WORKING PAPER

Monitoring convergence in the European Union

Monitoring upward convergence in the EU with R: The convergEU package

Upward convergence in the EU:
Concepts, measurements and indicators

Authors: Federico M. Stefanini, Nedka D. Nikiforova, Chiara Litardi, Martina Bisello, Eleonora Peruffo, Massimiliano Mascherini

Research Manager: Massimiliano Mascherini

Eurofound reference number: WPEF20008

The European Foundation for the Improvement of Living and Working Conditions (Eurofound) is a tripartite European Union Agency established in 1975. Its role is to provide knowledge in the area of social, employment and work-related policies according to Regulation (EU) 2019/127.

European Foundation for the Improvement of Living and Working Conditions

Telephone: (+353 1) 204 31 00

Email: information@eurofound.europa.eu

Web: www.eurofound.europa.eu

Contents

1. Introduction

Upward convergence in the economic and social dimensions has always been a 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.

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 structure 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 et al., 2018) has provided a set of statistical tools in order to monitor upward convergence for a given indicator through various convergence measures, each of which with different policy meaning. This Eurofound toolbox has been firstly implemented as set of Stata script and is available here https://www.eurofound.europa.eu/data/convergence-hub/convergence-methodology.

In this work, the Eurofound statistical toolbox for monitoring upward convergence has been implemented in the ConvergEUpackage. This work includes the four main measures of convergence, which have been implemented in R, but the **convergEU** R package also contains supporting functions to filter, impute, smooth indicators and to prepare country and indicator fiches in Html format. Changes of an indicator along time may be mapped to qualitative labels and shown in a graphical format for ease of interpretation. Qualitative patterns of change along time can be defined for indicators with respect to the EU average, and they are grouped into labelled classes to emphasize the underlying pattern. To the best of our knowledge, at the moment no other software package allows to automatically produce such a type of measures in the analyses of convergence.

The manuscript is organized as follows. In Section 2, we describe the main measures of convergence. The core functions of the package are illustrated in Section 3 through several examples. Section 4 is dedicated on finding patterns of convergence and divergence within the **convergEU** package. Other auxiliary functions developed in the package are presented in Section 5. Discussion and final remarks end the manuscript.

2. Measuring upward 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 orientation of the indicator towards, for example, better living and working conditions. For example, in the case of employment rates, upward convergence is observed when the indicator increases and disparities among countries decline, while for unemployment rates, upward convergence is observed when the indicator falls and disparities among countries decrease.

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 is adopted in order to give 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 overrepresented 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. Upward convergence, or upward convergence in the weak sense, occurs if only the EU average records an improvement in its performance, while reducing its disparities. Here, in this case while the EU average is improving, not every Member State records an improvement and some of them may be at risk of being 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 three statistical measures: Beta, Sigma and Delta convergence. The three measures investigating similar, but different aspects of the convergence process.

2.1 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 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 (Signorelli, 2005) analyses convergence for employment rates. The computation of beta-convergence is performed according to the following linear regression model:

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

where m represent a country, i refers to an indicator of interest, t is the reference time, and $\tau \in \{1,2,\dots\}$ is the length of the time window, which is typically equal to 1 or more years.

Despite that 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 analysed in practice. The current implementation of the beta-convergence function in the **convergEU** package is able to make calculations for several target years, but it always maintains the same reference t. The left-hand side of equation (1) is divided by the amount of time elapsed as a default option:

$$\tau^{-1}\{ln(y_{m,i,t+\tau}) - ln(y_{m,i,t})\} = \beta_0 + \beta_1 ln(y_{m,i,t}) + \epsilon_{m,i,t} \quad (2)$$

The computation of beta-convergence, formulas (1) and (2), 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 beta-convergence exists; otherwise, if $\hat{\beta}_1 > 0$, then there is an evidence of divergence.

2.2 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 $y_{m,i,t}$ be the value of an indicator i for member state m at time t. Moreover, let $y_{i,t}^{(M)}$ be the maximum value over member states in the reference set C (for example $C = EU28$), that is:

$$y_{i,t}^{(M)} = max(\{y_{m,i,t}: m \in C\}). \quad (3)$$

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

$$y_{i,t}^{(M)} - y_{m,i,t} \quad (4)$$

Thus, the overall distance at time t, called delta, is the sum of distances over the reference set C of member states:

$$\delta_{i,t} = \sum_{m \in C} (y_{i,t}^{(M)} - y_{m,i,t}) \quad (5)$$

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 C are diverging.

2.3 Gamma-convergence

We now introduce gamma convergence, which is an index based on ranks.

Let $y_{m,i,t}$ be the value of indicator i for member state m at time $t = 0,1, \dots, T$, and $\{\tilde{y}_{m,i,t}: m \in C)$ the ranks for indicator i over member states in the reference set C (for example $C = EU27$) at a given time t. The sum of ranks within member state m is:

$$\tilde{y}_{m,i}^{(c)} = \sum_{t=0}^{T} \tilde{y}_{m,i,t}. \tag{6}$$

Thus, the variance of the sum of ranks over the given interval, defined as $Var[\{\tilde{y}_{m,i}^{(c)}: m \in C\}]$ may be compared to $Var[\{\tilde{y}_{m,i,0}: m \in C\}]$, the variance of ranks in the reference time $t = 0$. The Kendall Index of rank concordance of member states with respect to C for the indicator i over a given time interval T is defined as:

$$KI(C, i, T) = \frac{Var[\{\tilde{y}_{m,i}^{(c)}: m \in C\}]}{(T+1)^2 Var[\{\tilde{y}_{m,i,0}: m \in C\}]}. \tag{7}$$

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 interval T, while a high index value indicates that few changes have been observed.

2.4 Sigma-convergence

The concept of sigma-convergence refers to a reduction in disparities between statistical observations, in this case Member States, over time. The decrease in a function of variability over time indicates that a variable is becoming increasingly homogeneous cross-nationally.

Let $y_{m,i,t}$ be the value of indicator i for member state m at time t, and let $\overline{y}_{C,i,t}$ be the average over the set C, for example $C = EU28$, that is:

$$\overline{y}_{C,i,t} = n(C)^{-1} \sum_{m \in C} y_{m,i,t} \tag{8}$$

where $n(C)$ is the number of member states within set C. The reduction in heterogeneity (possibly disparity) is typically assessed using two statistical summaries:

- the standard deviation

$$s_{C,i,t} = \sqrt{\left(n(C)^{-1} \sum_{m \in C} (y_{m,i,t} - \overline{y}_{C,i,t})^2\right)}\,; \tag{9}$$

and

- the coefficient of variation

$$CV(C, i, t) = 100 \cdot \frac{s_{C,i,t}}{\overline{y}_{C,i,t}}. \tag{10}$$

Therefore, for each year, 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 indicates that member states belonging to the set C are diverging. It must be noted that the coefficient of variation (equation (10)) rescale the standard deviation in formula (9) according to the mean, so that the same amount of variability becomes more and more negligible as the mean increases.

3. Working with the **convergEU** R package

The **convergEU** R package is a set of S3 functions and of data objects suited for the analysis of economic and social convergence of member states within the EU. In this Section we illustrate through several examples details on how to calculate convergence through the core functions implemented in the **convergEU** R package.

3.1 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’. This means that the dataset on which the analysis is performed must always have the following features:

- the dataset should be a tibble object, see for instance (Wickham and Grolemund, 2017) and the references therein; thus, dataframes should be converted into 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 more recent;

In the **convergEU** package, 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 is a list of the following three components: `$res` that returns TRUE if the dataset is ready for calculations, and `NULL` if it is not; `$msg` that possibly displays messages for the user, and `$err` which is the error component stating the reasons why the dataset did not pass these checks. The error component may be one in the following list:

- ‘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.’

Otherwise, the `$err` components takes value `NULL`. Examples on how to transform the data to achieve the required format are described in the following Subsections.

The **convergEU** package allows to work with different sets of member states, and they are all available by invoking the `convergEU_glb()` function. More precisely, this function generates a list which contains several collections of countries, in addition to some other constants and auxiliary information.

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

##  [1] "EUcodes"     "Eurozone"    "EA"          "EU12"
##  [5] "EU15"        "EU19"        "EU25"        "EU27_2007"
##  [9] "EU27_2019"   "EU27_2020"   "EU27"        "EU28"
## [13] "geoRefEUF"   "metaEUStat"  "tmpl_out"    "paralintags"
## [17] "rounDigits"  "epsilonV"    "scoreBoaTB"
```

The objects "Eurozone", "EA", "EU12", "EU15", "EU19", "EU25", "EU27", "EU27_2007", "EU27_2019", "EU27_2020" and "EU28" refer to the different sets 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 must be noted that "EU27" refers to Member States after 1 February 2020, while "EU28" is a valid tag for data collected up to 3 March 2020. The clusters "EU27_2020" and "EU27_2007" as defined by Eurostat are also available; for further details see https://ec.europa.eu/eurostat/help/faq/brexit.

For example, the set "EU12" refers to the first twelve member states. It must be noted that "Eurozone", "EA" and "EU19" are synonyms. To access a given set of member states, say for example the set "EU12", the user should invoke:

```
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, it must be also noted that the **convergEU** package can be used also for the analysis of any kind of dataset where convergence can be measured, for example regions or extra-EU countries; to this end, the argument custom should be used to properly define a selection of countries different from the standard aggregates (for example EU12, EU19 and so on)

The data available within the convergEU package refer to indicators of the EU. A given EU indicator could be of type "lowBest", so that the best possible situation is when the indicator takes the lowest value; for example, the minimisation of the Poverty rate is a positive result. Otherwise, a given indicator could be of type "highBest", so that the best situation is reached when the indicator reaches the highest value (maximisation), for example in the case of the Employment rate.

Finally, within the **convergEU** package there are four available datasets:

```
data(package = "convergEU")
```

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 3.2 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. The last object is the `emp_20_64_MS` dataset exploited during the development of this package. This dataset refer to the Employment rate of the lfsa_ergan Eurostat indicator from 2002 to 2018 and within the age class 20 to 64. It contains source data downloaded from the Eurostat repository which are in the required 'time by countries' format. Note that in the `emp_20_64_MS` dataset there are no missing values, and it is always ready for the analysis without any further data processing. Should you wish to ge the latest employment rate data, please use the download function for the lfsa_ergan indicator (section 3.2).

It is important to note that within the **convergEU** package, Eurofound data are locally (statically) stored. Thus, the user should keep this package updated to have the most recent version of Eurofound data.

3.2 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 which are Eurofound's surveys data, and available without any active Internet connection, and 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 extracted through the `extract_indicator_EUF()` function. EU indicators for this type of data refer to two dimensions: quality of life (EQLS survey) and working conditions (EWCS survey). For each dimension, metainformation is provided for the various indicators in the object `dbEUF2018meta():`

```
names(dbEUF2018meta)

##  [1] "DIMENSION"           "SUBDIMENSION"        "INDICATOR"
##  [4] "Code_in_database"    "Official_code"       "Unit"
##  [7] "Source_organisation" "Source_reference"    "Disaggregation"
## [10] "Bookmark_URL"
```

And van be visualised by invoking:

```
View(dbEUF2018meta)
```

Further details on the `extract_indicator_EUF()` function are provided in the documentation help:

```
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 by setting gender ="Total".

As an example, suppose that we choose to extract locally accessible data for the Mean Life Satisfaction indicator for which the Code_in_database from the dbEUF2018meta objects is "lifesatisf". As a time interval, we choose the period 2003-2016; lastly, the "EU19" is chosen for the set of member states. Thus, the syntax is as follows:

```
myTBlf <- extract_indicator_EUF(
  indicator_code = "lifesatisf", #Code_in_database
  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    EL    ES    FI    FR
##   <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  2003 Total  7.85  7.52  7.22  7.36  5.94  6.69  7.49  8.14  6.96
## 2  2007 Total  6.95  7.54  7.05  7.16  6.72  6.58  7.25  8.23  7.32
## 3  2011 Total  7.66  7.38  7.16  7.20  6.28  6.16  7.47  8.08  7.23
## 4  2016 Total  7.92  7.31  6.54  7.31  6.73  5.26  6.95  8.07  7.17
## # ... with 10 more variables: 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 output is a list where the first component is named $res. It 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. As already stated, the function takes as argument gender, thus in this example for the "lifesatisf" indicator, the data are extracted for both males and females. Otherwise, data of females are extracted by:

```
feTBlf <- extract_indicator_EUF(
  indicator_code = "lifesatisf", #Code_in_database
  fromTime=2003,
  toTime=2016,
```

```
  gender= "Females",
  countries= convergEU_glb()$EU19$memberStates$codeMS)
```

The second source of data available in the **convergEU** R package are downloadable data from Eurostat. The heterogeneity in the structure of different indicators normalized into the database requires some care. 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'.

Two functions are implemented in the **convergEU** R package function for downloading data from Eurostat: the `download_indicator_EUS()` and the `down_lo_EUS()` functions. The `download_indicator_EUS()` function allows to download data from the Eurostat repository:

```
help(download_indicator_EUS)
```

Function's arguments include an indicator name and its Eurostat code to be specified in the argument `indicator_code` as a string. Differently from the `extract_indicator_EUF()` function, the indicators' codes for this function are now available in two `convergEU_glb()` list components:

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

It must be noted that when 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" if the argument `rawDump` is set to `FALSE`.

