TECHNICAL REPORT

ESPON

June 2013

A strategy for elaboration and update of coherent time series of hierarchical territorial units.

SUMMARY

This technical report proposes a coherent strategy for the collection of coherent time series of core data. This strategy involves the following topics:

1. Identification of problems encountered until now in the estimation of missing values and outlier check for time series of count data
2. Definition of conditions for a long term strategy
3. Proposal of a coherent solution to be implemented before the end of M4D project and likely to be further developed and improved by next ESPON program.

ESPON M4D -
MULTI DIMENSIONAL DATABASE DESIGN & DEVELOPMENT

EUROPEAN UNION
Part-financed by the European Regional Development Fund
INVESTING IN YOUR FUTURE

32 pages

LIST OF AUTHORS

Claude Grasland, UMS 2414 RIATE

Ronan Ysebaert, UMS 2414 RIATE

Timothée Giraud, UMS 2414 RIATE

Martin Charlton, NCG

Alberto Caimo, NCG

Hélène Mathian, UMR Géographie-cités (?)

Contact

ronan.ysebaert@ums-riate.fr

UMS 2414 RIATE
Tel. (+ 33) 1 57 27 65 35

TABLE OF CONTENT

1 Introduction

The Core Database Strategy is an important part of the ESPON M4D project, corresponding, in general, to activities developed in work package B (Thematic group) but also, to some extent, to activities developed in work package A (e.g. storage of time series, identification of core data in the interface) and work package C (identification of outliers value, quality check).

This activity which is normally supposed to cover half of the activity of the ESPON M4D project (according to contract definition) has been in practical terms delayed to the final period of activity 2013-2014. The reason of this delay was the priority decided by ESPON CU on storage and diffusion of data collected by ESPON project through the web interface. As a result, the major part of available workforce has been concentrated on this part of the work in 2011-2012.

The opportunity of this priority will not to be discussed here. But it is nevertheless clear that Core Database Strategy remains a major contractual obligation for M4D (deliverables related to this task has been delayed, not removed). And it is also very clear that, in a long term perspective, the Core Database Strategy is as important as the storage of data collected by ESPON 2013 projects.

If a new ESPON is launched for the period 2014-2020, the priority in terms of data collection will necessarily be the collection of the basic count data (population, activity, production, land use) and their elaboration in time series as long as possible. If such data are not available immediately, many difficulties will be encountered by new project, as we now by experience of what happened in the beginning of the programming period 1999-2006 and 2007-2013.

Moreover, if we consider a cross-programming period perspective, we can consider that a major added value of the ESPON program could be to produce cumulative efforts which mean, in practical terms, to enlarge past time series in order to be able in the future to propose more accurate previsions. With coherent time series covering the period 1990-2010, it is reasonably possible to expect accurate predictions for the period 2010-2030, which is a major wish of stakeholders and policy makers.

The problem is that building such long term time series is a very difficult and complicated task, that can only be engaged for a limited set of indicators. The aim of the Core Database Strategy is precisely to define what are the indicators to be completed in priority in order to derive many others by intelligent procedures of aggregation, disaggregation, spatial analysis, etc…

Because of pressure on other objectives in 2011-2012, the elaboration of the long term time series of core data has been until now limited to few indicators. This is not really a problem as long as we have demonstrate that a lot of information can be derived from a limited number of core indicators. But what is more tricky is the fact that :

- Estimation of time series of core data is actually based on a manual procedure that consumes a lot of time and is difficult to replicate
- Outlier check of time series of count data is difficult to realize because the indicators are not ratio but absolute count, which limit the use of a lot of methods of outlier detection.

2 Diagnosis of the difficulties encountered with time series of count data

Time series of count data are very specific statistical object that cannot be handled with the same procedure of outlier check and estimation. To illustrate this point, let us start with a basic analysis of the time series of population at NUTS3 level from 1990 to 2010 delivered by M4D project.

2.1 Heterogeneous sources and heterogeneous methods of estimation

The elaboration of a complete table of population for all NUTS3 regions implies a very huge amount of empirical work by human specialist in order to remove every missing value from the table. Until now, the strategy developed by M4D has been (1) *to choose the best available data and the best method of estimation for the estimation of each missing value and (2) to store all metadata related to the various sources and the various methods of estimation*. This strategy is illustrated in Figure 1 where sample of data and metadata are displayed.

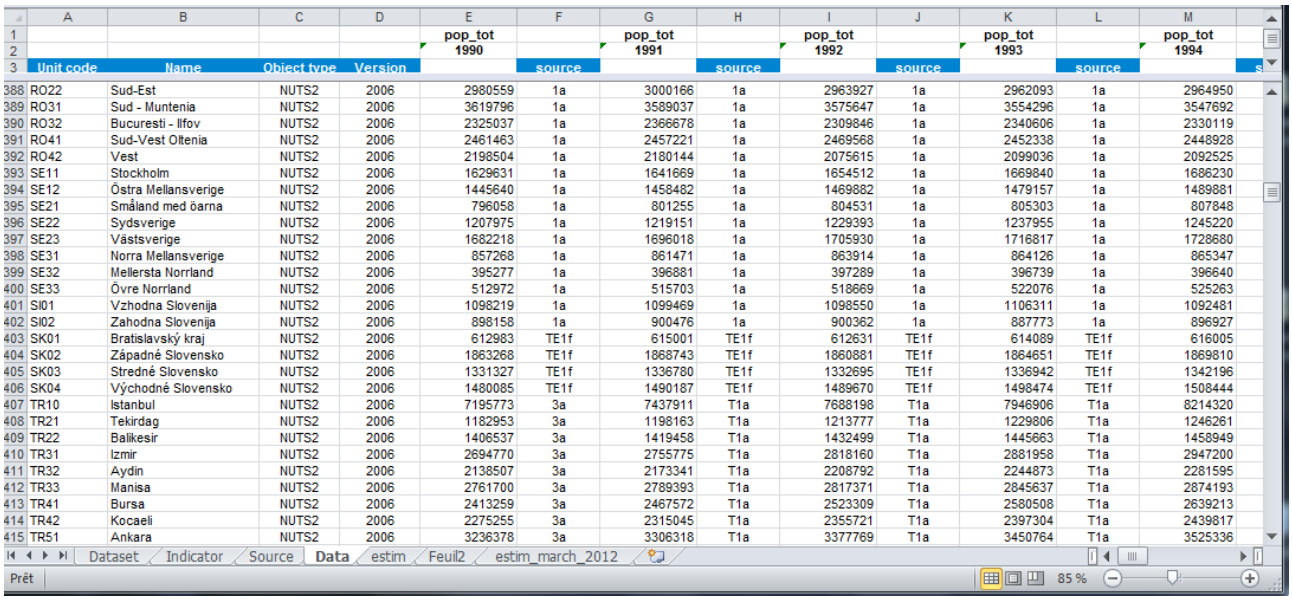
What are the strengths or weaknesses of this solution? The answer is not obvious because each strategic choice has a double face.

The multiplication of sources is apparently an obligation because no data provider is able to provide complete time series for the period at the target territorial level (NUTS3, Version 2006). Eurostat is generally chosen has prior provider but in many cases the missing data are necessarily collected through complementary sources provided by National Statistical Offices (NSI) of the countries. The problem is of course that NSI does not necessary use the same territorial division than Eurostat (risk of error) or the same definition. Even when it is the case, it can happen that updates are made by NSI on data that are not transmitted to Eurostat, or only with delays. In this case different figures will characterize the same territorial unit at the same time, according to NSI and Eurostat. And it is not obvious to decide on what is the best one: Eurostat has a political legitimacy at EU level, but NSI are the highest legitimacy at national level and are at less the responsible of initial data collection. Our purpose is not to solve this theological question but simply to underline the fact that mixture of several sources can increase the risk of "breaks" in time series.

The multiplication of estimation methods for missing values lead to the same dilemma. M4D project has proposed a catalogue of solutions that are well documented and help the human experts to choose the best one in each particular situation. But some of these methods are very sophisticated and can only be applied without error by very few human experts. Moreover, the work of estimation is actually done manually (by "click" in an excel sheet) and cannot be automatically reproduced, even when the detailed method is précised in metadata file. This is a real problem when the point is to update time series because new information are added (e.g. publication by Eurostat of new figures of population for 2010) or when old information are modified (e.g. replacement of provisional figures by definitive ones). A classic example is the "break" in time series introduce by the result of a census : the estimation used between two census date should normally be modified but in practical term it is generally not the case, creating automatically a time outlier at census date.

Figure 1 : The M4D estimated time series of population (1990-2010)

(a) sample of data

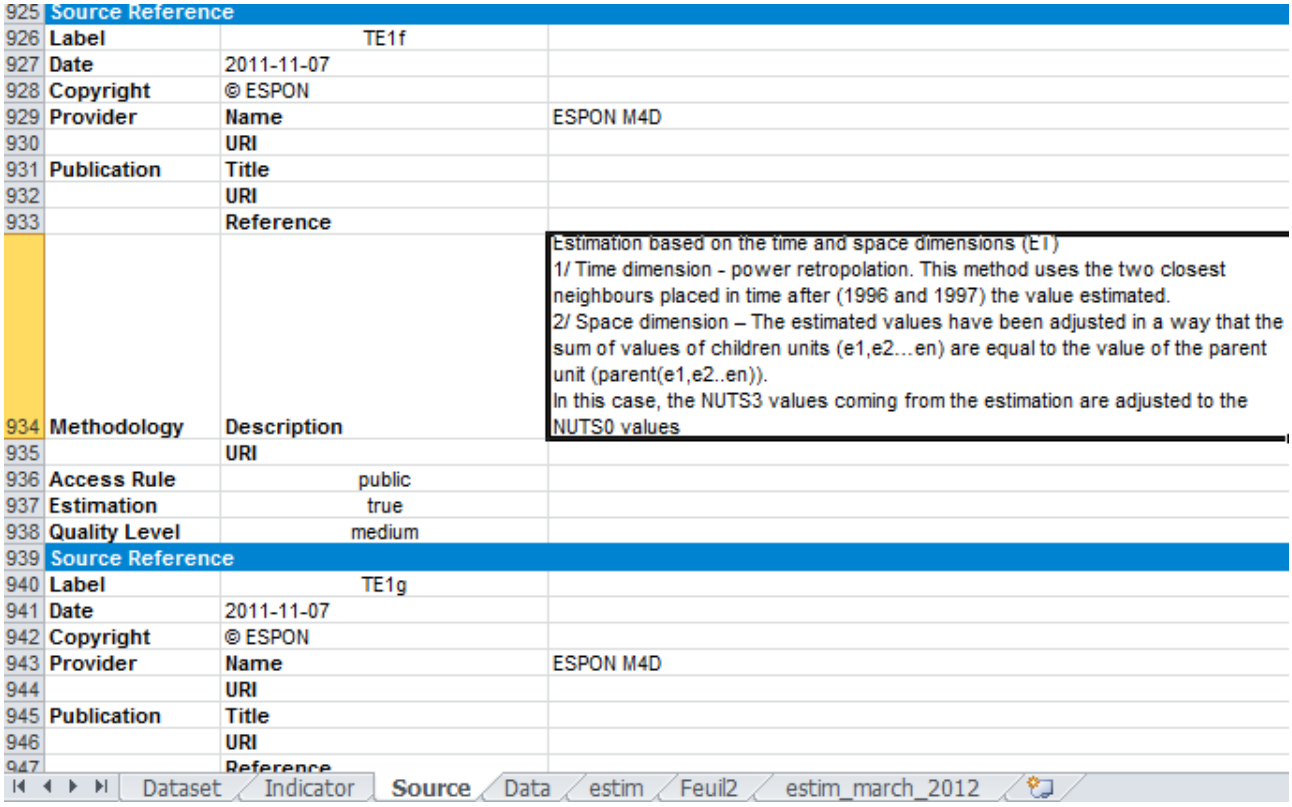


	A	B	C	D	E	F	G	H	I	J	K	L	M
1					pop_tot		pop_tot		pop_tot		pop_tot		pop_tot
2					1990		1991		1992		1993		1994
3	Unit code	Name	Object type	Version		source		source		source		source	
388	RO22	Sud-Est	NUTS2	2006	2980559	1a	3000166	1a	2963927	1a	2962093	1a	2964950
389	RO31	Sud - Muntenia	NUTS2	2006	3619796	1a	3589037	1a	3575647	1a	3554296	1a	3547692
390	RO32	Bucuresti - Ilfov	NUTS2	2006	2325037	1a	2366678	1a	2309846	1a	2340606	1a	2330119
391	RO41	Sud-Vest Oltenia	NUTS2	2006	2461463	1a	2457221	1a	2469568	1a	2452338	1a	2448928
392	RO42	Vest	NUTS2	2006	2198504	1a	2180144	1a	2075615	1a	2099036	1a	2092525
393	SE11	Stockholm	NUTS2	2006	1629631	1a	1641669	1a	1654512	1a	1669840	1a	1686230
394	SE12	Östra Mellansverige	NUTS2	2006	1445640	1a	1458482	1a	1469882	1a	1479157	1a	1489881
395	SE21	Småland med öarna	NUTS2	2006	796058	1a	801255	1a	804531	1a	805303	1a	807848
396	SE22	Sydsverige	NUTS2	2006	1207975	1a	1219151	1a	1229393	1a	1237955	1a	1245220
397	SE23	Västsverige	NUTS2	2006	1682218	1a	1696018	1a	1705930	1a	1716817	1a	1728680
398	SE31	Norra Mellansverige	NUTS2	2006	857268	1a	861471	1a	863914	1a	864126	1a	865347
399	SE32	Mellersta Norrland	NUTS2	2006	395277	1a	396881	1a	397289	1a	396739	1a	396640
400	SE33	Övre Norrland	NUTS2	2006	512972	1a	515703	1a	518669	1a	522076	1a	525263
401	SI01	Vzhodna Slovenija	NUTS2	2006	1098219	1a	1099469	1a	1098550	1a	1106311	1a	1092481
402	SI02	Zahodna Slovenija	NUTS2	2006	898158	1a	900476	1a	900362	1a	887773	1a	896927
403	SK01	Bratislavský kraj	NUTS2	2006	612983	TE1f	615001	TE1f	612631	TE1f	614089	TE1f	616005
404	SK02	Západné Slovensko	NUTS2	2006	1863268	TE1f	1868743	TE1f	1860881	TE1f	1864651	TE1f	1869810
405	SK03	Stredné Slovensko	NUTS2	2006	1331327	TE1f	1336780	TE1f	1332695	TE1f	1336942	TE1f	1342196
406	SK04	Východné Slovensko	NUTS2	2006	1480085	TE1f	1490187	TE1f	1489670	TE1f	1498474	TE1f	1508444
407	TR10	Istanbul	NUTS2	2006	7195773	3a	7437911	T1a	7688198	T1a	7946906	T1a	8214320
408	TR21	Tekirdag	NUTS2	2006	1182953	3a	1198163	T1a	1213777	T1a	1229806	T1a	1246261
409	TR22	Balikesir	NUTS2	2006	1406537	3a	1419458	T1a	1432499	T1a	1445663	T1a	1458949
410	TR31	Izmir	NUTS2	2006	2694770	3a	2755775	T1a	2818160	T1a	2881958	T1a	2947200
411	TR32	Aydin	NUTS2	2006	2138507	3a	2173341	T1a	2208792	T1a	2244873	T1a	2281595
412	TR33	Manisa	NUTS2	2006	2761700	3a	2789393	T1a	2817371	T1a	2845637	T1a	2874193
413	TR41	Bursa	NUTS2	2006	2413259	3a	2467572	T1a	2523309	T1a	2580508	T1a	2639213
414	TR42	Kocaeli	NUTS2	2006	2275255	3a	2315045	T1a	2355721	T1a	2397304	T1a	2439817
415	TR51	Ankara	NUTS2	2006	3236378	3a	3306318	T1a	3377769	T1a	3450764	T1a	3525336

Dataset | Indicator | Source | Data | estim | Feuil2 | estim_march_2012

Prêt 85 %

(b) sample of metadata



925	**Source Reference**		
926	**Label**	TE1f	
927	**Date**	2011-11-07	
928	**Copyright**	© ESPON	
929	**Provider**	**Name**	ESPON M4D
930		**URI**	
931	**Publication**	**Title**	
932		**URI**	
933		**Reference**	
934	**Methodology**	**Description**	Estimation based on the time and space dimensions (E1) 1/ Time dimension - power retropolation. This method uses the two closest neighbours placed in time after (1996 and 1997) the value estimated. 2/ Space dimension – The estimated values have been adjusted in a way that the sum of values of children units (e1,e2…en) are equal to the value of the parent unit (parent(e1,e2..en)). In this case, the NUTS3 values coming from the estimation are adjusted to the NUTS0 values
935		**URI**	
936	**Access Rule**	public	
937	**Estimation**	true	
938	**Quality Level**	medium	
939	**Source Reference**		
940	**Label**	TE1g	
941	**Date**	2011-11-07	
942	**Copyright**	© ESPON	
943	**Provider**	**Name**	ESPON M4D
944		**URI**	
945	**Publication**	**Title**	
946		**URI**	
947		**Reference**	

