OVERCOMING R ANGST: THE TOOLS THAT HELP STATISTICIANS LEARN AND USE R EFFECTIVELY

USING R IN THE STATISTICAL OFFICE: THE EXPERIENCE OF STATISTICS NETHERLANDS AND STATISTICS AUSTRIA

AUTOMATION OF PUBLICATIONS IN OFFICIAL STATISTICS USING R

SMALL AREA ESTIMATION METHODOLOGY (SAE) APPLIED ON BOGOTA MULTIPURPOSE SURVEY (EMB)

SIMPLY CLUSTERING. MAKING NEW SENSE IN THE FIVE FACETS MINDFULNESS QUESTIONNAIRE

BRIEF OVERVIEW OF SURVEY SAMPLING TECHNIQUES WITH R

BAYESSLANDMOFOREST: AN R IMPLEMENTATION OF BAYESIAN RANDOM FOREST FOR REGRESSION ANALYSIS OF HIGH-DIMENSIONAL DATA

BAYESIAN METHOD WITH CLUSTERING ALGORITHM FOR CREDIT CARD TRANSACTION FRAUD DETECTION

THE PERFORMANCES OF R GPU IMPLEMENTATIONS OF THE GMRES METHOD

ECONOMIC INTERDEPENDENCE BETWEEN THE GDP AND EXTERNAL TRADE IN ROMANIA - A VEC METHOD ANALYSE

1/2018

www.revistadestatistica.ro
CONTENTS 1/2018

OVERCOMING R ANGST. THE TOOLS THAT HELP STATISTICIANS LEARN AND USE R EFFECTIVELY 3
Adrian OŢOIU
Bucharest University of Economic Studies, Department of Statistics and Econometrics
Emilia ŢIŢAN
Bucharest University of Economic Studies, Department of Statistics and Econometrics

USING R IN THE STATISTICAL OFFICE: THE EXPERIENCE OF STATISTICS NETHERLANDS AND STATISTICS AUSTRIA 15
Alexander KOWARIK
Statistics Austria
Mark van der LOO
Statistics Netherlands

AUTOMATION OF PUBLICATIONS IN OFFICIAL STATISTICS USING R 31
Guido SCHULZ
Federal Statistical Office of Germany (Destatis)

SMALL AREA ESTIMATION METHODOLOGY (SAE) APPLIED ON BOGOTA MULTIPURPOSE SURVEY (EMB) 45
José F. ZEA
Santo Tomás University, Bogota, Colombia
Felipe ORTIZ
Santo Tomás University, Bogota, Colombia

SIMPLY CLUSTERING. MAKING NEW SENSE IN THE FIVE FACETS MINDFULNESS QUESTIONNAIRE 61
Elena DRUICĂ
University of Bucharest, Romania
Rodica IANOLE - CALIN
University of Bucharest, Romania

BRIEF OVERVIEW OF SURVEY SAMPLING TECHNIQUES WITH R 83
Camelia GOGA
Université de Bourgogne Franche-Comté

BAYESRANDOMFOREST: AN R IMPLEMENTATION OF BAYESIAN RANDOM FOREST FOR REGRESSION ANALYSIS OF HIGH-DIMENSIONAL DATA 95
Oyebayo Ridwan OLANIRAN
Universiti Tun Hussein Onn Malaysia
Mohd Asrul Affendi Bin ABDULLAH
Universiti Tun Hussein Onn Malaysia

Romanian Statistical Review nr. 1 / 2018
BAYESIAN METHOD WITH CLUSTERING ALGORITHM FOR CREDIT CARD TRANSACTION FRAUD DETECTION
Luis Jose S. SANTOS
De La Salle University, Philippines
Shirlee R. OCAMPO
De La Salle University, Manila, Philippines

THE PERFORMANCES OF R GPU IMPLEMENTATIONS OF THE GMRES METHOD
Bogdan OANCEA
University of Bucharest
Richard POSPISIL
Palacky University of Olomouc

ECONOMIC INTERDEPENDENCE BETWEEN THE GDP AND EXTERNAL TRADE IN ROMANIA - A VEC METHOD ANALYSE
PhD student Artur - Emilian SIMION
The Romanian Academy / National Institute of Statistics Romania
PhD Student Florentina Viorica GHEORGHE
The Romanian Academy/National Institute of Statistics-Romania
Overcoming R angst. The tools that help statisticians learn and use R effectively

Adrian Oţoiu (adrian.otoiu@csie.ase.ro) - corresponding author
Bucharest University of Economic Studies, Department of Statistics and Econometrics

Emilia Ţiţan (emilia.titan@csie.ase.ro)
Bucharest University of Economic Studies, Department of Statistics and Econometrics

ABSTRACT

This paper makes a review of the most popular graphical user interfaces (GUIs) in order to help new and unexperienced users learn and use R. Although R is free and benefits from state-of-the-art implementations of statistical algorithms and methods, it is not the first choice among statistical software users. This can be attributable to a lack of user-friendly interfaces similar to the ones available for other statistical packages. However, there are several GUIs that enable R users to use it quickly and effectively without having to use interactive programming and learn R code. For standard statistical analysis, RCommander proves to be the best solution that not only offers standard functionalities that are similar to commercial statistical software, but allows users to visualize, use, and customize the codes that perform standard data and statistical procedures, thus helping them to effectively learn R. For more advanced users, there are two main specialized GUIs RStudio, which facilitates the use of R through the use of scripts and generation of interactive applications and reports, and Rattle, specialized for data mining procedures and algorithms.

