  heter_fun = "var")
```

where the sample variance is selected to summarize the dispersion through the string argument `heter_fun`. The `res` component in the output is a list object whose components are:

- a statement if convergence or divergence occurred:

```
Empconv$res$declaration_type
## [1] "Convergence"
```

- the type of convergence/divergence, i.e. strict or weak, downward or upward:

```
Empconv$res$declaration_strict
## [1] "none"

Empconv$res$declaration_weak
## [1] "Weak downward"
```

- a vector of Member States labels for which the differences between indicator values taken at target time and at reference time are greater than zero:

```
Empconv$res$declaration_split$names_incre
```

```
## [1] "AT" "DE" "LU" "MT" "RO"
```

- a vector of Member States labels for which the differences between indicator values taken at target time and at reference time are smaller than zero:

```
Empconv$res$declaration_split$names_decre
```

```
## [1] "BE" "BG" "CY" "CZ" "DK" "EE" "EL" "ES" "FI"
## [10] "FR" "HR" "HU" "IE" "IT" "LT" "LV" "NL" "PL"
## [19] "PT" "SE" "SI" "SK" "UK"
```

- a vector of differences between indicator values taken at target time and at reference time for each Member State:

```
Empconv$res$diffe_MS
```

```
##   AT  BE BG  CY CZ DE  DK   EE  EL  ES
## 1 0.1 -0.4 -6 -1.5 -2  1 -3.8 -10.3 -2.5 -5.7
##   FI  FR  HR  HU IE  IT  LT  LU  LV  MT
## 1 -2.8 -1.2 -2.8 -1.6 -8 -1.9 -7.7 1.9 -11.1 0.9
##   NL  PL  PT  RO  SE  SI  SK  UK
## 1 -0.7 -0.7 -2.8 0.4 -2.3 -2.7 -4.2 -1.7
```

- the average of such differences:

```
Empconv$res$diffe_averages
```

```
## [1] -2.860714
```

- the dispersions at reference time and at target time computed on the basis of the dispersion summary specified in the argument `heter_fun` (e.g. the variance in this example):

```
Empconv$res$dispersions
```

```
## Time: 2008 Time: 2010
## 29.76417 28.44423
```

Note that other possible functions in the argument `heter_fun` are `pop_var` and `sd` related to the population variance and the standard deviation respectively. User-developed functions are also allowed, for example:

```
diffQQmu <- function(vettore){
  (quantile(vettore,0.75) -
   quantile(vettore,0.25))/mean(vettore)
}
```

Once the user-developed function for summarizing dispersion is defined, it is specified in the argument `heter_fun` as a string:

```
Empconv1<-upDo_CoDi(emp_20_64_MS,
  timeName = "time",
  indiType = "highBest",
  time_0 = 2008,
  time_t = 2010,
  heter_fun = "diffQQmu")
```

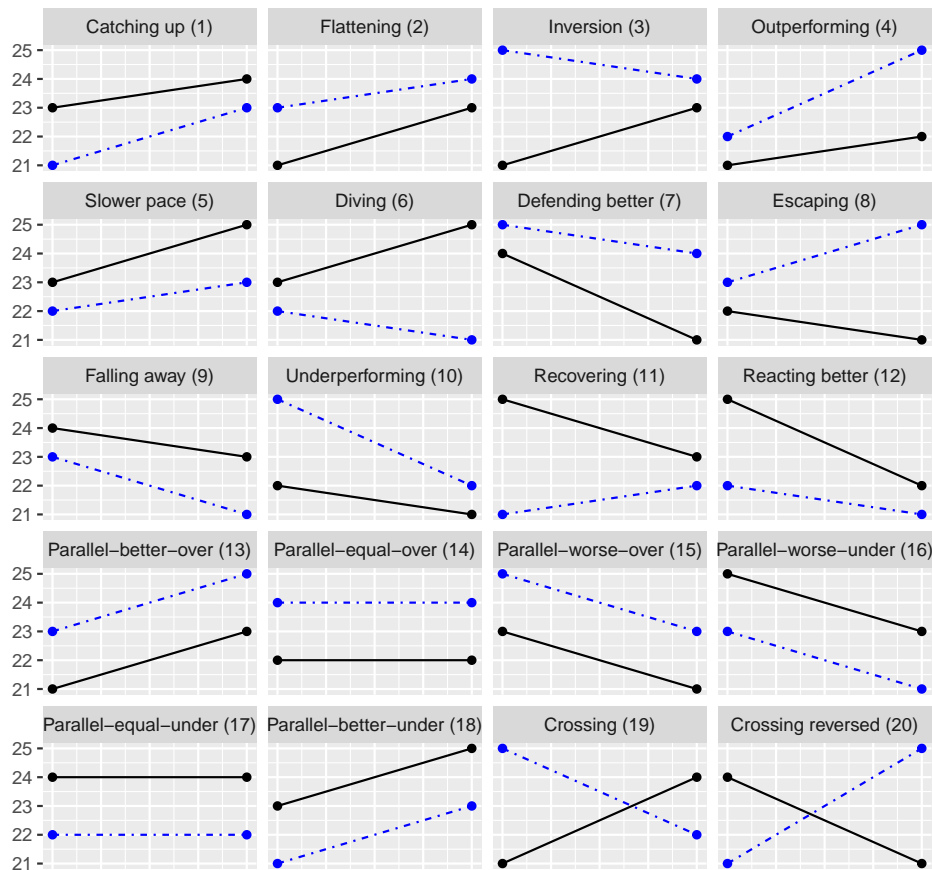


Figure 5: Patterns for an indicator of type "highBest" (dashed lines: Member State; solid lines: EU average)

Identification of patterns

Patterns of change along time for indicators in the EU are obtained by the function `ms_pattern_ori` (see `help(ms_pattern_ori)`, for both `lowBest` and `highBest` types of indicators. Thus, in the `ms_pattern_ori` function, the 20 patterns defined in Section 2.6 are defined through their numerical labels and corresponding string labels. Reference plots for the defined patterns are plotted by invoking the `patt_legend` function (see figures of Section 2.6), taking into account the type of indicator ("lowBest" or "highBest"). For indicators of type `highBest`, the reference plot (Figure 5) of the patterns is obtained as follows:

```
highind<-patt_legend(indiType="highBest")
fig5<-highind+geom_point()
figure5<-fig5+geom_point(ggplot2::aes(x = highind$data$time,
y = highind$data$measureM),
colour = "blue",linetype = "3313")
# figure5
```

Similarly, the reference plot of patterns for indicators of type "lowBest" is obtained by specifying the argument `indiType` (Figure 6):

```
lowhind<-patt_legend(indiType="lowBest")
fig6<-lowhind+geom_point()
figure6<-fig6+geom_point(ggplot2::aes(x = lowhind$data$time,
y = lowhind$data$measureM),
colour = "blue",linetype = "3313")
# figure6
```

The `emp_20_64_MS` dataset contains data of an indicator of type `highBest`, thus we run the `ms_pattern_ori` function with the following arguments:

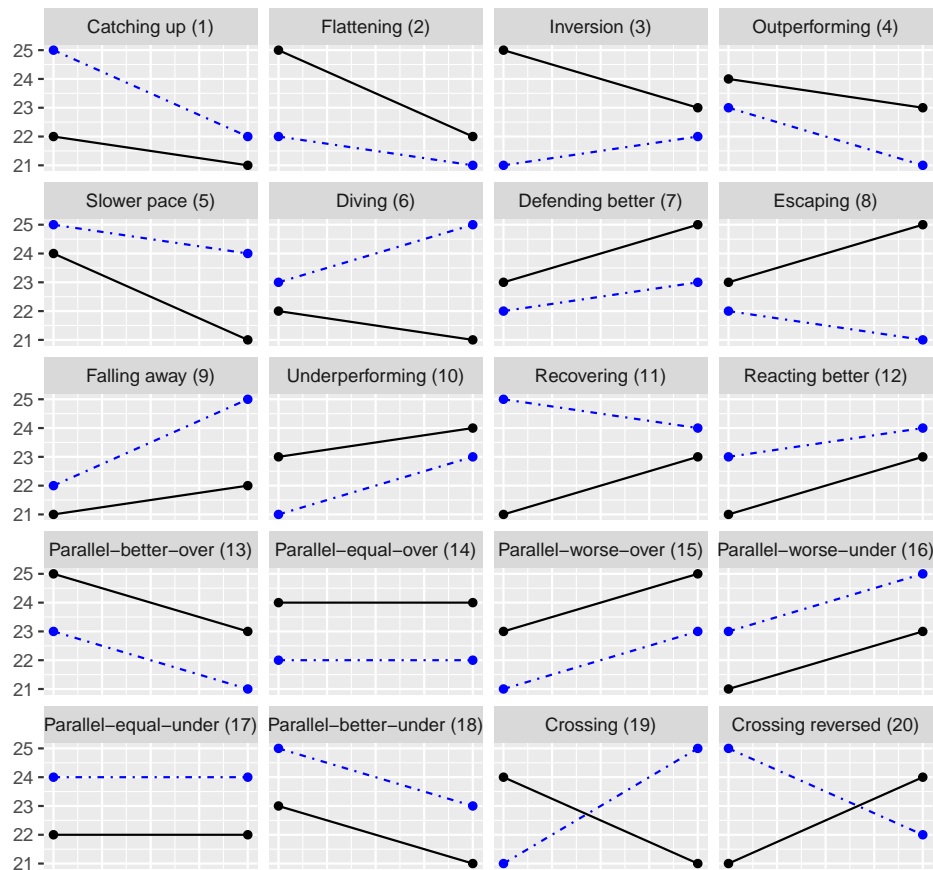


Figure 6: Patterns for an indicator of type "lowBest" (dashed lines: Member State; solid lines: EU average).

```
myemp <- ms_pattern_ori(emp_20_64_MS, "time",
  type="highBest")
mypattemp<-myemp$res$mat_label_tags
mypattempn<-myemp$res$mat_without_summaries
```

The `res` component of the output contains the numerical labels for the patterns as well as their string labels, as shown for the first six rows of the result:

```
head(mypattempn)

## # A tibble: 6 x 17
##   Country `2002/2003` `2003/2004` `2004/2005`
##   <chr>      <int>      <int>      <int>
## 1 AT          4          3          4
## 2 BE          6          1          1
## 3 BG          1          1          1
## 4 CY          4          4          3
## 5 CZ          3          3          4
## 6 DE          3          19         20
## # ... with 13 more variables:
## #   `2005/2006` <int>, `2006/2007` <int>,
## #   `2007/2008` <int>, `2008/2009` <int>,
## #   `2009/2010` <int>, `2010/2011` <int>,
## #   `2011/2012` <int>, `2012/2013` <int>,
## #   `2013/2014` <int>, `2014/2015` <int>,
## #   `2015/2016` <int>, `2016/2017` <int>,
## #   `2017/2018` <int>
```

```
head(mypattemp)
```

```
## # A tibble: 6 x 17
```

```
## Country `2002/2003` `2003/2004` `2004/2005`
## <chr> <chr> <chr> <chr>
## 1 AT Outperform~ Inversion Outperform~
## 2 BE Diving Catching up Catching up
## 3 BG Catching up Catching up Catching up
## 4 CY Outperform~ Outperform~ Inversion
## 5 CZ Inversion Inversion Outperform~
## 6 DE Inversion Crossing Crossing r~
## # ... with 13 more variables:
## # `2005/2006` <chr>, `2006/2007` <chr>,
## # `2007/2008` <chr>, `2008/2009` <chr>,
## # `2009/2010` <chr>, `2010/2011` <chr>,
## # `2011/2012` <chr>, `2012/2013` <chr>,
## # `2013/2014` <chr>, `2014/2015` <chr>,
## # `2015/2016` <chr>, `2016/2017` <chr>,
## # `2017/2018` <chr>
```

France is a Member State located at row 12 in the output (not shown) thus for the time interval 2006-2007 the pattern is:

```
mypattemp$`2006/2007`[12]
## [1] "Slower pace"
```

The resulting plot of the calculated pattern is shown in Figure 7, where the dashed line refers to France and the solid line refers to EU:

```
matRaw1 <- emp_20_64_MS[,-1]
EUavemp <- cbind(emp_20_64_MS[,1],
  EUavemp=apply(matRaw1,1,mean))
EUavemph<-EUavemp[5:6,]
avehu<-emp_20_64_MS[5:6,c("FR")]
gFR<-ggplot() + geom_point(aes(x=EUavemph$time,
  y=EUavemph$EUavemp),
  color='black') +
  geom_point(aes(x=EUavemph$time,y=avehu$FR),
  color='blue') +
  geom_line(aes(x=EUavemph$time,y=EUavemph$EUavemp),
  color='black') +
  geom_line(aes(x=EUavemph$time,y=avehu$FR),
  color='blue',linetype = 2) +
  labs(y="Employment rate indicator for France", x="")+
  scale_x_continuous(breaks = c(2006, 2007))+theme_classic()
#gFR
```

Thus, for this type of indicator, the interpretation of the "Slower pace" pattern is that in the period 2006-2007, the performance of France is initially lower than the EU average and grows at a slower rate, increasing the gap over time.

Scoreboards of countries

Differences of indicator values between subsequent years are at core of the scoreboard of countries, that is $y_{m,i,t} - y_{m,i,t-1}$ with $t \in \Omega_T$. The function `scoreb_yrs` calculates these differences and classifies them with respect to their distance from the mean; an example with the `emp_20_64_MS` dataset is:

```
data(emp_20_64_MS)
resTB <- scoreb_yrs(emp_20_64_MS,timeName = "time")
```

The result is a list of three components: (i) the summary statistics; (ii) the numerical labels to indicate the interval of the partition a level belongs to; (iii) the interval of the partition a change belongs to.

Numerical labels are assigned as described in Mascherini *et al.* (2018):

- value -1 if the original level or change is $y_{m,i,t} \leq \bar{y}_{i,t} - 1 \cdot s_{i,t}$;

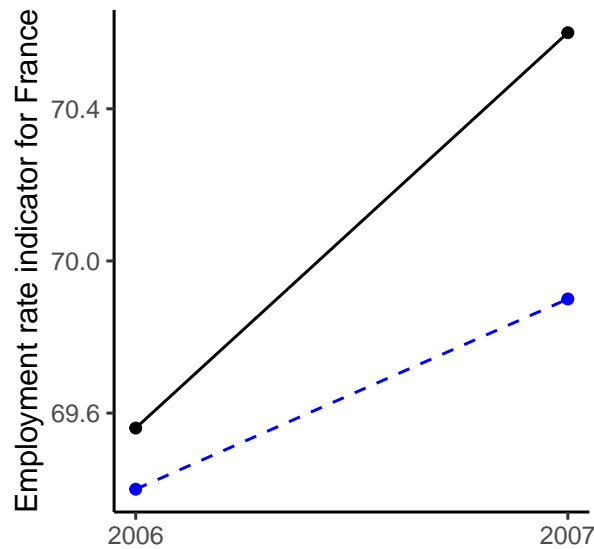


Figure 7: Pattern of France for years 2006-2007 (dashed line: France; solid line: EU average)

- value -0.5 if the original level or change is $\bar{y}_{i,t} - 1 \cdot s_{i,t} < y_{m,i,t} \leq \bar{y}_{i,t} - 0.5 \cdot s_{i,t}$;
- value 0 if the original level or change is $\bar{y}_{i,t} - 0.5 \cdot s_{i,t} < y_{m,i,t} \leq \bar{y}_{i,t} + 0.5 \cdot s_{i,t}$;
- value $+0.5$ if the original level or change is $\bar{y}_{i,t} + 0.5 \cdot s_{i,t} < y_{m,i,t} \leq \bar{y}_{i,t} + 1 \cdot s_{i,t}$;
- value $+1$ if the original level or change is $y > \bar{y}_{i,t} + 1 \cdot s_{i,t}$.

where $\bar{y}_{i,t}$ is the average at time t of the indicator i , and $s_{i,t}$ is the standard deviation. To compare a country with the EU average (Figure 8), the following steps are recommended, when starting from raw data:

```

selectedCountry <- "IT"
timeName <- "time"
outSig <- sigma_conv(emp_20_64_MS, timeName = timeName,
  time_0=2002,time_t=2016)
estrattore<- emp_20_64_MS[[timeName]] >= 2002 &
  emp_20_64_MS[[timeName]] <= 2016;
ttmp <- cbind(outSig$res,
  dplyr::select(emp_20_64_MS[estrattore,],
  -contains(timeName)))
miniY <- min(emp_20_64_MS[,
  - which(names(emp_20_64_MS) == timeName )])
maxiY <- max(emp_20_64_MS[,
  - which(names(emp_20_64_MS) == timeName )])
myx_angle <- 45
myG2 <- ggplot(ttmp) + ggtitle(
  paste("EU average (black, solid) and country",
  selectedCountry , " (red, dotted)") +
  geom_line(aes(x=ttmp[,timeName], y =ttmp[,"mean"]),
  colour="black") +
  geom_point(aes(x=ttmp[,timeName],y =ttmp[,"mean"]),
  colour="black") +
  ylim(c(miniY,maxiY)) +
  xlab("") +
  ylab("Employment rate Indicator") +
  theme(legend.position = "none")+
  geom_line( aes(x=ttmp[,timeName],
  y = ttmp[,"IT"],
  colour="red"),
  linetype="dotted") +
  geom_point( aes(x=ttmp[,timeName],
  y = ttmp[,"IT"],
  colour="red")) +
  ggplot2::scale_x_continuous(breaks = ttmp[,timeName],
  labels = ttmp[,timeName]) +

```

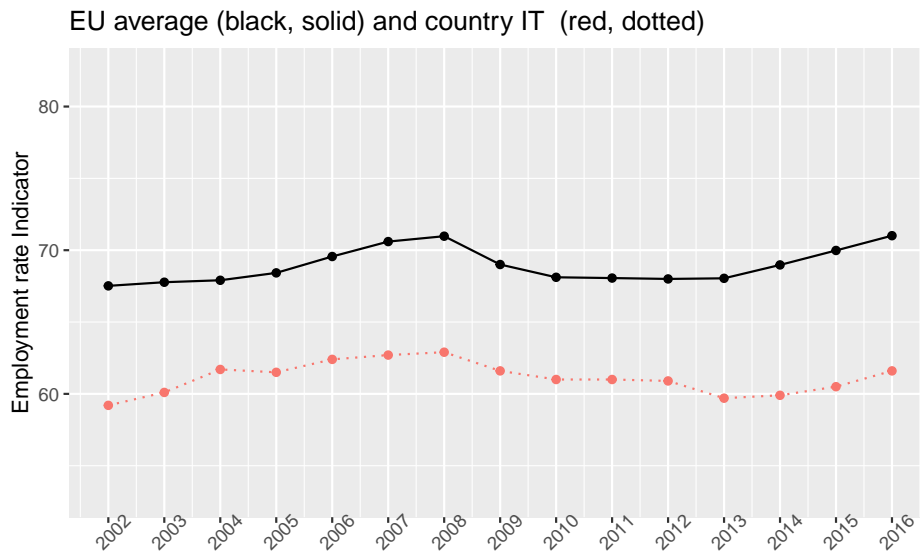


Figure 8: Comparison of Member States with EU (dotted line for the selected Member State)

```
ggplot2::theme(
  axis.text.x=ggplot2::element_text(
    angle = myx_angle
  ))
```

```
# myG2
```

It is also possible to graphically show departures in terms of the above defined numerical labels assigned to the partition (Figure 9):

```
obe_lvl <- scoreb_yrs(emp_20_64_MS,
  timeName = timeName)$res$sco_level_num
# select subset of time
estrattore <- (obe_lvl[[timeName]] >= 2009 &
  obe_lvl[[timeName]] <= 2016)
scobelvl <- obe_lvl[estrattore,]
my_MSstd <- ms_dynam( scobelvl,
  timeName = "time",
  displace = 0.25,
  displaceh = 0.45,
  dimeFontNum = 3,
  myfont_scale = 1.35,
  x_angle = 45,
  axis_name_y = "",
  axis_name_x = "",
  alpha_color = 0.9
)
```

```
# my_MSstd
```

3.2. Smoothing raw data

Changes of values observed for a selected indicator in a given Member State might be due to year-specific and local fluctuations. It may be of interest to explore this hypothesis, in particular when the goal is not prediction but the analysis of variance reduced indicators. A smoothing procedure removes sudden large changes and produces a less variable collection of indicator values over time. Given that in the EU context very short time series (panel data) are processed, a three points weighted average smoother is typically adopted, that is the smoother substitutes an original raw value $y_{m,i,t}$ with the following weighted average:

$$\check{y}_{m,i,t} = y_{m,i,t-1} (1 - w)/2 + w y_{m,i,t} + y_{m,i,t+1} (1 - w)/2 \quad (13)$$

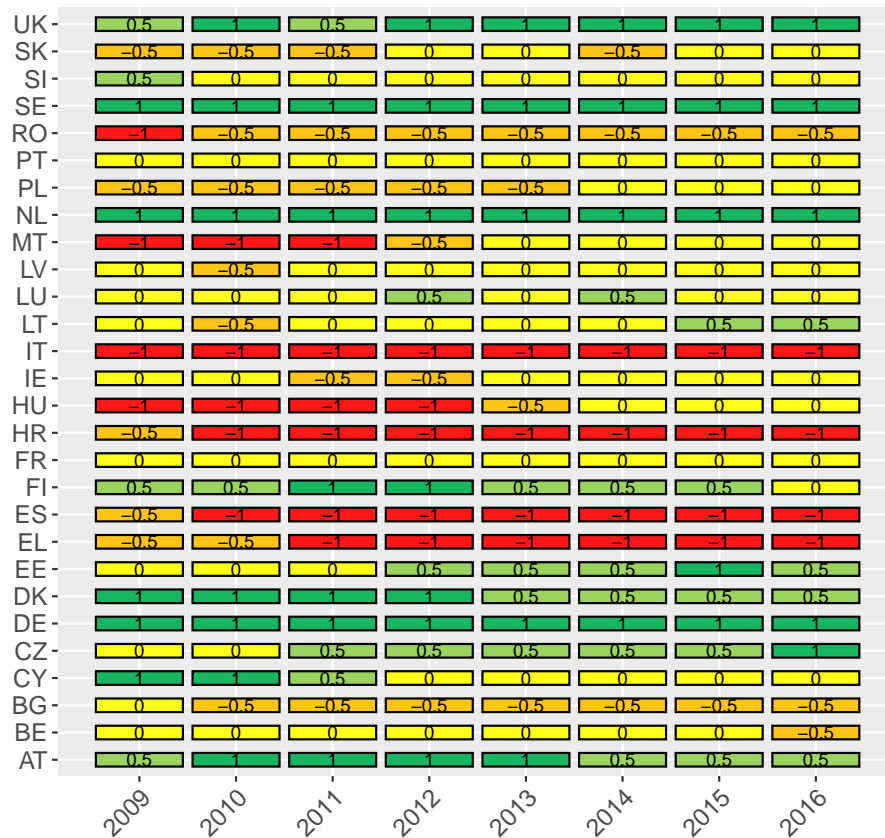


Figure 9: Qualitative representation of departure.

with $0 < w \leq 1$. The special case $w = 1$ corresponds to no smoothing. In case of missing values or if the weight is outside the interval $(0, 1]$ then an NA is returned. The first and last values are smoothed using weights w and $1 - w$.

The weighted average smoothing is implemented in the `smoo_dataset` function that requires a dataset without missing values. Furthermore, the time variable must be deleted from the dataset before running the `smoo_dataset` function. For example, the smoothing of the `TBhealthF` dataset is performed after missing values are imputed (Section A.3). Here the following three Member States are selected to illustrate the smoothing: Italy, France and Germany, coded as “IT”, “FR” and “DE” respectively. A tibble object is created with just these three countries and the time variable:

```
workTB <- dplyr::select(TBhealthF,time,IT,DE,FR)
check_data(workTB)$res

## [1] TRUE
```

The `check_data` function confirms that the dataset is ready for smoothing. Thus, the smoothing step can take place after removing the time variable:

```
require(dplyr)
resSM <- smoo_dataset(dplyr::select(workTB,-time),
  leadW = 0.15,
  timeTB= dplyr::select(workTB,time))
resSM <- rename(resSM, IT1=IT, DE1=DE, FR1=FR)
```

Note that the first argument is the complete dataset `workTB` with just country columns, while the third argument `timeTB` is the one-column dataset to be bound to smoothed data.

The argument `leadw` refers to the leading positive weight where $0 < w \leq 1$ as described above. In the above example the smoothing is firstly performed with a weight equal to 0.15.

To carry out a comparison between the original indicator and the smoothed versions, the following dataset is built:

```
compaTB <- bind_cols(workTB, dplyr::select(resSM,-time))[,
  c("time", "IT", "IT1", "DE", "DE1", "FR", "FR1")]
```

The graphical output shows changes for Italy (“IT”), Germany (“DE”) and France (“FR”) with the original indicator as straight line and the smoothed version as dotted line (Figure 10):

```
library(gridExtra)
library(ggplot2)
library(ggpubr)
ITq<-qplot(time,IT,data=compaTB, xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=IT1),colour="red", linetype = "dotted") +
  geom_point(aes(x=time,y=IT1),colour="red",shape=8)+
  labs(caption = "IT = Italy")
DEq<-qplot(time,DE,data=compaTB, xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=DE1),colour="red",linetype = "dotted") +
  geom_point(aes(x=time,y=DE1),colour="red",shape=8)+
  labs(caption = "DE = Germany")
FRq<-qplot(time,FR,data=compaTB, xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=FR1),colour="red",linetype = "dotted") +
  geom_point(aes(x=time,y=FR1),colour="red",shape=8)+
  labs(caption = "FR = France")

fig10<-grid.arrange(ITq, DEq, FRq, nrow=3)
```

```
figure10<-annotate_figure(fig10,
  left = text_grob("Self-perceived health indicator",
    rot = 90, vjust = 1)
)

# figure10
```

If the value of weight is equal to 0.50 the resulting amount of smoothing is reduced (Figure 11):

```
resSM2 <- smoo_dataset(dplyr::select(workTB,-c(time)),
  leadW = 0.5,
  timeTB= dplyr::select(workTB,time))
resSM2 <- rename(resSM2, IT1=IT, DE1=DE, FR1=FR)

compaTB2 <- dplyr::select(bind_cols(workTB,
  dplyr::select(resSM2,-time)),
  time,IT,IT1,DE,DE1, FR, FR1)
ITq2<-qplot(time,IT,data=compaTB2,xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=IT1),colour="red",linetype = "dotted") +
  geom_point(aes(x=time,y=IT1),colour="red",shape=8)+
  labs(caption = "IT = Italy")
DEq2<-qplot(time,DE,data=compaTB2, xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=DE1),colour="red",linetype = "dotted") +
  geom_point(aes(x=time,y=DE1),colour="red",shape=8)+
  labs(caption = "DE = Germany")
FRq2<-qplot(time,FR,data=compaTB2,xlab = "", ylab = "") +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=FR1),colour="red",linetype = "dotted") +
  geom_point(aes(x=time,y=FR1),colour="red",shape=8)+
  labs(caption = "FR = France")
#grid.arrange(ITq2, DEq2, FRq2, nrow=3)
fig11<-grid.arrange(ITq2, DEq2, FRq2, nrow=3)

figure11<-annotate_figure(fig11,
  left = text_grob("Self-perceived health indicator",
    rot = 90, vjust = 1)
)

#figure11
```

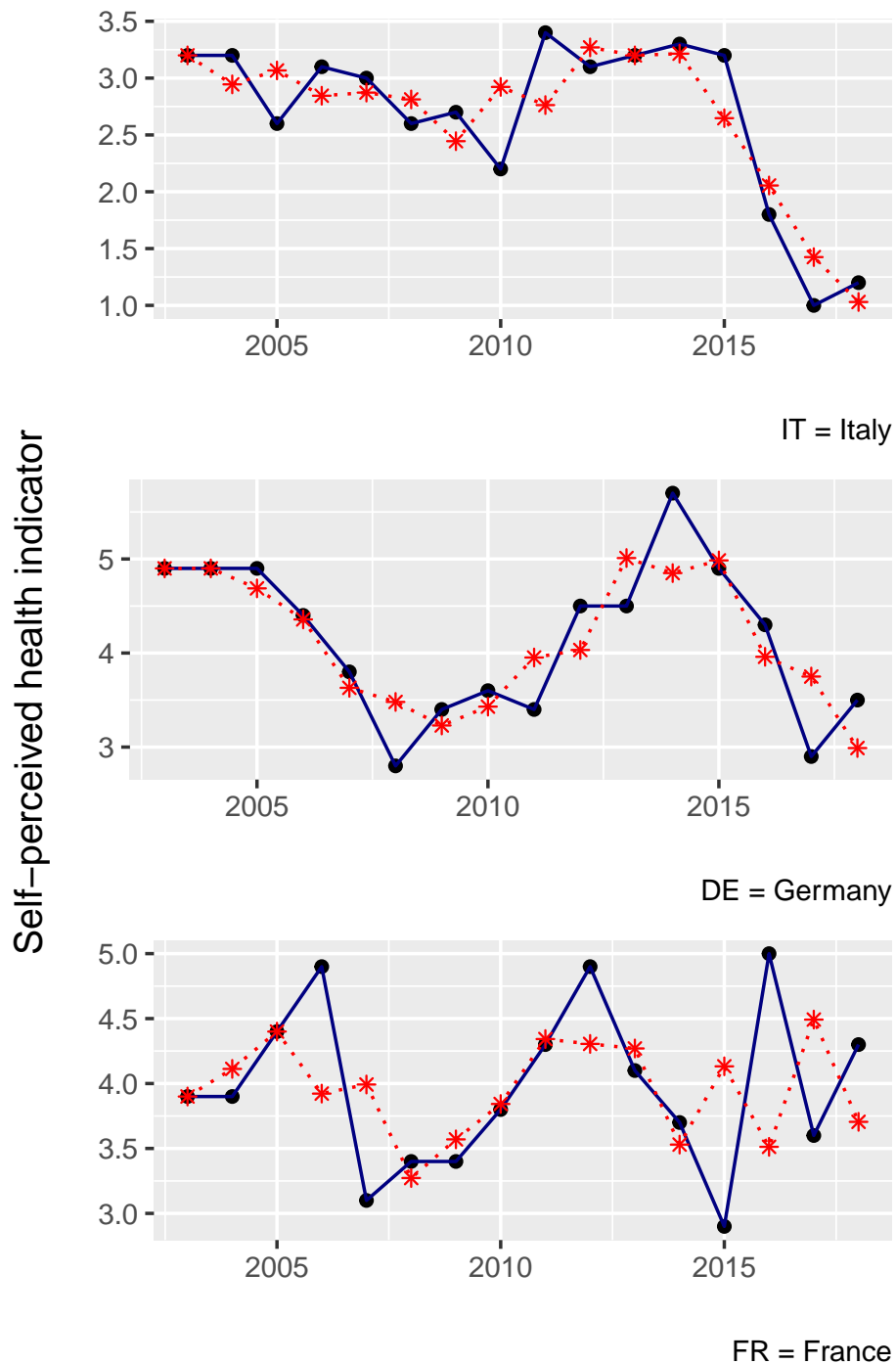


Figure 10: Plot of the original indicator as straight line and the smoothed version as dotted line, (weight 0.15).

It is worth noticing that a time window larger than 3 could be applied, but careful consideration is recommended about economic and social changes that may happen in EU during 5 or more consecutive years before smoothing.

Several alternative smoothing algorithms are available in R. Classical moving average smoothers are described in the **caTools** R package (Tuszynski 2020) (see for example the `runmean` function). In the **convergEU** package, smoothing based on moving average is implemented in the function `ma_dataset`, where the time window of the moving average is selected through the argument `kappa` (`help(ma_dataset)`).

Below, a smoother based on moving average for the `TBhealthF` dataset is applied to the following countries: Italy (“IT”), Germany (“DE”) and France (“FR”):

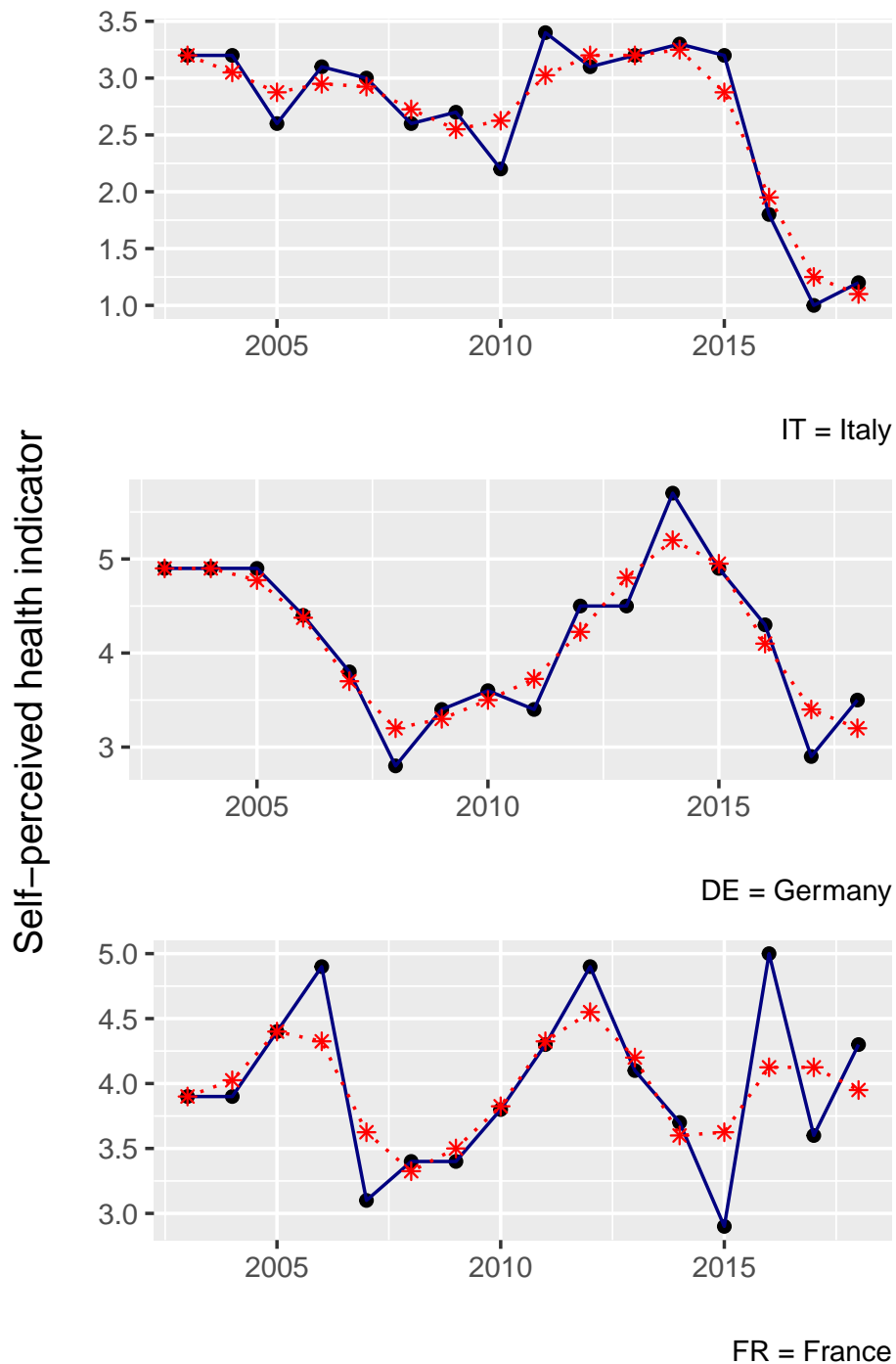


Figure 11: Plot of the original indicator as straight line and of the smoothed version as dotted line (weight 0.50)