Dataset | Indicator | Source | Data | estim | Feuil2 | estim_march_2012

2.2) The dilemma: local precision versus global homogeneity

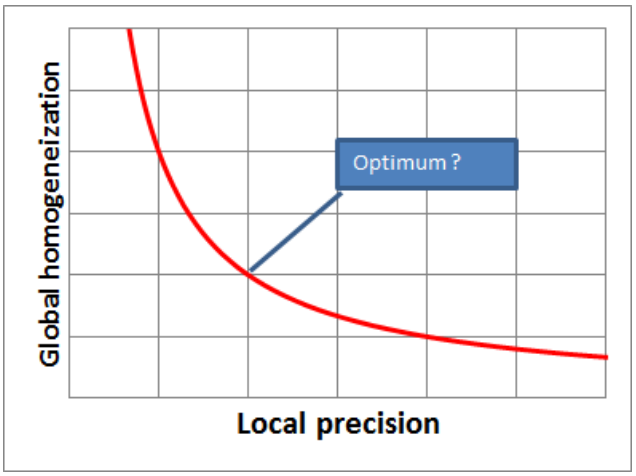
We propose to reconsider the strategic choice made until now to use the best available data or methods for the *local precision* of estimation of missing value and to examine an alternative where the focus is made on the global homogeneity of the solution. The main problem with the initial strategy (local precision) is the fact that all sources or methods employed can be perfectly correct, but at the end the emerging global result is not good. The problem clearly appeared when UMS RIATE started to realize a very basic outlier check of the time series of population, before to transmit to NCG for an in depth analysis of outlier combining all criteria. The very simple method we used revealed so much anomalies in time series that we decided to postpone the transmission of data to NCG and also decided to reconsider the opportunity to disaggregate or aggregate data with OLAP cube as long as we would not have understood the reason of the apparition of such a big number of time outliers (ex. Figure 2)

Figure 2 : Example of time outlier check for population (1990-2010)

code	Name	vpr1990	vpr1991	vpr1992	vpr1993	vpr1994	vpr1995	vpr1996	vpr1997	vpr1998	vpr1999	vpr2000	vpr2001	vpr2002	vpr2003	vpr2004	vpr2005	vpr2006	vpr2007	vpr2008	vpr2009	vpr2010	vpr2011
SK010	Bratislavský kraj	-1	1	0	0	0	0	0	0	-4	4	0	0	0	0	0	0	0	0	0			
SK021	Trnavský kraj	-4	3	0	0	0	-1	1	0	-1	0	0	0	1	0	0	0	1	1	-1	0		
SK022	Trenciansky kraj	-3	3	0	0	0	-1	0	0	1	-2	1	0	0	0	0	0	0	0	-1	0		
SK023	Nitriansky kraj	-4	3	0	0	0	-1	0	0	0	-1	1	0	0	1	0	0	0	0	0	0		
SK031	Zilinský kraj	-4	3	0	0	-1	1	-1	1	-1	-2	1	0	0	0	0	0	0	0	0	0		
SK032	Banskobystrický kraj	-4	3	0	0	0	-1	0	-1	0	1	-1	0	0	0	-1	0	0	0	0	0		
SK041	Presovský kraj	-4	3	0	0	-3	2	0	0	0	1	-1	0	0	0	0	0	0	0	1	0		
SK042	Kosický kraj	-4	4	0	0	-2	2	0	0	0	-2	1	0	0	0	0	0	0	0	1	-1		

The provisional diagnosis that we have made on population data lead to an interesting but striking conclusion: **the more we try to obtain exact value of isolated figure, the more we increase the number of outliers in time series**. To be sure, the objective of local optimization is to some extent contradictory with the objective of global homogenization of time series and we have to explore the possibility to define an optimum which is necessarily a compromise (Figure 3)

Figure 3: The compromise between local precision and global homogeneity of time series



2.3) A strategy based on the joint operation of data estimation and outlier check.

Our solution to the dilemma is not to choose one strategy against the other because both approaches are admittedly possible, depending on the user's needs.

- **The strategy of best local estimation** is typically convenient for users looking for official isolated figures and trying to answer to precise question like "*what was the population of Flanders in 1991?*". What is important for such users is to have very precise metadata defining the original sources (*e.g. Belgium NSI*) or the method of estimation used (*e.g. interpolation between year 1990 and year 1995 under the assumption of exponential growth*). For such a user, the discontinuities in time series are not a problem, precisely because they are looking for single time period or single units.
- **The strategy of global homogenisation** is typically convenient for users not interested in the analysis of specific situation but the examination of global trends in space or time. For example, a user trying to answer to a question like "*What has been the profile of population growth or decline of EU regions between 1990 and 2010*". In this case the degree of precision of a specific figure is not important at all. But discontinuities or outlier in time series are on the contrary a real danger for the analysis because they can introduce the apparition of specificities in time series that are purely artificial and related only to a change of sources or methods of estimation.

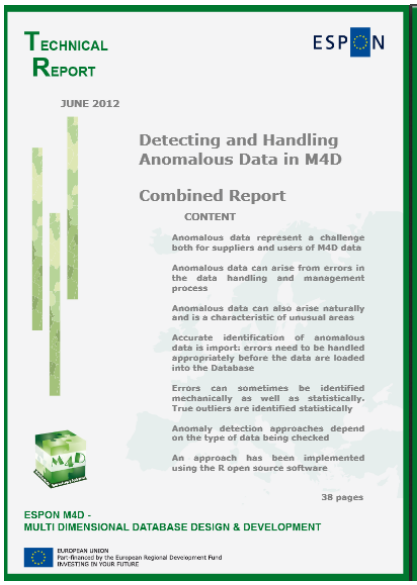
It is very clear that for the majority of data currently involved in the ESPON database, the strategy of local estimation appears as the best solution. But it is not the case for the long term time series called core data where the major interest is precisely to produce global evolutions and prospective results. We suggest therefore to adopt a new approach for this specific group of data that will be explained and developed in the next section.

The major originality of the approach proposed in this strategy is the fact that estimation of missing values and outlier check are not realized in separate steps but together. What we try to obtain is time series free that are eventually simplified or but that are fully consistent in temporal and territorial terms. And that can be also easily update and recomputed when new information are made available or when changes occur in territorial divisions (like the reform of NUTS).

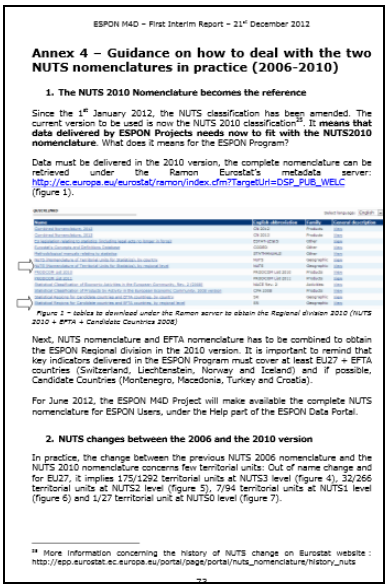
2.4) Relation with previous technical report or work done by M4D

The strategy proposed in the next section is the result of previous research done by NCG and RIATE. The reader can find more details in the following technical reports or annex of the interim report of M4D.

Work previously done by NCG



Word previously done by RIATE


ESPON M4D – First Interim Report – 21st December 2012

Annex 4 – Guidance on how to deal with the two NUTS nomenclatures in practice (2006-2010)

1. The NUTS 2010 Nomenclature becomes the reference

Since the 1st January 2012, the NUTS classification has been amended. The current version to be used is now the NUTS 2010 classification[34]. **It means that data delivered by ESPON Projects needs now to fit with the NUTS2010 nomenclature**. What does it means for the ESPON Program?

Data must be delivered in the 2010 version, the complete nomenclature can be retrieved under the Ramon Eurostat's metadata server: http://ec.europa.eu/eurostat/ramon/index.cfm?TargetUrl=DSP_PUB_WELC (figure 1).

Figure 1 – tables to download under the Ramon server to obtain the Regional division 2010 (NUTS 2010 + EFTA + Candidate Countries 2008)

Next, NUTS nomenclature and EFTA nomenclature has to be combined to obtain the ESPON Regional division in the 2010 version. It is important to remind that key indicators delivered in the ESPON Program must cover at least EU27 + EFTA countries (Switzerland, Liechtenstein, Norway and Iceland) and if possible, Candidate Countries (Montenegro, Macedonia, Turkey and Croatia).

For June 2012, the ESPON M4D Project will make available the complete NUTS nomenclature for ESPON Users, under the Help part of the ESPON Data Portal.

2. NUTS changes between the 2006 and the 2010 version

In practice, the change between the previous NUTS 2006 nomenclature and the NUTS 2010 nomenclature concerns few territorial units: Out of name change and for EU27, it implies 175/1292 territorial units at NUTS3 level (figure 4), 32/266 territorial units at NUTS2 level (figure 5), 7/94 territorial units at NUTS1 level (figure 6) and 1/27 territorial unit at NUTS0 level (figure 7).

[34] More information concerning the history of NUTS change on Eurostat website : http://epp.eurostat.ec.europa.eu/portal/page/portal/nuts_nomenclature/history_nuts

3. What is a time series?

3.1 Introduction

A time series is a collection of observations made sequentially in time[1].Time series arise in a number of areas of application and examples include:

- Economic time series – a classics examples include the Beveridge wheat price series and IBM daily closing stock price series
- Physical time series – hourly air temperatures at airports are collected and made available by the National Oceanic and Atmospheric Administration in the USA[2].
- Marketing time series – weekly sales figures for a particular product
- Demographic time series – mid year population estimates for a country or regions
- Process control – regular measurements are taken during a manufacturing process: should a parameter exceed a threshold, corrective action can be taken
- Binary processes – the position of a switch or valve can be monitored regularly
- Point processes – the dates of flood events at a particular location can be recorded with their dates (these series are irregular and the disribution of time intervals is of interest

The characteristics of a time series are that the measurements are order and that the observations are not independent. The exact prediction of future values of the series is rarely possible, because the series have a random component.

The objectives of time series analysis are twofold: description and prediction.

3.2 Time series description

In order to be able to consider making forecasts from a time series, it is necessary to know some it its characteristics. A series can be thought of as being composed of a number of sources of variation which come together additively. Each source of variation can be measured at each time period to yield the final series.

There are four sources of variation in a time series:

Trend: Chatfield defines trend as a 'long term change in the mean level' of the series. Trend may be increasing or decreasing, but it will be necessary to make some estimate of the trend in the series.

[1] Chatfield, C, 1989, The analysis of time series, 4th edition, London: Chapman and Hall
[2] For instance hourly time series of atmospheric conditions are Dublin Airport are available at http://weather.noaa.gov/weather/current/EIDW.html

Cyclic changes: business series are sometime observed to be influenced by long term business cycles, perhaps of 5-7 years duration. The solar magnetic activity cycle have as average duration of 11 years, and may affect climatic series. Hourly temperature measurements show diurnal variation, with warmer temperatures during daylight hours, and colder temperatures at night.

Seasonal variation: unemployment levels are typically higher in winter than summer; this seasonal component is well known, and can be measured and removed from the series to provide 'seasonally adjusted' measurements.

Residual variation: once the trend, cyclic and seasonal components have been removed from the series we are left with a set of values – these are the residuals, and may or may not show random variation.

A series where the trend and other systematic variation have been removed and which has constant variance is said to be **stationary**. One technique for removing trend is to difference the series:

$y_t = x_{t+1} - x_t$

This technique works well with non-seasonal series. If there is still trend in the series, or the variance changes, it may be necessary to difference the series again.

3.3 Modelling time series

There is a range of techniques available for modelling the variation in the series.

The **moving average** models the series as the average of the last k values, and this may or may not have different weights for each term in the series

$$s_t = \sum_{n=1}^{k} w_n x_{t+1-n}$$

If the weights $\{w_1, w_2, \ldots w_k\}$ and chosen to be 1/k, then the result is a **simple moving average**. In the **weighted moving average** are usually chosen to give more prominence to recent observations rather ones more temporally distant. The simple moving average does use all the information in the series at each estimate, since the equation can be rewritten as

$$s_t = s_{t-1} + \frac{x_t - x_{t-k}}{k}$$

The exponential moving average in its simplest form is:

$$s_1 = x_0$$
$$s_t = \alpha x_{t-1} + (1-\alpha)s_{t-1}$$

The question arises as to the 'correct' value for α. It can be computed from the data to yield a minimum value for the sum of the squared residuals $(x_t\text{-}s_t)^2$. The larger the value of α, the lower the smoothing in the resulting series. Again, the assumption is that the series has no trend. There are more complex versions of exponential smoothing, double and triple, which allow for trend and seasonal variation in the series.

A second class of models are known as autoregressive (AR) models. An autoregressive model of order p has the form

$$x_t = \alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + Z_t$$

The forecast series is regressed on itself. Z is a random process with a mean of zero and a variance of σ^2_Z. The moving average (MA) model of order q can also be written as

$$x_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q}$$

Which leads to the a mixed autoregressive integrated moving average model (ARIMA) of order (p,d,q):

$$x_t = \alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q}$$

The model is 'integrated' in the sense that the stationary model is fitted to the differenced data, and then must be summed (i.e. integrated) to give a model for the non-stationary data. Such models are closely associated with the forecasting methodology developed by Box and Jenkins[3].

A mathematical treatment of time series analysis would be both extensive and challenging, and while Chatfield's survey is accessible to an informed reader, Box & Jenkins' text is not for the faint hearted.

[3] Box GEP and Jenkins GM, 1970, *Time Series Analysis, Forecasting and Control*, San Francisco: Holden-Day

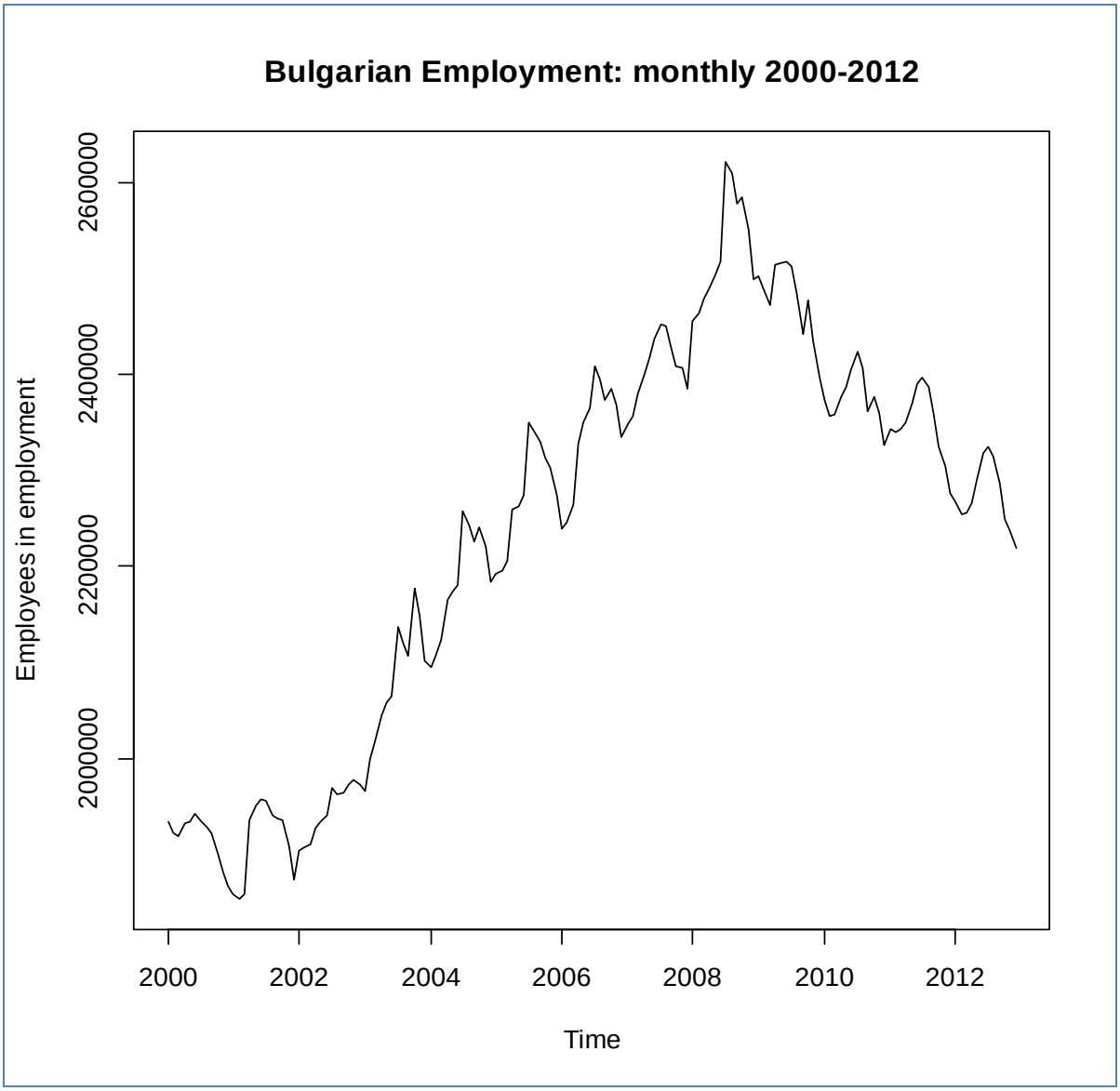
3.4 Example

An example time series might help to elucidate. The data represent monthly national counts of employees in employment for Bulgaria, from January 2000 to December 2012. The data may be downloaded from the Bulgarian National Statistical Institute website[4].