Keywords: software, open source, R, GUI.

JEL classification: C87

1. INTRODUCTION

R has distinguished itself as one of the most powerful statistical packages available. R has become the dominant analytics package as the share of analytics professionals using it has increased from 20% in 2007 to over 70% in 2015 (Rexer, 2015), being also the primary tool for 36% of the analytics professionals, far ahead from SAS, IBM SPSS Modeler and Knime, with 7% each (Rexer, 2015).

Its versatility and advanced capabilities are often superior to mainstream tools such as SPSS, SAS, Stata, EViews, not only in terms of being able to accommodate different data formats and statistical methods, but also to produce superior graphs. However, its steep learning curve is preventing...
many users to adopt it and makes them revert, or keep on using, other user-
friendly, commercial software.

While having these qualities and being available free of charge recommends R as the ideal choice for statistical analysis, its basic interface, a 
console with minimal options, prevents most users from learning and using it. The Rexer Analytics 2015 Data Science Survey shows that, although R is by far 
the most popular analytics software, and over 70% of analytics professionals use R, more than half of them do not use it as their primary analysis tool (Rexer, 2015).

However, the widespread use of several Graphic User Interfaces (GUI), which offer several alternatives to the then dominant use of R command line, 
have reduced the share of the latter from 51% in 2011 to only 17% in 2015, 
with the top R interface, R Studio, being used by 68% of R users in 2015.

Today, several R GUIs offer extensive point-and-click functionalities, 
which can help flatten, or even skip, the learning curve and allow users to take 
advantage of Rs diverse powerful capabilities in much the same way as in 
other commercial statistical packages. In this paper, we attempt to discuss the 
most popular R interfaces available, and focus on some powerful capabilities 
that are often not available outside the chosen interfaces.

2. LITERATURE REVIEW AND GENERAL 
FRAMEWORK

A survey of data miners prepared annually by Rexer Analytics puts R 
and its popular interfaces at the top of the most used analytics tools available, 
with 76% of the analytics professionals using it, also being the primary tool for 
36% of them (Rexer, 2015). Also, the same source shows that, between 2011 and 
2015, about 1/3rd of the users have switched to a GUI tool and abandoned using R 
console in their work in favor of one of its most popular interfaces (Rexer, 2015).

The primary graphical interface that is the most commonly used R 
environment is RStudio. Except for the command line, the next most popular 
interfaces are RCommander, Emacs and Knime. Among graphical user interfaces 
in R, we distinguish general GUI interfaces, RCommander, Deducer and RKWard 
(Valero-Mora and Ledesma, 2012), and specialized interfaces that are either 
for programming (e.g. gWidgets, ESS/Emacs) or extensions of other software 
packages into R, among which Knime, Statistica, IBM SPSS Suite (Rexer, 2015).

Among the most popular analytics software, R is perceived as one of 
the most difficult ones, with a steep learning curve (Muenchen, 2014), due to 
a variety of reasons, among which: 1) Help files are not always designed for 
beginners, 2) Misleading function or parameter names, 3) Inconsistent syntax, 
4) Difficult manipulations of variables and data management, 5) Unformatted 
output and different output formats, 6) Lack of GUI.
Also, while R is the most used analytics software, 64% of analytics professionals still use other software as their primary work tool (Rexer, 2015).

3. A STOCK-TAKING OF R INTERFACES

Based on our review, there are three types of GUI’s for R: interfaces to use R from different programs, interfaces for completing programming tasks in R, and, the most useful type, dedicated GUI interfaces, a classification based mostly on the work of Valero-Mora and Ledesma (2012).

As the scope of the current paper is to examine the interfaces that enable analysts and statisticians to use R in much the same way as commercial software packages are (e.g. SPSS, SAS, Stata, EViews), we will focus on the dedicated GUI Interfaces that can help harness the power of R, make the R learning curve smoother, and are available for free.

The interfaces to be examined are RStudio, the most popular R environment, RCommander, an interface similar to commercial software packages, and Rattle, an interface dedicated for performing data mining tasks.

3.1. Essential capabilities of interfaces

All GUIs have basic capabilities built in that support customary operations. First of all, there are file manipulation capabilities, that enable importing, exporting, visualization and editing of data files, and allow the use of different formats, among which text, CSV, SAS, SPSS and Stata.

The modular structure of R, which require installing and loading specialized packages to perform specific operations, is integrated in these GUIs. RStudio and RCommander have quick and easy package management functionalities. Rattle makes an exception as it does not offer explicit loading and dropping packages.

In addition, all of them feature windows which display the code used to perform needed commands, and a console which displays the results of commands and operations performed.

Data used can be visualized in convenient ways in all GUIs, allowing for quick examination of data, finding out their properties, changing them into different types (e.g. text to numeric, numeric to factor levels, etc.) and convenient editing if needed.

3.2. Distinctive features and specialisations

What makes the difference between the three interfaces, RStudio, RCommander and Rattle, are their specific capabilities, the targeting for specific needs, and focus for different user types and levels of proficiency.
RStudio is essentially programming-oriented, providing the options of creating scripts, projects and documents. It also has useful capabilities for importing and exporting files, managing datasets, display of the workspace structure, plots, packages and command history (see Figure 1).