To illustrate in details how the `download_indicator_EUS()` function works, we take as a n example the "Employment rate" indicator. The Eurostat code for this indicator is 'lfsa_erga ed', to be specified as a string in the argument indicator_code. We choose to download the data from 1992 to 2018; this time period is specified in the arguments 'fromTime' and 'toTim e' respectively. Lastly, the "EU28" set of member states is chosen, and the corresponding co untry labels are specified in the argument countries. Thus, the following arguments are speci fied in the function call: `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' because `rawDump=F`. The result is a list with the following components:

- `$res` which contains the selected data as a tibble (dataset);
- `$msg$gender` is a component describing results; `$msg$gender` states that gender was not selected; thus, it was automatically set to both males and females;
- `$msg$Age` states that age class was automatically set;
- `$msg$Further_Conditioning` contains further variables whose value must be set in or- der to obtain a tidy dataset; in particular `$msg$Further_Conditioning$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 could download the data, check the uniqueIdentif argument and choose the scales and contexts of his/her interest. It is also possible to download data only for females or only for males by specifying it in the argument gender. Moreover, the data could be also downloaded by considering explicitly a specific interval of ages. For example, the following syntax allows to download the data in the age interval Y30-54, for the time interval 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)
```

Note that the ageInterv argument is specified as 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. Details on down_lo_EUS() function are available in the documentation help:

```
help(down_lo_EUS)
```

where 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 next argument is the time interval, as already described above. The function allows to also condition to gender, age and other variables depending on the specific indicator. For this function, the 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.

As an example, suppose that we choose to download data for the Self-perceived health indicator. The indicator's code provided on 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:

- `$res` contains the selected data as a tibble;
- `$msg$Age` states that age class was automatically set;
- `$msg$Further_Conditioning` contains further variables to set in order to obtain a tidy dataset; in particular: `$msg$Further_Conditioning$current` states the value of conditioning taken from `$msg$Further_Conditioning$available_seleTagLs`;
- `$msg$Conditioning` 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 specify the argument `uniqueIdentif` as an integer among those in the first column left of `$msg$Further_Conditioning$available_seleTagLs` to obtain the same indicator under different scales and contexts. For example, for degree "3" of urbanization and level "fair":

```
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)
```

The user could also download data by considering explicitly a specific interval of ages among those available for the Self-perceived health indicator:

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

## [1] Y16-24
## 17 Levels: Y16-24 Y16-29 Y16-44 Y16-64 Y25-29 Y25-34 ... TOTAL
```

Thus, to download data in the age interval Y35-44 and time interval 2003-2018 the function is invoked as:

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

It must be noted that the data downloaded through the functions `download_indicator_EUS()` and `down_lo_EUS()` are not filtered by country members or EU clusters. 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()`:

```
help(dow_soc_scor_boa)
```

This function is specifically implemented for downloading social scoreboard indicators: it is an ‘envelope function’ that automatically download 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
```

Note that the result is a list whose components are similar to those of the functions

`download_indicator_EUS()` and `down_lo_EUS()`.

Lastly, it must be noted 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 et al., 2018) packages. Furthermore, examples and further details on this procedure are available in a specific **convergEU** vignette (Stefanini et al., 2020).

3.3 Imputation of missing data

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". Thus, if data contains missing values, imputation procedures must be invoked before performing the analysis of convergence. In this Subsection, the procedure for imputation of missing values provided by the **convergEU** package is described through a function called `impute_dataset()`:

```
help(impute_dataset)
```

The output of the `impute_dataset()` is a list with the usual three components: `$res`, `$msg` and `$err`. It must be emphasized that typical EU indicators cover a time span of not many years, thus 10% of missing values within a country may already represent a context that requires substantive reasoning before interpreting results after imputation. The user may or may not go ahead with the analysis depending on the considered context.

The function `impute_dataset()` provides two options for imputation of missing values located at the beginning (e.g. starting missing values, also called `tail`) or at the end (final missing values, also called head) of an indicator, 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 body of member state, a deterministic linear imputation is performed to obtain a complete dataset. The two examples below help to better illustrate the missing values imputation procedure within the package. The imputation of missing values in the **convergEU** package requires at least two non-missing values for a given member state. If the values for a given member state are completely missing, the imputation is not performed. More general strategies of imputation are under consideration but not yet available in this package.

As a first example, the dataset `emprTB5` related to the Employment rate indicator is considered for both males and females in the age interval "Y30-54" (Subsection 3.2). First, we assign a meaningful name of the data and we check it for unsuited features by invoking 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 option `constant` as follows:

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

Therefore, starting missing values for a given member state have been imputed equal to an indicator value in its first observed year.

Now, we are going to impute the missing values with the option `cut` as follows:

```
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    EL    IE    IT    LU
##   <fct>  <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Y30-54 T      2002  76.3  85.5  79.9  79.5  72.4  75.2  71.5  78.4
## 2 Y30-54 T      2003  75.8  84.7  81.2  79    73.6  74.7  72.4  77.1
## 3 Y30-54 T      2004  77.2  85    80.7  78.2  74.3  75.5  74    78.9
## 4 Y30-54 T      2005  78.2  85.6  81.4  78.6  74.5  76.9  73.9  80.5
## 5 Y30-54 T      2006  78.3  86.9  81.8  79.9  75.7  77.2  74.7  81.2
## 6 Y30-54 T      2007  79.7  86.8  82.8  81.4  76.1  77.7  74.9  82.2
## # ... with 20 more variables: 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>
```

Note that now all countries from 1992 to 2001 are deleted. This is due to a member state having initial missing values from year 1992 to year 2001. For all other missing values within the dataset, deterministic linear imputation is applied. Note also that if a country is processed but it has no missing, then no numerical value changes. 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: qualitative 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(EmpImp,-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.

A second example of missing values imputation within the **convergEU** package refers to the dataset `myTBheal5`, e.g. the data for the indicator "Self-perceived health" for both males and females between 35 and 44 years old. First, the set `"EU28"` of member states is selected from the data and the `check_data()` function is invoked:

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

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

that reveals the presence of missing values. Note that in the `TBhealth` dataset are present both missing values starting at the oldest year and missing values ending at the last year. Therefore, the arguments `tailMiss` and `headMiss` should be specified according to the user’s preference on how to impute them. Suppose that for the `TBhealth` dataset, missing values starting at the oldest year should be cloned (i.e. option `constant` in the `tailMiss` argument), while those ending at the last year should be deleted (i.e. option `cut` in the `headMiss` argument). Thus, the syntax is:

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

Finally, we call the `check_data()` function on the imputed dataset that confirms that the dataset `TBhealthF` is ready for the analysis:

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

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

3.4 Analysis of convergence

In Section 2, we have introduced the main measures of convergence implemented in the **counvergEU** 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 described in Subsection 3.2 choosing to select the set "EU12" of member states only:

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

## # A tibble: 6 x 13
##    time    BE    DK    FR    DE    EL    IE    IT    LU    NL    PT
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1992  72.1  84.3  77.7  78.7  67.8  58.4  68.7  72.7  72.2  78.6
## 2  1993  72.4  82.2  77.6  77.6  68.1  59.2  68    71.4  72.4  78.8
## 3  1994  72.2  81.6  76.9  77.4  69.1  60.8  67.4  72.5  73    78.4
## 4  1995  72.9  82.6  77.8  77.7  69.3  62.8  67.1  71.3  73.5  78.4
## 5  1996  72.9  83.1  78    77.5  70.1  64.5  67.5  72.4  74.8  78.3
## 6  1997  73.8  83.4  77.5  77.2  70.3  65.5  67.7  73.4  76.6  78.7
## # ... with 2 more variables: ES <dbl>, UK <dbl>

check_data(EmpImpF)

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

```
## $err
## NULL
```

The `check_data()` function confirms that the dataset is ready for the analysis of convergence.

Beta-convergence

The `beta_conv()` function is implemented in the **counvergEU** package for calculating beta- convergence (Subsection 2.1). 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; thus, the syntax is:

```
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 (Intercept)    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 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 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 (Intercept)   -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 ($\hat{\beta}_1 = 0.113$), and this result indicates divergence in this time period for the "EU12" of member states, even though the p-value is not significant. Furthermore, if we calculate the beta-convergence in the recent years after the crisis of 2008, e.g. from 2014 to 2018, we have:

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

```
EmpBC3$res$summary
```

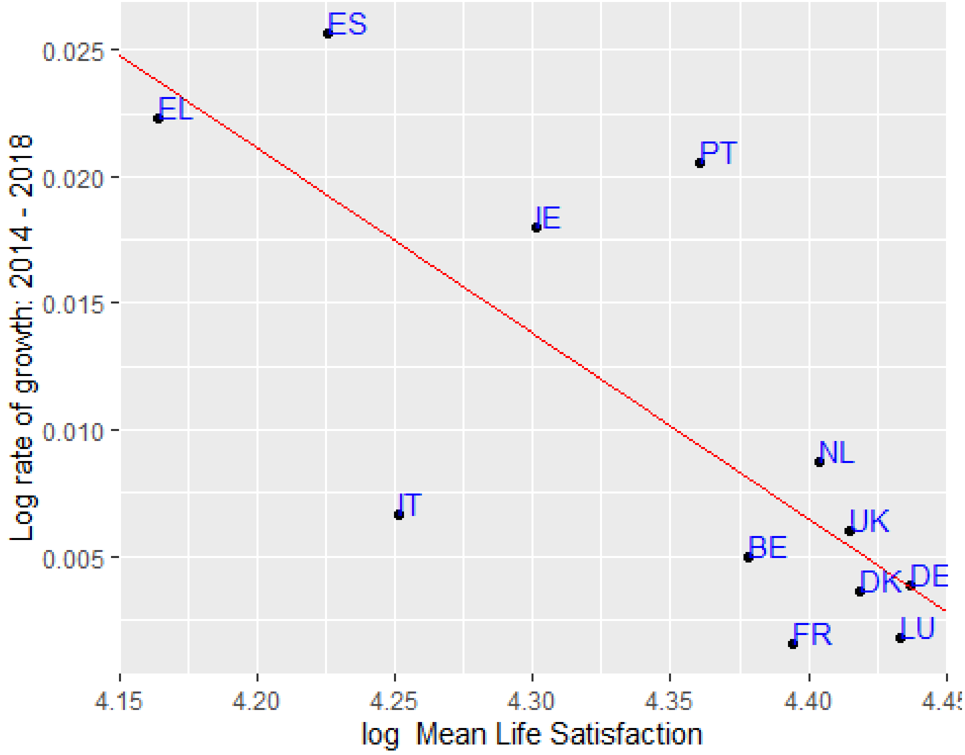
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   0.328     0.0853      3.85 0.00322
## 2 indic        -0.0732    0.0196     -3.73 0.00392
```

The estimated β_1 coefficient is highly significant and negative indicating a convergence for the "EU12" of member states in the period 2014-2018.

A plot of the transformed data and the straight line for the results obtained for the beta convergence may also be 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)
mybetaplot
```

Figure 1: regression plot for beta-convergence



where in the argument `betaRes`, the user should specify the output of the `beta_conv()` function, `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`.

Delta-convergence

Delta-convergence is calculated in the **convergEU** package by invoking the `delta_conv()` function. For the `EmpImpF` dataset, the delta-convergence is obtained as follows:

```
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
```

```
##
## $msg
## NULL
##
## $err
## NULL
```

The output is the usual list of three components, e.g. `$res`, `$msg` and `$err`. To better illustrate the results for the example here considered, the values of delta-convergence are plotted for each year (Figure 2):

```
deltac<-delta_conv(EmpImpF,"time")
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 (i.e. the overall distances in the period 2014-2018 decrease), while divergence is observed in the period 2010-2013, that is during the crisis of 2008 (i.e. the overall distances in the period 2010-2013 increase).

When considering delta convergence, it is useful to also evaluate for a given indicator how much a set of member states deviates from the EU average. To this end, the `demea_change()` function is implemented in the **convergEU** package:

```
help(demea_change)
```

More precisely, the `demea_change()` function calculates the differences from the EU average at subsequent time within each member state, and afterwards these negative and/or positive differences are added over years within the considered member states. As an example, consider the `EmpImpF` dataset for which we choose to calculate the deviations from the EU mean for the countries Italy, France, Germany and Spain in the time period 2003-2016; thus, the syntax is as follows:

```
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 must be noted that if the argument `sele_countries` is equal to NA, then all countries in the dataset are considered. The output of the `demea_change()` function consists of the usual three list components, i.e. `res, msg` and `err`. The component `res` in turn 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    IT     LU    NL
##    <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>
##  1  2003 -2.03   6.87  3.37  1.17   -4.23 -3.13 -5.43 -0.733  4.07
##  2  2004 -1.2    6.60  2.30 -0.2    -4.1  -2.9  -4.4   0.5    3.5
##  3  2005 -0.792  6.61  2.41 -0.392  -4.49 -2.09 -5.09  1.51   2.01
##  4  2006 -1.44   7.16  2.06  0.158  -4.04 -2.54 -5.04  1.46   2.36
##  5  2007 -0.717  6.38  2.38  0.983  -4.32 -2.72 -5.52  1.78   2.98
##  6  2008 -0.15   6.95  3.25  1.35   -4.05 -3.85 -5.75  0.25   4.55
##  7  2009  0.733  5.93  3.83  2.63   -3.27 -6.77 -5.67  2.23   5.63
##  8  2010  1.72   5.32  4.02  3.82   -4.68 -8.28 -5.78  3.72   5.62
##  9  2011  1.61   5.51  4.21  5.71   -7.79 -8.29 -5.19  4.11   5.71
## 10  2012  2.39   5.79  4.49  6.79  -11.8  -7.61 -5.21  6.39   6.09
## 11  2013  2.69   5.99  4.59  7.29  -13.7  -5.31 -6.31  6.99   5.09
## 12  2014  2.04   5.34  3.34  6.84  -13.4  -3.86 -7.46  6.54   4.14
## 13  2015  1.12   5.02  2.42  6.52  -12.0  -2.88 -7.68  4.72   4.02
## 14  2016  1.03   4.53  1.83  6.13  -11.4  -2.77 -7.77  3.73   4.03
## # ... with 3 more variables: 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       IE       IT
##    <dbl>  <dbl>    <dbl>  <dbl>  <dbl>  <dbl>    <dbl>    <dbl>
##  1  2004 -0.833 -0.267   -1.07  -0.967 -0.133 -0.233   -1.03
##  2  2005 -0.408  0.00833  0.108  0.192  0.392 -0.808    0.692
##  3  2006  0.65   0.55    -0.35  -0.233 -0.45   0.45    -0.0500
```