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2000	1934335	1923257	1920010	1932633	1934915	1942462	1934792	1929736	1923424	1901373	1882562	1867701
2001	1859801	1853911	1859421	1936101	1951598	1957874	1956898	1940798	1937798	1937008	1909720	1874537
2002	1904439	1907412	1911829	1928135	1934001	1941057	1969447	1963176	1964232	1972368	1978177	1973357
2003	1966409	2000342	2018247	2044893	2057766	2065991	2137397	2120412	2107736	2177785	2148702	2102561
2004	2095675	2109524	2124639	2166242	2173868	2180858	2257969	2243242	2226104	2241607	2221459	2183464
2005	2191894	2195119	2205970	2259663	2263378	2275258	2350430	2341150	2330249	2312621	2303541	2274631
2006	2239864	2246053	2263898	2328772	2349561	2365344	2408565	2395432	2373853	2385030	2367891	2335545
2007	2347755	2355835	2379551	2400343	2417169	2436369	2451607	2450865	2429024	2408166	2406337	2384903
2008	2455536	2464259	2478101	2488316	2503391	2517729	2621733	2608601	2578059	2584129	2551354	2499126
2009	2502133	2487997	2472024	2514441	2514767	2516415	2512344	2485137	2442424	2476424	2434357	2396144
2010	2373349	2355926	2357992	2375879	2386180	2405586	2423337	2406704	2362277	2377274	2360232	2326123
2011	2342509	2340307	2342500	2349300	2370022	2389415	2397351	2386131	2358628	2324646	2305109	2276111
2012	2268146	2254946	2255534	2265885	2289972	2318780	2325325	2314832	2285469	2248545	2238034	2220070

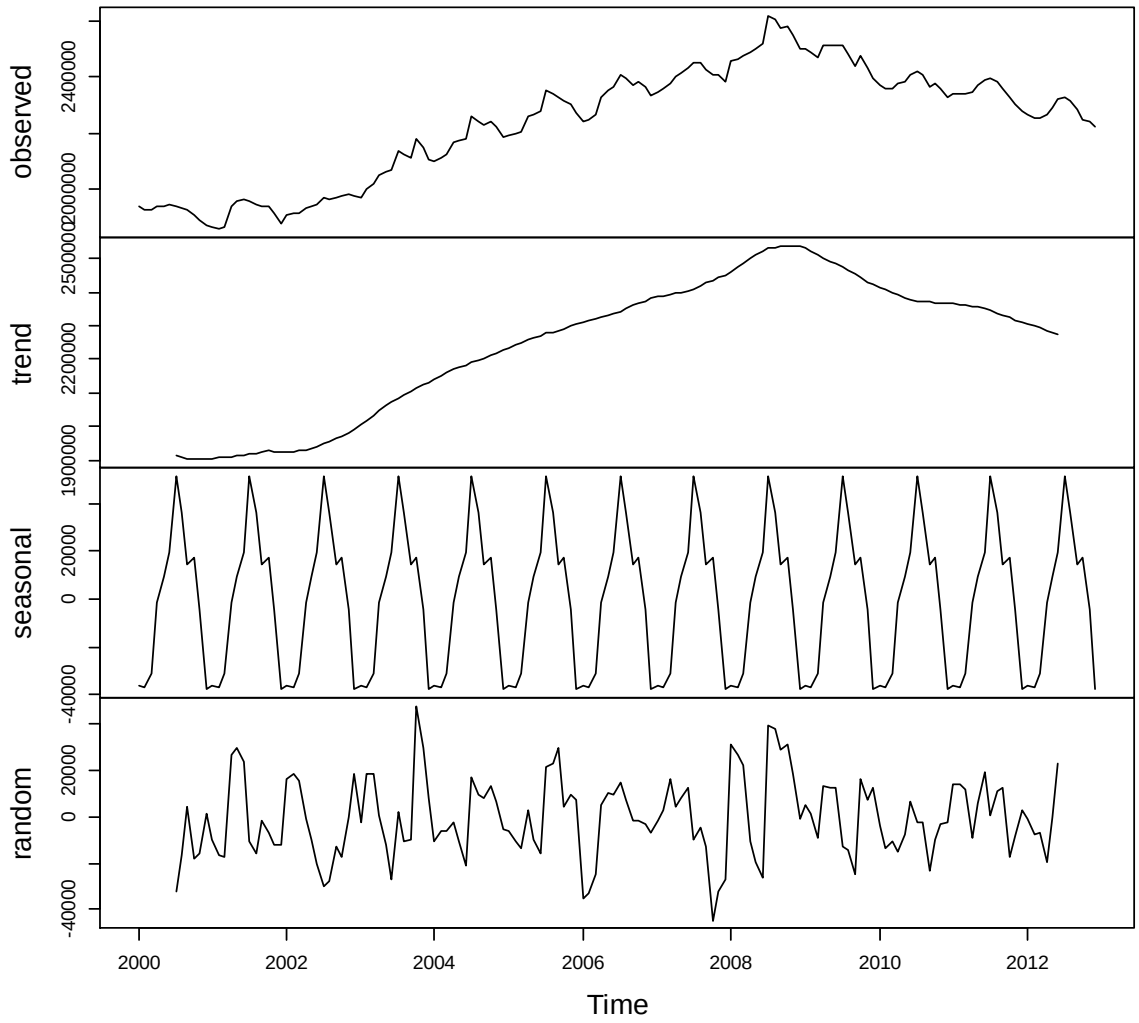
Much exploratory time series analysis involve visualisation, and a useful starting point is to examine a plot of the variation in the data over time:

[4] Employees under Labour Contract: http://www.nsi.bg/otrasalen.php?otr=51



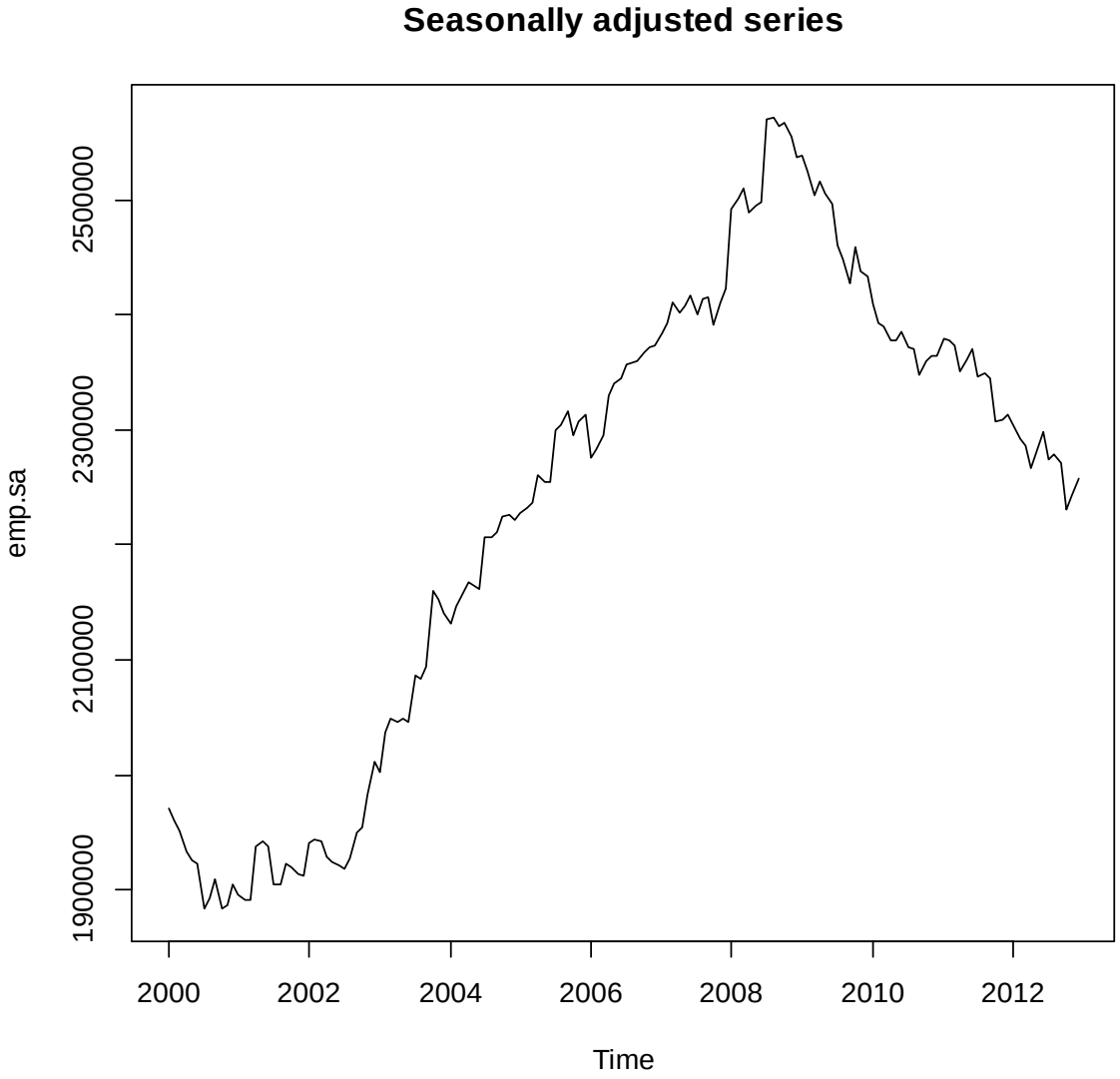
A number of features are immediately obvious from an examination of the plot. First, the series rises to a peak in late 2008, and following the trajectory of several European economies, drops. In other words, there is an upward trend during the first part of the series, followed by a downward trend. Second, within each annual time period, the employment peaks in the summer and troughs in the winter – there is evidence of seasonal variation.

The R language provides several functions to assist with the analysis an forecasting of time series. One of these, decompose(), will extract the trend and seasonal variation from the series, and compute the residuals.

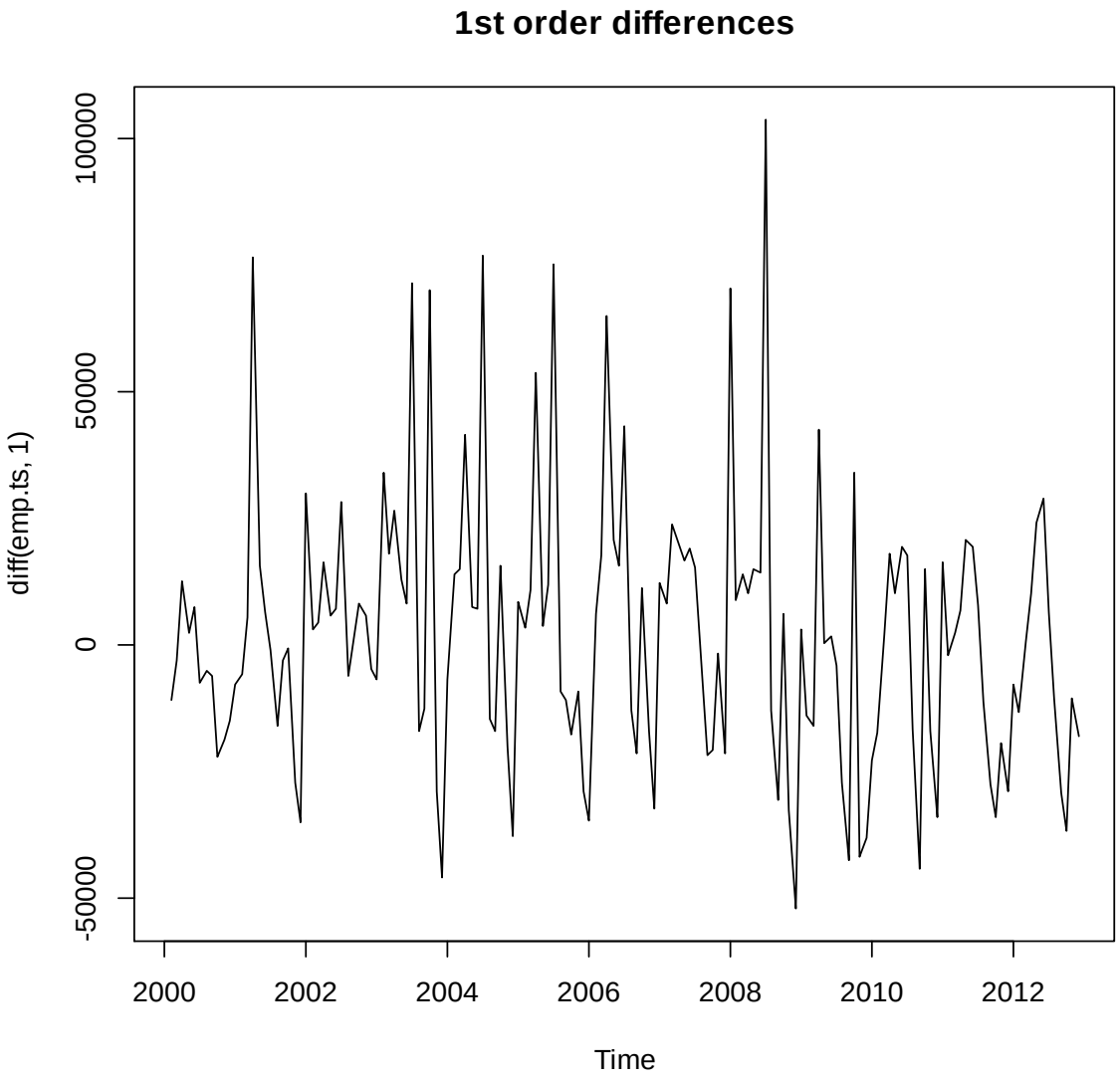


The topmost 'observed' panel in the figure shows the raw series - as in the previous figure. The second panel shows the trend component that has been extracted from the decomposed series. We can see that from 2000 through 2002 this is little growth after which there is rapid, and reasonably constant growth through the 2000s until 2009 when the economy starts contracting and sections of the workforce lose their jobs. This decline continues to the end of the series, December 2012. The third panel shows the seasonal component. An annual cycle is quite evident: after a couple of months of little change, growth is rapid rising to a mid-year peak. Employment starts to fall, apart from a small autumnal discontinuity, back to the winter levels. This seasonal component is more or less constant during the period of the series. The final panel shows the residual series - this is what remains after the trend and seasonal components have been removed.

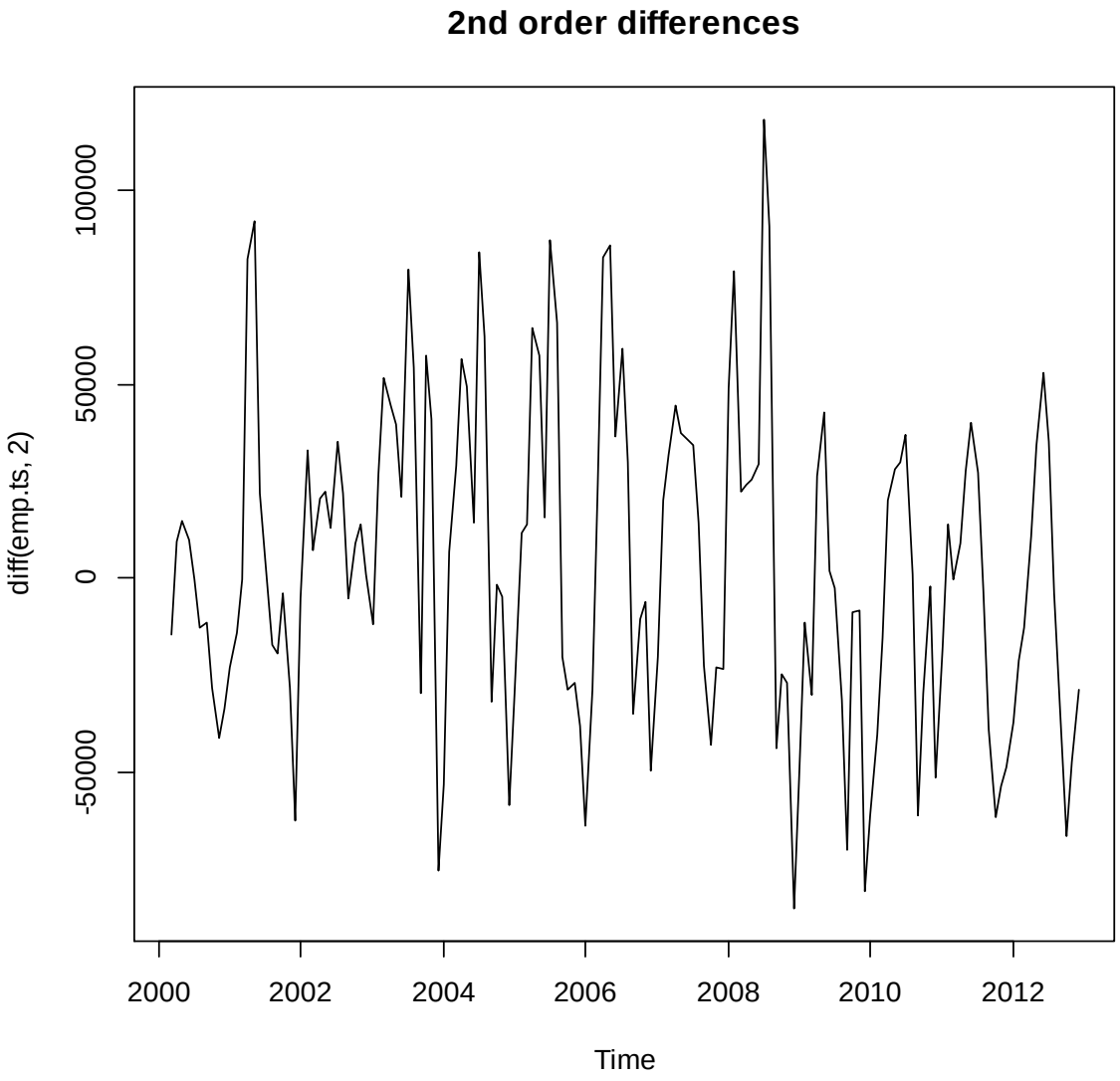
Further modelling would take place on the residual series - of particular interest is the relationship between successive elements of the series, not just elements with are adjacent, but elements which are several lags apart. A measure of the relationship is known as autocorrelation; a plot fo autocorrelation levels (the 'autocorrelation function') over time can be used a s diagnostic to yields some insights into the behaviour of the series, and which are the appropriate models for describing this behaviour.