This fits the initial purpose of RStudio, designed as an interface to improve the use of R through several features aimed at enhancing productivity through coding and debugging facilities. Everything else that RStudio offers revolves around these features, which aim at providing enhanced functionalities to easy manage the workspace, see available files and variables, packages that are available and can be activated or deactivated, and create documents in HTML, pdf and Word, and interactive applications using the package Shiny.

So far RStudio has emerged as the most popular GUI for R. Its share among the R tools has jumped from 8% in 2011 to 62% in 2015 and has become the dominant environment to use R (Rexer 2015).

RCommander is an point-and-click interface (Fox, 2017), which renders some of the essential capabilities of R and allows users focus more on statistical work without having to learn the specific R command for performing statistical analysis (Fox and Bouchet-Valat, 2016).

File and data management capabilities enable point-and-click data imports from a variety of formats (text, csv, Excel, Stata, SPSS, Minitab and SAS), merge datasets, subsetting datasets based on user-defined criteria, transforming and ordering variables, exclude cases with missing data, etc.

RStudio interface overview

Figure 1

Source: RStudio software.
Beside the essential file and data management capabilities, and package management capabilities, RCommander has a collection of statistical functions, graphs, modelling options, statistical tests, and distributions. They cover most of the statistical tools and procedures required for regular quantitative work: summaries, tests for means, variances, and normality of distributions, ANOVA, non-parametric tests, correlations, and contingency tables. Among the advanced statistical methods, RCommander offers convenient implementations of regression analysis, GLM models, ordinal regression models, multinomial logit, factor analysis, cluster analysis and principal components analysis. Scale reliability for testing composite index components is also available following an implementation that is unique to RCommander. Besides a complete set of graphs, it also has a strong modelling set of options, which includes coefficients testing, model selection tools which implements stepwise algorithms, specification and diagnostic tests, among which Durbin-Watson, Breusch-Pagan, VIF factors, and diagnostic plots.

To sum up, RCommander is the closest GUI to established commercial software, offering comparable tools and options for data and statistical analysis. Another feature that is not easily accessible is the possibility to customize menu options to include commands that are not currently available, or new procedures. There are also some plugins, such as RcmdrPlugin.BCA, which reconfigures the menu to make it amenable for performing business analytics tasks, and RcmdrPlugin.epack, which implements time series analysis, added as two main menu options in addition to the regular RCommander plugin.
Rattle is a data mining interface, which, aside from automating data mining tasks through a simple, easy-to-use interface, also provides a stepping stone for the use of R for advanced data analysis (Williams, 2009).

Its purpose is somewhat similar to RCommander, as it aims to provide a user-friendly environment for data mining and modelling. However it appears to be a lot more automated, customized for data mining, starting from data import, and continuing with data exploration, testing and transformation. Three tabs, cluster, association and model provide a wide range of modelling options, which are tested for their performance in Evaluate tab. All operations appear in logs with comments, and scripts can be exported so as to be used later on (Williams, 2011), either in R Console or in RStudio (Togawave, 2015).

Although it makes available a lot of point-and-click capabilities, Rattle is not very flexible in terms of customization; thus, besides data miners and modelers, Rattle remains off limits, or presents little interest for regular data analysis or statistical work. Another drawback are the recent issues concerning its recent smooth installation and operation, which can be successfully solved by power users but remain off-limits for regular researchers. Also, for more advanced work, the company that produced Rattle, Togawave, recommends
the use of RStudio to create and run data mining tasks that go beyond the capabilities offered by Rattle (Togawave, 2015).

**Rattle interface**

![Rattle interface](image)

Source: RCommander package

### 3.3. Strengths and weaknesses for each interface

A SWOT analysis can shed light on each of the examined interfaces so far, putting in perspective all advantages and shortcomings of each of the examined interfaces.
### SWOT analysis of the analytics GUI for R

<table>
<thead>
<tr>
<th></th>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RStudio</strong></td>
<td>One of the best development environments</td>
<td>Except basic file and package manipulation, requires knowledge of R code.</td>
</tr>
<tr>
<td></td>
<td>Allows production of applications and documents in different formats (Rpubs,</td>
<td>Issues with graphs as size of graph window prevents their proper display.</td>
</tr>
<tr>
<td></td>
<td>Word, Pdf, Html)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allows free posts of R output and files in RPubs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quick manipulation and selection of active files and packages in right-hand</td>
<td></td>
</tr>
<tr>
<td></td>
<td>windows</td>
<td></td>
</tr>
<tr>
<td><strong>RCommander</strong></td>
<td>Does not require prior knowledge of R</td>
<td>Changes of active files difficult</td>
</tr>
<tr>
<td></td>
<td>Allows performing most of the operations available in commercial software</td>
<td>Undloading packages is done using RCode</td>
</tr>
<tr>
<td></td>
<td>packages</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model selection tools and capabilities</td>
<td>Poor integration of standard R commands which are not always functional</td>
</tr>
<tr>
<td></td>
<td>Customizable interface for specific tasks (e.g. time series, business</td>
<td></td>
</tr>
<tr>
<td></td>
<td>analytics)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parameter changes of menu functions</td>
<td></td>
</tr>
<tr>
<td><strong>Rattle</strong></td>
<td>Integrated layout for performing data mining</td>
<td>Poor flexibility for regular data analysis and statistical work</td>
</tr>
<tr>
<td></td>
<td>Detailed data mining framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementation of social analysis and natural language processing solutions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SWOT analysis of the analytics GUI for R