```
## 4 2007 -0.725 -0.775 0.325 0.825 0.275 0.175 0.475
## 5 2008 -0.567 0.567 0.867 0.367 -0.267 1.13 0.233
## 6 2009 0.583 -1.02 0.583 1.28 -0.783 2.92 -0.0833
## 7 2010 0.983 -0.617 0.183 1.18 1.42 1.52 0.117
## 8 2011 -0.108 0.192 0.192 1.89 3.11 0.00833 -0.592
## 9 2012 0.783 0.283 0.283 1.08 4.02 -0.683 0.0167
## 10 2013 0.300 0.2 0.1 0.5 1.90 -2.30 1.10
## 11 2014 -0.650 -0.650 -1.25 -0.45 -0.35 -1.45 1.15
## 12 2015 -0.925 -0.325 -0.925 -0.325 -1.38 -0.975 0.225
## 13 2016 -0.0833 -0.483 -0.583 -0.383 -0.617 -0.117 0.0833
## # ... with 5 more variables: 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
```

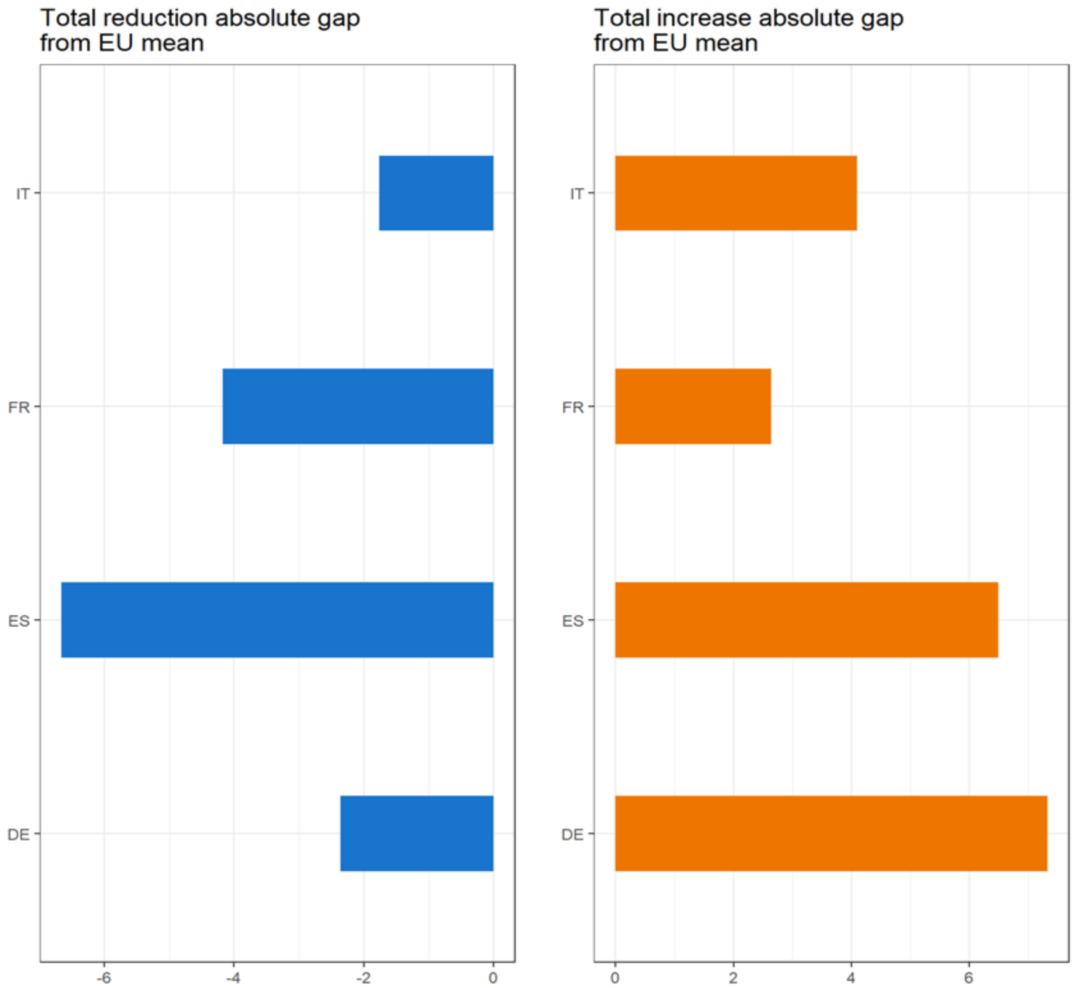
- the minimum and the maximum of the calculated differences:

```
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):

```
plot(res1$res$res_graph)
```

Figure 3: plot of total reduction and total increase



Gamma-convergence

Gamma-convergence is implemented in the function `gamma_conv()` of the **convergEU** package. For the `EmpImpF` dataset, the gamma-convergence is calculated as follows:

```
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 output is the usual list of the three components `$res`, `$msg` and `$err`:

```
gammac1

## $res
## [1] 0.7712996
##
## $msg
## NULL
##
## $err
## NULL
```

The result indicates a high value approaching one for this measure; thus, we can state that few changes have been observed for the Employment rate indicator in the period 2008-2013 for the "EU12" of member states.

Note that it is also possible to print the ranks of the member states for each time by specifying the option `printRanks=T` as follows:

```
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
##   [6,] 6.0 12  9.0  8.0 4.0 2.0  3  5.0  7.0 10.5 1.0 10.5
##   [7,] 5.0 12  8.0  7.0 4.0 3.0  2  6.0  9.0 11.0 1.0 10.0
##   [8,] 5.0 12  7.0  8.0 3.0 4.0  2  6.0  9.0 11.0 1.0 10.0
##   [9,] 5.0 12  7.0  8.0 3.0 4.0  2  6.0 10.0 11.0 1.0  9.0
##  [10,] 5.0 12  7.0  8.0 3.0 4.0  2  6.0 10.0 11.0 1.0  9.0
##  [11,] 5.0 12  8.0  7.0 3.0 4.0  2  6.0 11.0 10.0 1.0  9.0
##  [12,] 5.0 12  9.5  7.0 3.0 4.0  2  6.0 11.0  9.5 1.0  8.0
##  [13,] 5.0 12  8.0  6.0 3.0 4.0  2  7.0 11.0 10.0 1.0  9.0
##  [14,] 5.0 12 10.0  6.0 2.5 4.0  1  7.0  8.0  9.0 2.5 11.0
##  [15,] 5.0 12  9.0  6.0 2.0 4.0  1  7.0 11.0 10.0 3.0  8.0
##  [16,] 5.0 12 10.0  6.0 2.0 4.0  1  9.0 11.0  8.0 3.0  7.0
##  [17,] 5.0 12 10.0  8.0 3.0 4.0  1  6.0 11.0  9.0 2.0  7.0
##  [18,] 5.0 12 10.0  9.0 4.0 2.0  3  8.0 11.0  6.0 1.0  7.0
##  [19,] 7.0 11 10.0  9.0 4.0 1.0  3  8.0 12.0  5.0 2.0  6.0
##  [20,] 6.0 10  9.0 11.5 3.0 1.5  4  8.0 11.5  5.0 1.5  7.0
##  [21,] 6.0  9  8.0 12.0 1.0 3.0  4 11.0 10.0  5.0 2.0  7.0
##  [22,] 6.0 10  8.0 12.0 1.0 4.0  3 11.0  9.0  5.0 2.0  7.0
##  [23,] 6.0 10  7.0 12.0 1.0 4.0  3 11.0  8.0  5.0 2.0  9.0
##  [24,] 5.0 11  7.0 12.0 1.0 4.0  2  9.5  8.0  6.0 3.0  9.5
##  [25,] 5.0 11  6.0 12.0 1.0 4.0  2  8.0  9.0  7.0 3.0 10.0
##  [26,] 5.0  8  6.0 12.0 1.0 4.0  2 11.0  9.0  7.0 3.0 10.0
##  [27,] 5.0  7  6.0 12.0 1.0 4.0  2 10.0  8.5 11.0 3.0  8.5
```

In the **convergEU** package, it is also possible to calculate the gamma-convergence between pairs of subsequent years through the `gamma_conv_msteps()` function. For example for the `EmpImpF` dataset and time period 2008-2018:

```
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
##
## $msg
## NULL
##
```

The results indicate that a high number of changes in the position of the member states of `"EU12"` have been observed in the period 2008-2009, while few changes have occurred for example in the period 2017-2018.

Sigma-convergence

Sigma-convergence is calculated in the **convergEU** package by invoking the `sigma_conv()` function. For the Employment rate indicator, the sigma-convergence is obtained as follows:

```
empSC <- sigma_conv(EmpImpF,timeName="time")
names(empSC)

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

The output of the `sigma_conv()` function is the usual list of the components `res, msg` and `err`. Note that 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. 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.
```

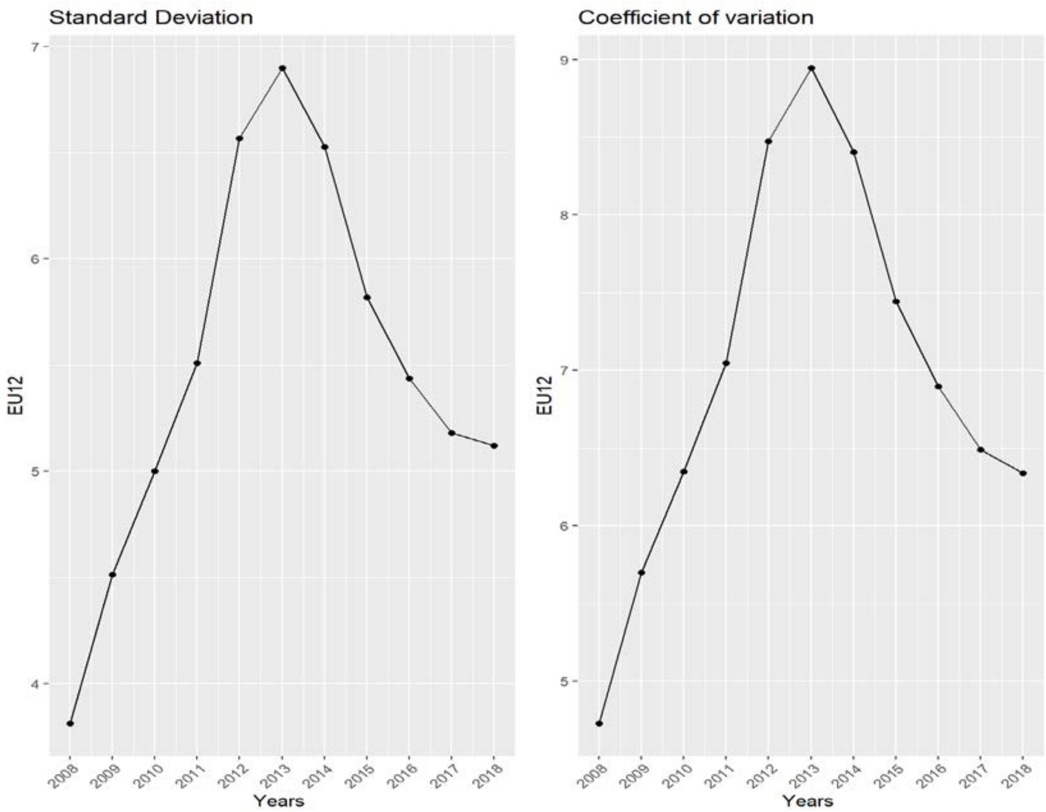
```
## 4   2011    5.51 0.0705  78.2       364.
## 5   2012    6.57 0.0847  77.5       518.
## 6   2013    6.90 0.0895  77.1       571.
## 7   2014    6.53 0.0840  77.7       511.
## 8   2015    5.82 0.0744  78.2       406.
## 9   2016    5.44 0.0689  78.9       355.
## 10  2017    5.18 0.0649  79.8       322.
## 11  2018    5.12 0.0634  80.8       315.
##
## $msg
## NULL
##
## $err
## NULL
```

The `$res` component, is a tibble containing the values of sigma-convergence (i.e. `stdDev` and CV) along time and the deviance.

It is also possible to obtain a graphical representation of the standard deviation and the coefficient of variation obtained for the sigma convergence by invoking the `sigma_conv_graph()` function specifically implemented for this aim in the **convergEU** package. Thus, the plot of the standard deviation and the coefficient of variation in the considered period 2008-2018 is obtained as follows (Figure 4):

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

Figure 4: standard deviation and coefficient of variation by year



where `sigmaconvOut` requires the output of the `sigma_conv()` function, and where `time_0` and `time_t` refer to the starting and ending time respectively. It must be also noted that the considered cluster 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 indicating divergence.

3.5 Weighted average smoothing of a complete dataset and Moving Average smoother

It may be of interest to assume that part of the variability observed in a country on a given index is not structural, i.e. not due to causal determinants but to transient fluctuations.

Furthermore, the interest here is not directed towards prediction but on smoothing values observed in the whole considered time interval. In such a case, a smoothing procedure remove sudden large changes showing a less variable time series than the original. Given that here very short time series (panel data) are considered, a three points weighted average smoother is adopted. The smoother substitutes an original raw value $y_{m,i,t}$. of country m, indicator i at time t, with the weighted average

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

where $0 < w \leq 1$. The special case $w = 1$corresponds to no smoothing. In case of missing values, an NA is returned. 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. It requires a dataset without missing values, e.g. eventually an imputed dataset. Furthermore, the time variable must be deleted from the dataset before the smoothing with the `smoo_dataset()` function.

For example, below is displayed the smoothing of the `TBhealthF` dataset in which the missing values have been already imputed (Section 3.3). We choose the following three member states on which the smoothing is to be performed: Italy, France and Germany, coded “IT”, “FR” and “DE” respectively. Thus, we select the data with just these three countries and the time variable:

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

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

The `check_data()` function confirms that the dataset is ready for smoothing. Thus, once checking is passed, we go with the smoothing step after deleting the time variable:

```
library(dplyr)
resSM <- smoo_dataset(select(workTB,-time),
                      leadW = 0.15,
                      timeTB= select(workTB,time))
```

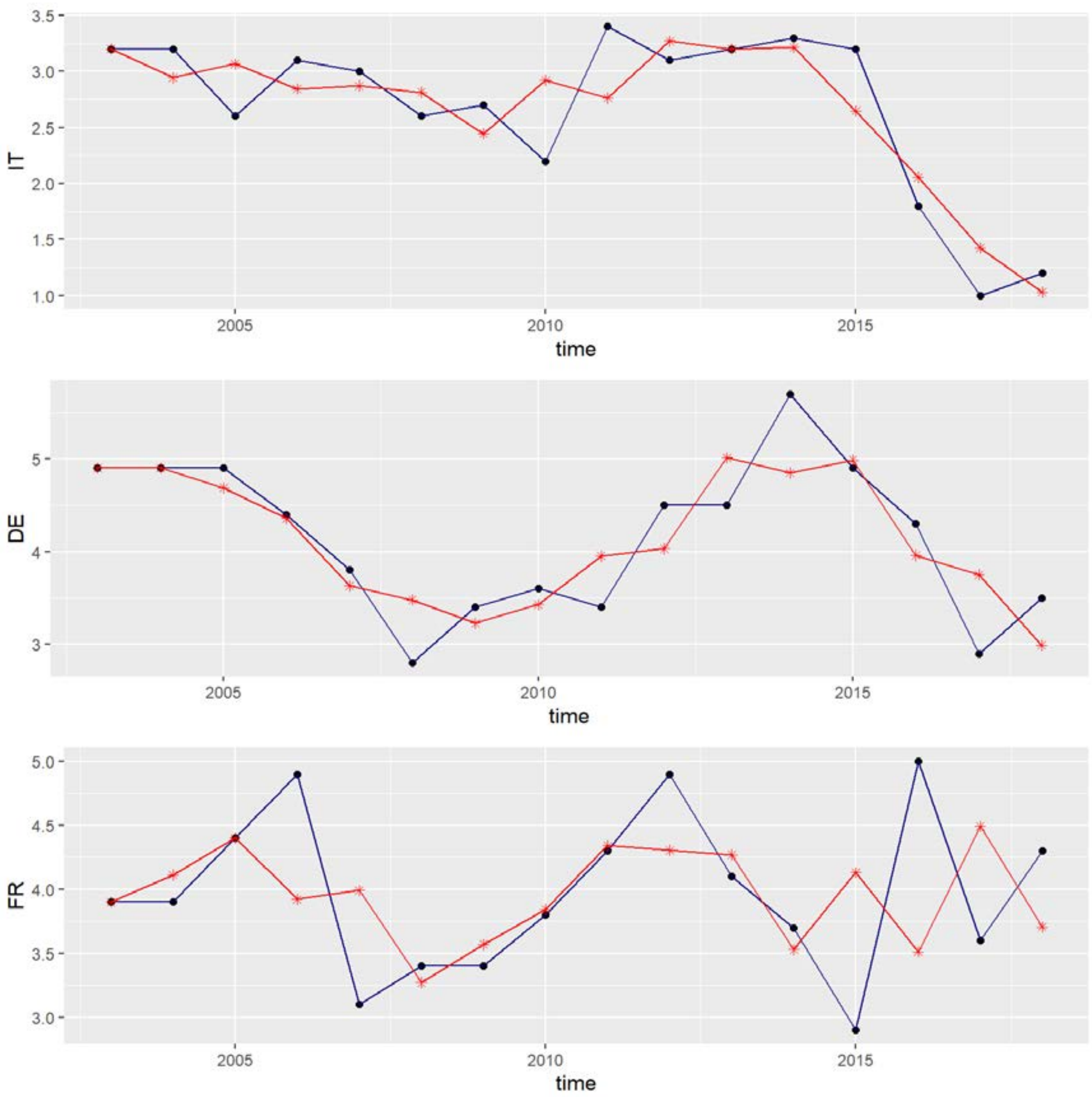
Note that, the first argument is the complete dataset `workTB` with just country columns, while the third argument `timeTB` is the dataset that will be displayed in the output and in which the time variable has been added after performing smoothing. The argument `leadw` refers to the leading positive weight where $(0 < w \leq 1)$ as described above. In our example, the smoothing is firstly performed with an weight equal to 0.15. To carry out comparison between the original indicator's values and the smoothed ones, the following dataset is built:

```
compaTB <- select(bind_cols(workTB, select(resSM,-time)),
                  time,IT,IT1,DE,DE1, FR, FR1)
```

Therefore, a graphical output shows changes for Italy ("IT"), Germany ("DE") and France ("FR") with the original indicator in blue and the smoothed version in red (Figure 5):

```
ITq<-qplot(time,IT,data=compaTB) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=IT1),colour="red") +
  geom_point(aes(x=time,y=IT1),colour="red",shape=8)
DEq<-qplot(time,DE,data=compaTB) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=DE1),colour="red") +
  geom_point(aes(x=time,y=DE1),colour="red",shape=8)
FRq<-qplot(time,FR,data=compaTB) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=FR1),colour="red") +
  geom_point(aes(x=time,y=FR1),colour="red",shape=8)

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

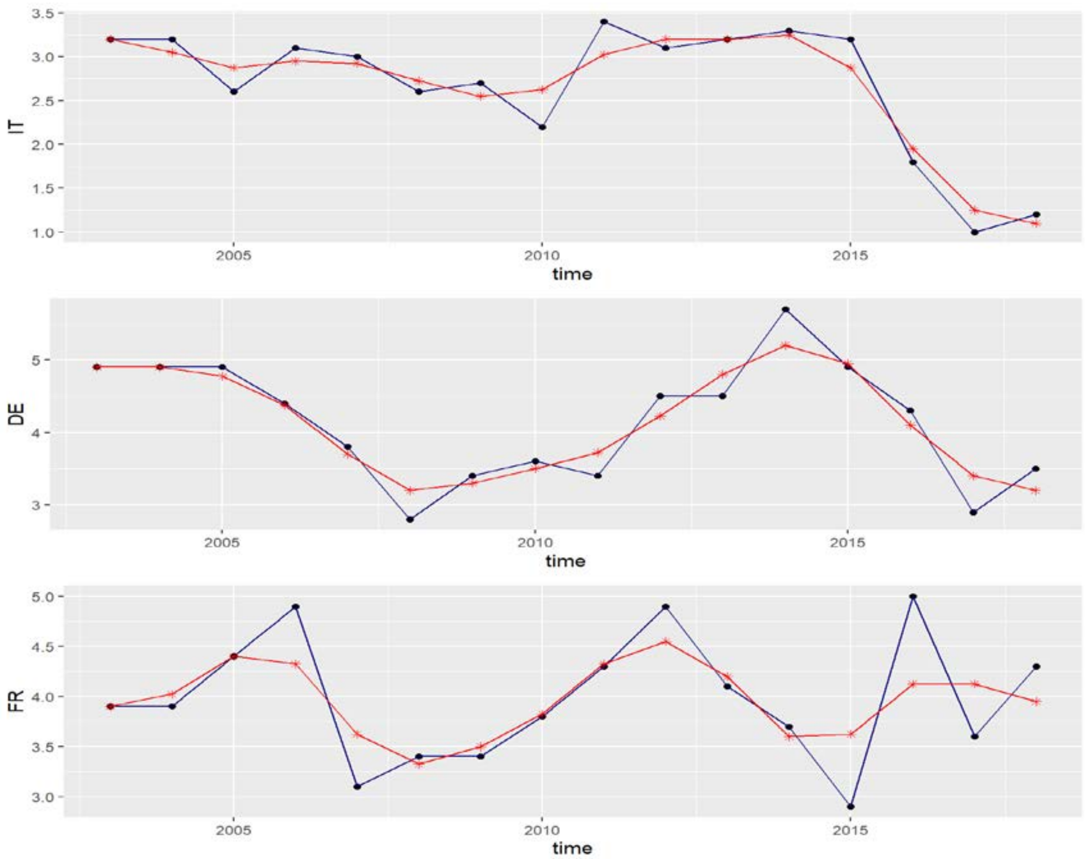
Figure 5: plot of the original indicator in blue and the smoothed version in red (weight 0.15)



Note that, higher the selected weight less smoothed are the indicator's values. For example, by choosing a weight equal to 0.50 a smoother plot is obtained (Figure 6):

```
resSM2 <- smoo_dataset(select(workTB,-time),
                       leadW = 0.5, timeTB= select(workTB,time))
compaTB2 <- select(bind_cols(workTB, select(resSM2,-time)),
                   time,IT,IT1,DE,DE1, FR, FR1)
ITq2<-qplot(time,IT,data=compaTB2) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=IT1),colour="red") +
  geom_point(aes(x=time,y=IT1),colour="red",shape=8)
DEq2<-qplot(time,DE,data=compaTB2) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=DE1),colour="red") +
  geom_point(aes(x=time,y=DE1),colour="red",shape=8)
FRq2<-qplot(time,FR,data=compaTB2) +
  geom_line(colour="navyblue") +
  geom_line(aes(x=time,y=FR1),colour="red") +
  geom_point(aes(x=time,y=FR1),colour="red",shape=8)
grid.arrange(ITq2, DEq2, FRq2, nrow=3)
```

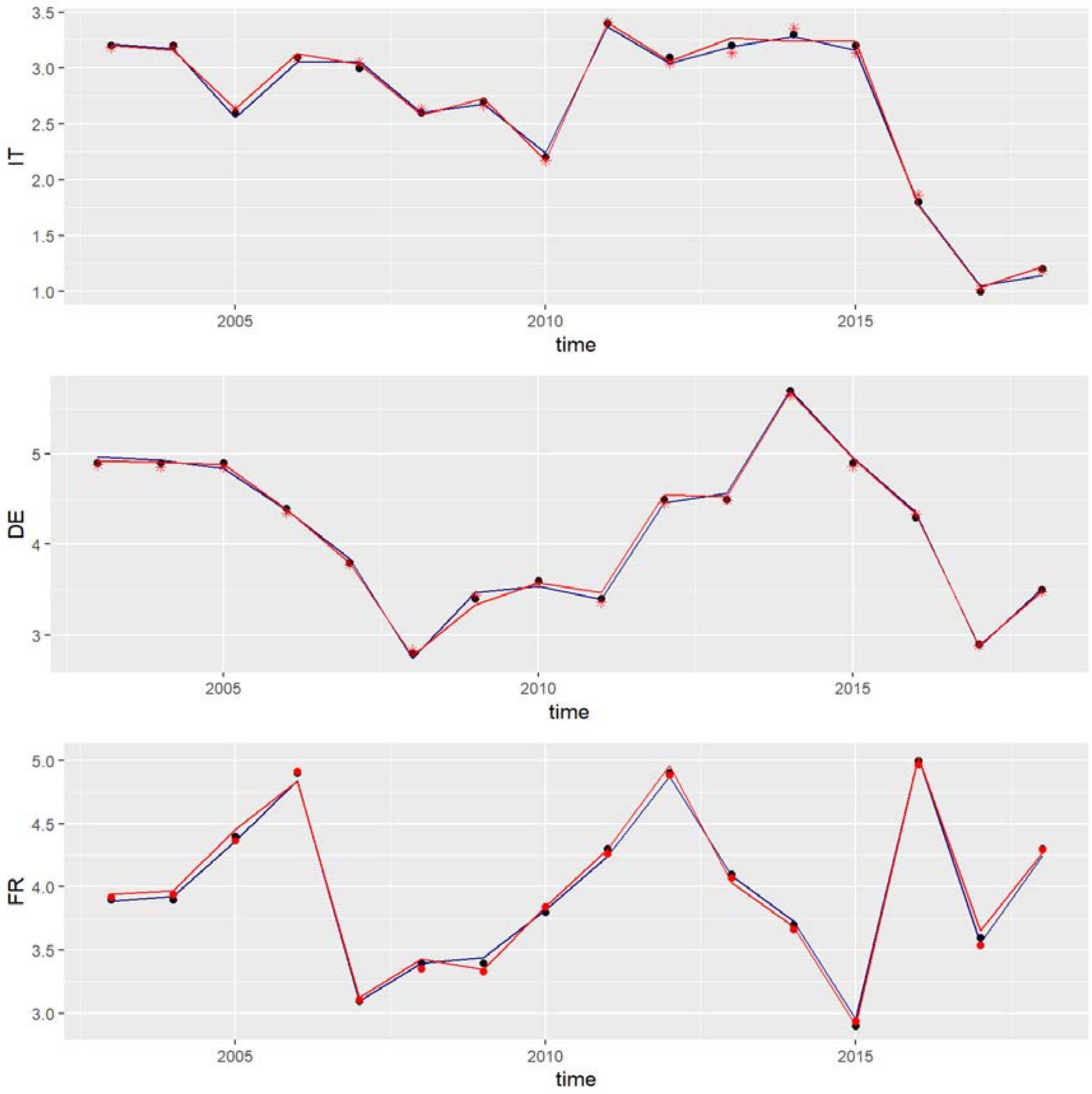
Figure 6: plot of the original indicator in blue and the smoothed version in red (weight 0.50)