The figure above shows the series with the seasonal component removed - there is still trend, and clearly there is also some variability about the series. To attempt to remove the trend the series can be differenced; that is, $d_t = x_t - x_{t-1}$. It may be necessary to difference the series twice is the trend itself is not constant. Forecasts are then made using the differenced de-seasoned series, and then the forecasts adjusted to add the trend and seasonal components back in.



The 1st order differences suggest that the variability around the series mean is not zero. To obtain a stationary series, we would difference it again.



Typical series models are from the ARIMA family: recall that this as the acronym for **A**uto**R**egressive **I**ntegrated **M**oving **A**verage, usually written ARIMA(p,d,q), where p is the degree of the autoregressive component, d is the degree of differencing, and q is the degree of the moving average component. The degree of the each component and the differencing is usually one of 0, 1, or 2. The various combinations allow 27 alternative models:

p,d,q	p,d,q	p,d,q
0,0,0	1,0,0	2,0,0
0,0,1	1,0,1	2,0,1
0,0,2	1,0,2	2,0,2
0,1,0	1,1,0	2,1,0
0,1,1	1,1,1	2,1,1
0,1,2	1,1,2	2,1,2
0,2,0	1,2,0	2,2,0
0,2,1	1,2,1	2,2,1

0,2,0 1,2,2 2,2,2

A plot of autocorrelation and partial autocorrelation functions may often suggest which might be the most appropriate model. This would be both time consuming and labour intensive for all 27 models.

A second approach is to use the fact that the fitting each model is not a task which occupies much computer time. It is possible to enumerate all 27 models and choose that which fits the observed series most closely is a short period of time.

A principle of fitting models generally is that of parsimony - it if preferable to fit a simpler model than a more complex model. However, if we add more terms to a model, in general the fit improves, and does not get any less good. To prevent overfitting, a measure of the goodness of fit is required, but one which also includes a penalty for the complexity of the model. A commonly used criterion is choosing between different models is the Akaike Information Criterion: - (2logL -2k) where L is the maximised likelihood and k is the number of parameter that have been fitted. If n is small relative to k a further penalty is the addition of (2k(k + 1) / (n - k- 1)).

The AIC is a measure of the unknown distance between the fitted model and the unknown true model, and as such is a relative distance. It can be used to compare the performance of two models (used the predict the same dependent data). If the AICs are subtracted that which has the lower AIC is held to be closer to the true model. If the difference between the AICs is less than 3, a rule of thumb has it that there is little to choose between the models[5].

The auto.arima() function in R can be used to find the parameters of the ARIMA model which best fits the observed data. It can be argued that in following such a course of action the analyst is abrogating all responsibility for choosing the most appropriate model to the computer. However, following the prescription of evaluating every one of the twenty-seven models would also lead the analyst to the same conclusion.

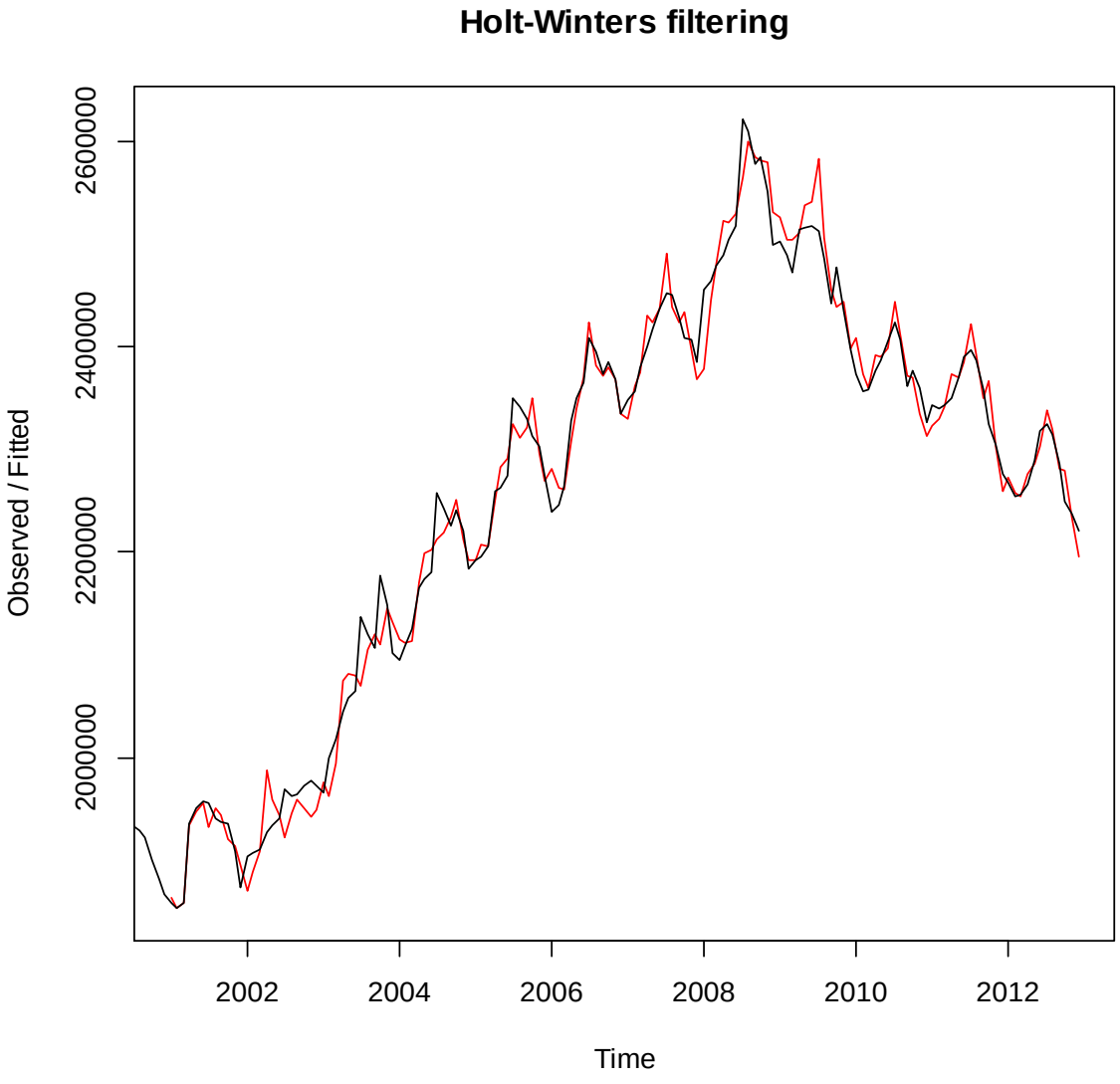
A simple model of a time series is known as the Holt Winters model. Tjis can be used for description as well as forecasting. Fitting an H-W model to the Bulgarian employment series yields the following parameters.

```
Smoothing parameters:
alpha: 0.696412
beta : 0.05633729
gamma: 0.9477416

Coefficients:
a   2251290.635
b     -4457.237
s1   -39630.142
s2   -46592.827
```

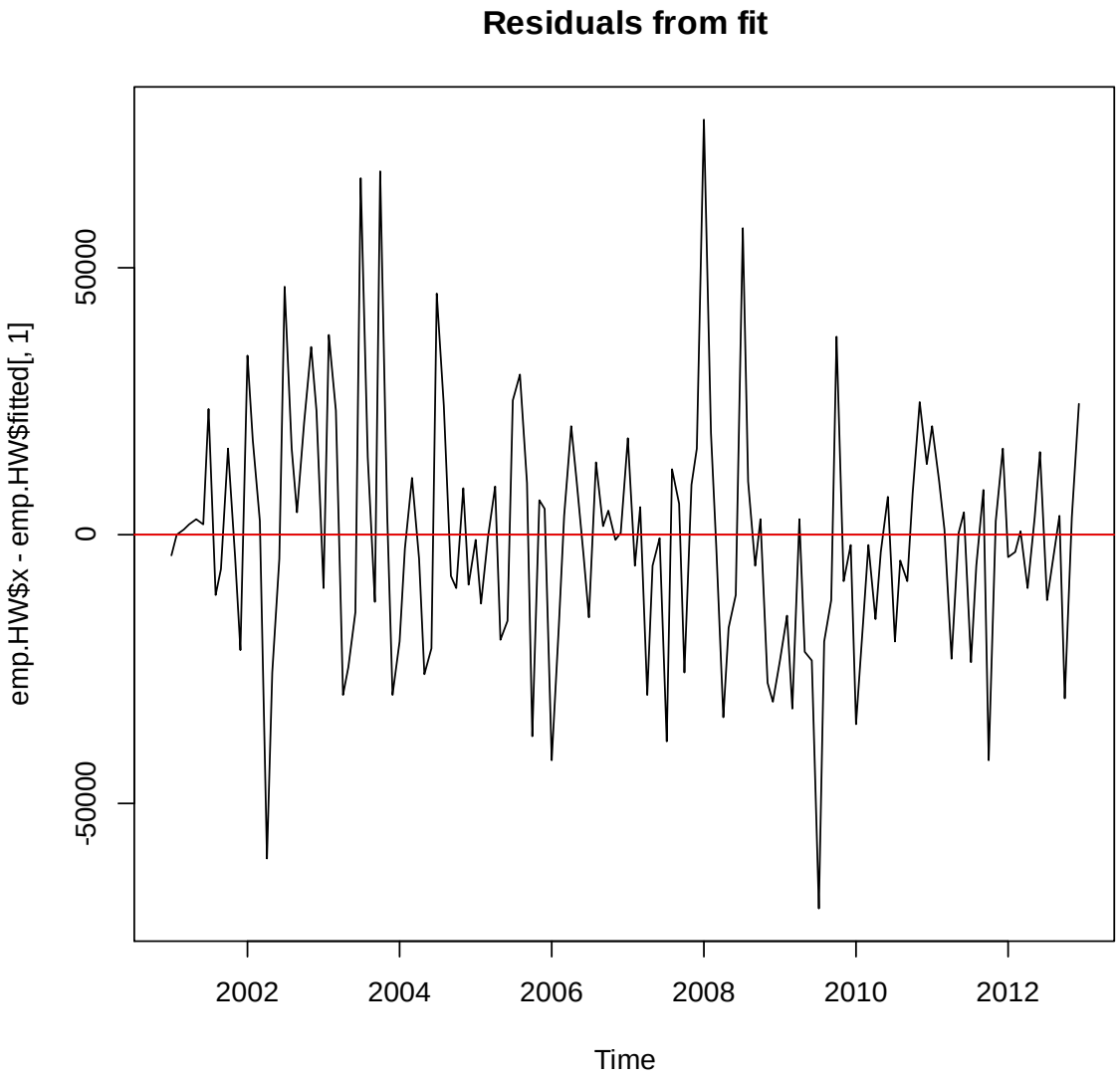
[5] If two models for the same dependent data have two parameters, it can be show that the difference between the AICs becomes a likelihood ratio test - the critical value is χ^2 with one degree of freedom: 3.84.

```
s3   -42355.593
s4   -20680.484
s5     5477.774
s6    27715.386
s7    46890.248
s8    44404.251
s9    16804.820
s10    6071.568
s11   -1694.399
s12  -31608.607
```

The parameters α, β, and γ are those which control the level, trend and seasonal components of the series. The parameters theselves are usually in the range (0,1). Note the high γ parameter which draws our attention to the strong seasonal variation within the series.



The prevous figure shows the original series and also the forecast for that series from the Holt-Winters fit. The predictions appear to be reasonable, but an examination of the plot of the residual suggests that we could probably do somewhat better.



The previous plot shows the residuals from the Holt-Winters fit to the Bulgarian employment data. Whilst there are some large residuals (over 50000), in general the predictions are within +/- 25000 of the real values - an error of about +/- 1%.

The Bulgarian series has 156 observations, so the estimates of the parameters are reasonably reliable. The ESPON series sometimes rarely have more than 20 observations, sometime less. This means that we have around 13% or less of the data for a reliable fit. The fact that many series a part of a hierarchy means that we may required other methods which are less dependent on a abundance of data and which lend themselves to constraint.

4 Missing Data and Imputation

4.1 Introduction

A characteristic of data is the occurrence of missing values. In a sample survey a missing value can arise because a respondent has chosen not to answer some of all of the questions. Missing values may also arise where combinations of alternatives are impossible: males do not get cervical cancer, females do not get testicular cancer, but both can suffer from breast cancer.

A missing value may arise because a measurement could not be taken - either the circumstances did not permit it (for example the 2001 Irish Census of Population took place in 2002 due t he prevalence of bovine 'foot and mouth' disease in the summer of 2001 with consequent restrictions on rural movement), or perhaps the equipment was faulty or was out of calibration. Another possibility which would give rise to missing data might be non-availability of a secondary source.

The question therefore arises as to the treatment of missing data. This then in turn, raises questions as to the model which led to the generation of the missing data.

We begin by considering the options for **cross-sectional data**. Much ESPON data is cross-sectional.

Available case (AC) analysis is a strategy for dealing with missing data in which only those observations with no missing data in the variables for the analysis are employed. This might appear on the surface to be a reasonable strategy, but it does rely on the missing data being both sparse and random. If there are 100 observations and 10 variables being used for a multivariate analysis, and for 50 of the observations at least one variable has missing data, then this strategy would lead to only half the observations being used for the analysis. Observation of the data that have been received for data checking as part of the ESPON Database project would suggest that the generating model is not random. This being the case, the application of AC would lead to potential bias in any analysis. However, it is widely used. The SPSS system allows for 'listwise' deletion in multivariate analysis; this is the AC approach.

A second approach is to use as much of the data as possible; a **modified AC** analysis. For example, we might be computing a correlation matrix. Indeed the crossproduct matrix (X^TX) is widely used in many multivariate techniques. This deals with variables a pair at a time; only cases for which both variables V1 and V2 are present are used in the computation of the V1V2 crossproduct. If there is missing data in V3 for different observations, then only those observations for which V1 and V3 are present are used, but these will be different pairs from those used in the computation of V1V2. Th eresult is that different cases are used to complete the cells in the matrix, although the same cases will be used for the V1V2 crossproduct as V2V1. And different

numbers of cases will be used in the completion of the diagonal elements, V1V1, V2V2, V3V3 and so on. This is sometimes known as 'pairwise' deletion, and can lead to instability in the subsequent computations. For these reasons this strategy cannot be recommended.

A third approach is to replace the missing data by the mean of the observations - sometimes known as **'mean substitution'**. This does not affect the subsequent computation of the means, but does affect the variances. It also may affect local spatial patterns – an imputed missing value in area of generally low values would be become an outlier. An alternative approach with spatial data would be to use the mean of the values in the neighbouring areas – again this may lead to incorrect imputation if the area with the missing value would have had a particularly high local value (for example Liechtenstein's GDP).

A fourth approach is known as **hotdeck**[6]. Comparable cases to those with missing values are identified (k-nearest neighbours can be used), and the imputation is from either taking the values directly or computing a mean of the comparable cases. The advantage of the hotdeck is that no new data is used, and all imputed values are taken from the dataset.

A final approach would be to use a **cross-sectional regression** fitted to data cases with all the variables, and then impute the missing data from the model. This does require a plausible model, however, but it does give the analyst some freedom in imputing the data. The disadvantage is that the imputed value is the expectation of the Y – a mean – so that some smoothing takes place in the imputation process. **Geographically weighted regression** would allow a model to be fitted to those cases with all variables, and then parameter estimates for the X data estimated at those locations where the Y data requires imputation. This may lead to a more plausible imputation that relying on spatiall global parameter estimates. Again, the analyst needs to develop a plausible model.

If there is a **longitudinal** component then there are a number of options. **Last Observation Carried Forward** (LOCF) is perhaps the simplest method. The value to be imputed for a variable at time t+1 is the value for the same observation at time t. This assumes that the values of the variables are more or less constant over time – differencing might be used if there is a trend in the data, and the series reconstructed after the imputation for the differenced values.