```
cuTB1 <- TBhealthF[,c("time", "IT", "DE", "FR")]
```

Time windows of the moving average are selected setting the `kappa` argument, for example $k = 3$, $k = 4$ and $k = 5$:

```
ma3<-ma_dataset(cuTB1, kappa=3, timeName= "time")
ma4<-ma_dataset(cuTB1, kappa=4, timeName= "time")
ma5<-ma_dataset(cuTB1, kappa=5, timeName= "time")
```

where the output is a standard tidy dataset of smoothed values. In Figure (12), plots after smoothing are shown for the three levels of smoothing: original data (straight line), $k = 3$ (dotted line), $k=4$ (dashed line), $k=5$ (dot-dash line) (Figure 12):

```

myGit <- ggplot(cuTB1, aes(x=time, y=cuTB1$IT))+
  geom_line()+geom_point()+
  geom_line(aes(x=time, y=ma3$res$IT), colour="red",
            linetype="dotted")+
  geom_point(aes(x=time, y=ma3$res$IT), colour="red")+
  labs(y="", x="")+
  #
  geom_line(aes(x=time, y=ma4$res$IT), colour="blue",
            linetype="dashed")+
  geom_point(aes(x=time, y=ma4$res$IT), colour="blue")+
  #
  geom_line(aes(x=time, y=ma5$res$IT), colour="green",
            linetype="dotdash")+
  geom_point(aes(x=time, y=ma5$res$IT), colour="green")+
  labs(caption = "IT = Italy")

myGde <- ggplot(cuTB1, aes(x=time, y=cuTB1$DE))+
  geom_line()+geom_point()+
  geom_line(aes(x=time, y=ma3$res$DE), colour="red",
            linetype="dotted")+
  geom_point(aes(x=time, y=ma3$res$DE), colour="red")+
  labs(y="", x="")+
  #
  geom_line(aes(x=time, y=ma4$res$DE), colour="blue",
            linetype="dashed")+

  geom_point(aes(x=time, y=ma4$res$DE), colour="blue")+
  #
  geom_line(aes(x=time, y=ma5$res$DE), colour="green",
            linetype="dotdash")+
  geom_point(aes(x=time, y=ma5$res$DE), colour="green")+
  labs(caption = "DE = Germany")

myGfr <- ggplot(cuTB1, aes(x=time, y=cuTB1$FR))+
  geom_line()+geom_point()+
  geom_line(aes(x=time, y=ma3$res$FR), colour="red",
            linetype="dotted")+
  geom_point(aes(x=time, y=ma3$res$FR), colour="red")+
  labs(y="", x="")+
  #
  geom_line(aes(x=time, y=ma4$res$FR), colour="blue",
            linetype="dashed")+
  geom_point(aes(x=time, y=ma4$res$FR), colour="blue")+
  #
  geom_line(aes(x=time, y=ma5$res$FR), colour="green",
            linetype="dotdash")+
  geom_point(aes(x=time, y=ma5$res$FR), colour="green")+
  labs(caption = "FR = France")

fig12<-grid.arrange(myGit, myGde, myGfr, nrow=3)

figure12<-annotate_figure(fig12,
  left = text_grob("Self-perceived health indicator",
                  rot = 90, vjust = 1)
)
# figure12

```

We remark that time series here considered are typically quite short, say 15 observations, therefore if $k = 7$ then a lot of observations are smoothed with a different number of observations (shorter at the start and at the end). Therefore, in the considered context, smoothing is typically performed over a time window of maximum 5 consecutive years.

3.3. Automatic compilation of fiches

The `counvergEU` package makes possible to prepare indicator fiches as well as country fiches, and to automatically produce all tables and figures assembled into the HTML output file. These two helper functions were specifically developed to facilitate the work of the users who regularly deal with EU indicators.

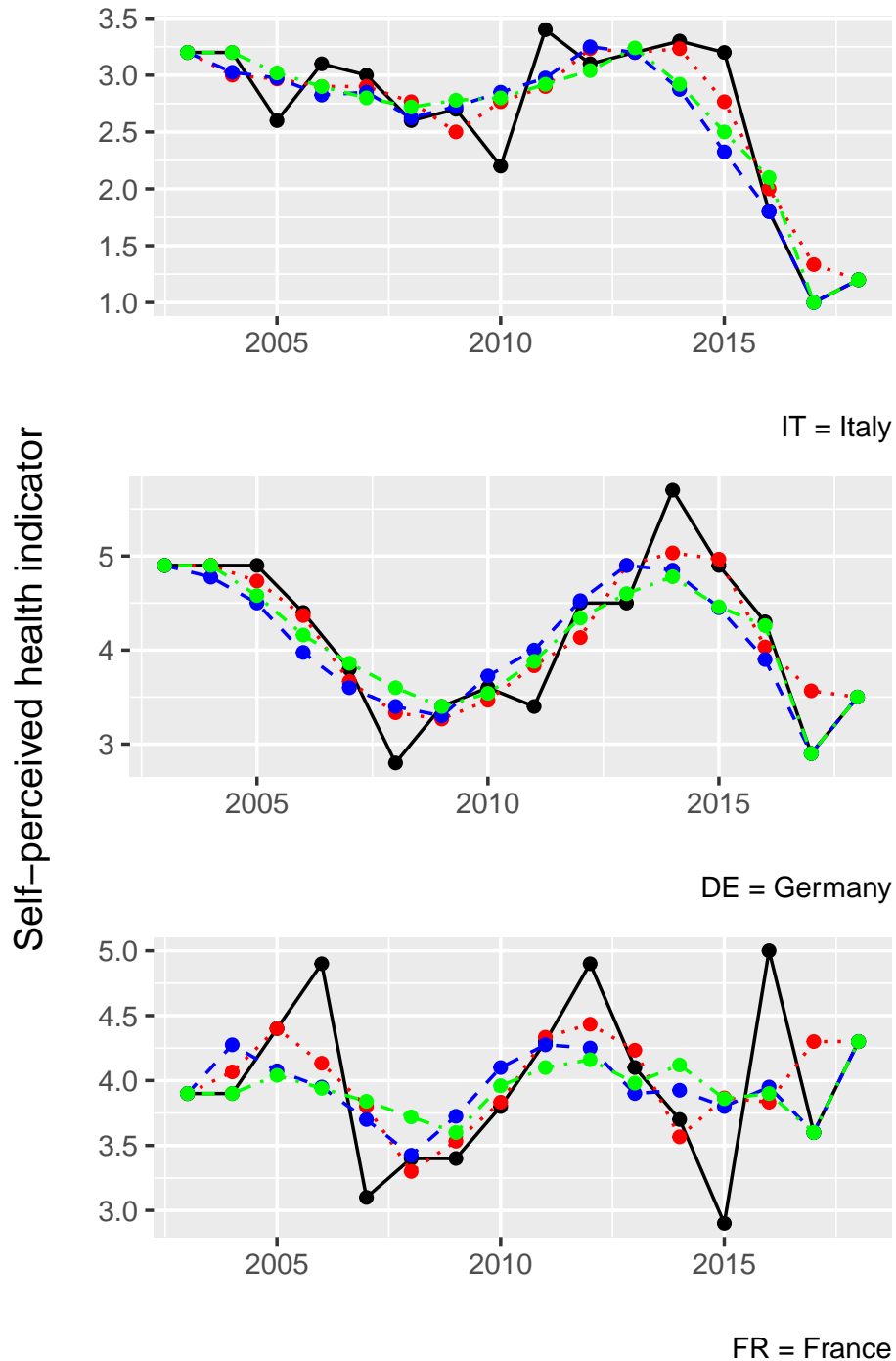


Figure 12: Plot of an indicator for Italy, Germany and France under different degrees of smoothing: original (solid line), $k = 3$ (dotted line), $k = 4$ (dashed line), $k = 5$ (dot-dash line)

It is important to emphasize that an active Internet connection is mandatory while invoking both functions described below in order to obtain a properly rendered HTML file.

Country fiches

The `go_ms_fi` function returns a country fiche, which is a report focused on statistical summaries pertaining to a selected country and a given indicator. This function produces an HTML file that shows several statistical summaries for a selected reference country and for another set of Member States, selected for meaningful comparison. Scoreboards of the se-

lected countries, patterns of convergence and divergence as well as country ranking tables are all parts of this fiche type.

In addition to automatically compiling country fiches, the `go_ms_fi` function creates a folder along an existing path in the local disk and copies Rmarkdown source files within it. The main Rmarkdown file is parameterized, thus the content of the final document depends on the dataset, time interval and some other context information provided as input.

It is very important to prepare complete data in the required format "time by countries"; that is, the dataset is made by a time variable and as many other variables as countries that enter into the calculation of averages. Failing to satisfy these requirements causes the use of a wrong mean value at each time point.

Below, a call to the function `go_ms_fi` illustrates the syntax for the analysis of the `EmpImpF` dataset:

```
go_ms_fi(
  workDF = 'EmpImpF',
  countryRef = 'DE',
  otherCountries = "c('IT','UK','FR')",
  time_0 = 2005,
  time_t = 2015,
  tName = 'time',
  indiType = 'highBest',
  aggregation= 'EU27',
  x_angle= 45,
  dataNow= Sys.time(),
  author = 'A.Student',
  outFile = 'Germany-up2-2016-EMP',
  outDir = 'F:/analysis/IT2018',
  indiName= 'EmpImpF')
```

It is worth noting the presence of some constraints and of a rather uncommon way of passing R objects to this function. In fact, the first argument is the name of the R object as string enclosed within quotes, instead of just the name of such R object: such string contains the name of the dataset that must be available in the R workspace before invoking the `go_ms_fi` function. The second argument `countryRef` is a string with the short name of a the reference Member State that will appear in one-country plots. The argument `indiType` specifies if the considered indicator is of type "lowBest" or "highBest".

Of particular importance is the argument `outFile`, which is a string indicating the name of the output file. Similarly, `outDir` is the path (possibly with disk unit and folders) in which the final compiled HTML file will be stored. The syntax of the path depends on the operating system; for example `outDir='F:/analysis/IT2018'` indicates that in the usb disk called 'F', within the folder 'analysis' is located folder 'IT2018' where R will write the country fiche. Note that a disk called 'F' must exist and also the folder 'analysis' must exist in such unit, while folder 'IT2018' is created by the function if it does not already exist.

Within the above mentioned output directory, besides the compiled HTML, a file that contains data, statistical results and ggplot2 objects (images) produced during the compilation of the country fiche is also stored; its name is obtained by pasting the string specified in the `outFile` argument to the string '-workspace.RData'.

Indicator fiches

The `go_indica_fi` function builds and compiles indicator fiches in HTML format. Arguments of this function are: an indicator, a time window, a set of countries. The resulting HTML file reports measures of convergence as well as time series plots, box-plots, and patterns of convergence and divergence. An indicator fiche focuses on the selected indicator and a selected aggregation of Member States, without preferential emphasis on some Member States (no weights applied). Note that some arguments are passed as strings instead of R objects, as described for country fiches.

Below, an example of the syntax needed for the analysis of the `EmpImpF` dataset is shown:

```

go_indica_fi(
  time_0 = 2005,
  time_t = 2015,
  timeName = 'time',
  workDF = 'EmpImpF' ,
  indicaT = 'lfsa_ergaed',
  indiType = c('highBest','lowBest')[1],
  seleMeasure = 'all',
  seleAggre = 'EU15',
  x_angle = 45,
  data_res_download = FALSE,
  auth = 'A.Student',
  dataNow = Sys.time(),
  outFile = 'EmpImp',
  outDir = 'F:/analysis/IT2018')

```

Note that the argument `seleMeasure` refers to the set of measures of convergence for which the analysis should be performed; if `seleMeasure = 'all'`, as shown in the example above, then results include all four measures of convergence. The set of countries for which the analysis of convergence should be performed is selected with the argument `seleAggre`, as a string object; the default is the set of Member States labeled as “EU27”. To note that if the selection is “custom” the calculation of convergence measures can be performed on any type of datasets whose format is time by region, where regions may be European countries, US States, a collection of big cities, or any other collection of meaningful units.

Similarly to the country fiche, the option `outDir` specifies an existing unit and a number of existing folders, i.e. the path in which the compiled fiche will be stored. Only the last folder may be created by the function if it does not exist before compilation.

A web application to prepare fiches

Eurofound web site hosts a web application devoted to the preparation of fiches, therefore it is possible to obtain indicator and country fiches without installing any software and to read the results on tablets or smartphones, if an Internet connection is available. The web application is not a comprehensive implementation of all functionalities available in the **convergEU** R package, but it is focused on the compilation of fiches by providing arguments through a graphical interface, that is without studying the R language. A selected subset of indicators and EU Member States can be processed using the web-app. Further details are available at the following link: <https://www.eurofound.europa.eu/data/convergence-hub/convergeu-app/>

4. Discussion

The **convergEU** package enables fast calculation of several measures of convergence for EU Members States, as well as for other aggregates of regions. Thanks to a set of specialized function, this package can be used without advanced programming skills. Nevertheless, data preparation and cleaning (Appendix I) may involve less than trivial scripting skills, especially if data do not come from official EU providers like Eurostat.

Future work could include the improvement of several features of this package. R developers might notice that the code was written rather to be easily read by non-programmers and to be user-friendly for social and economics scientists, than for an efficient implementation; anyway, performances in terms of speed and RAM space are not critical in the current implementation. S3 generic functions, R error handling and even R6 classes will be considered in future code refactoring, with the aim of making the extension of this package easier.

A software implementation to calculate measures of convergence is also available in STATA (StataCorp LLC. 2019), a widespread proprietary statistical software, but our R package, as of October 2021, has several additional features including:

- imputation of missing data and smoothing of original raw data;

- automated downloading of Eurostat data in real time;
- functions to fill and compile templates into downloadable HTML "fiches", which contain all the results pertaining to an indicator or to a Member State;
- a derived Shiny web app able to prepare fiches without coding.

The **convergEU** R package may be also considered a tool to support research on convergence, for example in view of a possible extension towards multivariate analysis. The refinement of criteria to classify Member States performances, like qualitative patterns and the EPSR scoreboard, is another area of paramount importance due to the role that such classification may play in policy making.

Acknowledgments

Our thanks to Eurofound for the co-operation during the development and testing of this package. The information and views set out in this article are those of the authors and do not necessarily reflect the official opinion of Eurofound and of the European Commission. Federico M. Stefanini and Nedka D. Nikiforova acknowledge the financial support provided by Eurofound and by the "Dipartimenti Eccellenti 2018-2022", ministerial funds, Italy.

Note: Research carried out prior to the UK's withdrawal from the European Union on 31 January 2020, and published subsequently, may include data relating to the 28 EU Member States.

References

- Barro RJ, Sala-i-Martin X (1992). "Convergence." *Journal of Political Economy*, **100**(2), 223–252. doi:<https://doi.org/10.1086/261816>.
- Boyle GE, McCarthy TG (1999). "Simple Measures of Convergence in Per Capita GDP: A Note on Some Further International Evidence." *Applied Economics Letters*, **6**(6), 343–347. doi:<https://doi.org/10.1080/135048599353041>.
- European Commission SG (2018). "European Pillar of Social Rights." Publications Office of the European Union, Luxembourg. doi:[10.2792/154364](https://doi.org/10.2792/154364). URL <https://data.europa.eu/doi/10.2792/154364>.
- Friedman M (1992). "Do Old Fallacies Ever Die?" *Journal of Economic Literature*, **30**(4), 2129–2132.
- Heichel S, Pape J, Sommerer T (2005). "Is There Convergence in Convergence Research? An Overview of Empirical Studies on Policy Convergence." *Journal of European Public Policy*, **12**(5), 817–840. doi:[10.1080/13501760500161431](https://doi.org/10.1080/13501760500161431).
- Kenworthy L (1999). "Economic Integration and Convergence: A Look at the US States." *Social Science Quarterly*, **80**(4), 858–869.
- Mascherini M, Bisello M, Dubois H, Eiffe F (2018). "Upward Convergence in the EU: Concepts, Measurements and Indicators." Publications Office of the European Union, Luxembourg. doi:[10.2806/68012](https://doi.org/10.2806/68012). URL <http://eurofound.link/ef18003>.
- Quah D (1993). "Galton's Fallacy and Tests of the Convergence Hypothesis." *Scandinavian Journal of Economics*, **95**(4), 427–443.

- Sala-i-Martin X (1996). “The Classical Approach to Convergence Analysis.” *The Economic Journal*, **106**(437), 1019–1036. doi:<https://doi.org/10.2307/2235375>.
- Signorelli M (2005). “Growth and Employment: Comparative Performance, Convergences and Co-movements.” *Technical report*, University of Perugia, Perugia, Italy.
- Solow RM (1956). “A Contribution to the Theory of Economic Growth.” *The Quarterly Journal of Economics*, **70**(1), 65–94. doi:[10.2307/1884513](https://doi.org/10.2307/1884513).
- StataCorp LLC (2019). *Stata Statistical Software: Release 16*. College Station, TX, USA. URL <https://www.stata.com/>.
- Stefanini FM, Nikiforova ND, Litardi C, Bisello M, Peruffo E, Mascherini M (2020a). “Monitoring Upward Convergence in the EU with R: The *convergEU* Package.” Eurofound Working Paper, Eurofound reference number: WPEF20008. URL <https://www.eurofound.europa.eu/sites/default/files/wpef20008.pdf>.
- Stefanini FM, Nikiforova ND, Litardi C, Peruffo E, Mascherini M (2020b). “Tutorial: Analysis of Convergence with the *convergEU* Package.” Vignette, *convergEU* R package. URL <https://local.disia.unifi.it/stefanini/RESEARCH/coneu/tutorial-conv.html>.
- Tuszynski J (2020). *caTools: Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc.* R package version 1.18.0, URL <https://CRAN.R-project.org/package=caTools>.
- Wickham H, Bryan J (2019). *readxl: Read Excel Files*. R package version 1.3.1, URL <https://CRAN.R-project.org/package=readxl>.
- Wickham H, Golemund G (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. 1st edition. O’Reilly Media, Inc. ISBN 1491910399.
- Wickham H, Hester J, Francois R (2018). *readr: Read Rectangular Text Data*. R package version 1.3.1, URL <https://CRAN.R-project.org/package=readr>.

A. Appendix I: how to prepare data

In this Appendix, a general introduction is provided on data preparation to work with the **convergEU** R package, including the description on how to impute missing values.

A.1. General advices on how to use the **convergEU** R package

The analysis of convergence within the **convergEU** package is performed on clean and imputed data, i.e. a tidy dataset in the format years by countries, thus the dataset on which the analysis is performed must always have the following features:

- its a tibble object, see for instance [Wickham and Grolemund \(2017\)](#) and the references therein; objects in other formats should be transformed in tibbles;
- qualitative variables are not allowed in the dataset, nor vectors of strings;
- missing values are not allowed;
- a time variable has to be present and rows of the dataset are ordered by time from oldest to the more recent;

The `check_data` function is specifically implemented to check for the presence of these unsuitable features that must be addressed before starting the analysis. The output of the `check_data` function, as well as of most functions of this package, is a list of the following three components: `$res`, `$msg` and `$err`. The `$res` of `check_data` returns `TRUE` if the dataset is ready for calculations, and `NULL` if it is not, while the `$msg` component possibly displays messages for the user; the `$err` component contains the following messages to explain the reasons why the dataset did not pass these checks:

- “Error: one or more missing values in the dataframe.”
- “Error: qualitative variables in the dataframe.”
- “Error: string variables in the dataframe.”
- “Error: timeName variable absent.”
- “Error: time variable is not ordered.”

Examples on how to transform the data to achieve the required format are described in the following Subsections.

The **convergEU** package works with different sets of Member States, and they are all available by invoking the `convergEU_glb` function. More precisely, this function generates a vector which contains several collections of countries, in addition to some other constants and auxiliary information.

```
library(convergEU)
names(convergEU_glb())

## [1] "EUcodes"      "Eurozone"
## [3] "EA"           "EU12"
## [5] "EU15"         "EU19"
## [7] "EU25"         "EU27_2007"
## [9] "EU27_2019"   "EU27_2020"
## [11] "EU27"         "EU28"
## [13] "geoRefEUF"    "metaEUStat"
## [15] "tmp1_out"     "paralintags"
## [17] "rounDigits"   "epsilonV"
## [19] "scoreBoaTB"   "labels_clusters"
```

The objects "Eurozone", "EA", "EU12", "EU15", "EU19", "EU25", "EU27", "EU27_2007", "EU27_2019", "EU27_2020" and "EU28" refer to the different aggregations of Member States stored within the `convergEU_glb` function (Member States names and corresponding labels); these country labels correspond to those used by Eurostat (https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:EU_enlargements). It is worth noting that "EU27" refers to Member States after the 1st February 2020, while "EU28" is a valid tag for data collected up to the 3rd March 2020; the "EU27_2020" and "EU27_2007" clusters as defined by Eurostat are also available (details at <https://ec.europa.eu/eurostat/help/faq/brexit>). For convenience, "Eurozone", "EA" and "EU19" are also defined as synonyms.

In order to access a given aggregate of Member States, say for example the set "EU12", the user should invoke an "EU" list component:

```
convergEU_glb()$EU12

## $dates
## [1] "01-11-1993" "31/12/1994"
##
## $memberStates
## # A tibble: 12 x 2
##   MS           codeMS
##   <chr>        <chr>
## 1 Belgium      BE
## 2 Denmark      DK
## 3 France       FR
## 4 Germany      DE
## 5 Greece       EL
## 6 Ireland      IE
## 7 Italy        IT
## 8 Luxembourg   LU
## 9 Netherlands NL
## 10 Portugal    PT
## 11 Spain       ES
## 12 United-Kingdom UK
```

For each available set of Member States within the `convergEU_glb` function, the object `codeMS` contains the corresponding Member States labels. This object is of particular importance, since several functions implemented in this package requires the specification of the Member States labels. However, the `convergEU` package can be also used in the analysis of any kind of regions (i.e. also countries that are not Member States); to this end, the argument `custom` should be used to properly define a selection of countries different from the standard aggregates (e.g. EU12, EU27 etc.), to be provided as string vector.

Finally, within the `convergEU` R package there are four datasets that collect metadata:

```
# data(package = "convergEU")
Data sets in package 'convergEU':

dbEUF2018meta:
Metainformation on Eurofound dataset

dbEurofound:
Eurofound dataset

dbMetaEUStat:
Eurostat metainformation

emp_20_64_MS:
Dataset emp_20_64_MS
```

The first one, `dbEUF2018meta`, contains a description (metadata) of the locally available data within the `convergEU` package, without any downloading from the Eurofound website; see

Subsection A.2 for further details on this dataset. The second locally accessible object in the **convergEU** package is `dbEurofound` which is a raw local Eurofound database. The third object in this package is `dbMetaEUStat` which provides metainformation on downloadable data from the Eurostat repository. Lastly, the `emp_20_64_MS` object is an Eurofound dataset exploited during the development of this package: it refers to the Employment rate indicator from 2002 to 2018 in the age class from 20 to 64. It contains source data downloaded from the Eurostat repository and formatted as a "time by countries" table. Note that in the `emp_20_64_MS` dataset missing values have been already imputed, and it is always ready for the analysis without any further data processing. It is worth noting that within the **convergEU** package Eurofound data are locally (statically) stored, thus the user must update this package to access the latest version of data released by Eurofound.

A.2. Locally accessible Eurofound data and downloadable data from the Eurostat website

Two types of data sources are available within the **convergEU** R package: (i) locally accessible data gathered by Eurofound, and available without any active Internet connection; (ii) Eurostat data that can be downloaded on demand if an Internet connection is active, using functions in this package.

The locally available Eurofound data are accessed through the `extract_indicator_EUF` function. EU indicators for this type of data refer to two dimensions: quality of life and working conditions. For each dimension, metainformation regarding the various indicators is contained in the object `dbEUF2018meta`:

```
names(dbEUF2018meta)

## [1] "DIMENSION"          "SUBDIMENSION"
## [3] "INDICATOR"          "Code_in_database"
## [5] "Official_code"      "Unit"
## [7] "Source_organisation" "Source_reference"
## [9] "Disaggregation"     "Bookmark_URL"
```

(see `help(extract_indicator_EUF)`).

The `extract_indicator_EUF` function requires the specification of the name of an indicator, a time interval, and a set of Member States. An indicator is chosen from those available in the object `dbEUF2018meta$Code_in_database`, and its corresponding code is specified in the argument `indicator_code` as a string. The time interval is specified in the arguments `fromTime` for the initial time, and `toTime` for the final time. The set of Member States is specified in the argument `countries` by choosing one among those provided by the function `convergEU_glb`. Note that Member States labels, rather than names, are specified in the argument `countries`. The argument `gender` also allows to extract data only for males or for females, otherwise both gender types are extracted with the function argument `gender="Total"`.

Suppose, for example, that locally accessible data for the Mean Life Satisfaction indicator has to be extracted; `Code_in_database` from the `dbEUF2018meta` objects is "lifesatisf"; the time interval is 2003-2016; lastly, EU19 is chosen for the set of Member States; the R code to obtain data is:

```
myTblf <- extract_indicator_EUF(
  indicator_code="lifesatisf",#database code
  fromTime=2003,
  toTime=2016,
  gender= c("Total","Females","Males")[1],
  countries=
    convergEU_glb()$EU19$memberStates$codeMS
)
```

myTblf

```
## $res
## # A tibble: 4 x 21
##   time sex      AT      BE      CY      DE      EE
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2003 Total  7.85  7.52  7.22  7.36  5.94
## 2 2007 Total  6.95  7.54  7.05  7.16  6.72
## 3 2011 Total  7.66  7.38  7.16  7.20  6.28
## 4 2016 Total  7.92  7.31  6.54  7.31  6.73
## # ... with 14 more variables: EL <dbl>,
## #   ES <dbl>, FI <dbl>, FR <dbl>, IE <dbl>,
## #   IT <dbl>, LT <dbl>, LU <dbl>, LV <dbl>,
## #   MT <dbl>, NL <dbl>, PT <dbl>, SI <dbl>,
## #   SK <dbl>
##
## $msg
## NULL
##
## $err
## NULL
```

The `res` component in the output contains the dataset in the format "time by countries"; the remaining two components are `msg` that possibly carries messages for the user, and `err` which is a string containing an error message, if an error occurs.

Eurostat is the second data source accessible using the **convergEU** R package. The heterogeneity in the structure of different indicators requires some cares. A list of covariates for each indicator is sometimes present besides age and gender, thus their values must be set to produce a tidy dataset in the format "time by countries". The `download_indicator_EUS` function downloads data from the Eurostat repository using arguments like an indicator name and the Eurostat indicator code, both to be specified as strings. Indicator codes for this function are available in two `convergEU_glb` list components:

- `convergEU_glb()$metaEUStat$selectorUser`
- `convergEU_glb()$metaEUStat$Official_code_purified`

If the argument `rawDump` is `TRUE`, raw data are downloaded and returned in the original (raw) format, which is not a tidy dataset; thus, a tidy dataset is returned in the format "years by countries" only if `rawDump` argument is set to `FALSE`.

To illustrate details on how the `download_indicator_EUS` function works, let's consider the "Employment rate" indicator. The Eurostat code for this indicator is "lfsa_ergaed" and this is the string specified through the argument `indicator_code`. In order to download the data from 1992 to 2018 the arguments `fromTime` and `toTime` are provided; the "EU28" set of Member States is chosen and the corresponding country labels are specified in the argument `countries`; the resulting code is shown below:

```
emprTB <- download_indicator_EUS(
  indicator_code= "lfsa_ergaed",
  fromTime = 1992,
  toTime = 2018,
  countries=
  convergEU_glb()$EU28$memberStates$codeMS,
  rawDump=F)
```

which returns the data in the required format "time by countries". The result is a list with the following components:

- `$res` which contains the selected data as a tibble (dataset);
- `msggender` is a component describing results; `msggender` states that gender was not selected; thus, it was automatically set to both males and females;
- `msgAge` states that age class was automatically set;

- `msgFurther_Conditioning` contains further variables whose value must be set in order to obtain a tidy dataset; in particular `msgFurther_Conditioning$current` states the value that was exploited during the download of the returned data; the set of allowed conditioning values (strings) is contained within the component `$available_seleTagLs`, more precisely `msgFurther_Conditioning$available_seleTagLs`;
- `msgConditioning` contains common conditioning variables, such as `indicator_code`, `ageInterv` and `gender`.

It is therefore possible to call several times the same function and to specify the argument `uniqueIdentif` as an integer among those listed in the first column left of list component `msgFurther_Conditioning$available_seleTagLs` to obtain the same indicator under different scales and contexts. That is, the user is expected to download data, check the `uniqueIdentif` argument and choose the scales and contexts of interest. It is also possible to download data only for females or only for males by specifying it in the argument `gender`. An interval of ages may be also specified, for example the following code allows to download the data in the age interval Y30-54, a time interval from 1992 to 2018 and `uniqueIdentif=5`:

```
emprTB5 <- download_indicator_EUS(
  indicator_code= 'lfsa_ergaed',
  fromTime = 1992,
  toTime = 2018,
  gender='T',
  ageInterv = 'Y30-54',
  countries=
  convergEU_glb()$EU28$memberStates$codeMS,
  rawDump=F,uniqueIdentif=5)
```

where the `ageInterv` argument is specified by a string.

The `down_lo_EUS` function allows to download data directly from the Eurostat website (<https://ec.europa.eu/eurostat/data/database>), if an Internet connection is active: the first argument is the Eurostat code (string) referred to the chosen indicator, and it is available on the Eurostat web site next to each indicator's name within brackets. The time interval is another argument which has to be specified as already described above. The function allows to condition with respect to gender, age and other variables depending on the specific indicator. The special argument `rawDump` set to `TRUE` determines the downloading of raw data, thus the result is not a tidy dataset; to obtain a tidy dataset in the format "years by countries", the argument `rawDump` is set to `FALSE`. All indicators are supported but the data are not filtered by country members (original geo variable) and/or for EU clusters. Therefore, it is up to the user to proceed with further filtering/selection so that the desired collection of countries is obtained.

Suppose, for example, that we choose to download data for the Self-perceived health indicator. The indicator's code provided by the Eurostat website is "hlth_silc_18". For this indicator, we choose to download the data from 2003 to 2018 as follows:

```
myTBheal <- down_lo_EUS(
  indicator_code = "hlth_silc_18",
  fromTime = 2003,
  toTime = 2018,
  gender= "T",
  ageInterv = NA,
  rawDump=F,
  uniqueIdentif = 1)
```

The result is similar to the list created by the `download_indicator_EUS` function:

- `$res` contains the selected data as a tibble;
- `msgAge` states that age class was automatically set;

- `msgFurther_Conditioning` contains further variables to set in order to obtain a tidy dataset; in particular: `msgFurther_Conditioning$current` states the value of conditioning taken from `msgFurther_Conditioning$available_seleTagLs`;
- `msgConditioning` contains common conditioning variables, such as `ageInterv` and `gender`.

Similarly to the `download_indicator_EUS` function, it is therefore possible to call several times the same function and to specify the argument `uniqueIdentif` as an integer among those in the first column left of `msgFurther_Conditioning $available_seleTagLs` in order to obtain the same indicator under different scales and contexts; for example, degree “3” of urbanization and level “fair” is obtained as follows:

```
myTBheal2 <- down_lo_EUS(
  indicator_code = "hlth_silc_18",
  fromTime = 2003,
  toTime = 2018,
  gender= "T",
  ageInterv = NA,
  rawDump=F,
  uniqueIdentif = 17)
```

It is also possible to condition to `gender` in order to download data only for females or only for males, for example:

```
myTBheal3 <- down_lo_EUS(
  indicator_code = "hlth_silc_18",
  fromTime = 2003,
  toTime = 2018,
  gender= "F",
  ageInterv = NA,
  rawDump=F,
  uniqueIdentif = 1)
```

```
myTBheal4 <- down_lo_EUS(
  indicator_code = "hlth_silc_18",
  fromTime = 2003,
  toTime = 2018,
  gender= "M",
  ageInterv = NA,
  rawDump=F,
  uniqueIdentif = 1)
```

Specific intervals among those available for the Self-perceived health indicator may be specified:

```
unique(myTBheal$msg$Conditioning$ageInterv)

## [1] "Y16-24"
```

To download data in the age interval Y35-44 and time interval 2003-2018, arguments of this function are:

```
myTBheal5 <- down_lo_EUS(
  indicator_code = 'hlth_silc_18',
  fromTime = 2003,
  toTime = 2018,
  gender= 'T',
  ageInterv = 'Y35-44',
  rawDump=F,
  uniqueIdentif = 1)
```

Data downloaded through the functions `download_indicator_EUS` and `down_lo_EUS` are not filtered by EU aggregation of Member States. Therefore, some further data processing is needed if the user wishes to obtain the data only for a given collection of countries.

Users exploiting automated access to the database through web services or bulk download should pay particular attention to changes of labels. Further information about EU aggregates is available at <https://ec.europa.eu/eurostat/help/faq/brexit>.

The last function of the **convergEU** package suited to download data from Eurostat is `dow_soc_scor_boa` (see `help(dow_soc_scor_boa)`). This function is specifically implemented for downloading all the indicators involved in the social scoreboard from Eurostat. Thus, the user should pay particular attention to the speed of the Internet connection before invoking it, given that a large quantity of data is downloaded. For example, to download all the indicators involved in the social scoreboard from 2007 to 2017, the syntax is:

```
socInd<-dow_soc_scor_boa(fromTime= 2007, toTime= 2017)
```

In this example, 37 social scoreboard indicators are downloaded and stored in the user's local repository. The user may select one of these 37 indicators in the subsequent analysis, say for example the first one, e.g. *I.01.01.00*:

```
socInd1<-socInd$I.01.01.00
```

The `socInd1` object is a list whose components are similar to those of the functions `download_indicator_EUS` and `down_lo_EUS`.

Lastly, it is noteworthy to notice that the user could download data directly from the Eurostat website in some external source file format, like `.csv`, `.xls` and `.xlsx` formats. The user may refer to the standard procedures available in several R packages to import the data with this type of file formats, such as **readxl** (Wickham and Bryan 2019) and **readr** (Wickham, Hester, and Francois 2018) packages. Furthermore, examples and further details on this procedure are available in a specific **convergEU** vignette (Stefanini, Nikiforova, Litardi, Peruffo, and Mascherini 2020b).

A.3. Imputation of missing values

As already stated in Subsection 3, the analysis of convergence is performed on clean and imputed data without missing values: a tidy dataset in the format "years by countries". Typical EU indicators cover a time span of not many years, thus 10% of missing values within a Member State may already represent a context that requires substantive reasoning before imputing missing values: the user may or may not proceed in the analysis, depending on the considered context.

In the **convergEU** package the function called `impute_dataset` (see `help(impute_dataset)`) performs this task: two options are available for imputation of missing values located at the first rows of the dataset (also called `tail` missing values) or at the end (also called `head` missing values), that is:

- they can be completely deleted, using the `cut` option in the arguments `tailMiss` (missing values starting at the oldest year) and `headMiss` (missing values in the most recent subsequent years);
- they could be 'cloned', that is copied from the first observed-available year by setting the `constant` option as argument in `tailMiss` and `headMiss`.

For all other missing values within the table body of a Member State, a deterministic linear imputation is performed to obtain the complete dataset. The imputation of missing values in the **convergEU** package requires at least two non-missing values for a given Member State: if values describing a Member State are completely missing, the imputation is not performed.

The examples below help to better illustrate the missing values imputation procedure within the package. In the first example, the dataset `emprTB5` related to the Employment rate

indicator is considered for both males and females, with age interval "Y30-54" (Subsection A.2). First, we assign a meaningful name to the dataset and the presence of unsuitable features is checked by a call to the `check_data` function:

```
empTBFin<-emprTB5$res
check_data(empTBFin, timeName = "time")

## $res
## NULL
##
## $msg
## NULL
##
## $err
## [1] "Error: one or more missing values in the dataframe."
```

The `check_data` function reveals the presence of missing values. Thus, we impute the starting missing values in the `empTBFin` dataset choosing the `constant` option:

```
EmpImp <- impute_dataset(
  empTBFin,
  timeName = "time",
  countries=
    convergEU_glb()$EU28$memberStates$codeMS,
  tailMiss = c("cut", "constant")[2],
  headMiss = c("cut", "constant")[2])$res
```

where the option `cut` could be selected instead:

```
EmpImp1 <- impute_dataset(
  empTBFin, timeName = "time",
  countries=
    convergEU_glb()$EU28$memberStates$codeMS,
  tailMiss = c("cut", "constant")[1],
  headMiss = c("cut", "constant")[1])$res
head(EmpImp1)

## # A tibble: 6 x 31
##   age sex time BE DK FR DE
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Y30-54 T 2002 76.3 85.5 79.9 79.5
## 2 Y30-54 T 2003 75.8 84.7 81.2 79
## 3 Y30-54 T 2004 77.2 85 80.7 78.2
## 4 Y30-54 T 2005 78.2 85.6 81.4 78.6
## 5 Y30-54 T 2006 78.3 86.9 81.8 79.9
## 6 Y30-54 T 2007 79.7 86.8 82.8 81.4
## # ... with 24 more variables: EL <dbl>,
## # IE <dbl>, IT <dbl>, LU <dbl>, NL <dbl>,
## # PT <dbl>, ES <dbl>, UK <dbl>, AT <dbl>,
## # FI <dbl>, SE <dbl>, CY <dbl>, CZ <dbl>,
## # EE <dbl>, HU <dbl>, LV <dbl>, LT <dbl>,
## # MT <dbl>, PL <dbl>, SK <dbl>, SI <dbl>,
## # BG <dbl>, RO <dbl>, HR <dbl>
```

so that all countries from 1992 to 2001 are deleted. This is due to the fact that a Member State has initial missing values from year 1992 to year 2001. If a Member State is processed but it has no missing values then no change takes place. Once the missing values are imputed, the `check_data` function is called again on the imputed data:

```
check_data(EmpImp)

## $res
## NULL
##
## $msg
## NULL
##
## $err
## [1] "Error: string variables in the dataframe."
```

In this case, the check fails due to the presence of the string variables *age* and *sex* that should be deleted as follows:

```
EmpImpf <- dplyr::select(EmpImpf,-sex,-age)
check_data(EmpImpf)

## $res
## [1] TRUE
##
## $msg
## NULL
##
## $err
## NULL
```

Now, the `check_data` function confirms that `EmpImpf` is ready for the analysis of convergence, that is the `$res` component is `TRUE`.

In the second example, the indicator "Self-perceived health" is considered for both males and females between thirty five and forty four years, and stored in the `myTBheal5` dataset. After selecting the "EU28" aggregation, the `check_data` function is invoked:

```
TBheal<-myTBheal5$res
EU28estr<-convergEU_glb()$EU28$memberStates$codeMS
TBhealth<- dplyr::select(TBheal,time,all_of(EU28estr))
check_data(TBhealth, timeName = "time")

## $res
## NULL
##
## $msg
## NULL
##
## $err
## [1] "Error: one or more missing values in the dataframe."
```

and it reveals the presence of missing values. Given that in the `TBhealth` dataset both types of missing values, in starting and ending years, are present then both arguments `tailMiss` and `headMiss` have to be specified:

```
TBhealthF <- impute_dataset(TBhealth,
  timeName = "time",
  countries=
    convergEU_glb()$EU28$memberStates$codeMS,
  tailMiss = c("cut", "constant")[2],
  headMiss = c("cut", "constant")[1])$res
```

Finally, a call to the `check_data` function on the imputed dataset confirms that the dataset `TBhealthF` is ready for the analysis:

```
check_data(TBhealthF, timeName = "time")

## $res
## [1] TRUE
##
## $msg
## NULL
##
## $err
## NULL
```

B. Appendix II: auxiliary functions

In the `convergEU` package, several auxiliary functions are implemented to further enrich the analysis of convergence. These functions are described in details in the following Sections.

B.1. Absolute changes across countries

The absolute change for Member State m , indicator i at time t is defined as:

$$\Delta y_{m,i,t} = y_{m,i,t} - y_{m,i,t-1} \quad (14)$$

The `abso_change` function implements the above calculations by processing a complete and sorted dataset in the format "time by countries". The dataset `emp_20_64_MS` contained in the `convergEU` package and described in Subsection A.2 can be used for illustrating how to use it. Absolute changes over the 28 Member States of EU28 below are calculated by considering the reference year 2007 (argument `time_0`) and the target year 2016 (argument `time_t`):

```
myAC <- abso_change(emp_20_64_MS,
                    time_0 = 2007,
                    time_t = 2016,
                    all_within=TRUE,
                    timeName = "time")
names(myAC$res)

## [1] "abso_change"      "sum_abs_change"
## [3] "average_abs_change"
```

The resulting object `myAC` consists of:

1. (i) the list `myACresabso_change` of the absolute changes for each Member State;
2. (ii) a list of the sum of absolute values (`myACressum_abs_change`), defined in general as $\sum_{t=t_0+1} |\Delta y_{m,i,t}|$;
3. (iii) a list of the average per pairs of years, i.e. the sum of absolute values is divided by the number of pair of years, e.g. `myACresaverage_abs_change`.

B.2. Summaries of clusters of countries

An important summary is obtained as unweighted average of country values through the `average_clust` function. For example, the unweighted average in the `emp_20_64_MS` dataset for the EU28 set of Member States is computed as follows (Figure 13):

```
wwTB <- average_clust(emp_20_64_MS,
                     timeName = "time",
                     cluster = "EU28")$res[,c(1,30)]
```

Time series can be plotted with the `ggplot2` `qplot` function:

```
mini_EU <- min(wwTB$EU28)
maxi_EU <- max(wwTB$EU28)

tmp <- qplot(time, EU28, data=wwTB,
            ylim=c(mini_EU,maxi_EU))+geom_line()+
      ylab("Employment rate indicator: EU28 unweighted average")+xlab("")+theme_classic()
```

The resulting plot shows that there is an increase of the EU28 average of the Employment rate indicator from 2002 to 2008. Later, with the advent of the 2008 crisis, the EU28 average for this indicator decreases from 2008 to 2013, and then it begins to rise again.

B.3. Departures from the average

The `departure_mean` function has been implemented to calculate departures of indicator values from the EU average. Actual calculations are based on the function implemented for the sigma-convergence. More precisely, if the `emp_20_64_MS` dataset is considered, first the sigma-convergence is calculated:

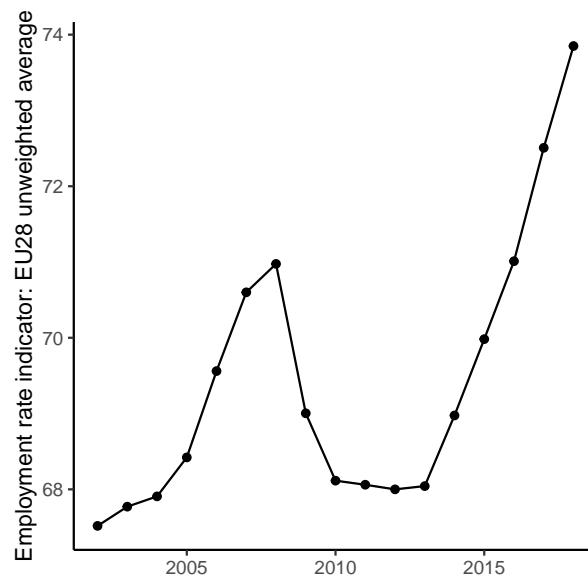


Figure 13: Unweighted average of the employment rate for the cluster of countries EU28

```
mySTB <- sigma_conv(emp_20_64_MS,timeName="time")
```

The departure from the mean can be characterized as follows:

```
res <- departure_mean(oriTB = emp_20_64_MS,
                     sigmaTB = mySTB$res)
names(res$res)

## [1] "departures"      "squaredContrib"
## [3] "devianceContrib"

res$res$departures

## # A tibble: 17 x 33
##   time stdDev    CV mean devianceT   AT
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2002  6.34 0.0939 67.5  1125.  0
## 2 2003  5.95 0.0878 67.8   991.  0
## 3 2004  5.70 0.0839 67.9   909.  0
## 4 2005  5.54 0.0809 68.4   858.  0
## 5 2006  5.57 0.0801 69.6   869.  0
## 6 2007  5.47 0.0775 70.6   838.  0
## 7 2008  5.36 0.0755 71.0   804.  0
## 8 2009  5.03 0.0730 69.0   710.  0
## 9 2010  5.24 0.0769 68.1   768.  1
## 10 2011  5.59 0.0821 68.1   875.  1
## 11 2012  5.98 0.0880 68    1002.  1
## 12 2013  6.28 0.0922 68.0  1103.  1
## 13 2014  5.98 0.0867 69.0  1000.  0
## 14 2015  5.74 0.0820 70.0   922.  0
## 15 2016  5.60 0.0789 71.0   879.  0
## 16 2017  5.37 0.0741 72.5   808.  0
## 17 2018  5.30 0.0717 73.8   786.  0
## # ... with 27 more variables: BE <dbl>,
## # BG <dbl>, CY <dbl>, CZ <dbl>, DE <dbl>,
## # DK <dbl>, EE <dbl>, EL <dbl>, ES <dbl>,
## # FI <dbl>, FR <dbl>, HR <dbl>, HU <dbl>,
## # IE <dbl>, IT <dbl>, LT <dbl>, LU <dbl>,
## # LV <dbl>, MT <dbl>, NL <dbl>, PL <dbl>,
## # PT <dbl>, RO <dbl>, SE <dbl>, SI <dbl>,
## # SK <dbl>, UK <dbl>
```

where $(-1, 0, 1)$ indicates values respectively below -1 , within the interval $(-1, 1)$ and above $+1$. Details on the contribution of each MS to the variance at a given time t is evaluated by

the square of the difference $(y_{m,i,t} - \bar{y}_{EU28,i,t})^2$ between the indicator i of country m at time t and the unweighted average over Member States, say EU28:

```
res$res$squaredContrib

## # A tibble: 17 x 28
##   AT      BE      BG      CY      CZ      DE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 11.4   7.94 1.21e+2 57.5   17.5 1.64e+0
## 2 12.5   10.7 8.23e+1 58.2   10.4 3.95e-1
## 3 0.243  4.44 4.50e+1 60.7    4.81 5.10e-5
## 4 3.91   3.69 4.25e+1 35.7    5.19 9.58e-1
## 5 4.16   9.37 1.99e+1 38.9    2.69 2.37e+0
## 6 4.84   8.41 4.84e+0 38.4    1.96 5.29e+0
## 7 7.98   8.85 7.56e-2 30.5    2.03 9.15e+0
## 8 19.3   3.62 4.14e-2 39.6    3.60 2.70e+1
## 9 33.5   0.264 1.17e+1 47.4    5.22 4.74e+1
## 10 37.7   0.579 2.66e+1 28.5    8.06 7.12e+1
## 11 41.0   0.640 2.50e+1 4.84   12.2 7.92e+1
## 12 43.0   0.710 2.06e+1 0.710  19.9 8.57e+1
## 13 27.3   2.81  1.50e+1 1.89   20.5 7.61e+1
## 14 18.6   7.74  8.31e+0 4.34   23.2 6.43e+1
## 15 14.4   11.0  1.10e+1 5.34   32.4 5.76e+1
## 16 8.37   16.1  1.46e+0 2.91   35.9 4.48e+1
## 17 5.52   17.2  2.10e+0 0.0025 36.6 3.66e+1
## # ... with 22 more variables: DK <dbl>,
## # EE <dbl>, EL <dbl>, ES <dbl>, FI <dbl>,
## # FR <dbl>, HR <dbl>, HU <dbl>, IE <dbl>,
## # IT <dbl>, LT <dbl>, LU <dbl>, LV <dbl>,
## # MT <dbl>, NL <dbl>, PL <dbl>, PT <dbl>,
## # RO <dbl>, SE <dbl>, SI <dbl>, SK <dbl>,
## # UK <dbl>
```

The numerator of the variance, called deviance, can be decomposed for each time point in order to appreciate the percentage of contribution provided by each Member State to the total deviance,

$$100 \cdot \frac{(y_{m,i,t} - \bar{y}_{EU28,i,t})^2}{\sum_s (y_{s,i,t} - \bar{y}_{EU28,i,t})^2}$$

where the Member State of interest is m , and where s enumerates Member States in the considered aggregation:

```
res$res$devianceContrib

## # A tibble: 17 x 28
##   AT      BE      BG      CY      CZ      DE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1.02   0.706 1.08e+1 5.11e+0 1.56 1.46e-1
## 2 1.26   1.08  8.30e+0 5.87e+0 1.05 3.99e-2
## 3 0.0267 0.488 4.95e+0 6.68e+0 0.529 5.61e-6
## 4 0.456  0.430 4.95e+0 4.16e+0 0.605 1.12e-1
## 5 0.479  1.08  2.29e+0 4.48e+0 0.309 2.73e-1
## 6 0.578  1.00  5.78e-1 4.59e+0 0.234 6.31e-1
## 7 0.993  1.10  9.41e-3 3.80e+0 0.253 1.14e+0
## 8 2.72   0.511 5.84e-3 5.59e+0 0.507 3.81e+0
## 9 4.36   0.0344 1.52e+0 6.17e+0 0.680 6.17e+0
## 10 4.31   0.0662 3.04e+0 3.26e+0 0.922 8.14e+0
## 11 4.09   0.0639 2.50e+0 4.83e-1 1.22 7.91e+0
## 12 3.90   0.0644 1.87e+0 6.44e-2 1.80 7.77e+0
## 13 2.73   0.280 1.50e+0 1.89e-1 2.05 7.61e+0
## 14 2.02   0.839 9.01e-1 4.70e-1 2.52 6.97e+0
## 15 1.63   1.25  1.25e+0 6.08e-1 3.68 6.55e+0
## 16 1.04   1.99  1.80e-1 3.60e-1 4.44 5.54e+0
## 17 0.703  2.19  2.68e-1 3.18e-4 4.66 4.66e+0
## # ... with 22 more variables: DK <dbl>,
## # EE <dbl>, EL <dbl>, ES <dbl>, FI <dbl>,
## # FR <dbl>, HR <dbl>, HU <dbl>, IE <dbl>,
## # IT <dbl>, LT <dbl>, LU <dbl>, LV <dbl>,
## # MT <dbl>, NL <dbl>, PL <dbl>, PT <dbl>,
## # RO <dbl>, SE <dbl>, SI <dbl>, SK <dbl>,
## # UK <dbl>
```

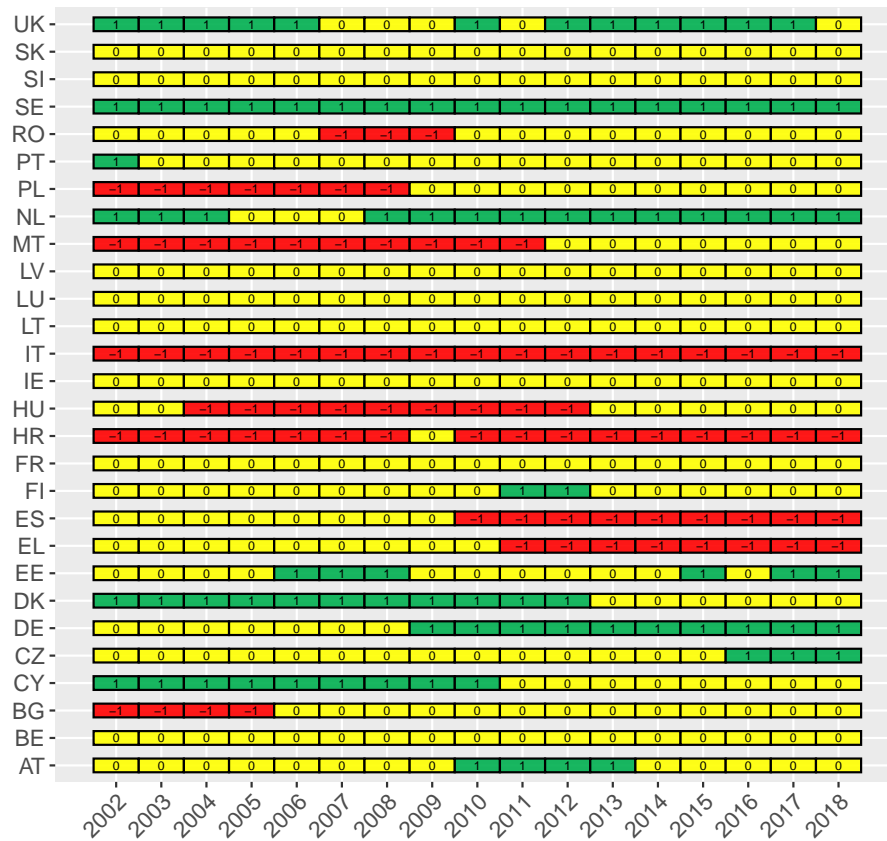


Figure 14: Qualitative features of time series.

It follows that each row adds to 100. Moreover, it is possible to produce a graphical output about the main features of country time series, using the `graph_departure` function, for example (Figure 14):

```
myGD <- graph_departure(res$res$departures,
  timeName = "time",
  displace = 0.25,
  displaceh = 0.5,
  dimeFontNum = 2,
  myfont_scale = 1.25,
  x_angle = 45,
  color_rect = c("-1"='red1',
                 "0"="#ffff00",
                 "1"="#00b050"),
  axis_name_y = "",
  axis_name_x = "",
  alpha_color = 0.9
)
# myGD
```

B.4. Countries ranking

Country ranking at a given year is implemented by the `country_ranking` function (see `help(country_ranking)`). The argument `typeInd` sets the type of indicator, `highBest` or `lowBest`. In the `emp_20_64_MS` dataset, the ranking from 2005 to 2010 is obtained as follows:

```
data(emp_20_64_MS)
myCR<-country_ranking(emp_20_64_MS,
  timeName = "time",
  time_0 = 2005,
  time_t = 2010,
  typeInd= "highBest")
```

```
myCR

## $res
## # A tibble: 6 x 29
##   time   AT   BE   BG   CY   CZ   DE
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2005    13    19    24    4    11    14
## 2 2006    11    19    22    4    14    15
## 3 2007    11    20    19    4    15    10
## 4 2008    10    21    16    5    14    9
## 5 2009     8    17    15    4    11    5
## 6 2010     6    14    18    3    10    3
## # ... with 22 more variables: DK <dbl>,
## #   EE <dbl>, EL <dbl>, ES <dbl>, FI <dbl>,
## #   FR <dbl>, HR <dbl>, HU <dbl>, IE <dbl>,
## #   IT <dbl>, LT <dbl>, LU <dbl>, LV <dbl>,
## #   MT <dbl>, NL <dbl>, PL <dbl>, PT <dbl>,
## #   RO <dbl>, SE <dbl>, SI <dbl>, SK <dbl>,
## #   UK <dbl>
##
## $msg
## NULL
##
## $err
## NULL
```

B.5. Departures from the best performing country

The `departure_best` function calculates the departure of each Member State from the best performing Member State, thus for the `emp_20_64_MS` dataset the function may be invoked as follows:

```
mySTB2 <- departure_best(emp_20_64_MS, timeName="time",
                        indiType = "highBest")
```

where the `$res` component of `mySTB2` is a list of the departures from the best performer (component `resraw_departures`), and of the cumulated differences over years (`rescumulated_dif`).

A plot of the deviations from the best performer is produced by the `departure_best_plot` function, for example the plot of deviations for Italy, considered as country of interest, with Germany, France and Sweden as further countries to plot is obtained as follows (Figure 15):

```
depa_res <- departure_best_plot(mySTB2$res$cumulated_dif,
                              mainCountry = "IT",
                              countries=c("DE", "FR", "SE"),
                              displace = 0.25,
                              axis_name_y = "",
                              val_alpha = 0.95,
                              debug=FALSE)
```

with `mainCountry` argument representing the country of interest and the `countries` argument to choose the set of countries with respect to which the deviations are plotted.

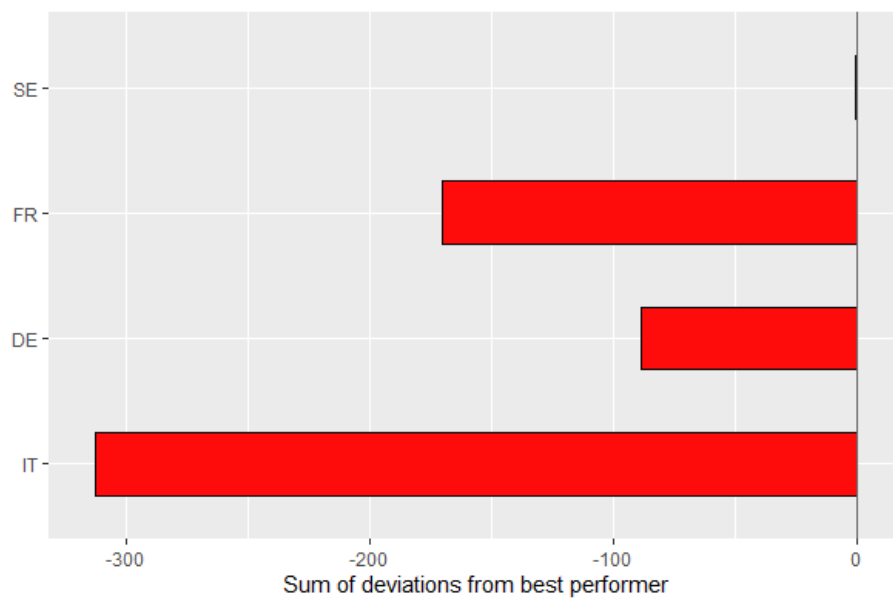


Figure 15: Departures from best performers

Affiliation:

Federico M. Stefanini
Department of Environmental Science and Policy
University of Milan
via Celoria, 10, Milan, Italy
E-mail: federico.stefanini@unimi.it
URL: <https://www.unimi.it/it/ugov/person/federico-stefanini>

Nedka D. Nikiforova
Department of Statistics, Computer Science, Applications "G. Parenti"
University of Florence
viale Morgagni, 59, Florence, Italy
E-mail: n.nikiforova@unifi.it

Eleonora Peruffo
Eurofound
Wyattville Road, Loughlinstown, Co. Dublin, D18 KP65, Ireland
<https://www.eurofound.europa.eu/>

Martina Bisello
Eurofound
Wyattville Road, Loughlinstown, Co. Dublin, D18 KP65, Ireland
<https://www.eurofound.europa.eu/>

Chiara Litardi
Eurofound
Wyattville Road, Loughlinstown, Co. Dublin, D18 KP65, Ireland
<https://www.eurofound.europa.eu/>

Massimiliano Mascherini
Eurofound
Wyattville Road, Loughlinstown, Co. Dublin, D18 KP65, Ireland
<https://www.eurofound.europa.eu/>