**Table 1 (continued)**

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>RStudio Development excellence</td>
<td>RStudio Other user-friendly interfaces and proprietary solutions</td>
</tr>
<tr>
<td>Integration with other R tools and other analytics software</td>
<td></td>
</tr>
<tr>
<td>Addition of user-friendly functionalities to enlarge user base</td>
<td></td>
</tr>
<tr>
<td>RStudio Other user-friendly interfaces and proprietary solutions</td>
<td></td>
</tr>
<tr>
<td>RCommander Becoming the standard entry-level R environment</td>
<td>Lack of developmental advances</td>
</tr>
<tr>
<td>Having a larger collection of plug-ins that can automate a larger number of R commands, and enable performance of latest analysis tools (e.g. natural language processing, social media analysis)</td>
<td>Features and evolution of other open-source software offering similar user-friendly analytics solutions</td>
</tr>
<tr>
<td>Support-based solutions for less tech-savvy users</td>
<td>Lack of appropriate popularization of its features and capabilities</td>
</tr>
<tr>
<td>Capabilities available for advanced users such as document and application production</td>
<td></td>
</tr>
<tr>
<td>RCommander Support-based solutions for less tech-savvy users</td>
<td></td>
</tr>
<tr>
<td>RCommander Lack of appropriate popularization of its features and capabilities</td>
<td></td>
</tr>
<tr>
<td>Rattle Improvements in flexibility of uses for data analytics and statistical work</td>
<td>Other open-source data mining solutions offering</td>
</tr>
<tr>
<td>Widening the target users range to include statisticians and data analysts which do not need the standard data mining framework</td>
<td>Poor support for bugs for regular users may undermine its usability for less tech-savvy users to which it addresses</td>
</tr>
<tr>
<td>Constant acquisition and integration of latest analytics methods</td>
<td>Improved basic functionality to make installation and use more easy and bug-free</td>
</tr>
</tbody>
</table>

Source: authors’ compilation

Some important considerations must also be made, to complement the essentially concise nature of the SWOT analysis. RStudio requires knowledge and a good mastery of R commands, and can be thought of as a programming environment rather than a tool to learn R. Although it helps learn R through its interactive script window, the user still has to learn each and every command used using R reference sources, among
which the most important are: package documentations, help files, books, among which UseR! Springer series, and specialized forums. Thus, it is not much help for R beginners and less tech-savvy users, and its emergence as the main interface for R does not ensure that R will become the preferred software used for data analysts and data scientists.

Rattle is a point-and-click interface with powerful data mining capabilities. Its substantial tools for data processing and manipulations can be of help for intensive data mining and modelling work, which is well supported by a wide range of modelling options, among which: cluster analysis, association analysis, decision trees, linear regression, SVM, neural networks. Also, it allows testing of the obtained models on different splits of the original data, using advanced tools (e.g. ROC curves, lift, etc.). However, for more advanced work, commands have to be exported and modified/used separately, either in the R console or in RStudio. Also, its recent issues are likely to affect its popularity and chase away users towards similar open-source data mining solutions (e.g. RapidMiner, Weka).

R Commander is also a point and click interface, which exhibits the highest degree of flexibility and versatility compared to the other two interfaces. Its two-window interface allows the user to see immediately the script generated by the point-and-click commands, and the corresponding output in the window below it. Based on the R documentation, the user can modify the parameters of the GUI-generated commands to suit her needs. Its commands range from data processing commands (importing, variable transformation, merging datasets, deleting observations, etc.), to statistical tests and models, graphs, model testing, selection and comparison, and statistical distributions. R Commander also benefits for dedicated built-in commands, such as Scale Reliability, and from plugins that change the menu options to make it suitable for particular types of analyses. Among the most well-known plugins are epack, with time series modeling options, and BCA for customer and business analytics, which has data mining capabilities similar to Rattle. Other specialized plugins are available at http://www.rcommander.com. So far it is the best environment that can smoothen the steep learning curve for R and make it accessible to wide range of users.

4. CONCLUSION

Although R is a programming-oriented language, there are tools that make its use fairly simple and similar to commercial software packages such as SPSS, STATA, etc. These tools offer a wide range of options when it comes to how and how fast much one wants to learn R and, at the same time, do regular statistical and data analytics work in an easy way comparable with dedicated statistical packages.
For learning R, the most helpful interface is RCommander. With a lot of import and data manipulation options, it allows the user to see the scripts used with point-and-click commands and modify them based on options available, or to use pre-define scripts loaded with the file menu. The specific plug-ins, and options to customize RCommander makes it the best starting point for any user without a programming background, who is accustomed to using commercial software packages, to do statistical work and at the same time learn R. However, its layout is not meant to run larger scripts, however, this can be overcome by using it under RStudio.

For data mining professionals, Rattle may be a better option since it offers an integrated environment that allows the use of selected R commands with no knowledge of R required. However, the transition from point-and-click to script-based work is not as straightforward as in RCommander, therefore the learning curve is still steep and cumbersome for those without any programming background and skills, and it may actually induce a gap that prevents the average user make the transition. Its occasional issues with respect to its installation and use may pose a threat to its future as a viable open-source data mining environment.

RStudio is an interface for users who already have knowledge of R, or want first to learn R and perform statistical/analytical work as they gain the necessary knowledge. Its capabilities make it the best choice for using R for experienced users. RStudio can also integrate RCommander or Rattle, facilitating the transition to advanced use for those who learned to use R with either of, or both of, these two interfaces.