Lastly, note that a weight equal to 1 leaves data unchanged (Figure 7):

```
resSM3 <- smoo_dataset(select(workTB,-time),
                       leadW = 1, timeTB= select(workTB,time))
compaTB3 <- select(bind_cols(workTB, select(resSM3,-time)),
                   time,IT,IT1,DE,DE1, FR, FR1)
ITq3<-qplot(time,IT,data=compaTB3) +
  geom_line(colour="navyblue",
            position=position_jitter(width=0, height=0.07)) +
  geom_line(aes(x=time,y=IT1),colour="red",
            position=position_jitter(width=0, height=0.07)) +
  geom_point(aes(x=time,y=IT1),colour="red",
             shape=8,position=position_jitter(width=0, height=0.07))
DEq3<-qplot(time,DE,data=compaTB3) +
  geom_line(colour="navyblue",
            position=position_jitter(width=0, height=0.07)) +
  geom_line(aes(x=time,y=DE1),colour="red",
            position=position_jitter(width=0, height=0.07)) +
  geom_point(aes(x=time,y=DE1),colour="red",
             shape=8,position=position_jitter(width=0, height=0.07))
FRq3<-qplot(time,FR,data=compaTB3) +
  geom_line(colour="navyblue",
            position=position_jitter(width=0, height=0.07)) +
  geom_line(aes(x=time,y=FR1),colour="red",
            position=position_jitter(width=0, height=0.07)) +
  geom_point(aes(x=time,y=FR1),colour="red",
             position=position_jitter(width=0, height=0.07))
grid.arrange(ITq3, DEq3, FRq3, nrow=3)
```

Figure 7: plot of the original indicator in blue and the smoothed version in red (weight 1)



It must be noted that a time window larger than 3 could be considered, but deep thoughts are recommended about economic and social changes that may happen in EU during 5 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 **caTools** package. In the **convergEU** package, smoother based on moving average is implemented in the function `ma_dataset()` where the time window of the moving average is set through the argument `kappa`:

```
help(ma_dataset)
```

Below, we are going to compute smoother based on moving average for the `TBhealthF` dataset by considering the countries Italy (“IT”), Germany (“DE”) and France (“FR”). Thus, the "time by countries" dataset is:

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

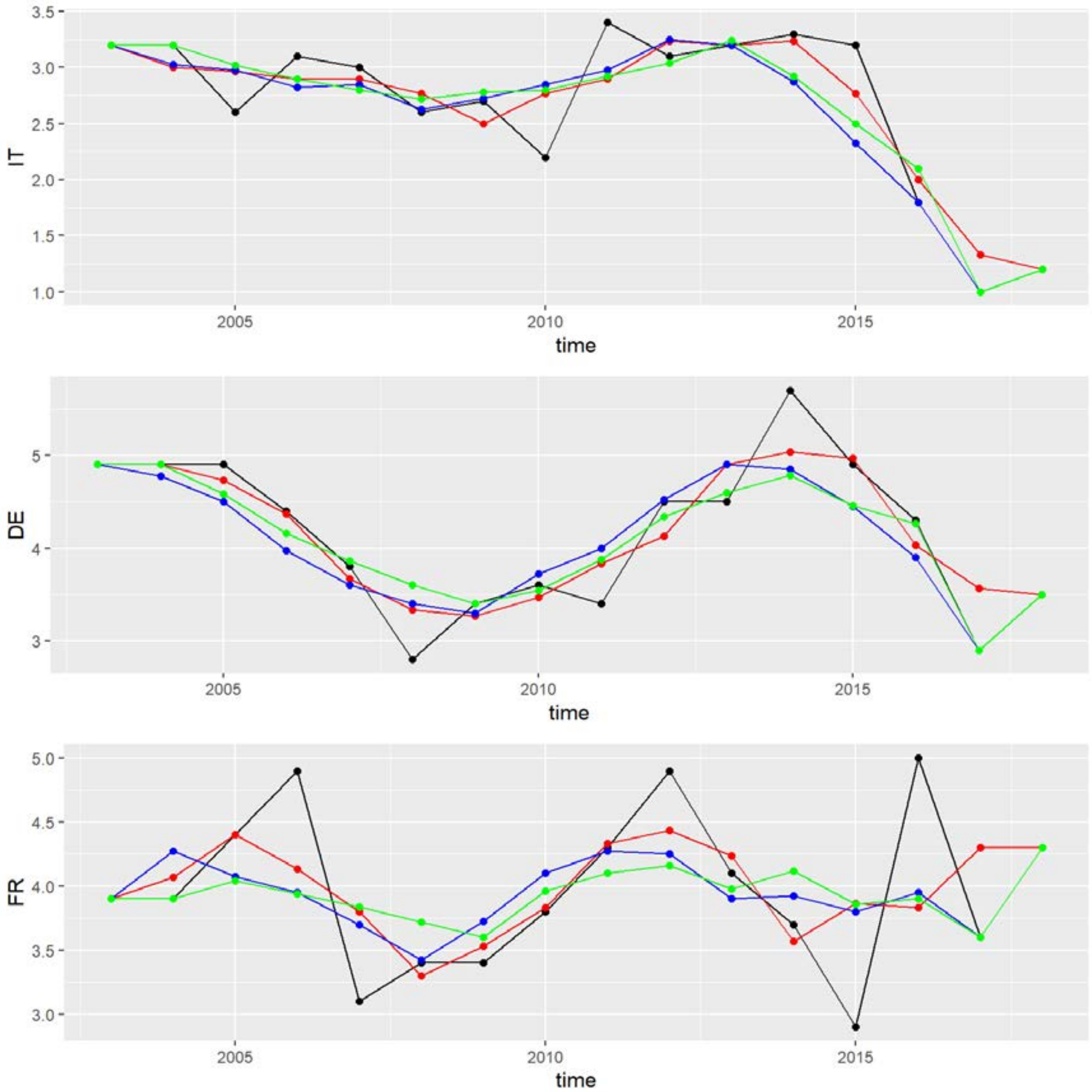
We proceed by considering the following time windows of the moving average: $k = 3$, $k = 4$ and $k = 5$.

```
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. The plot below shows results for Italy, Germany and France for different degrees of smoothing: original (black), $k = 3$ (red), $k = 4$ (blue), $k = 5$ (green) (Figure 8):

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

myGde <- ggplot(cuTB1,aes(x=time,y=cuTB1$DE))+geom_line()+geom_point()+
  geom_line(aes(x=time,y=ma3$res$DE),colour="red")+
  geom_point(aes(x=time,y=ma3$res$DE),colour="red")+
  labs(y="DE")+
  #
  geom_line(aes(x=time,y=ma4$res$DE),colour="blue")+
  geom_point(aes(x=time,y=ma4$res$DE),colour="blue")+
  #
  geom_line(aes(x=time,y=ma5$res$DE),colour="green")+
  geom_point(aes(x=time,y=ma5$res$DE),colour="green")

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

grid.arrange(myGit, myGde, myGfr, nrow=3)
```

Figure 8: plot of an indicator for Italy, Germany and France under different degrees of smoothing: original (black), $k = 3$ (red), $k = 4$ (blue), $k = 5$ (green).



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

3.6 Country fiches

The **counvergEU** package provides the function `go_ms_fi()` to prepare country fiches developed specifically to facilitate the work of the users who regularly deal with EU indicators. This function produces an `Html` file that shows several results for a given reference country, compared to one or more other countries of interest: for instance, scoreboards for the selected countries, patterns of convergence and divergence as well as country ranking tables. By invoking the `go_ms_fi(),` the `Html` file can be created for different indicators, timeframes and countries.

The go_ms_fi() function automatically prepares the country fiches but it is also able to create a directory along an existing path and to copy the `Rmarkdown` files representing the template within it. The `Rmarkdown` file is parameterized so that passing different arguments before compilation into Html format, different data can be analysed, say different indicators and countries.

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 the time average. Failing to satisfy these requisites causes the use of a wrong mean value at each year. Nevertheless, one key country is specified, and some other countries of interest may be listed, to better decorate graphs and compare performances on just a subsets of all member states.

Below, a call to the function `go_ms_fi()` illustrates the syntax for 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= 'EU28',
  x_angle= 45,
  dataNow= Sys.time(),
  author = 'A.Student',
  outFile = 'Germany-up2-2016-EMP',
  outDir = 'F:/analysis/IT2018',
  indiName= 'EmpImpF')
```

It is very important to emphasize the presence of some constraints and a quite unusual way to pass values as arguments of this function. In fact, the first argument is the working dataset which is passed not as an R object but as a string, 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 member state that will be shown in one-country plots. The argument `indiType` specifies if the considered indicator is built so that a low value is good for a country (`indiType = “lowBest”`) or if a high value is good (`indiType = “highBest”).`

Of particular importance is the argument `outFile` that can be a string indicating the name of the output file. Similarly, `outDir` is the path (possibly a 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 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`, it is also stored a file called like specified by `outFile` but with added the string `‘-workspace.RData’` that contains data and plots produced during the compilation of the country fiche for further subsequent use in other technical reports and/or documents.

It is important to note that an active Internet connection is mandatory while invoking the `go_ms_fi()` function to obtain a properly rendered `Html` file.

3.7 Indicator fiches

An auxiliary function `go_indica_fi()` is provided in the R package **convergEU** to build indicator fiches in `Html` format. This means that the user chooses a given indicator, a time window and a set of countries, and the corresponding fiche is automatically produced by invoking the `go_indica_fi()` function. The resulting `Html` file reports measures of convergence as well as time series plots, box-plots, and patterns of convergence and divergence. Differently from country fiches that besides a reference country also requires the specification of a set of other countries for comparison, an indicator fiche is built just from a specified indicator and a given set of member states. When invoking the `go_indica_fi()` function, an output directory must be also specified. Note that some arguments are passed as strings instead of objects, as described in the last section above for the country fiches.

An example on how to invoke the function for the `EmpImpF` dataset is:

```
go_indica_fi(
  time_0 = 2005,
  time_t = 2015,
  timeName = 'time',
  workDF = 'EmpImpF' ,
  indicaT = 'lfsa_ergaed',
  indiType = c('highBest','lowBest')[1],
  seleMeasure = 'all',
  seleAggre = 'EU28',
  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 in the syntax 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”.`

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.

Similarly to the `go_ms_fi()` function, it is important to note that an active Internet connection is mandatory while invoking the `go_indica_fi()` function to obtain a properly rendered `Html` file.

4. Patterns and types of convergence and divergence

4.1 Basic theory on patterns

Some notation is required to describe qualitative patterns of change along time for indicators in EU. The following random variables are defined:

- $Y_{EU,i,t}$ is the random variable for EU average, indicator i at time $t \in T_{EU,i}$;
- $Y_{m,i,t}$ is the random variable for indicator i of member state m at time $t \in T_{m,i}$;
- observed value are $y_{EU,i,t}$ and $y_{m,i,t}$, with $m = 1, \ldots, 28$.

Let's assume that we have just two subsequent time points, for example, $T_{EU,i} = T_{m,i} = \{2003, 2004\}$, with the observed values $y_{EU,i,2003}, y_{EU,i,2004}, y_{m,i,2003}, y_{m,i,2004}$.