Linear interpolation is another commonly used method. The value to be imputed at time t is the average of the values at time t-1 and time t+1. Again this assumes linearity – or at least local linearity.

Longitudinal regression models the relationship between the time component and the variable in question: $y_t = \beta_0 + \beta_1 t$. This assumes linearity in the model adequately describes the relationship between the time counts and the data. Again, if the series is stationary, or has been differenced to make it so, then this may be a

[6] Myers T, 2011, Goodbye, listwise deletion: presenting hot deck imputation as an easy and effective tool for handling missing data, Communication Methods and Measures, 5(4), 297-310

reasonable model. A variant would allow the inclusion of additional covariates, each of which would themselves be time dependent. The model would be fitted to observations where the variables are all represented by non-missing values, as a training set, and the missing values for the y_t variable imputed accordingly.

All the method provide a single point estimate of the imputed value, which means that there is potential for over-smoothing in the results. A technique known as **multiple imputation**[7] exists which the imputation is carried out using repeated random subsets of the data. M subsets are taken, and the result is M imputed values for the missing value (with whatever model is used for imputation [hotdeck, for example]). A point estimate of the imputed value can be calculated from the M repetitions, and a variance estimated can also be made.

Twisk et al 2002 have observed that cross-sectional imputation leads to under-estimation of the standard errors[8]. They also found that the success of the multiple imputation method was highly dependent on the selection of the model for missingness. They also pointed out that where the data does not lend itself to the computation of a mean, that is, when it is dichotomous or categorical, then LOCF is the most frequently used imputation method. Finally where the data have both longitudinal and cross-sectional components, longitudinal methods are preferable to cross-sectional methods.

Tang el al report that AC analysis will show serious bias if the missing data mechanism departs substantially from 'missing completely at random'[9]. If repeated measures change substantially, LOCF may introduce bias. They also suggest that multiple imputation techniques perform better than expedient approaches such as AC or LOCF, although if many variables are not normal, then the hotdeck approach is a better choice again.

[7] Fay RE, 1996, Alterantive paradigms for the analysis of imputed survey data, *Journal of the American Statistical Association*, 91, 490-498

[8] Twisk J and de Vente W, 2002, Attrition in longitudinal studies: how to deal with missing data, *Journal of Clinical Epidemiology*, 55, 329-337

[9] Tang L, Song J, Belin TR and Unützer J, 2005, A comparison of imputation methods in a longitudinal randomized clinical trial, Statistics in Medicine, 24, 2111-2128.

5 Solution proposed for homogenization and update of times series of core data

5.1 General rules and objectives

The solution that will be developed in the following section is based on a limited number of rules that should normally be followed without exceptions, in order to fulfill precise objectives

1. *Only one primary source is normally used for the production of time series*. The fact to use different sources for the same territorial unit is indeed a major factor of creation of "breaks" or heterogeneity. It means that we will normally prefer to estimate values rather than use alternative data source.

2. *All times series should be perfectly consistent in terms of hierarchical aggregation of territories*. The different subdivision of data provided by a primary producer should be perfectly exact. If data provided by the initial producer does not follow this rule, they will be modified in order to fulfill perfectly the aggregation rules of the nomenclature.

3. *All time series should be free of time outlier, except when the outlier can be explained by concrete and real facts*. It means that we prefer to obtain values that are different from the official one when an obvious statistical bias is present in time series of the data producer. Typically, when a new census creates a discontinuity in the time series, we will recalculate the values between theses census and the previous one. More generally we will try in the majority of case to obtain stationary time series as long as we have no reason to suspect that specific event has created discontinuities.

4. *All estimation of missing values should be made by mean of an automatic procedure that can be repeated quickly and – ideally - without manual intervention*. This rule is the most difficult but also the most important because time series should be regularly modified for different reasons : (1) introduction of recent data provided by data producer ; (2) discovery of errors in existing data or modification of provisional values in definitive ones ; (3) discovery of new estimation methods that could improve previous ones.

5. *All procedures and methods used in the estimation should be transparent and added in the metadata field*. This general rule of the ESPON database is just reminded here but remains very important. The user of time series should be perfectly aware of the fact that data that are sometime different from "official statistics" because of the target of global homogeneity.

6. *An estimation of uncertainty should be ideally added to all figures of time series*. In principle, we do not need to introduce here an outlier check of this data because we have precisely decided to remove outliers. But we should ideally indicate the 95% confidence interval of values present in time series, not only for estimated values but also for the other ones.

5.2 A data model combining time and territorial hierarchy

The application of previous rules (1) and (2) lead us to propose a specific data model for the storage of time series describing hierarchical territorial units like NUTS. To illustrate the strategy, we will take the example of estimation of missing data for active population of Bulgaria between 1999 and 2010 on the basis of EUROSTAT data at NUTS0, NUTS1 and NUTS2 levels (version 2006). This specific data model can be firstly presented in tabular format (Figure 4) but is more clear if presented in form of hierarchical trees of data linked through time (Figure 5).

Figure 4 : Illustration of the strategy of hierarchical data reconstitution

Step1 : Initial data

code	name	level	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
BG	Bulgaria	NUTS0		2794.7	2702.8	2741.0	2834.7	2922.6	2981.9	3110.0	3252.6	3360.7	3253.6	3052.8
BG3	Severna i iztochna Bulgari	NUTS1					1412.0	1445.6	1476.3	1529.5	1581.7	1632.2	1571.9	1465.9
BG4	Yugozapadna i yuzhna tse	NUTS1					1422.8	1477.0	1505.6	1580.5	1670.9	1728.5	1681.7	1586.9
BG31	Severozapaden	NUTS2					315.7	318.3	314.6	327.7	345.4	359.3	341.3	313.7
BG32	Severen tsentralen	NUTS2					335.1	344.9	344.4	352.0	368.3	374.4	365.6	336.0
BG33	Severoiztochen	NUTS2					350.5	361.3	389.3	405.0	413.4	429.1	409.5	387.5
BG34	Yugoiztochen	NUTS2					410.6	421.1	428.0	444.8	454.6	469.4	455.6	428.7
BG41	Yugozapaden	NUTS2					855.4	894.5	920.7	974.1	1025.3	1060.2	1042.4	991.3
BG42	Yuzhen tsentralen	NUTS2					567.3	582.5	584.9	606.4	645.6	668.3	639.2	595.7

Step2 : Estimation and check

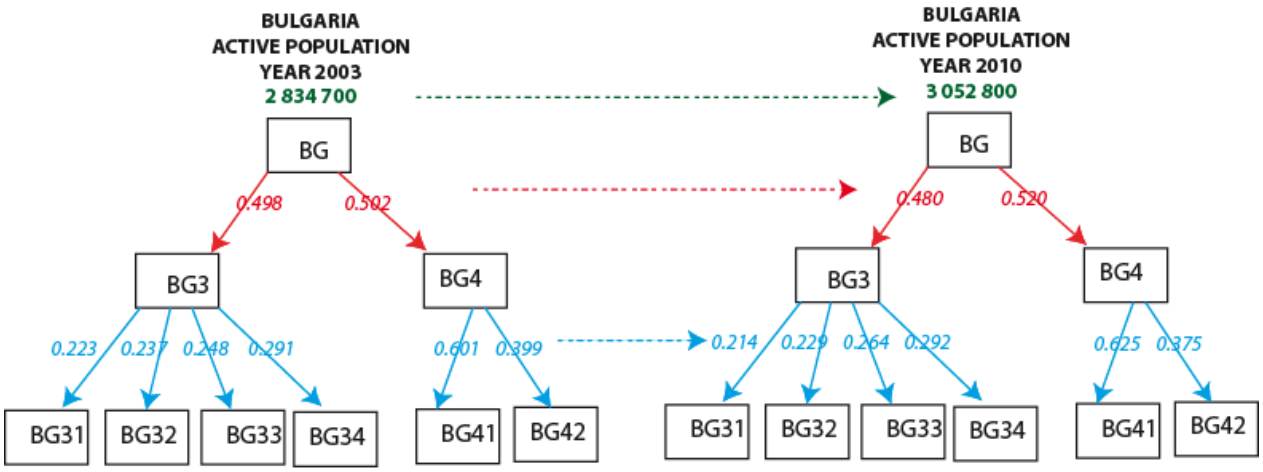
code	name	level	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
BG	Bulgaria	NUTS0	**2794.7**	**2794.7**	**2702.8**	**2741.0**	**2834.7**	**2922.6**	**2981.9**	**3110.0**	**3252.6**	**3360.7**	**3253.6**	**3052.8**
BG3	Severna i iztochna Bulgari	NUTS1	0.4981	0.4981	0.4981	0.4981	0.4981	0.4946	0.4951	0.4918	0.4863	0.4857	0.4831	0.4802
BG4	Yugozapadna i yuzhna tse	NUTS1	0.5019	0.5019	0.5019	0.5019	0.5019	0.5054	0.5049	0.5082	0.5137	0.5143	0.5169	0.5198
BG31	Severozapaden	NUTS2	0.2236	0.2236	0.2236	0.2236	0.2236	0.2202	0.2131	0.2143	0.2184	0.2201	0.2171	0.2140
BG32	Severen tsentralen	NUTS2	0.2373	0.2373	0.2373	0.2373	0.2373	0.2386	0.2333	0.2301	0.2329	0.2294	0.2326	0.2292
BG33	Severoiztochen	NUTS2	0.2482	0.2482	0.2482	0.2482	0.2482	0.2499	0.2637	0.2648	0.2614	0.2629	0.2605	0.2643
BG34	Yugoiztochen	NUTS2	0.2908	0.2908	0.2908	0.2908	0.2908	0.2913	0.2899	0.2908	0.2874	0.2876	0.2898	0.2924
BG41	Yugozapaden	NUTS2	0.6013	0.6013	0.6013	0.6013	0.6013	0.6056	0.6115	0.6163	0.6136	0.6134	0.6199	0.6246
BG42	Yuzhen tsentralen	NUTS2	0.3987	0.3987	0.3987	0.3987	0.3987	0.3944	0.3885	0.3837	0.3864	0.3866	0.3801	0.3754

Step3 : Core data

code	name	level	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
BG	Bulgaria	NUTS0	**2794.7**	**2794.7**	**2702.8**	**2741.0**	**2834.7**	**2922.6**	**2981.9**	**3110.0**	**3252.6**	**3360.7**	**3253.6**	**3052.8**
BG3	Severna i iztochna Bulgari	NUTS1	1392.0	1392.0	1346.3	1365.3	1412.0	1445.6	1476.3	1529.5	1581.7	1632.2	1571.9	1465.9
BG4	Yugozapadna i yuzhna tse	NUTS1	1402.7	1402.7	1356.5	1375.7	1422.7	1477.0	1505.6	1580.5	1670.9	1728.5	1681.7	1586.9
BG31	Severozapaden	NUTS2	311.3	311.3	301.0	305.3	315.7	318.3	314.6	327.7	345.4	359.3	341.3	313.7
BG32	Severen tsentralen	NUTS2	330.4	330.4	319.5	324.0	335.1	344.9	344.4	352.0	368.3	374.4	365.6	336.0
BG33	Severoiztochen	NUTS2	345.6	345.6	334.2	338.9	350.5	361.3	389.3	405.0	413.4	429.1	409.5	387.5
BG34	Yugoiztochen	NUTS2	404.8	404.8	391.5	397.0	410.6	421.1	428.0	444.8	454.6	469.4	455.6	428.7
BG41	Yugozapaden	NUTS2	843.4	843.4	815.6	827.2	855.4	894.5	920.7	974.1	1025.3	1060.2	1042.5	991.2
BG42	Yuzhen tsentralen	NUTS2	559.3	559.3	540.9	548.6	567.3	582.5	584.9	606.4	645.6	668.3	639.2	595.7

- Count variable is estimated only for the top level of hierarchy: for example, the population of Bulgaria in 1999 is estimated in number of active.
- All other territorial units are described by a frequency of the unit of upper level: for example the unit BG3 is described by the evolution of its share of BG and the unit BG31 as a share of BG3.
- Consistency of hierarchy is imposed. When the sum of frequency of child units depending from the same parent is different from 1.00, the value is adjusted. For example, the sum of BG31+BG32+BG33+BG34 in 2003 is equal to 1411.9 when BG3 is declared as 1412.0. This is not an error but simply a question of rounded value. Nevertheless it is corrected.

Figure 5 : Tree representation of the data model



The tree structure make more clear the way we propose to solve the problem of homogenization and estimation of missing values by dividing a big problem in smaller parts more easy to solve, according to René Descartes' method: "*The second [principle] is to divide each of the difficulties under examination into as many parts as possible, and as might be necessary for its adequate solution*". Basically, the problem that we have to solve is reduced to a combination of vertical and horizontal analysis of the trees.

- **Vertical analysis** will allow at the same time to check for logical errors (are sum of frequency of child of the same parent always equal to 1) and to estimate some missing values (definition of the value of a parent by sum of child or estimation of one missing child value by difference between the parent and the other childs …).

- **Horizontal analysis** will allow estimating missing values by mean of method of time series analysis and also to check for time outlier and provide margin of errors. But the important point is the fact that this estimation are made for small groups of time series that are typically the frequency of all the child of the same parent. This frequency is related to internal redistribution and not to external or general trends that are only taken into account for the estimation of the raw count at the top level of the tree.

With these reduction of problem in smaller parts, it appears more easy to propose automatic procedure of data check and data estimation that verify objectives (1) and (2) but can be implemented in a computer program, fulfilling the objectives (4),(5),(6). The most important difficulty remains the objective (3) related to the decision on what are real "breaks" in time series explainable by concrete fact and what are simple noise or biases to be eliminated by the procedure. To fulfill this final objectives, it would certainly be necessary to couple the estimation procedure with an expert system where human are invited to give advices on ambiguous cases where the algorithm cannot decide alone of the solution. For more details on this point, see the work realized by C. Plumejeaud (2010) in its Ph'D.

According to the time remaining in ESPON M4D project, we will focus on the production of a fully automatic solution without human expertise. The optimization of the procedure with expert intervention will be let opened for future work in ESPON 2014-2020.

6 Characteristics of ESPON Time Series

6.1 Introduction

The ESPON time series are have two challenging characteristics. First, they are relatively short; typically a series may be no more than 20 years in length, and many are far shorter. The poses challenges for estimating level, trend, and any seasonality in the data. The second characteristic is that they are often part of a spatial hierarchy. In the example used later in this report, the series has 12 time periods, and has measurements for NUTS0, NUTS1, and NUTS2 levels. The NUTS codes for the series are shown below:

BG			NUTS0
	BG3		NUTS1
		BG31	NUTS2
		BG32	NUTS2
		BG33	NUTS2
		BG34	NUTS2
	BG4		NUTS1
		BG41	NUTS2
		BG42	NUTS2

Thus each series has a longitudinal component, represented by the variation across the time domain, and a cross-sectional component, represented by variation within and between the various levels in the NUTS hierarchy. This means that there are some constraints: the total for the units at NUTS level s must sum to the value of the parent unit at NUTS level s-1. This means that for any time period a series of constraints are in place:

$X_{BG} = X_{BG3} + X_{BG4}$

$X_{BG3} = X_{BG31} + X_{BG32} + X_{BG33} + X_{BG34}$

$X_{BG4} = X_{BG41} + X_{BG42}$

This gives rise to a mechanism for estimating any missing components in each series.

6.2 Proportions

If the counts are converted to proportions, these can be used in a LOCF or weighted autoregression to fill in the gaps. In any time period, the following relations need to hold

NUTS0: $1 = P_{BG3} + P_{BG4}$

NUTS1: $1 = P_{BG31} + P_{BG32} + P_{BG33} + P_{BG34}$

NUTS1: $1 = P_{BG41} + P_{BG42}$

The missing proportions for time period t can be estimated from those either in the past (in extrapolation) or in the future (in retropolation). Then, the proportions may need small adjustments to ensure that the constraints above hold (there may be rounding error in the computations). Once the NUTS0 totals are known, the rest are easily computed.

The estimation approach is outlined in the next section, and the code is available in appendix 1. It should be noted that this code deals with a particular *pattern* of missing data. In this example, all top level data is available except for the first year, and no lower level data is available for the first four years. The strategy that is adopted is:

1. Compute the proportions between the various levels in the hierarchy (NUTS0:NUTS1, NUTS1:NUTS2).
2. Estimate the missing NUTS0 value
3. Estimate and constrain the missing NUTS1 and NUTS2 proportions
4. Convert the proportions to counts for NUTS1 then NUTS2

Step 3 can either assume the LOCF model in retropolation, or an autoregressive form, as detailed in the previous section.

A challenge is recognising the patterns of missing data, in order to bring the appropriate model into play for extra- or retro- polation. In the final section, we consider some alternative scenarios for imputation.