Based on our review, we believe that the three GUIs are sufficient for overcoming the difficulties of using R, depending on the types of users and on the specific tasks to be performed. In the case of RCommander, its functionalities and features are very similar to other commercial statistical packages. However, for advanced users, these interfaces are not a perfect substitute for actually learning R and become familiar with its commands and options. Nevertheless, learning commands and code for statistical software is also needed for other leading commercial software (e.g. SAS, SPSS) for the purpose of carrying out advanced statistical and analytics work. Therefore, the limitations of the three GUIs for R should not be seen as a major disadvantage, but rather as a state of affairs that is consistent with making the transition to advanced use straightforward and accessible, perhaps more so than other commercial packages that do not benefit from specialized information disseminated via online documentation and forums, and an user community as helpful as the one for R.
Acknowledgment

A presentation of this paper was made at the 11th International Conference on Applied Statistics held in Brasov, on June 2, 2017.

References
Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria

Alexander Kowarik
Statistics Austria

Mark van der Loo
Statistics Netherlands

ABSTRACT
Over the last decade the R language and statistical environment has received a surge in popularity. Indeed, R has become one of the central tools for modern statistics and data science. Slowly but certainly, statistical offices are introducing R as a valid tool for statistical production as well. The Austrian and Dutch offices were amongst the first national statistical institutes to approve R as a tool for production. The aim of this paper is to describe how R was introduced and currently used in our offices. On one hand, we focus on practical issues such as infrastructural considerations, the use of R-based software by non-R programmers, and update policies. On the other hand we describe the activities that were undertaken to educate new users in the use of R. We also discuss work that was contributed back to the R community in general and the official statistics community in particular. Finally, we discuss how collaboration and standardization take place in an open source environment.

Keywords: R Software, Official Statistics
JEL Classification: C80, C88

INTRODUCTION
The R language exists since 1993, and back then it was a new and exotic thing. Nowadays, at least for over a decade it is the dominant programming language and statistical software in academia in the area of Statistics. Indeed, IEEE (2017) now ranks R as the 6th most popular programming language worldwide. Similarly, the TIOBE (2017) index currently lists R as the number 15 language while the PYPL (2017) popularity index ranks R as the number 8 language. This is impressive especially when one considers that the special purpose language R competes in these rankings with general purpose languages such as Python, C, C# and Java. In each of these ranking, R has risen over the
last years and has surpassed all other statistical languages, including SAS, in popularity.

It is natural to ask why R became so popular. Although formal research on this question appears to be lacking, a few properties and circumstances that make such a growth plausible readily present themselves. On one hand, R may be 'riding the wave of data science' that is currently washing over industry, government and academia. R probably was simply there at the right place at the right time -- offering a wide range of statistical and data handling functionality with a convenient and programmable interface for free when data science took off as a field. Additionally, the choice to publish R as an open source tool was an important cornerstone for its success.

Before continuing however, it is worthwhile pointing out a few properties typical to R, its developers, and its community. First, R is developed by a dedicated core team with a strong focus on the software’s correctness, stability, and backward compatibility. The R core team consists mostly academics working in the field of (computational) statistics --- a field that benefits greatly from the availability of such software. For users, this means that R can be relied upon both for functionality and API stability. Second, R is developed under the GNU general public license. This entails that anyone is permitted to copy, study, alter, and redistribute R and its source code. Third, R is extensible with code bundles ('R packages') that are easily exchanged. A valid R package is endowed with a description of its purpose, version, dependencies, and authorship. Every function exposed to the user must be described in a standardized manner for a package to be valid. Package authors can offer their work for publication on a central repository such as CRAN. At CRAN, packages are subjected to intense scrutiny, both prior to acceptance and afterwards. In particular, CRAN ensures that no package entering the repository breaks any other package that depends on it, directly or indirectly. Users can therefore rely on the fact that all of the 11,000+ packages currently on CRAN have a minimum amount of documentation and a certain amount of interface stability. Fourth, R has attracted attention from the industry, which lead to the development of many tools that facilitate the workflow from initial analyses to publishing results, deploying models, or sharing software/applications. Examples include integrated development environments, database connectivity tools, software test suites, visualisation tools, connectivity with web applications and more.

The growth of R has led to an active international (online) community of users and developers who efficiently exchange information via mailinglist, fora (stackoverflow), and social media (Twitter). There are several conference series devoted to the use of R. The most well-known include the useR!
conference, which is considered the main worldwide user meeting, the satRday initiative, the twice-a-year Effective Applications of the R Language (EARL) conference, and of course the ‘use of R in official statistics’ conference (uRos). R is widely supported by the industry with major software vendors offering an interfaces to and from R from their products. The software industry aims to advance R and its community by combining efforts through the R consortium, an organization that ‘support[s] the worldwide community of users, maintainers and developers of R software’ (R consortium, 2015).

Still, the official statistics branch and especially national statistical institutes seem quite hesitant to make the switch from a traditional, often commercial, solution to an open-source alternative. Of course any switch of software product will demand resources to handle legacy code, staff (re) education, and so on. Reasons, particular to R or other open source products, seem to include the lack of formal procedures to handle free software with open source licenses and the perceived risk of working with software that is apparently not backed by commercial support.

The purpose of this paper is to discuss the experience of introducing and using R at the statistical office, from the perspective of two statistical institutes. Specific production steps or statistical products are not described in this paper. In Templ and Todorov (2016) you find an overview of R packages aligned to the general statistical business process model.