From the Eurofound report (Mascherini et al., 2018), the gradient is numerically calculated as follows:

- Member State (MS):

$$\beta_m := \nabla f_{MS}(y) = \frac{y_{m,i,t+\delta} - y_{m,i,t}}{\delta} \tag{11}$$

- EU average (EU):

$$\beta_{EU} := \nabla \mu_{EU}(y) = \frac{y_{EU,i,t+\delta} - y_{EU,i,t}}{\delta} \tag{12}$$

while the value of delta is:

$$\Delta_{t+\delta,t}\sigma^2 := (y_{m,i,t+\delta} - y_{EU,i,t+\delta})^2 - (y_{m,i,t} - y_{EU,i,t})^2 \tag{13}$$

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). In the following list, each considered condition of a `"highBest"` indicator (on the left) is associated to a numerical code (on the right):

- $(\beta_{EU} > 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 1$
- $(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 2$
- $(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 3$
- $(\beta_{EU} > 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 4$
- $(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 5$
- $(\beta_{EU} > 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 6$

- $(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 7$
- $(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 8$
- $(\beta_{EU} < 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 > 0) \Longrightarrow 9$
- $(\beta_{EU} < 0) \wedge (\beta_{EU} > \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 10$
- $(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m > 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 11$
- $(\beta_{EU} < 0) \wedge (\beta_{EU} < \beta_m) \wedge (\beta_m < 0) \wedge (\Delta_{t+\delta,t}\sigma^2 < 0) \Longrightarrow 12$

Further numerical labels are here introduced to recognize parallelism, crossing and any other configuration not included above:

- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} > 0) \wedge (y_{m,i,t} > y_{EU,i,t}) \Longrightarrow 13$
- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} = 0) \wedge (y_{m,i,t} > y_{EU,i,t}) \Longrightarrow 14$
- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} < 0) \wedge (y_{m,i,t} > y_{EU,i,t}) \Longrightarrow 15$
- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} < 0) \wedge (y_{m,i,t} < y_{EU,i,t}) \Longrightarrow 16$
- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} = 0) \wedge (y_{m,i,t} < y_{EU,i,t}) \Longrightarrow 17$
- $(\beta_{EU} = \beta_m) \wedge (\beta_{EU} > 0) \wedge (y_{m,i,t} < y_{EU,i,t}) \Longrightarrow 18$
- $(y_{m,i,t} > y_{EU,i,t}) \wedge (y_{m,i,t+1} < y_{EU,i,t+1}) \Longrightarrow 19$
- $(y_{m,i,t} < y_{EU,i,t}) \wedge (y_{m,i,t+1} > y_{EU,i,t+1}) \Longrightarrow 20$
- Other (Inspection) $\Longrightarrow 21$

The 21 patterns defined above through numerical labels are also associated to corresponding string labels as shown in in Table 1.

In order to find patterns for indicators of type `"lowBest",` we assume that the higher the indicator value, the worse the considered socio-economic feature in a member state. 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".` To better explain these points in Table 1, we report a summary of the 21 patterns according to the type of the indicator, i.e. `"lowBest"` or `"highBest".`

Moreover, the empirical values leading to a given pattern could be very close to values that are labelled into a different way; thus a graphical display may show if a given index presents values close to more than one pattern. That is, despite that each pattern has its own mathematical definition, it may happen that a given indicator behaves like an extreme case of a given pattern so that it is graphically very close to another extreme case of a differently labelled pattern.

Table 1: Summary of the 21 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	EU	MS	Type	EU better	EU	MS
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

4.2 Finding patterns within the convergEU package

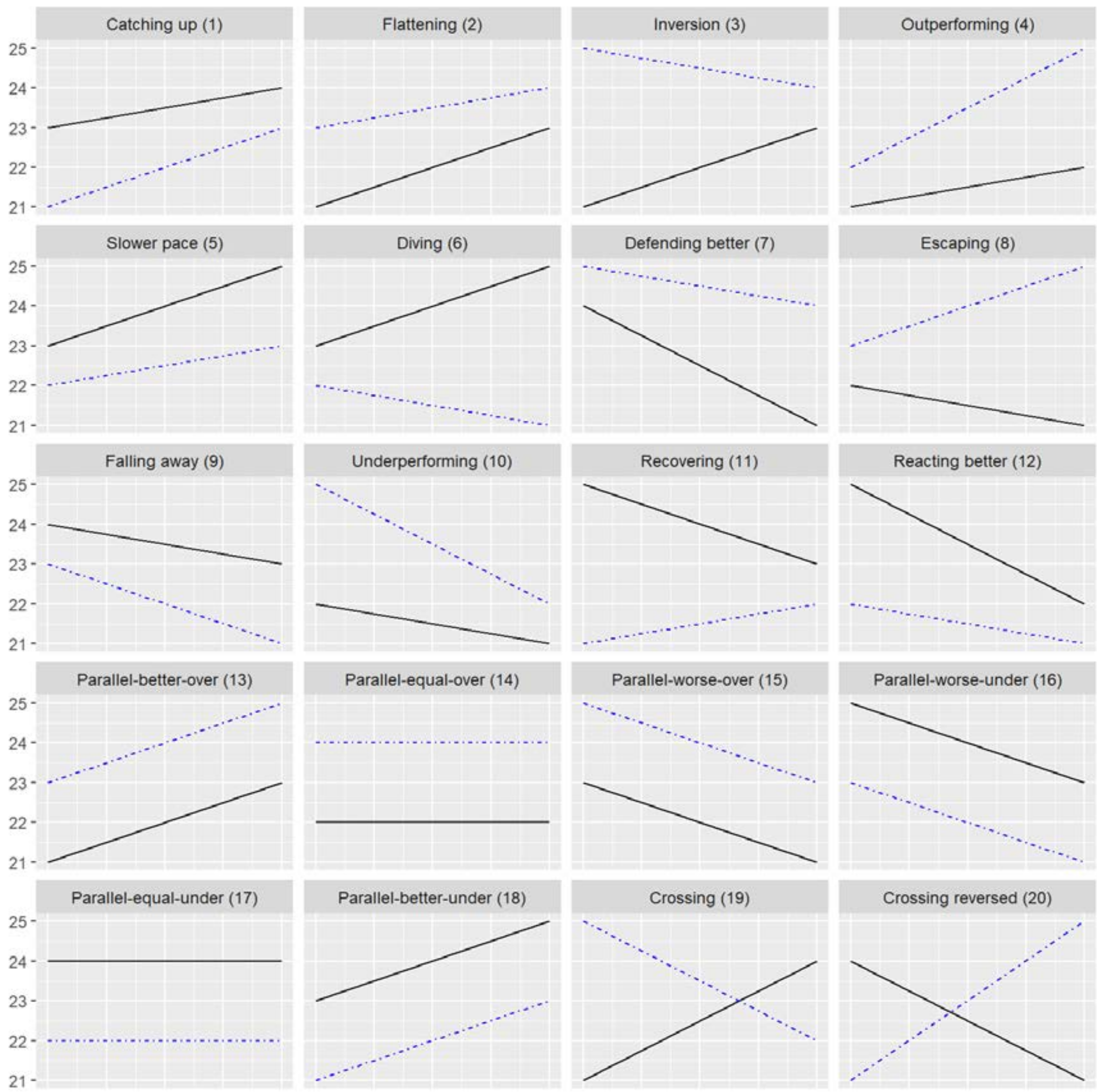
The **convergEU** package allows to obtain patterns of change along time for indicators in the EU, and to this end, the function `ms_pattern_ori()` has been implemented in this package:

```
help(ms_pattern_ori)
```

for both `lowBest` and `highBest` types of indicators. Thus, in the `ms_pattern_ori()` function, the 20 + 1 patterns defined in the previous Subsection are defined through their numerical labels and corresponding string labels. Moreover, graphical plots for the defined patterns depending on the type of indicators (`"lowBest"` or `"highBest"`) are also available within this package by invoking the `patt_legend()` function. For indicators of type `highBest`, the graphical plot of the patterns is obtained as follows (Figure 9):

```
highind<-patt_legend(indiType="highBest")
highind
```

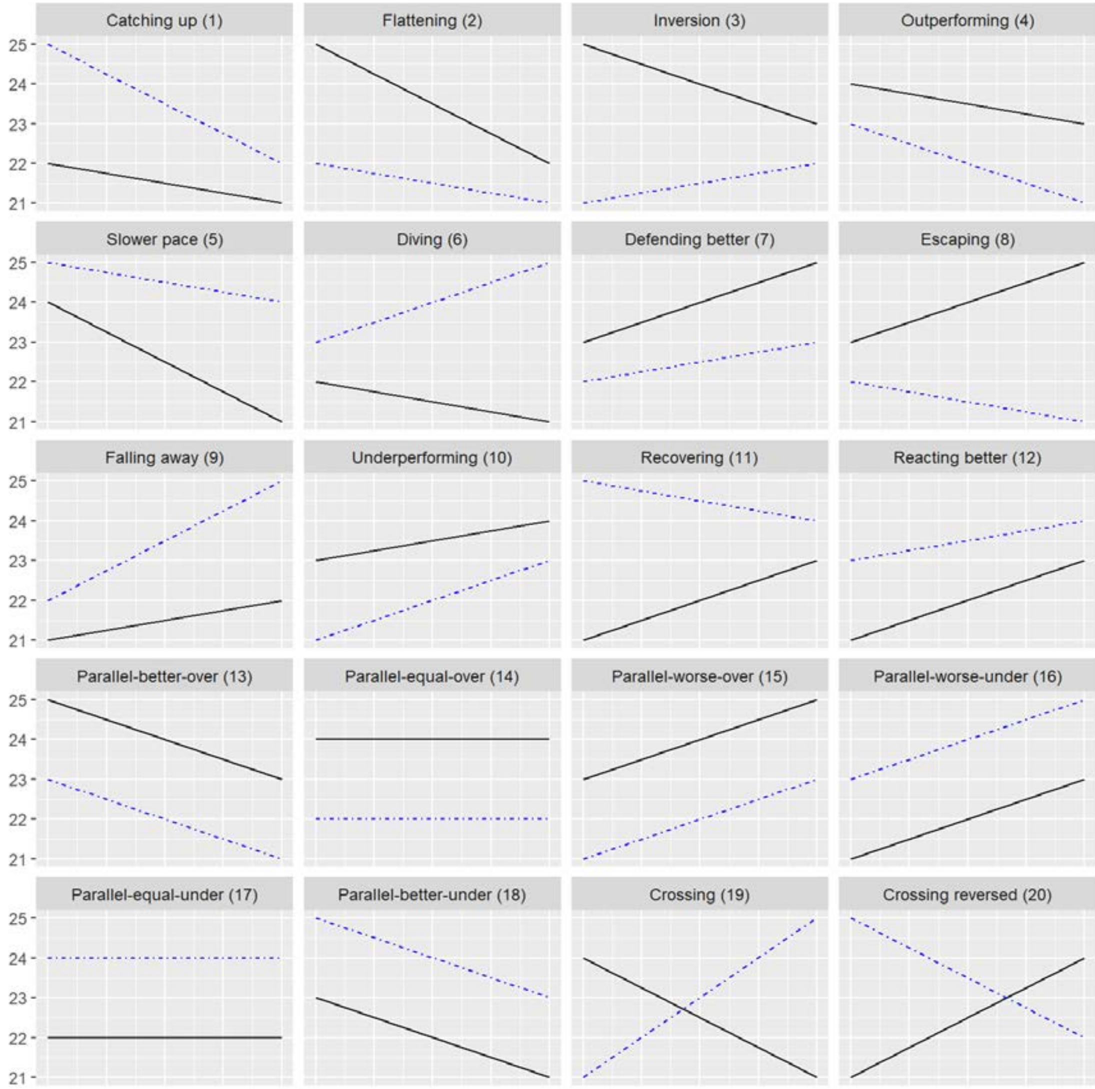
Figure 9: patterns for an indicator of type "highBest" (dashed blue lines: Member State; solid lines: EU average)



Similarly, in order to obtain the plot of patterns for `lowBest` type of indicators, the `indiType` argument must be properly specified (Figure 10):

```
lowhind<-patt_legend(indiType="lowBest")
lowhind
```

Figure 10: patterns for an indicator of type "lowBest" (dashed blue lines: Member State; solid lines: EU average)



Let's consider an example to find patterns for the `emp_20_64_MS` dataset for which the indicator is of type `highBest` by invoking the `ms_pattern_ori()` function as follows:

```
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:

```
head(mypattempn)
```

```
## #A tibble: 6 x 17
##   Country `2002/2003` `2003/2004` `2004/2005` `2005/2006` `2006/2007`
##   <chr>         <int>       <int>       <int>       <int>       <int>
## 1 AT                4           3           4           4           4
## 2 BE                6           1           1          21           1
## 3 BG                1           1           1           1           1
## 4 CY                4           4           3           4           2
## 5 CZ                3           3           4           2           2
## 6 DE                3          19          20           4           4
## # ... with 11 more variables: `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` `2005/2006` `2006/2007`
##   <chr>   <chr>       <chr>       <chr>       <chr>       <chr>
## 1 AT      Outperform~ Inversion   Outperform~ Outperform~ Outperform~
## 2 BE      Diving      Catching up Catching up Other (Ins~ Catching up
## 3 BG      Catching up Catching up Catching up Catching up Catching up
## 4 CY      Outperform~ Outperform~ Inversion   Outperform~ Flattening
## 5 CZ      Inversion   Inversion   Outperform~ Flattening  Flattening
## 6 DE      Inversion   Crossing    Crossing r~ Outperform~ Outperform~
## # ... with 11 more variables: `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>
```

Let's consider the time period 2006-2007 and the country France ("FR") for which the obtained pattern is "Slower pace":

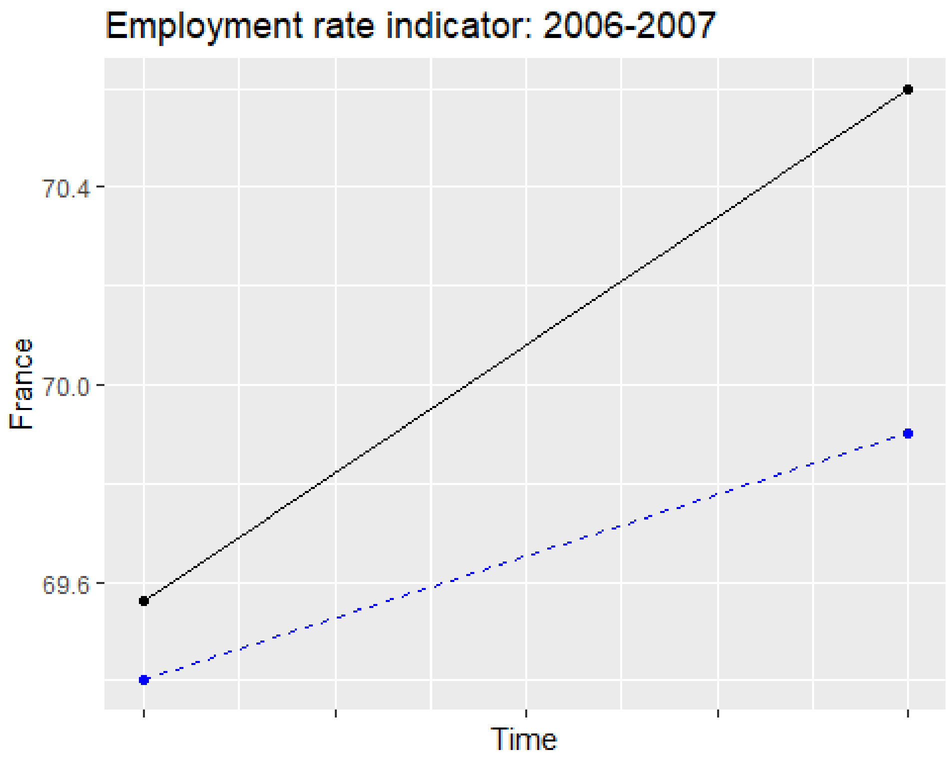
```
mypattemp$`2006/2007`[12]

## [1] "Slower pace"
```

The resulting plot of the calculated pattern is shown in Figure 11, where the dashed blue line refers to the France and the black solid line refers to the EU:

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

```
  theme(axis.text.x=element_blank())
gFR
```

Figure 11: pattern of France for years 2006-2007 (dashed blue line: France; solid line: EU average)



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.

4.3 Types of convergence/ divergence within the convergEU package

Convergence and divergence may be strict or weak, upward or downward. In the **convergEU** package, the function `upDo_CoDi()` is specifically implemented for assessing the type of convergence/ divergence which occurs for a given indicator, a collection of member states and a given period of time. The interpretation depends on the type of indicator, that is `"highBest"` or `"lowBest"`. 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/ divergence occurring in the time period 2008-2010, and according to the sample variance for summarizing dispersion. Thus, the syntax for the `upDo_CoDi()` function is as follows:

```
Empconv<-upDo_CoDi(emp_20_64_MS,
                  timeName = "time",
```

```
              indiType = "highBest",
              time_0 = 2008,
              time_t = 2010,
              heter_fun = "var")
```

where the sample variance for summarizing dispersion is specified in the argument `heter_fun` as a string. The output of the `upDo_CoDi()` function consists of the usual three list components: `$res, $msg` and `$err`.

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

- a statement if a convergence or a 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 list of the member states labels for which the differences for a given indicator between the target time and the reference time are greater than zero:

```
Empconv$res$declaration_split$names_incre

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

- a list of the member states labels for which the differences for a given idicator between the target time and the reference time are smaller than zero:

```
Empconv$res$declaration_split$names_decre

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

- a list of the differences for a given idicator between the target time and the reference time for each member state:

```
Empconv$res$diffe_MS

##    AT   BE BG   CY CZ DE   DK    EE   EL   ES   FI   FR   HR   HU IE
## 1 0.1 -0.4 -6 -1.5 -2  1 -3.8 -10.3 -2.5 -5.7 -2.8 -1.2 -2.8 -1.6 -8
##     IT   LT  LU    LV  MT   NL   PL   PT  RO   SE   SI   SK   UK
## 1 -1.9 -7.7 1.9 -11.1 0.9 -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 for the reference time and the target time respectively, and computed on the basis of the type of dispersion 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")
```

5. Auxiliary functions implemented in the package

In the **convergEU** package, several auxiliary functions are implemented to further enrich the analysis of convergence. These functions are described in detail in the following Section.

5.1 Absolute changes across countries

Absolute change is defined as:

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

for member state m, indicator i at time t. In the **convergEU** package, absolute changes across countries are calculated through the `abso_change()` function on a sorted dataset in the format 'time by countries' and without missing values. To illustrate in detail this function, we take as an example the locally available dataset `emp_20_64_MS` described in Subsection 3.2.

Following, we calculate the absolute changes across the 28 member states of the set EU28 by considering as reference time 2007 (argument `time_0`) and as focus time 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"      "average_abs_change"
```

The results consist of: (i) the list `myAC$res$abso_change` of the absolute changes for each member state; (ii) a list of the sum of absolute values (`myAC$res$sum_abs_change`), defined in general as $\sum_{t=t_0+1} |\Delta y_{m,i,t}|$; (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. `myAC$res$average_abs_change`.

5.2 Summaries of clusters of countries

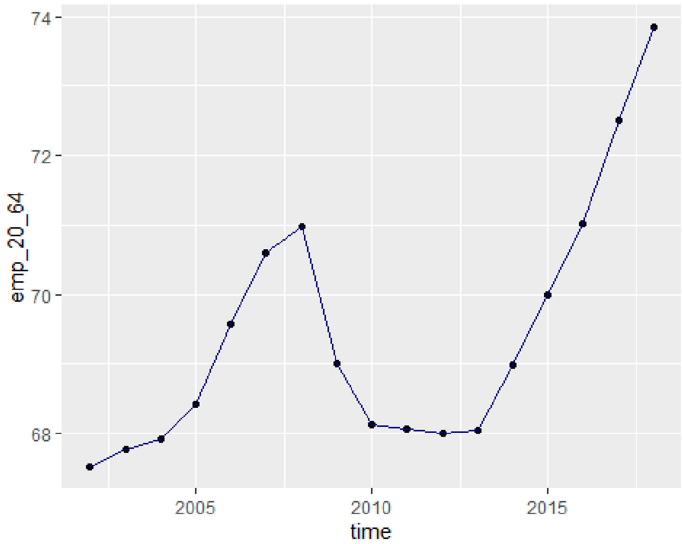
An important summary is obtained as unweighted average of country values through the `average_clust()` function. For example, the unweighted average in the `emp_20_64_MS()` dataset for the EU28 set of member states is computed as follows (Figure 12):

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

Time series can be also plotted as follows:

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

qplot(time, EU28, data=wwTB,
      ylim=c(mini_EU,maxi_EU))+geom_line(colour="navy blue")+
      ylab("emp_20_64")
```

Figure 12: unweighted average of the employment rate for the cluster of countries EU28



By considering more in details the results, there is an increase of the EU28 average of the Employment rate indicator from 2002 to 2008. Subsequently, with the advent of the crisis in 2008, the EU28 average for this indicator decreases from 2008 to 2013, and then begins to increase again.

5.3 Departures from the average

In the **convergEU** package, the function `departure_mean()`calculates for each member state the departure from the average. The calculations are performed with respect to the results obtained for sigma-convergence. An example follows for the `emp_20_64_MS()` dataset. First, the sigma-convergence is calculated:

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

Then, 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"  "devianceContrib"

res$res$departures

## # A tibble: 17 x 33
##     time stdDev     CV  mean devianceT    AT    BE    BG    CY    CZ
##    <dbl>  <dbl>  <dbl> <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  2002   6.34 0.0939  67.5     1125.     0     0    -1     1     0
##  2  2003   5.95 0.0878  67.8      991.     0     0    -1     1     0
```

```
## 3  2004   5.70 0.0839  67.9      909.     0     0    -1     1     0
## 4  2005   5.54 0.0809  68.4      858.     0     0    -1     1     0
## 5  2006   5.57 0.0801  69.6      869.     0     0     0     1     0
## 6  2007   5.47 0.0775  70.6      838.     0     0     0     1     0
## 7  2008   5.36 0.0755  71.0      804.     0     0     0     1     0
## 8  2009   5.03 0.0730  69.0      710.     0     0     0     1     0
## 9  2010   5.24 0.0769  68.1      768.     1     0     0     1     0
## 10 2011   5.59 0.0821  68.1      875.     1     0     0     0     0
## 11 2012   5.98 0.0880  68       1002.     1     0     0     0     0
## 12 2013   6.28 0.0922  68.0     1103.     1     0     0     0     0
## 13 2014   5.98 0.0867  69.0     1000.     0     0     0     0     0
## 14 2015   5.74 0.0820  70.0      922.     0     0     0     0     0
## 15 2016   5.60 0.0789  71.0      879.     0     0     0     0     1
## 16 2017   5.37 0.0741  72.5      808.     0     0     0     0     1
## 17 2018   5.30 0.0717  73.8      786.     0     0     0     0     1
## # ... with 23 more variables: 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 $(y_{m,i,t} - \overline{y}_{EU28,i,t})^2$ bewteen 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    DK     EE    EL
##     <dbl>  <dbl>   <dbl>   <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>
##   1 11.4    7.94 1.21e+2 57.5    17.5  1.64e+0 116.   0.612  22.3
##   2 12.5   10.7  8.23e+1 58.2    10.4  3.95e-1  92.7  1.77   15.8
##   3  0.243  4.44 4.50e+1 60.7     4.81 5.10e-5 104.   5.26   13.0
##   4  3.91   3.69 4.25e+1 35.7     5.19 9.58e-1  91.7 12.8    16.2
##   5  4.16   9.37 1.99e+1 38.9     2.69 2.37e+0  96.8 40.2    15.7
##   6  4.84   8.41 4.84e+0 38.4     1.96 5.29e+0  70.6 39.7    23.0
##   7  7.98   8.85 7.56e-2 30.5     2.03 9.15e+0  59.7 37.5    21.9
##   8 19.3    3.62 4.14e-2 39.6     3.60 2.70e+1  50.4  0.993  11.6
##   9 33.5    0.264 1.17e+1 47.4    5.22 4.74e+1  46.0  1.73   18.6
##  10 37.7    0.579 2.66e+1 28.5    8.06 7.12e+1  45.4  6.45   71.6
##  11 41.0    0.640 2.50e+1  4.84  12.2  7.92e+1  39.7 17.6   169
##  12 43.0    0.710 2.06e+1  0.710 19.9  8.57e+1  39.2 27.6   229.
##  13 27.3    2.81 1.50e+1  1.89   20.5  7.61e+1  32.8 28.4   246.
##  14 18.6    7.74 8.31e+0  4.34   23.2  6.43e+1  29.4 42.5   227.
##  15 14.4   11.0  1.10e+1  5.34   32.4  5.76e+1  24.9 31.2   219.
##  16  8.37  16.1  1.46e+0  2.91   35.9  4.48e+1  16.8 38.4   216.
##  17  5.52  17.2  2.10e+0  0.0025 36.6  3.66e+1  13.3 31.9   206.
## # ... with 19 more variables: 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>
```

It is also possible to decompose the numerator of the variance, called deviance, at each time in order to appreciate the percentage of contribution provided by each member state to the total deviance,

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

for member state m, indicator i at time t, and where s enumerates Member States in the considered cluster.

```
res$res$devianceContrib

## # A tibble: 17 x 28
##         AT     BE      BG      CY    CZ      DE    DK     EE    EL
##      <dbl>  <dbl>   <dbl>   <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>
##  1 1.02   0.706  1.08e+1 5.11e+0 1.56  1.46e-1 10.3  0.0544  1.98
##  2 1.26   1.08   8.30e+0 5.87e+0 1.05  3.99e-2  9.35 0.178   1.59
##  3 0.0267 0.488  4.95e+0 6.68e+0 0.529 5.61e-6 11.4  0.578   1.43
##  4 0.456  0.430  4.95e+0 4.16e+0 0.605 1.12e-1 10.7  1.49    1.88
##  5 0.479  1.08   2.29e+0 4.48e+0 0.309 2.73e-1 11.1  4.63    1.81
##  6 0.578  1.00   5.78e-1 4.59e+0 0.234 6.31e-1  8.42 4.74    2.75
##  7 0.993  1.10   9.41e-3 3.80e+0 0.253 1.14e+0  7.43 4.67    2.72
##  8 2.72   0.511  5.84e-3 5.59e+0 0.507 3.81e+0  7.10 0.140   1.63
##  9 4.36   0.0344 1.52e+0 6.17e+0 0.680 6.17e+0  6.00 0.225   2.42
## 10 4.31   0.0662 3.04e+0 3.26e+0 0.922 8.14e+0  5.19 0.737   8.18
## 11 4.09   0.0639 2.50e+0 4.83e-1 1.22  7.91e+0  3.96 1.76   16.9
## 12 3.90   0.0644 1.87e+0 6.44e-2 1.80  7.77e+0  3.55 2.51   20.8
## 13 2.73   0.280  1.50e+0 1.89e-1 2.05  7.61e+0  3.28 2.83   24.6
## 14 2.02   0.839  9.01e-1 4.70e-1 2.52  6.97e+0  3.18 4.61   24.7
## 15 1.63   1.25   1.25e+0 6.08e-1 3.68  6.55e+0  2.83 3.56   25.0
## 16 1.04   1.99   1.80e-1 3.60e-1 4.44  5.54e+0  2.07 4.74   26.8
## 17 0.703  2.19   2.68e-1 3.18e-4 4.66  4.66e+0  1.70 4.06   26.2
## # ... with 19 more variables: 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>
```

Thus, each row adds to 100.