7 Methodologies for estimating missing time series elements

The example uses employment data in 1000s for the NUTS0, NUTS1 and NUTS2 zones in Bulgaria. The goal of the estimation procedure is to be able to provide a complete series of employment totals for all NUTS units over the period 1999 to 2010 inclusive. At the national, NUTS0, level there are totals for each year except 1999. For the other NUTS units in the example, the totals for the period 1999 to 2002 inclusive are missing. We need to make an estimation first of the NUTS0 total for 1999 and then the NUTS1 levels for 1999 to 2002, followed by the NUTS3 levels for the same time period.

```
emp <- read.table("test_bulgaria_emp.txt", sep="\t", dec=".", header=TRUE)
```

This creates a *data frame* called **emp**. The rows in the data frame represent observations, in our case NUTS spatial units. The columns represent variables. Variables of different types may be collected together in a data frame; in the example we have alphameric data for the NUTS codes, NTS region names and the NUTS1 level. The other data items are numeric. Data which is not present is represented by the NA letter pair.

```
emp
```

```
  code                                name level emp1999 emp2000 emp2001 emp2002 emp2003 emp2004 emp2005
1   BG                            Bulgaria NUTS0      NA  2794.7  2702.8    2741  2834.7  2922.6  2981.9
2  BG3          Severna i iztochna Bulgaria NUTS1      NA      NA      NA      NA  1412.0  1445.6  1476.3
3 BG31                     Severozapaden NUTS2      NA      NA      NA      NA   315.7   318.3   314.6
4 BG32                 Severen tsentralen NUTS2      NA      NA      NA      NA   335.1   344.9   344.4
5 BG33                     Severoiztochen NUTS2      NA      NA      NA      NA   350.5   361.3   389.3
6 BG34                       Yugoiztochen NUTS2      NA      NA      NA      NA   410.6   421.1   428.0
7  BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA      NA      NA      NA  1422.8  1477.0  1505.6
8 BG41                           Yugozapaden NUTS2      NA      NA      NA      NA   855.4   894.5   920.7
9 BG42                      Yuzhen tsentralen NUTS2      NA      NA      NA      NA   567.3   582.5   584.9
  emp2006 emp2007 emp2008 emp2009 emp2010
1  3110.0  3252.6  3360.7  3253.6  3052.8
2  1529.5  1581.7  1632.2  1571.9  1465.9
3   327.7   345.4   359.3   341.3   313.7
4   352.0   368.3   374.4   365.6   336.0
5   405.0   413.4   429.1   409.5   387.5
6   444.8   454.6   469.4   455.6   428.7
7  1580.5  1670.9  1728.5  1681.7  1586.9
8   974.1  1025.3  1060.2  1042.4   991.3
9   606.4   645.6   668.3   639.2   595.7
```

A summary of the structure of the data reveals how R has organised ther transfer of the data from the external file into the data frame. The variables 'code', 'name', and 'level' are of type factor – they have a fixed number of categories, and the value for any observation is a member of the set of categories. Most of the other variables are typed as numeric, although the variable 'emp1999' is currently typed as logical. The act of making computations on the 'emp1999' variable will *coerce* its type to numeric.

```
str(emp)
```

```
'data.frame':   9 obs. of  15 variables:
 $ code   : Factor w/ 9 levels "BG","BG3","BG31",..: 1 2 3 4 5 6 7 8 9
 $ name   : Factor w/ 9 levels "Bulgaria","Severen tsentralen",..: 1 3 5 2 4 6 8 7 9
 $ level  : Factor w/ 3 levels "NUTS0","NUTS1",..: 1 2 3 3 3 3 2 3 3
 $ emp1999: logi  NA NA NA NA NA NA ...
 $ emp2000: num  2795 NA NA NA NA ...
 $ emp2001: num  2703 NA NA NA NA ...
 $ emp2002: num  2741 NA NA NA NA ...
 $ emp2003: num  2835 1412 316 335 350 ...
 $ emp2004: num  2923 1446 318 345 361 ...
```

```
 $ emp2005: num  2982 1476 315 344 389 ...
 $ emp2006: num  3110 1530 328 352 405 ...
 $ emp2007: num  3253 1582 345 368 413 ...
 $ emp2008: num  3361 1632 359 374 429 ...
 $ emp2009: num  3254 1572 341 366 410 ...
 $ emp2010: num  3053 1466 314 336 388 ...
```

Recall that the expedient methodologies for these short series include LOCF (last observation carried forward), autoregression (AR), and proportional assignment (PA). In this case, the totals for the national series and the proportions for the lower level series can be filled in using LOCF or AR; the totals may then be estimated using PA.

Next we specify the weight schemes for the LOCF or AR estimations. We can other forecast forwards from x_t to x_{t+1}, x_{t+2}, and so on, or backforecast from x_t to x_{t-1}, x_{t-2}. The terms extrapolation and retropolation may also be used in place of forecast and backforecast. The equations for a 4 term model are shown below, as a backforecast in the upper equation, and a forecast in the lower equation.

$$\hat{x}_{t-1} = \beta_0 x_t + \beta_1 x_{t+1} + \beta_2 x_{t+2} + \beta_3 x_{t+3}$$
$$\hat{x}_{t+1} = \beta_0 x_t + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \beta_3 x_{t-3}$$

A wide range of weighting schemes is possible, a selection of which are shown below:

```
weight <- c(1.00, 0.00, 0.00, 0.00
weight <- c(0.50, 0.30, 0.15, 0.05)
weight <- c(0.30, 0.30, 0.25, 0.05)
weight <- c(1.9656988, -0.9656988, 0, 0
nw <- length(weight)
```

The weights are chosen so that they always sum to 1: this is equivalent to a 4 point moving average. Indeed, a 4 point moving average would have the weights: (0.25, 0.25, 0.25, 0.25).

The first is equivalent to LOCF: $x_{t-1} = 1.0x_t + 0x_{t+1} + 0x_{t+2} + 0x_{t+3}$ is an example for the backforecasted case. In the second case, the weights are tapered: more weight is given to recent observations than ones which are more distant in time. This allows a contribution from time periods further forward/backward in the series to the estimate of the missing value. After a time these will settle down to a constant value. However, the tapered weights allow a slower, damped, convergence on the constant values. The third give less weight to the more recent observations, and the last would be applied to a series where the extrapolations are show a downward trend.

The technique requires backwards linkage up through the rows representing higher levels in the NUTS hierarchy. So, for a NUTS2 region we need to be able to link backwards to its parent NUTS1 region, and so on. These data have a variable which specifies the NUTS level, but this either may not be present, or may contain errors. For this reason, we shall compute the NUTS level from the NUTS code, and then use this to create a pointer to the parent NUTS node.

```
rownames(emp) <- emp$code
emp$NUTSlevel <- nchar (levels(emp$code)[emp$code]) - 2
```

The first instruction copies the NUTS code as the rowname – the row name is an attribute of the data frame structure. Usually it is just the sequence number of the row in the matrix, counting downwards from the top, but may be altered as the analyst desires. In this example the variable code contains the NUTS codes.

The second instruction extracts from NUTS code for each observation, determines the number of characters in the code, and subtracts 2: the result of the level of the NUTS region with that code. BG has a length of 2, and represents a NUTS0 region; BG31 has a length of 4, and represents a NUTS2 region.

At the end of these computations the matrix includes an extra column, NUTSlevel, and the rownames are the same as the code variable:

```
     code                                    name level emp1999 emp2010 NUTSlevel
BG     BG                                Bulgaria NUTS0      NA  3052.8         0
BG3   BG3              Severna i iztochna Bulgaria NUTS1      NA  1465.9         1
BG31 BG31                        Severozapaden NUTS2      NA   313.7         2
BG32 BG32                    Severen tsentralen NUTS2      NA   336.0         2
BG33 BG33                     Severoiztochen NUTS2      NA   387.5         2
BG34 BG34                       Yugoiztochen NUTS2      NA   428.7         2
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA  1586.9         1
BG41 BG41                        Yugozapaden NUTS2      NA   991.3         2
BG42 BG42                    Yuzhen tsentralen NUTS2      NA   595.7         2
```

The next step is to create the back-linkage, so that the name of the parent row is present in each row of the data. The NUTS0 row is the superparent (the 'root' of the tree), and its parent entry will be blank.

```
emp$parent    <- ifelse (emp$NUTSlevel > 0, substr(emp$code,1, emp$NUTSlevel+1), "")
```

In the instruction, and rows representing NUTS levels other than 0 have the characters representing its parent (the characters from 1 to NUTSlevel+1 inclusive) extracted from the NUTS code. The extra column appears in the data frame…

```
     code                                    name level emp1999 emp2010 NUTSlevel parent
BG     BG                                Bulgaria NUTS0      NA  3052.8         0
BG3   BG3              Severna i iztochna Bulgaria NUTS1      NA  1465.9         1     BG
BG31 BG31                        Severozapaden NUTS2      NA   313.7         2    BG3
BG32 BG32                    Severen tsentralen NUTS2      NA   336.0         2    BG3
BG33 BG33                     Severoiztochen NUTS2      NA   387.5         2    BG3
BG34 BG34                       Yugoiztochen NUTS2      NA   428.7         2    BG3
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA  1586.9         1     BG
BG41 BG41                        Yugozapaden NUTS2      NA   991.3         2    BG4
BG42 BG42                    Yuzhen tsentralen NUTS2      NA   595.7         2    BG4
```

The next stage is to compute the proportions for the NUTS1 and NUTS2 levels that the represent. Some initialisation is required:

```
maxRow <- nrow(emp)
dataCols <- 8:15
props <- emp
props[2:maxRow,dataCols] <- NA
```

There are maxRows rows in the dataframe, and the columns containing valid data for the NUTS1 and NUTS2 regions are, in this example, are 8 through 15. We make a

copy of the 'emp' data frame, named props, and set the entries which contain data at NUTS1/2 to NA. The props data frame then has the following entries:

```
     code                                    name level emp1999 emp2000 emp2001 emp2002
BG     BG                                Bulgaria NUTS0      NA  2794.7  2702.8    2741
BG3   BG3             Severna i iztochna Bulgaria NUTS1      NA      NA      NA      NA
BG31 BG31                          Severozapaden NUTS2      NA      NA      NA      NA
BG32 BG32                       Severen tsentralen NUTS2      NA      NA      NA      NA
BG33 BG33                           Severoiztochen NUTS2      NA      NA      NA      NA
BG34 BG34                            Yugoiztochen NUTS2      NA      NA      NA      NA
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA      NA      NA      NA
BG41 BG41                            Yugozapaden NUTS2      NA      NA      NA      NA
BG42 BG42                         Yuzhen tsentralen NUTS2      NA      NA      NA      NA

     emp2003 emp2004 emp2005 emp2006 emp2007 emp2008 emp2009 emp2010 NUTSlevel parent
BG    2834.7  2922.6  2981.9    3110  3252.6  3360.7  3253.6  3052.8         0
BG3       NA      NA      NA      NA      NA      NA      NA      NA         1     BG
BG31      NA      NA      NA      NA      NA      NA      NA      NA         2    BG3
BG32      NA      NA      NA      NA      NA      NA      NA      NA         2    BG3
BG33      NA      NA      NA      NA      NA      NA      NA      NA         2    BG3
BG34      NA      NA      NA      NA      NA      NA      NA      NA         2    BG3
BG4       NA      NA      NA      NA      NA      NA      NA      NA         1     BG
BG41      NA      NA      NA      NA      NA      NA      NA      NA         2    BG4
BG42      NA      NA      NA      NA      NA      NA      NA      NA         2    BG4
```

Next we compute the proportions that each region's employment is of the employment in its parent region.

```
for (i in 2:maxRow)
      parentRow <- which(rownames(emp) == emp$parent[i]
      props[i,dataCols] <- emp[i,dataCols] / emp[parentRow,dataCols]
}
```

The computations take place in a loop; we find the row number for the parent row; and this is followed by computing the proportions for the child row. The variable dataCols indexes the columns with valid NUTS0, NUTS1 and NUTS2 data.

At the end of this step, the proportions are:

```
     code                                    name level emp1999 emp2000 emp2001 emp2002
BG     BG                                Bulgaria NUTS0      NA  2794.7  2702.8    2741
BG3   BG3             Severna i iztochna Bulgaria NUTS1      NA      NA      NA      NA
BG31 BG31                          Severozapaden NUTS2      NA      NA      NA      NA
BG32 BG32                       Severen tsentralen NUTS2      NA      NA      NA      NA
BG33 BG33                           Severoiztochen NUTS2      NA      NA      NA      NA
BG34 BG34                            Yugoiztochen NUTS2      NA      NA      NA      NA
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA      NA      NA      NA
BG41 BG41                            Yugozapaden NUTS2      NA      NA      NA      NA
BG42 BG42                         Yuzhen tsentralen NUTS2      NA      NA      NA      NA

          emp2003      emp2004      emp2005      emp2006      emp2007      emp2008
BG   2834.7000000 2922.6000000 2981.9000000 3110.0000000 3252.6000000 3360.7000000
BG3     0.4981127    0.4946281    0.4950870    0.4918006    0.4862879    0.4856726
BG31    0.2235836    0.2201854    0.2131003    0.2142530    0.2183726    0.2201323
BG32    0.2373229    0.2385861    0.2332859    0.2301406    0.2328507    0.2293837
BG33    0.2482295    0.2499308    0.2636998    0.2647924    0.2613644    0.2628967
BG34    0.2907932    0.2912977    0.2899140    0.2908140    0.2874123    0.2875873
BG4     0.5019226    0.5053719    0.5049130    0.5081994    0.5137121    0.5143274
BG41    0.6012089    0.6056195    0.6115170    0.6163239    0.6136214    0.6133642
BG42    0.3987208    0.3943805    0.3884830    0.3836761    0.3863786    0.3866358

          emp2009      emp2010 NUTSlevel parent
BG   3253.6000000 3052.8000000         0
BG3     0.4831264    0.4801821         1     BG
BG31    0.2171258    0.2139982         2    BG3
BG32    0.2325848    0.2292107         2    BG3
BG33    0.2605128    0.2643427         2    BG3
BG34    0.2898403    0.2924483         2    BG3
BG4     0.5168736    0.5198179         1     BG
```

```
BG41    0.6198490    0.6246770         2    BG4
BG42    0.3800916    0.3753860         2    BG4
```

We can now begin to fill in the missing data. We start by working down the hierarchy from the top. Emp1999 for the national (NUTS0) level is missing, so we used the weights to compute a weighted average of the previous 4 values (emp2000 … emp2003).

```
final <- emp
final[1,4] <- sum(final[1,5:8] * weight
```

We make a copy of the original data frame, emp, and name it final. The weights are then applied to the relevant columns (in this case 5:8) of the employment totals. This produces a complete series at NUTS0 level:

```
   code     name level  emp1999 emp2000 emp2001 emp2002 emp2003 emp2004 emp2005 emp2006
BG   BG Bulgaria NUTS0 2761.075  2794.7  2702.8    2741  2834.7  2922.6  2981.9    3110

   emp2007 emp2008 emp2009 emp2010 NUTSlevel parent
BG  3252.6  3360.7  3253.6  3052.8         0
```

If we had used the LOCF approach, the imputed value for 1999 would be 2794.7, rather than 2761.1; there is some modelled growth before the series declines slightly.

We can impute the missing proportions in columns 7 downwards to 4 – we are backforecasting.

```
adjustCols <- 7:4

for (updateRow in 2:maxRow) {
   for (updateCol in adjustCols)
      props[updateRow,updateCol] <- sum(props[updateRow,seq(updateCol+1,
updateCol+4)] * weight)
   }
}
```

The backforecasted proportions are shown below

```
     code                                    name level   emp1999      emp2000      emp2001
BG     BG                                 Bulgaria NUTS0        NA 2794.7000000 2702.8000000
BG3   BG3              Severna i iztochna Bulgaria NUTS1 0.4966113    0.4966033    0.4965313
BG31 BG31                           Severozapaden NUTS2 0.2211714    0.2212146    0.2210204
BG32 BG32                       Severen tsentralen NUTS2 0.2370109    0.2370578    0.2370177
BG33 BG33                          Severoiztochen NUTS2 0.2509418    0.2508414    0.2510877
BG34 BG34                            Yugoiztochen NUTS2 0.2908351    0.2908456    0.2908352
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1 0.5034090    0.5034170    0.5034881
BG41 BG41                             Yugozapaden NUTS2 0.6040510    0.6040117    0.6041985
BG42 BG42                         Yuzhen tsentralen NUTS2 0.3959084    0.3959478    0.3957629

          emp2002      emp2003      emp2004      emp2005      emp2006      emp2007
BG   2741.0000000 2834.7000000 2922.6000000 2981.9000000 3110.0000000 3252.6000000
BG3     0.4962978    0.4981127    0.4946281    0.4950870    0.4918006    0.4862879
BG31    0.2205251    0.2235836    0.2201854    0.2131003    0.2142530    0.2183726
BG32    0.2367372    0.2373229    0.2385861    0.2332859    0.2301406    0.2328507
BG33    0.2518886    0.2482295    0.2499308    0.2636998    0.2647924    0.2613644
BG34    0.2908137    0.2907932    0.2912977    0.2899140    0.2908140    0.2874123
BG4     0.5037198    0.5019226    0.5053719    0.5049130    0.5081994    0.5137121
BG41    0.6048340    0.6012089    0.6056195    0.6115170    0.6163239    0.6136214
BG42    0.3951308    0.3987208    0.3943805    0.3884830    0.3836761    0.3863786

          emp2008      emp2009      emp2010 NUTSlevel parent
BG   3360.7000000 3253.6000000 3052.8000000         0
BG3     0.4856726    0.4831264    0.4801821         1     BG
```

```
BG31    0.2201323    0.2171258    0.2139982         2    BG3
BG32    0.2293837    0.2325848    0.2292107         2    BG3
BG33    0.2628967    0.2605128    0.2643427         2    BG3
BG34    0.2875873    0.2898403    0.2924483         2    BG3
BG4     0.5143274    0.5168736    0.5198179         1     BG
BG41    0.6133642    0.6198490    0.6246770         2    BG4
BG42    0.3866358    0.3800916    0.3753860         2    BG4
```