In the next section, we discuss how R was introduced into our offices as a valid tool for statistical production. After this we describe some developments in the infrastructure hosting the internal R distribution. In the fourth section we focus on how R packages are used at the statistical office and we provide a brief overview of the packages that have been contributed to the community. A special section is devoted the modes of collaboration that are common in a community-driven tool such as R. We conclude with a short summary and conclusion.

**REMARKS ABOUT ADOPTING R AT THE STATISTICAL OFFICE**

**Austria**

The process of adopting R as a standard tool at STAT was cumbersome and long, but we are now at stage with a lot of expertise and using R in different areas of the production process. Since 2004 it was used “unofficially” by one user at the methods unit, but new members of the methods unit soon joined him in using (and loving) the R environment. Basically, it was a grassroots movement and merely tolerated by the IT department, where still SAS was the only strategic statistical software. Soon after it was used to build our first
R package sdcMicro which was released to CRAN in May 2007 (VIM also in 2007 and sdcTable followed in 2009). Between 2004 and 2009 it was more or less tolerated that the methods unit used R and developed new methods with it, but it was not be used in production.

A wider audience got a first glimpse during the methodological training courses, where R was used since 2005 as a computer assisted training tool for visualizations, animations and to ask simple questions about the course content (see Dinges et al., 2011). The usage of the tool required no R knowledge but sparked the interest in several course participants.

In 2009 we got the official approval to use R, but only within the methods unit and we got our first server to use R on bigger problems, especially in the area of statistical disclosure control. Finally in 2013, R was approved as a standard tool for everyone at STAT and it could be used where deemed fit. In the same year a training programme for R was established. At first it was a week long course with everything in it from basic data manipulation, graphics to generating dynamic reports. The course programme was steadily improved and now consists of 6 modules. Over 100 participants took at least one of these courses. In addition to the training programme a wiki was set up, however it never really became lively and is not used anymore. To foster the exchange of knowledge and ideas we now have regular user group meetings (about twice a year). During these meetings a number of R users present their projects. About 40 employees use R at least once a week at the moment.

The methods unit, which was among the first user of R at STAT and has a lot of experience in using it, is providing support to all R users at STAT. The support requests are handled via the issue tracking system Jira. For IT-related issues, the methods unit interacts with the IT department through another Jira project, e.g. when a new version of the desktop R package is generated or some administrative tasks are necessary on the servers.

**Netherlands**

R was accepted as a standard (approved) tool for statistical production in 2010. The incentive for introducing the tool was the lack of a main tool or language for statistical programming at the office. At that time, tools such as Matlab, Stata, Ox, and Splus, were used in several niche places in the office while many production systems would rely on SPSS, standard office software, SN’s own software modules (Bascula, Blaise), and custom-build software components for statistical production. After a preselection, R and Matlab were chosen as two of the main candidates to provide a scientific programming language. A detailed comparison of needs and features surfaced R as the preferred language. The fact that R is open source was one of the deciding factors.
Once it was decided that R would be introduced, a project was set up to decide on deployment, develop programming guides, and a wiki for R users. The fact that introduction of R was an ‘official project’ with a dedicated project leader helped a tremendous amount in overcoming some organisational issues. For example, processes and contracts at the IT department were completely geared towards commercial products and vendors. There simply was no procedure to download and install an open source product for production. The fact that it was necessary to download R, CRAN and an interface separately (this was before RStudio, and Notepad++ was used with an R-interface) was also new. Finding solutions to such issues require some time and resources.

In the beginning, three different internal ‘R distributions’ were developed, distinguishing use for production, analyses and development. This practice that was eventually simplified into a single centralized distribution (to be discussed later). At the same time the initiative was taken to set up a local R user community. A few of the ‘expert’ programmers took it upon themselves to organise informal biweekly user meetings, set up trainings (twice per year) and to be available to help with operational questions. The driving aim was to have an internal community that mimics the international R community, in terms of openness and (informal) information exchange.

Currently, about two hundred of Statistics Netherlands’ employees have followed the internal training course. Initially the course was aimed at statisticians who need to program a statistical production system but gradually the course was adapted to more ad-hoc users (analysts) and people who need to run and adapt scripts rather than develop them from scratch. More advanced (optional) workshops are taught occasionally as well to still serve the more advanced user community. A wiki was set up as well. It never took off in the sense of users contributing a lot of content. The wiki is still useful to collect answers to questions, workarounds, information on upgrade procedures, package installations and so on.

Statistics Netherlands has been decentralizing development of statistical systems over the past few years. In this context, R was only the first tool that has been introduced. Other tools that have been introduced since R include Python (2012-13), git (for version control, 2014-15) and Spark (2016). In all these cases, the importance of setting up a community in the way that was pioneered during the introduction of R was recognized. Explicit attention was paid to set up user meetings and other communication channels such as a community wiki during introduction of all these tools.
INFRASTRUCTURE

Austria

In the future R will be only available to users via dedicated R servers and the interface will be RStudio Server Pro. This setup allows for central maintenance of the R installation and for a very flexible approach to resource management. The users can interact with the server through the familiar interface of RStudio in the browser, so the training efforts are minimal when switching from RStudio Desktop.

Every user is limited in available resources, e.g. a user will begin with 4 GB of memory available, but this could be extended easily on a per user basis. Since the servers are virtualized it is also possible to allocate additional resources to the server even for just a short amount of time when a peak usage is predicted.