Moreover, it is possible to produce a graphical output about the main features of country time series. To this end, the function `graph_departure()` is implemented in the **convergEU** package. For the example here considered, the graphical output is shown below (Figure 13):

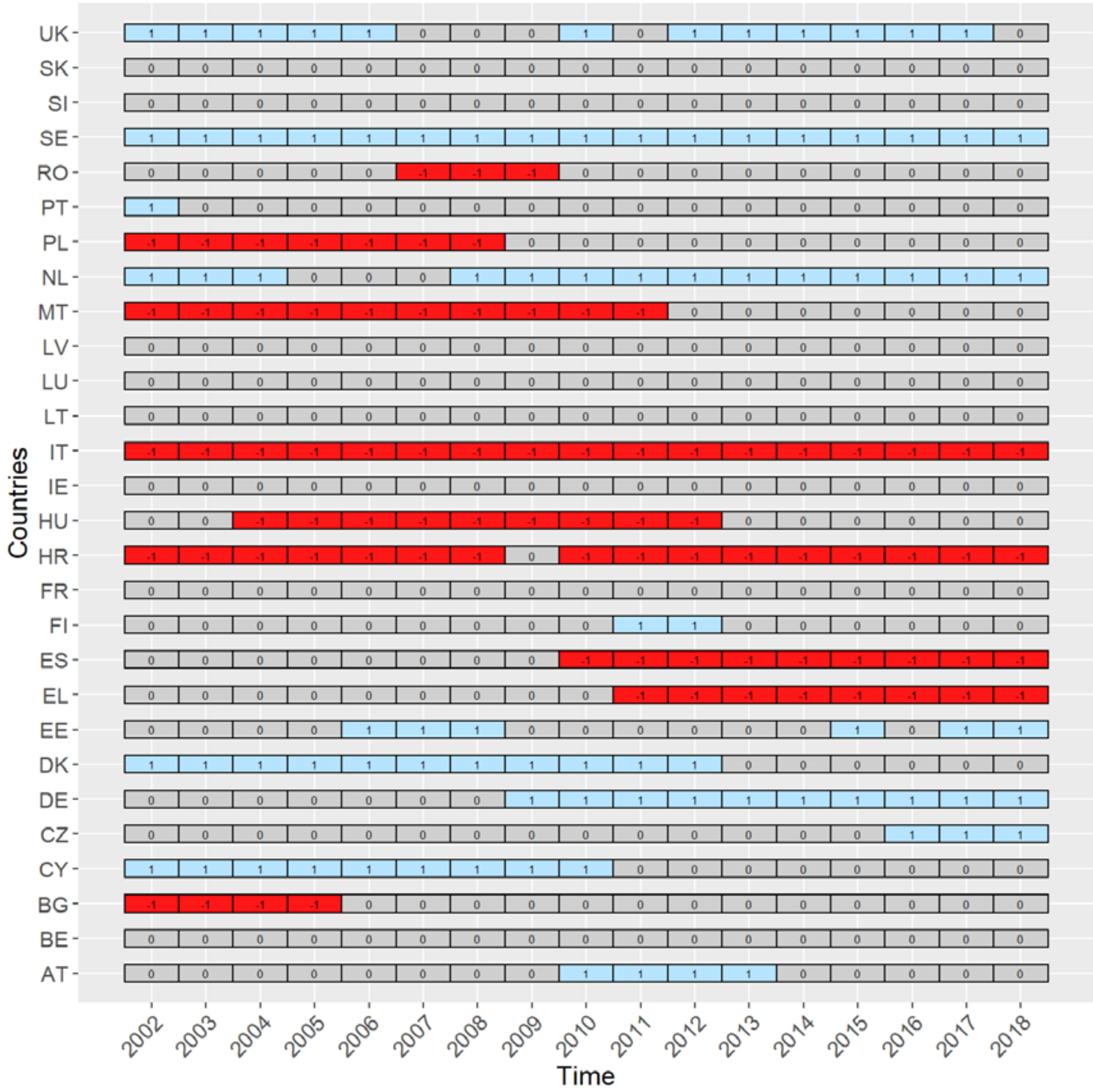
```
myGD <- graph_departure(res$res$departures,
              timeName = "time",
              displace = 0.25,
              displaceh = 0.5,
              dimeFontNum = 2,
              myfont_scale = 1.25,
              x_angle = 45,
              color_rect = c("-1"='red1',
                             "0"='gray80',
                             "1"='lightskyblue1'),
              axis_name_y = "Countries",
```

```
                axis_name_x = "Time",
                alpha_color = 0.9
                )
myGD

## $res
## $msg
## NULL
##
## $err
## NULL
```

Any selection of member states is feasible, as shown in the following example:

```
myGD1 <- graph_departure(res$res$departures[,1:10],
                timeName = "time",
                displace = 0.25,
                displaceh = 0.45,
                dimeFontNum = 4,
                myfont_scale = 1.35,
                x_angle = 45,
                color_rect = c("-1"='red1',
                               "0"='gray80',
                               "1"='lightskyblue1'),
                axis_name_y = "Countries",
                axis_name_x = "Time",
                alpha_color = 0.29)
```

Figure 13: qualitative features of time series



5.4 Countries ranking

In the **convergEU** package, ranking of countries by each considered years is implemented through the `country_ranking()` function:

```
help(country_ranking)
```

The argument `typeInd` allows to select according to which type of indicator the ranking should be performed: `highBest` ("higher is the best") or `lowBest` ("lower is the best"). For example, for the `emp_20_64_MS` data, the ranking is performed according to the `highBest` indicator; thus, the ranking from 2005 to 2010 is:

```
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    DK    EE    EL    ES
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  2005    13    19    24     4    11    14     2     9    21    18
## 2  2006    11    19    22     4    14    15     1     3    21    18
## 3  2007    11    20    19     4    15    10     2     3    22    17
## 4  2008    10    21    16     5    14     9     2     3    22    20
## 5  2009     8    17    15     4    11     5     3    13    21    24
## 6  2010     6    14    18     3    10     3     5    15    23    24
## # ... with 18 more variables: 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
```

5.5 Departures from the best performing country

The function `departure_best()` in the **convergEU** package allows to calculate for each country the departure from the best performing member state. For example, for the `emp_20_64_MS` data by considering the `highBest` indicator:

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

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

It is also possible to plot for each country the deviations from the best performer. To this end, the function `departure_best_plot()` is specifically implemented. Thus, the plot of the deviations from the best performed for Italy, considered as country of interest, and Germany, France and Sweden considered as reference countries, is as follows:

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

Note that in the argument `mainCountry`, the country of interest should be specified (that is Italy in the example), while the argument `countries` reports the set of countries with respect to which the deviations are plotted.

5.6 Scoreboards of countries

The basis of scoreboard are raw values of an indicator (level, $y_{m,i,t}$) for member state m at time t for indicator i. Differences among subsequent years (changes) are also very important, namely:

$$y_{m,i,t} - y_{m,i,t-1} \qquad (15)$$

Thus, the function `scoreb_yrs()` is implemented in this package to calculate these differences; for example for the `emp_20_64_MS` dataset:

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

The result is a list of three components: the summary statistics, the numerical labels to indicate the interval of the partition a level belongs to, the interval of the partition a change belongs to.

Numerical labels are assigned as described below (Mascherini et al., 2018):

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

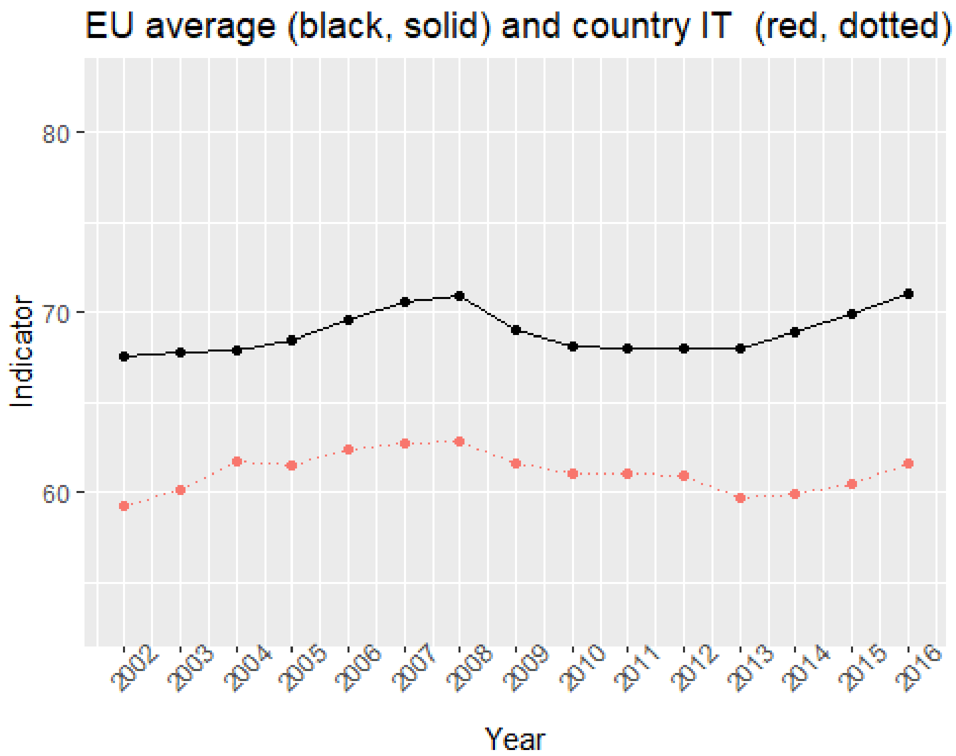
where $\overline{y}_{i,t}$ is the average at time t of the indicator i, and $s_{i,t}$ is the standard deviation.

For the comparison of a country with the EU average (Figure 14), the following steps are recommended, 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[[tim
eName]] <= 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("Year") + ylab("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]) +
   ggplot2::theme(
         axis.text.x=ggplot2::element_text(
         angle = myx_angle
         ))

myG2
```

Figure 14: comparison of Member States with EU



It is also possible to graphically show departures in terms of the above defined partition (Figure 15):

```
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 = "Countries",
                  axis_name_x = "Time",
                  alpha_color = 0.9
                  )
my_MSstd
```

Figure 15: qualitative representation of departure

6. Discussion and final remarks

The **convergEU** package enables a fast calculation of the measures of convergence for EU Members States. The use of the package does not require advanced programming skills to widen its usage to non-programmers who wish to analyse convergence. Nevertheless, a future implementation based on Shiny technology (Chang et al., 2019) could be even simpler to use, and accessible even on tablets and smartphones if connected to Internet.

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 economic scientists, than for an efficient implementation; anyway, performances in terms of speed and RAM space are not critical in the current implementation.

Some refactoring will take place in the future as well as the implementation of more output options, like pdf as native format of compiled fiches.

A software implementation to calculate measures of convergence is also available in STATA (Bisello and Mascherini, 2019), a widespread proprietary statistical software, but our R package, as of February 2020, 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.

The **convergEU** package may also be 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 is another area of paramount importance due to the role that a classification may play in policy making.

References

Biselllo, M., Mascherini, M. (2019), *Hitchhiker - version 3.0 january 2020 a toolkit for upward convergence analysis in Europe. An application to the employment rate (age 20-64) from 2002 to 2016.* Eurofound, Dublin.

Chang, W., Cheng, J., Allaire, J., Xie, Y. and McPherson, J. (2019), 'shiny: Web Application Framework for R', R package version 1.4.0, available at

https://cran.r-project.org/web/packages/shiny/index.html.

Heichel, S., Pape, J. and Sommerer, T. (2005), 'Is there convergence in convergence research? An overview of empirical studies on policy convergence', *Journal of European Public Policy*, Vol 12, No. 5, pp. 817–840.

Mascherini, M., Bisello, M., Dubois, H. and Eiffe, F. (2018), 'Upward convergence in the EU: concepts, measurements and indicators', Publications Office of the European Union, Luxembourg, available at http://eurofound.link/ef18003, doi: 10.2806/68012.

Signorelli, M. (2005), 'Growth and employment: comparative performance, convergences and co-movements', Technical report, University of Perugia, Perugia, Italy.

Solow, R. M. (1956), 'A contribution to the theory of economic growth', *The Quarterly Journal of Economics*, Vol 70, No. 1, pp. 65–94.

Stefanini, F. M., Nikiforova, N. D., Litardi, C., Peruffo, E. and Mascherini, M. (2020), 'Tutorial: analysis of convergence with the convergEU package', Vignette, **convergEU** R package, available at 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, available at

https://cran.r-project.org/web/packages/caTools/index.html.

Wickham, H. and Bryan, J. (2019), 'readxl: Read Excel Files', R package version 1.3.1, available at https://cran.r-project.org/web/packages/readxl/index.html.

Wickham, H. and Grolemund, 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. and Francois, R. (2018), 'readr: Read Rectangular Text Data', R package version 1.3.1, available at https://CRAN.R-project.org/package=readr.

WPEF20008

The European Foundation for the Improvement of Living and Working Conditions (Eurofound) is a tripartite European Union Agency established in 1975. Its role is to provide knowledge in the area of social, employment and work-related policies according to Regulation (EU) 2019/127.