We next check the extrapolations: the higher level proportions should sum to 1. There is some minor rounding error which we can use to constraint the proportions to 1.

```
checkColumns <- 4:15
summaryTable <- aggregate(props[,checkColumns],by=list(emp$parent),FUN=sum
rownames(summaryTable) <- summaryTable$Group.1
print(summaryTable)
```

```
    Group.1   emp1999      emp2000      emp2001      emp2002      emp2003 emp2004 emp2005
                 NA 2794.7000000 2702.8000000 2741.0000000 2834.7000000  2922.6  2981.9
BG       BG 1.0000204    1.0000203    1.0000194    1.0000176    1.0000353     1.0     1.0
BG3     BG3 0.9999591    0.9999593    0.9999610    0.9999646    0.9999292     1.0     1.0
BG4     BG4 0.9999594    0.9999596    0.9999613    0.9999649    0.9999297     1.0     1.0

    emp2006 emp2007 emp2008      emp2009     emp2010
       3110  3252.6  3360.7 3253.6000000 3052.800000
BG        1     1.0     1.0    1.0000000    1.000000
BG3       1     1.0     1.0    1.0000636    1.000000
BG4       1     1.0     1.0    0.9999405    1.000063
```

Finally we compute the totals using the extrapolated and constrained proportions:

```
for (level in 1:2) {
   adjustRows <- which(final$NUTSlevel == level)
   for (irow in adjustRows) {
      parentRow <- which(rownames(final) == final$parent[irow])
      final[irow,adjustCols] <- final[parentRow,adjustCols] * props[irow,adjustCols]
   }
}
```

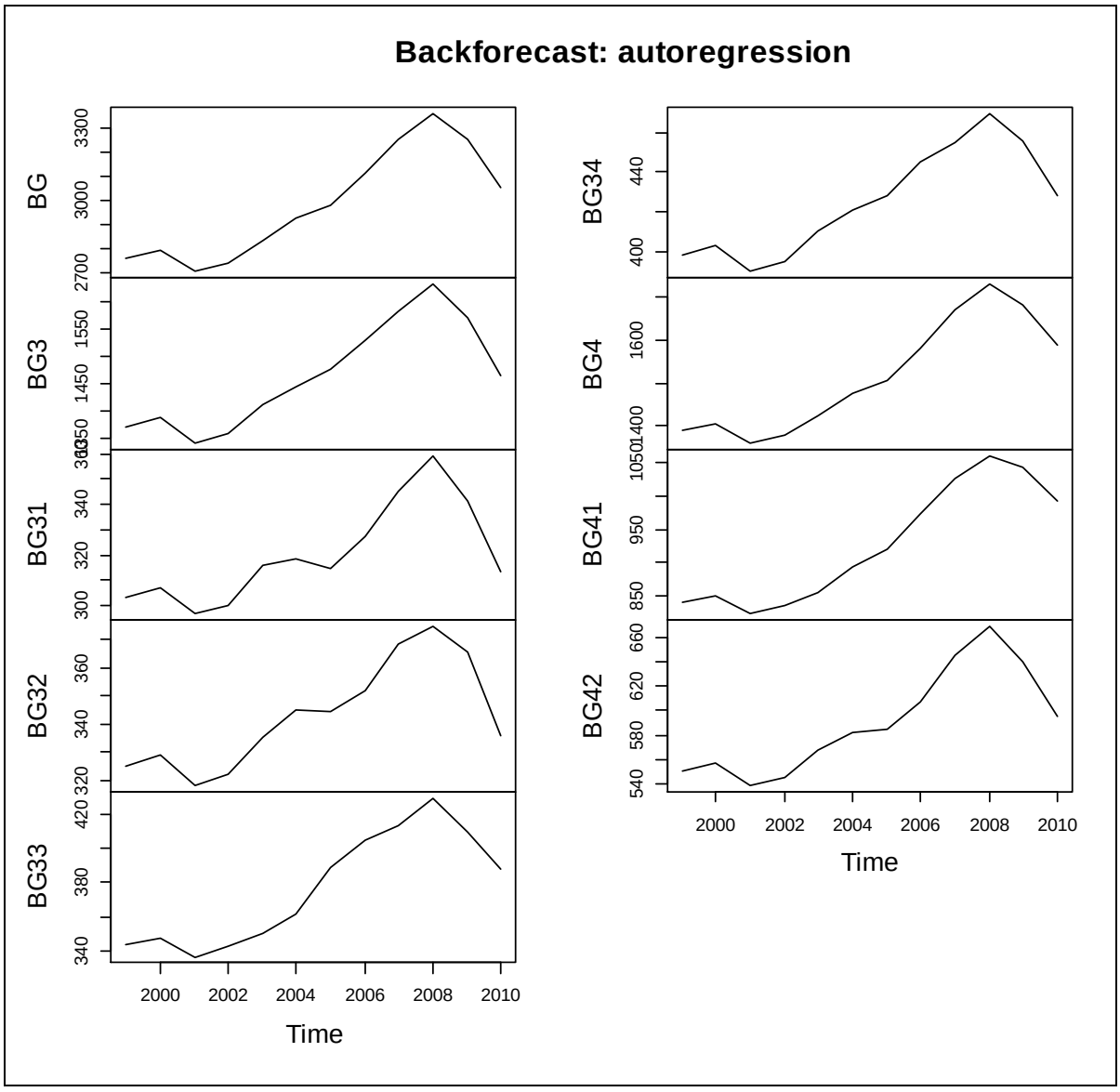
The backforecasted totals are in the dataframe, final, below:

```
     code                                    name level   emp1999   emp2000   emp2001
BG     BG                                Bulgaria NUTS0 2761.0750 2794.7000 2702.8000
BG3   BG3             Severna i iztochna Bulgaria NUTS1 1371.1532 1387.8291 1341.9987
BG31 BG31                        Severozapaden NUTS2  303.2722  307.0205  296.6207
BG32 BG32                    Severen tsentralen NUTS2  324.9916  329.0090  318.0898
BG33 BG33                        Severoiztochen NUTS2  344.0937  348.1392  336.9725
BG34 BG34                          Yugoiztochen NUTS2  398.7957  403.6604  390.3156
BG4   BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1 1389.9218 1406.8709 1360.8013
BG41 BG41                          Yugozapaden NUTS2  839.6177  849.8009  822.2258
BG42 BG42                       Yuzhen tsentralen NUTS2  550.3041  557.0700  538.5755

       emp2002 emp2003 emp2004 emp2005 emp2006 emp2007 emp2008 emp2009 emp2010 NUTSlevel
BG   2741.0000  2834.7  2922.6  2981.9  3110.0  3252.6  3360.7  3253.6  3052.8         0
BG3  1360.3284  1412.0  1445.6  1476.3  1529.5  1581.7  1632.2  1571.9  1465.9         1
BG31  299.9972   315.7   318.3   314.6   327.7   345.4   359.3   341.3   313.7         2
BG32  322.0517   335.1   344.9   344.4   352.0   368.3   374.4   365.6   336.0         2
BG33  342.6633   350.5   361.3   389.3   405.0   413.4   429.1   409.5   387.5         2
BG34  395.6162   410.6   421.1   428.0   444.8   454.6   469.4   455.6   428.7         2
BG4  1380.6716  1422.8  1477.0  1505.6  1580.5  1670.9  1728.5  1681.7  1586.9         1
BG41  835.1065   855.4   894.5   920.7   974.1  1025.3  1060.2  1042.4   991.3         2
BG42  545.5651   567.3   582.5   584.9   606.4   645.6   668.3   639.2   595.7         2

     parent
BG
BG3      BG
BG31    BG3
BG32    BG3
BG33    BG3
```

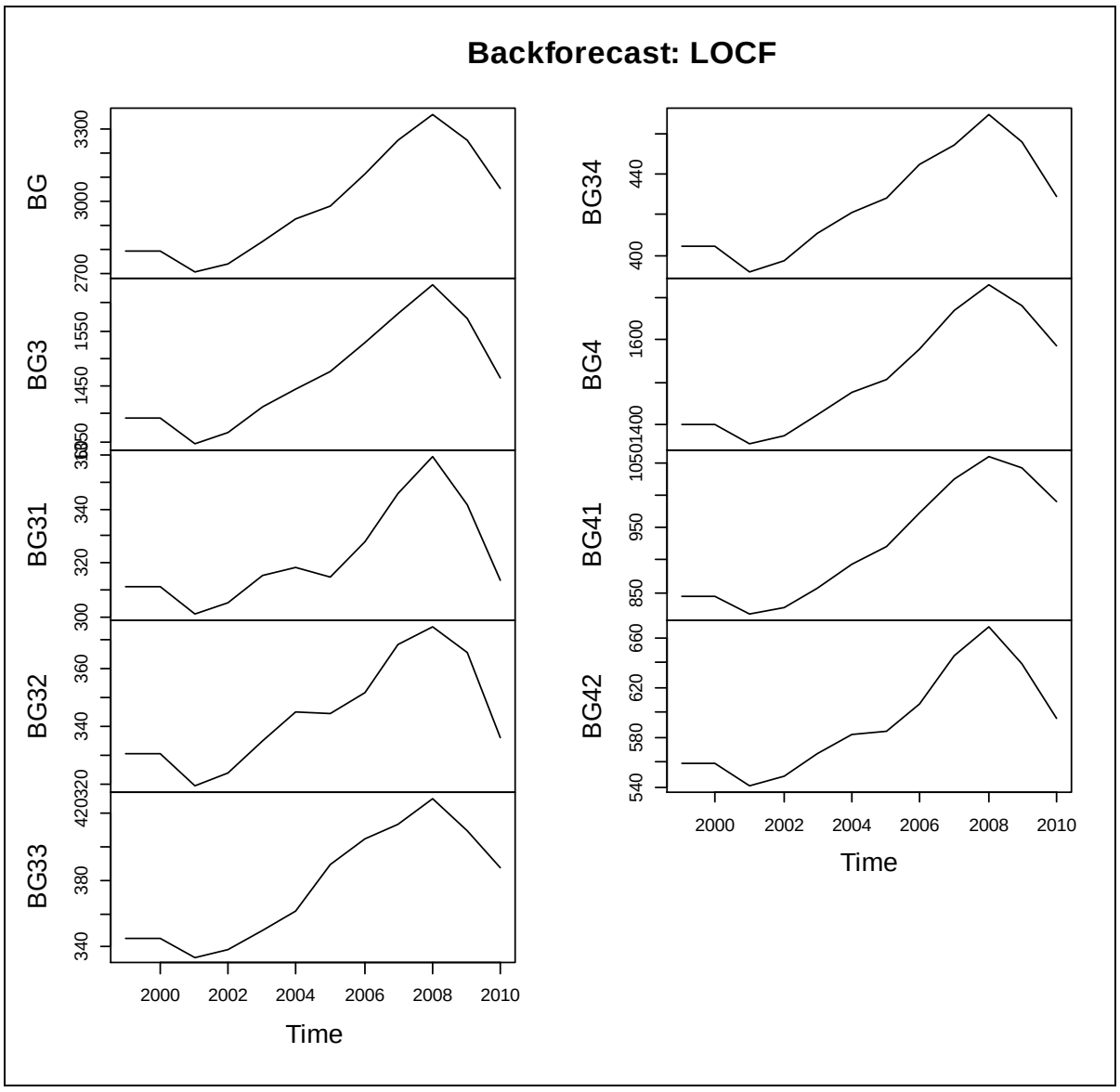
```
BG34     BG3
BG4       BG
BG41     BG4
BG42     BG4
```

Finally we can plot the various series – the plot below shows the NUTS0, NUTS1 and NUTS2 series using the 'autoregression' approach.

```
emp.ts <- ts(t(final[,4:15]), start=1999, frequency=1)
plot(emp.ts,main="Backforecast: autoregression")
```



We can compare this with the LOCF approach – notice that the first part of the series is level – there is no change. Whether this is close to what really happened is a matter for conjecture. The autoregression approach allows a small increase in the trend, which given the real series presented dearlier in this document, may be a closer reflection of reality.


Backforecast: LOCF
BG
BG3
BG31
BG32
BG33
BG34
BG4
BG41
BG42
Time
Time

8 Time series estimation scenarios and strategies

It is possible to envisage a number of scenarios with regard to time series data for which estimation may or may not be possible. We shall illustrate these with the Bulgaria example which we have used in the previous sections.

There are 5 scenarios which we consider. Each is illustrated with a sample of data. The missing values are highlighted in a bold red font thus: **NA**. The pattern of missing data is also shown in a matrixplot. The matrixplot is one of the missing data visualization tools available in the R VIM (**V**isualisation and **I**mputation of **M**issing Values) package[10]. In the matrixplot, the missing valued cells are shown colour solid red; cells with data are coloured various shades of grey according to their value. We discuss possible imputation strategies for the various scenarios. From this we may be able to develop some form of semi-automated imputation software.

8.1 Scenario 1

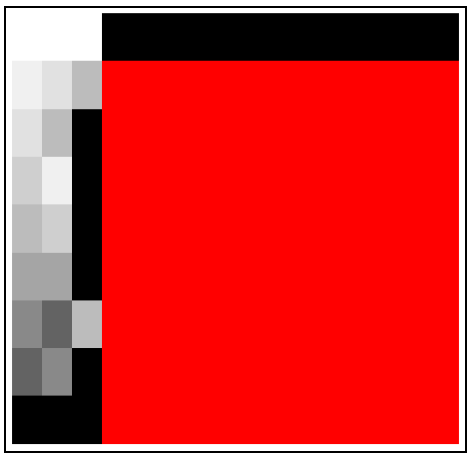
We begin with perhaps the most challenging scenario besides that of having no data at all. The scenario is that national data (NUTS0) is present, but data for no other spatial units are available. All of the entries for NUTS1 and NUTS2 are NA.

	code	Name	level	emp1999	emp2000	emp2001
1	BG	Bulgaria	NUTS0	2794.7	2794.7	2702.8
2	BG3	Severna i iztochna Bulgaria	NUTS1	NA	NA	NA
3	BG31	Severozapaden	NUTS2	NA	NA	NA
4	BG32	Severen tsentralen	NUTS2	NA	NA	NA
5	BG33	Severoiztochen	NUTS2	NA	NA	NA
6	BG34	Yugoiztochen	NUTS2	NA	NA	NA
7	BG4	Yugozapadna i yuzhna tsentralna Bulgaria	NUTS1	NA	NA	NA
8	BG41	Yugozapaden	NUTS2	NA	NA	NA
9	BG42	Yuzhen tsentralen	NUTS2	NA	NA	NA

	emp2002	emp2003	emp2004	emp2005	emp2006	emp2007	emp2008	emp2009	emp2010
1	2741	2834.7	2922.6	2981.9	3110	3252.6	3360.7	3253.6	3052.8
2	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	NA	NA	NA	NA	NA	NA	NA	NA	NA

The matrixplot shows this pattern in stark detail relative to the other columns:

[10] Templ M, Alfons A, Filzmoser P, 2012, Exploring incomplete data using visualisation tools, *Journal of Advances in Data Analysis and Classification*, 6(1), 29-47



The top (black) row represents the NUTS0 level, and the red area is that which contains missing data for the NUTS1 and NUTS2 levels. The first three columns represent the NUTS code, the region name, and the NUTS level.

Imputation from the available data is impossible. It would be unwise to assume that the 6 NUTSq2 regions shared 1/6th of the national totals each and the NUTS1 1/3 and 2/3 of the totals. Depending on the availability of other covariates at NUTS1/NUTS2 it may be possible to create a model by which the NUTS1 and NUTS totals could be estimated, and then the NUTS hierarchical constraints can be employed to ensure internal consistency.

Such socio-economic or economic model would require domain specific knowledge, and again it would be unwise to consider imputation without this contextual understanding.