The key feature of the commercial version of RStudio Server is the possibility to have multiple sessions per user (with the open source version only one session per user is possible). Additionally, there are a number of enterprise features implemented in the commercial version of RStudio Server. At the moment the setup of a new more powerful R server with Rstudio Server Pro installed is under development and when ready users will be transferred to this server incrementally.

STATs current approach is described in the following, however this approach is deprecated and will probably be abandoned in 2018:

Through the centralised IT service everybody is able to request an installation packages of R, this installation package contains the following:

- R
- A predefined RProfile which gives information on startup about getting support and sets the CRAN mirror to the Austrian mirror, which is defined as an exception in the firewall.
- A “curated” collection of R packages. This list is curated by the methods unit and additional packages can be request by every R user. However, every R user can install additional R packages in a user specific library.
- RStudio
- MikTex (TeX distribution for Windows)
- Slik-Subversion (SVN command line tool for Windows)

The TeX installation is needed to be able to generate PDF documents from R Markdown files or R Sweave files and the SVN tools are necessary to access the internal SVN server, which can be used as version control for R projects.
About twice a year the R installation package is updated. This update is prepared in cooperation between the IT department and the methods unit. Currently, a Linux server with R, RStudio Server and all necessary tools is available to a limited number of employees with the need for more memory (or for parallel processing). The server has 128 GB of memory and 16 cores and is used for some heavy-lifting tasks in R.

Netherlands

R is made available for common (desktop) use by installing it in a read-only central directory. Users of R and RStudio need to run a small script that sets up a shortcut to the centrally installed versions. The main advantage of this approach is that one user can write a script for a second (non-R) user that need not install anything. The R engine can be called from any virtual machine with access to the central directory. This means almost any VM in practice and it also includes the batch environment where (R) scripts can be scheduled to run.

Users also have access to a MikTeX installation so pdf documents can be generated from RStudio with ease. The methods department provides LaTeX, Wordsm, and HTML templates for internal reports and (externally published) discussion papers in CBS house style. These templates are made available from RStudio by default.

When upgrading, an old R version is not replaced but rather a new version is placed next to it. That way, production scripts need not to be updated and retested against the new distribution. Users are advised to upgrade scripts to the new R version during regular maintenance (e.g. updating for new data sets). This procedure is mainly developed to guarantee a degree of independence between updating R and applications of it. After a few years, older R versions can be removed as they are no longer used.

Statistics Netherlands has recently implemented virtual remote desktop servers with 64GB memory and 8 cores each. Any user can request access to such machines to run heavier applications or to work comfortably with larger datasets. These machines use the same centrally installed R distribution with a locally installed version of RStudio.

R users have access to a curated standard set of packages which are installed centrally as well. At each upgrade, the R expert group revisits the current set of packages, deprecating obsolete packages and adding new ones. For example, in the earliest versions, RGTK was the preferred tool to build graphical user interfaces while Shiny is recommended currently.

If so desired users can install packages in a personal library, using a local copy of CRAN that was downloaded at the same time as the central
R version. Since Statistical Departments have the ultimate responsibility for quality of statistics, users are warned that usage is at their own responsibility. R experts frequently help choosing or testing packages for specialized purposes.

PACKAGES

Since the number of available packages is huge, there is a need for some curated information with direct usage for NSIs. There are two main sources for this: The CRAN Task View Official Statistics and Survey Methodology (Templ, 2017) and the Official Statistics Awesome Software List (Ten Bosch and Van der Loo, 2017). The main difference between the two lists is that the CRAN Task View focusses on R packages, while the Awesome List includes any open source software that can be directly downloaded, is used at an NSI for production of statistics, and is free (in the sense of freedom). Both lists are open for suggestions by users.

Journals, conferences and workshops are other obvious ways to stay in touch. Interesting conferences for official statisticians are the ‘use of R in Official Statistics’ conference (uRos), the main useR! conference series, the new European R user meeting (eRum) and the biennial EARL conference (Effective Applications of the R Language). Additionally, there is a quite active R community on twitter, so it might be beneficial to follow people with similar interest there. A starting point for that could be to look at the followers of the two authors (@alexkowa and @markvdloo) and the accounts they follow. Most conferences have twitter handles as well and by searching for the relevant hashtag (e.g. #useR2017), useful information often surfaces.

Public Packages

An R package may be thought of as ‘the fundamental unit of R code’ (Wickham 2014). Because packages contain both code and documentation, they are ideal containers for implementing novel methodologies in an organisation. Publishing R packages on CRAN has the additional advantage of possibly reaching more users, allowing or greater opportunity to find bugs in code or documentation. If the code is developed through an open environment such as github or or gitlab, other developers can contribute code or documentation as well.

Both Statistics Netherlands and Statistics Austria have published packages on the Comprehensive R Archive Network (CRAN). In the following, a concise overview of this work is provided.
Austria

Employees from STAT (and more specifically the method unit) have (co-)developed several packages which are available for download from CRAN, the most relevant are (see Templ et al., 2014 for details):

- **sdcMicro** (Templ et al., 2015), **sdcTable** (Meindl, 2017) for statistical disclosure control for micro and tabular data (supported by the CENEX statistical disclosure control)
- **simPop** (Temp et al., 2017) for the generation of synthetic data sets
- **VIM** (Kowarik et al., 2016) for visualization and imputation of missing data
- **x12** (Kowarik et al., 2014) for applying X13-ARIMA-SEATS to time series in R

These packages are already used in production in several NSIs and the packages for statistical disclosure control are under governance of the centre of excellence for statistical disclosure control which is partly funded by Eurostat.