8.2 Scenario 2

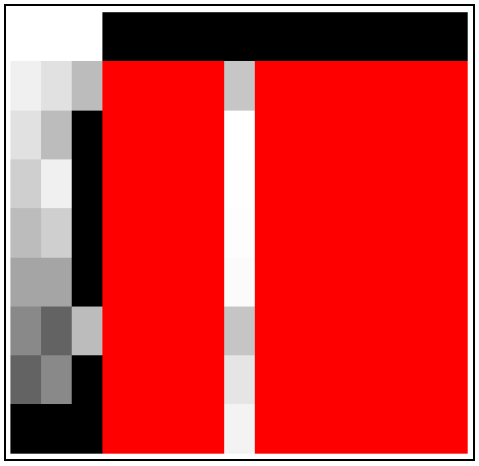
The second scenario provides some extra data in addition to the national totals. We assume that cross-sectional data is available for a single time period. Whilst there are many possibilities for this scenario we have chosen an arbitrary date in the middle of the overall time period. Clearly the cross-sectional observations could be at the beginning of the time period or at the end.

	code	Name	level	emp1999	emp2000	emp2001
1	BG	Bulgaria	NUTS0	2794.7	2794.7	2702.8
2	BG3	Severna i iztochna Bulgaria	NUTS1	NA	NA	NA
3	BG31	Severozapaden	NUTS2	NA	NA	NA
4	BG32	Severen tsentralen	NUTS2	NA	NA	NA
5	BG33	Severoiztochen	NUTS2	NA	NA	NA
6	BG34	Yugoiztochen	NUTS2	NA	NA	NA
7	BG4	Yugozapadna i yuzhna tsentralna Bulgaria	NUTS1	NA	NA	NA
8	BG41	Yugozapaden	NUTS2	NA	NA	NA
9	BG42	Yuzhen tsentralen	NUTS2	NA	NA	NA

	emp2002	emp2003	emp2004	emp2005	emp2006	emp2007	emp2008	emp2009	emp2010
1	2741	2834.7	2922.6	2981.9	3110	3252.6	3360.7	3253.6	3052.8
2	NA	1412.0	NA	NA	NA	NA	NA	NA	NA
3	NA	315.7	NA	NA	NA	NA	NA	NA	NA
4	NA	335.1	NA	NA	NA	NA	NA	NA	NA
5	NA	350.5	NA	NA	NA	NA	NA	NA	NA
6	NA	410.6	NA	NA	NA	NA	NA	NA	NA
7	NA	1422.8	NA	NA	NA	NA	NA	NA	NA
8	NA	855.4	NA	NA	NA	NA	NA	NA	NA

9	NA	567.3	NA	NA	NA	NA	NA	NA	NA

The matrixplot shows the single column of cross-sectional data cutting the panel of missing data in two; again the black bar at the top of the panel represents the NUTS0 data series is assumed to be present.



This presents a number of challenges. At best we must assume that the proportions of NUTS0/NUTS1/NUTS2 totals for the available cross-sectional remain constant, and estimate the missing data accordingly having used LOCF in each direction. If time based covariate(s) are available, then these might be used to modify the proportions through an appropriate model. However, because we are forced to use an expedient method here, and there is a high proportion of missing data, the reliability of any estimates beyond "indicative" may we be an important issue.

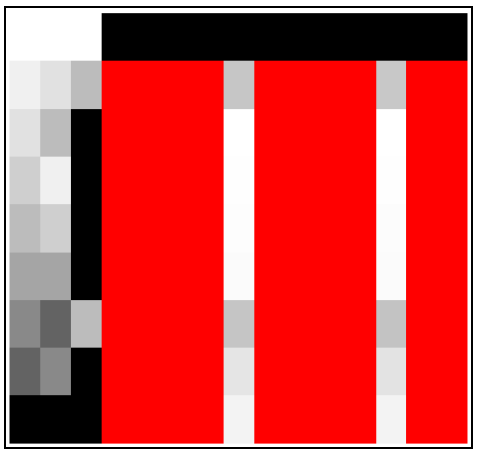
8.3 Scenario 3

Scenario 3 develops scenario 2 by the addition of an extra column of cross-sectional data. Depending on the time period, this may well represent the situation in which there are annual national estimates of an indicator, and that detailed cross-sectional data is available for two census dates. In some countries, Ireland notably, census of population data is available every 5 years. Elsewhere it is conventional for governments to undertake a decennial census.

	code	Name	level	emp1999	emp2000	emp2001
1	BG	Bulgaria	NUTS0	2794.7	2794.7	2702.8
2	BG3	Severna i iztochna Bulgaria	NUTS1	NA	NA	NA
3	BG31	Severozapaden	NUTS2	NA	NA	NA
4	BG32	Severen tsentralen	NUTS2	NA	NA	NA
5	BG33	Severoiztochen	NUTS2	NA	NA	NA
6	BG34	Yugoiztochen	NUTS2	NA	NA	NA
7	BG4	Yugozapadna i yuzhna tsentralna Bulgaria	NUTS1	NA	NA	NA
8	BG41	Yugozapaden	NUTS2	NA	NA	NA
9	BG42	Yuzhen tsentralen	NUTS2	NA	NA	NA

	emp2002	emp2003	emp2004	emp2005	emp2006	emp2007	emp2008	emp2009	emp2010
1	2741	2834.7	2922.6	2981.9	3110	3252.6	3360.7	3253.6	3052.8
2	NA	1412.0	NA	NA	NA	NA	1632.2	NA	NA
3	NA	315.7	NA	NA	NA	NA	359.3	NA	NA
4	NA	335.1	NA	NA	NA	NA	374.4	NA	NA
5	NA	350.5	NA	NA	NA	NA	429.1	NA	NA
6	NA	410.6	NA	NA	NA	NA	469.4	NA	NA
7	NA	1422.8	NA	NA	NA	NA	1728.5	NA	NA
8	NA	855.4	NA	NA	NA	NA	1060.2	NA	NA
9	NA	567.3	NA	NA	NA	NA	668.3	NA	NA

The matrix plot shows the missing data section divided into three panels, with the complete national (NUTS0) series, and the cross-sectional data available for the two census periods.



We have rather more data with which to make estimates. Again, we compute the proportions that the NUTS1 and NUTS2 data are of their parent NUTS0 and NUTS1 regions. A conservative assumption would be to use linear interpolation between the two census periods, assuming that there were no boundary changes in the intercensal period. Alternative views of filling the left and right hand panels might include (i) LOCF from the nearest census data, (ii) carrying the linear interpolation both backwards and forwards, or (iii) taking the mean of LOCF and linear interpolation. The extra-, inter- and retro-polated proportions would require adjustment to make such that they fitted the hierarchical summation constraints.

As with previous scenarios, if suitable time-dependent covariates were available, a suitable model might be used to modify the predictions of the proportions, with constraint adjustment as before.

Whilst there is increased reliability of the estimates, the proportion of missing relative to non-missing data is still uncomfortably large.

8.4 Scenario 4

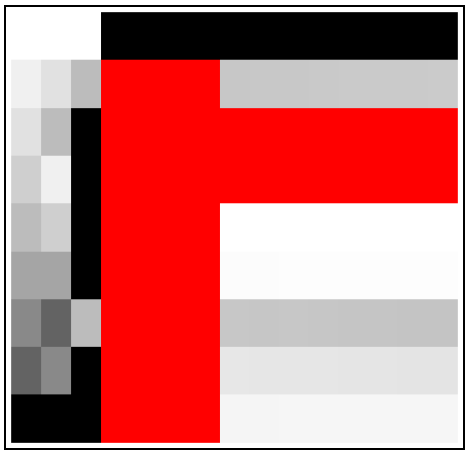
With more data, the possibility of more reliable estimates of the missing data is more certain. Another scenario would be to have relatively good cross-sectional coverage for about 2/3 of the data (both cross-sectional and longitudinal), but with a few early years not present at NUTS1/2 and the NUTS2 for part of a NUTS1 region missing.

	code	Name	level	emp1999	emp2000	emp2001
1	BG	Bulgaria	NUTS0	2794.7	2794.7	2702.8
2	BG3	Severna i iztochna Bulgaria	NUTS1	NA	NA	NA
3	BG31	Severozapaden	NUTS2	NA	NA	NA
4	BG32	Severen tsentralen	NUTS2	NA	NA	NA
5	BG33	Severoiztochen	NUTS2	NA	NA	NA
6	BG34	Yugoiztochen	NUTS2	NA	NA	NA
7	BG4	Yugozapadna i yuzhna tsentralna Bulgaria	NUTS1	NA	NA	NA
8	BG41	Yugozapaden	NUTS2	NA	NA	NA
9	BG42	Yuzhen tsentralen	NUTS2	NA	NA	NA

	emp2002	emp2003	emp2004	emp2005	emp2006	emp2007	emp2008	emp2009	emp2010

1	2741	2834.7	2922.6	2981.9	3110.0	3252.6	3360.7	3253.6	3052.8
2	NA	1412.0	1445.6	1476.3	1529.5	1581.7	1632.2	1571.9	1465.9
3	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	NA	350.5	361.3	389.3	405.0	413.4	429.1	409.5	387.5
6	NA	410.6	421.1	428.0	444.8	454.6	469.4	455.6	428.7
7	NA	1422.8	1477.0	1505.6	1580.5	1670.9	1728.5	1681.7	1586.9
8	NA	855.4	894.5	920.7	974.1	1025.3	1060.2	1042.4	991.3
9	NA	567.3	582.5	584.9	606.4	645.6	668.3	639.2	595.7

The characteristic pattern in the matrixplot shows the extent of the missing data.



The strategy is to complete the series for the later years, and then complete the cross-sectional data for the early missing years.

If only one of the NUTS2 zones was missing, then the totals could estimated quite easily since they would correspond to the NUTS1 total less the sum of the component NUTS2 counts. As well have no further data we can take the difference between the total for the NUTS2 zones with data and that the parent NUTS1 zones and, without additional data, allocate ½ of the residual to each zone. If other data were available, even for some of the missing time periods, we would be able to make a slightly more reasonable allocation.

Completing the earlier years might involve either LOCF, carried backwards, or the autoregressive approach outlined in the previous section. With two options, we might consider the mean of the two approaches. Again, if additional covariates were available, then a suitable model can be used to create modified forecasts. The hierarchical constraints will then ensure internal consistency.

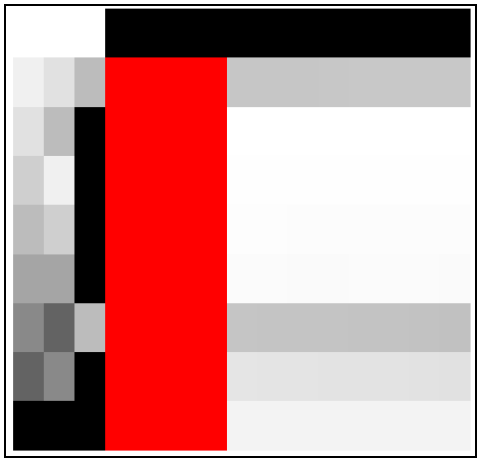
8.5 Scenario 5

A final alternative matches closely the example chosen earlier. The national NUTS0 data is present, but the NUTS1/2 data are missing for a panel in the earlier part of the time series. A comaprabe situation would arise if the panel were later in the series rather than earlier, except that we would extrapolate rather than retropolate.

	code	Name	level	emp1999	emp2000	emp2001
1	BG	Bulgaria	NUTS0	2794.7	2794.7	2702.8
2	BG3	Severna i iztochna Bulgaria	NUTS1	NA	NA	NA
3	BG31	Severozapaden	NUTS2	NA	NA	NA

```
4 BG32                          Severen tsentralen NUTS2      NA      NA      NA
5 BG33                              Severoiztochen NUTS2      NA      NA      NA
6 BG34                                Yugoiztochen NUTS2      NA      NA      NA
7  BG4 Yugozapadna i yuzhna tsentralna Bulgaria NUTS1      NA      NA      NA
8 BG41                                Yugozapaden NUTS2      NA      NA      NA
9 BG42                             Yuzhen tsentralen NUTS2      NA      NA      NA

  emp2002 emp2003 emp2004 emp2005 emp2006 emp2007 emp2008 emp2009 emp2010
1    2741  2834.7  2922.6  2981.9  3110.0  3252.6  3360.7  3253.6  3052.8
2      NA  1412.0  1445.6  1476.3  1529.5  1581.7  1632.2  1571.9  1465.9
3      NA   315.7   318.3   314.6   327.7   345.4   359.3   341.3   313.7
4      NA   335.1   344.9   344.4   352.0   368.3   374.4   365.6   336.0
5      NA   350.5   361.3   389.3   405.0   413.4   429.1   409.5   387.5
6      NA   410.6   421.1   428.0   444.8   454.6   469.4   455.6   428.7
7      NA  1422.8  1477.0  1505.6  1580.5  1670.9  1728.5  1681.7  1586.9
8      NA   855.4   894.5   920.7   974.1  1025.3  1060.2  1042.4   991.3
9      NA   567.3   582.5   584.9   606.4   645.6   668.3   639.2   595.7
```

The pattern in the matrixplot suggests the imputation strategy.



The strategy is similar to that outlined in the previous section. We use LOCF or AR to complete the proportional shares, adjust the proportions in terms of the hierarchical constraints, and then compute the totals. If time varying covariates are available, these could be used to sharpen the estimates, provided an appropriate model can be formulated.

8.6 Scenarios and strategies

The characteristics of the ESPON series suggest a blanket imputation strategy based on something such as hotdeck would not yield the best solutions. A combination of longitudinal and cross-sectional data means that a combination strategy will be required. A single missing cell can be completed using cross-sectional methods, as could a few cells missing in a column. Large quanities of data require different strategies, and we have outlined some of the possible strategies in the sections above.

The issue then is whether we can recognise a particular pattern of missing data, and then apply the appropriate strategy or combination of strategies. This remains a challenge.

Appendix1 Estimation R code

```
# Estimator_002.R
#
# C. Grasland, ESPON M4D, April 2013
# M. Charlton, NCG, May 2013
#


#
# (A.1) LOAD THE DATA
#
act<-read.table("test_bulgaria_act.txt", sep="\t",dec=".",header=TRUE)
act                                          # economically active
emp<-read.table("test_bulgaria_emp.txt", sep="\t",dec=".",header=TRUE)
emp
scenario <- emp                              # in employment


#
# (A.2) SET THE AUTOREGRESSION WEIGHTS
#
weight <- c(0.50, 0.30, 0.15, 0.05)                    # weights for for backwards autoregression
weight <- c(1.00, 0.00, 0.00, 0.00)                    # equivalent to LOCF
weight <- c(1.9656988, -0.9656988, 0, 0)               # from autoregressive fit, min AIC
nw <- length(weight)

# (B) CREATE CHILD -> PARENT REVERSE LINKAGE

rownames(emp) <- emp$code                             # label the rows with the NUTS codes
str(emp)                                              # check the data types and lengths
emp$NUTSlevel <- nchar (levels(emp$code)[emp$code]) - 2              # get the NUTS level (code is
a factor)
emp$parent    <- ifelse (emp$NUTSlevel > 0, substr(emp$code,1, emp$NUTSlevel+1), "") # get the
parent rows indices

#
# (C) COMPUTE HIERARCHICAL PROPORTIONS
#
maxRow <- nrow(emp)                          # length of the dataset
dataCols <- 8:15                             # columns with valid data
props <- emp                                 # copy the data frame
props[2:maxRow,dataCols] <- NA               # initialise results initially to NA
props
#
# Compute the proportions
#
for (i in 2:maxRow) {                                          # go through the rows
      parentRow <- which(rownames(emp) == emp$parent[i])       # find the parent row
      props[i,dataCols] <- emp[i,dataCols] / emp[parentRow,dataCols]     # compute the
                                                                         # proportions
}
print(props)

#
# (D) COMPLETE THE NATIONAL TIME SERIES
#

final <- emp                                          # copy the original data frame
final[1,4] <- sum(final[1,5:8] * weight)              # national totals

#
# (E) Update the proportions with the autoregressive adjustment
#

adjustCols <- 7:4                                  # columns to be adjusted

for (updateRow in 2:maxRow) {                      # do each row separately
   for (updateCol in adjustCols) {                 # backwards adjustment in each column
       props[updateRow,updateCol] <- sum(props[updateRow,seq(updateCol+1, updateCol+4)] * weight)
   }
}

#
```

```
# E.1 - deal with rounding errors - first sum proportions over the NUTx codes
#
checkColumns <- 4:15
summaryTable <- aggregate(props[,checkColumns],by=list(emp$parent),FUN=sum)     # check the totals
rownames(summaryTable) <- summaryTable$Group.1        # easy indexing - eventually divide children
by 1/summary
print(summaryTable)                                   # there's some rounding error

#
# E.2 - constrain the proportions to sum to 1
#

summaryColumns <- 2:13
for (checkRow in 2:maxRow) {
   checkParent <- props$parent[checkRow]                        # find the parent for this row
   whichParent <- which(rownames(summaryTable) == checkParent)  # find the row in the summary Table
   props[checkRow,checkColumns] <- props[checkRow,checkColumns] /
summaryTable[whichParent,summaryColumns]  # Update
}


#####################################
# (F) compute the populations
#####################################

for (level in 1:2) {                                              # do level 1 then level 2)
   adjustRows <- which(final$NUTSlevel == level)                  # rows for this level
   for (irow in adjustRows) {
      parentRow <- which(rownames(final) == final$parent[irow])    # find the parent row(s)
      final[irow,adjustCols] <- final[parentRow,adjustCols] * props[irow,adjustCols]
   }
}


#
# (G) Rounding error check
#
final[1,4:8] - colSums(final[c(2,7),4:8])        # NUTS1 columnSums should equal parent NUTS0
#     emp1999 emp2000 emp2001 emp2002 emp2003
#  BG       0       0       0       0    -0.1

#

final[2,4:8] - colSums(final[3:6,4:8])           # NUTS2 columnSums should equal parent NUTS1
#      emp1999 emp2000 emp2001 emp2002 emp2003
#  BG3       0       0       0       0     0.1


final[7,4:8] - colSums(final[8:9,4:8])            # NUTS2 columnSums should equal parent NUTS1
#      emp1999 emp2000       emp2001 emp2002 emp2003
#  BG4       0       0 -2.273737e-13       0     0.1



#########################################################
# H - plot the series
#########################################################;'

emp.ts <- ts(t(final[,4:15]), start=1999, frequency=1)
plot(emp.ts,main="Backforecast: LOCF")
```