All the public packages are hosted on Github, so it is easy to file an issue or to contribute to the development of any of the packages.

At the moment employees at STAT are working on a package that provides an interface to the EU GIS service GISCO, so it will provide the functionality of geocoding, reverse geocoding and routing. As STAT is part of the NETSILC3 project a package for bootstrapping and estimation of indicators and corresponding standard errors is also under development.

Netherlands

All published packages have been developed at the methods department. Most of them are a result of the annually updated research programme. Packages produced as a result of the research programme can be divided into packages focused on data editing, data visualisation, packages for handling large datasets and others.

In the area of data editing, the validate package (Van der Loo and de Jonge, 2017b) provides functionality to read, apply, investigate and maintain sets of edit rules. The results of confronting data with a set of edits can be conveniently summarized, exported or visualized. The validate package is fundamental to other data cleaning functionality. In particular, the errorlocate package (De Jonge and van der Loo, 2017) implements error localization functionality based on the paradigm of Fellegi and Holt (1976). A package for detection and removal of redundancies and inconsistencies in rule sets based on the validate package is currently under development and will be
published soon under the name validatetools. (De Jonge and Van der Loo, 2017). These three packages replace the editrules package (De Jonge and Van der Loo) which combined the functionality of rule application, management and manipulation. Work on a package that can export data validation results in the new ESS validation report standard (Van der Loo and Ten Bosch, 2017) is underway.

Several rule-driven data cleaning methods developed at SN have been implemented in the deductive package (van der Loo and De Jonge, 2017). This package provides methods for deductive imputation, finding typing errors in numerical data under linear (in)equality constraints. The package uses rule sets defined in the validate package. Package simputation (Van der Loo, 2017) offers a simple interface to many different imputation (model-based) methods, while rspa offers functionality to force imputed numerical values to satisfy linear (in)equality constraints. The recently published lumberjack package (Van der Loo, 2017b) can be used to trace (log) changes in a data set as it gets processed during data editing. A thorough discussion of both theory and application of data cleaning in R can be found in Van der Loo and De Jonge (2017c).

In the area of data visualisation, the tabplot package (Tennekes and De Jonge, 2017) implements a method for visualizing large, multivariate datasets while tabplotd3 (De Jonge and Tennekes, 2013) implements a web-based interactive version. The treemap package of Tennekes (2017) implements tree plots -- a space-filling visualisation of hierarchical data structures. Finally, packages tmap and tmaptools provide functionality for creating thematic maps (Tennekes, 2017b, 2017c).


Besides packages that are developed as part of the research programme, several employees at Statistics Netherlands have developed infrastructural packages for working with large data, either for other projects and/or in their free time. Worth mentioning are the LaF package (Van der Laan 2017) for handling large ASCII files, ffbase (De Jonge et al, 2016) for working with large binary files on disk (in ff format), and the lvec package by Van der Laan (2017b), which implements a disk-based representation of standard R vectors.

Finally, we note some packages developed by employees of statistics Netherlands that have been mostly personal interest and free time projects. The widely used whisker package of De Jonge (2013) implements the moustache standard for logicless templating, the stringdist package of Van der Loo implements a range of string distance metrics, and docopt (De Jonge, 2016) package facilitates documenting and parsing of command line options when running an R script from the shell.
**Internal Packages**

To get R code ready for the statistical production process it is advisable to wrap it in an R package. The distribution and the development process (versioning) of an R package is well defined and well manageable. Therefore, there are specific R packages for specific tasks in the process.

**Austria**

The methods unit develops internal R packages to facilitate repetitive tasks. One example is drawing a sample of household or persons, for that the R package sampSTAT was developed, the centrally stored rich frame can be easily downloaded from within R without SQL knowledge or even knowing how the database containing the frame looks like. Additional functionality is drawing samples with different designs and auxiliary functions such as telephone number search or accessing the address register to get the most current postal information.

Previously, many price-indices at STAT have been calculated using Microsoft Excel. Since this should be avoided, the package RPI allows defining a hierarchical structured price index and applying many useful methods such as switching the base period, splicing an index (e.g. combining indices with different base periods), applying quality adjustments or automated reporting/plotting on such index objects.

Based on the public R package for statistical disclosure control for tabular data sdcTable, the R packages anonLSE and anonKJE provide customized function for structural business survey and short term statistics. For the anonymization process, it is also taken into account that there are differences in classification of Austrian national publications versus European requirements and a coherent suppression pattern is computed.

**Netherlands**

At Statistics Netherlands, projects involving methodological advice sometimes include (co)writing an R package for a specific internal purpose. Examples include a package for statistical disclosure control in a specific area, a package for survey sampling capable of excluding units that were included earlier, a package for time series analysis, and most recently, a package for accessing and analyzing supply-and-demand tables for National Accounts. Some packages are (co-)developed by the methods department as a service to the statistical domain. In this case ownership is usually transferred when development is finished. In other cases, staff at the statistical domain develop the packages for local use.
COLLABORATION

Over the last years, R has surged in popularity both within and outside the Statistical Office. The rise of R (and other open source data science tools) went hand in hand with the rise of a very open and collaborative culture amongst a generation of developers in general and data scientists in particular. Statistics Netherlands and Statistics Austria have profited tremendously from the R community, both on- and offline. Our offices have also contributed back to the R community, for example by contributing packages. The informal way of collaborating in the open source community differs from the formalized collaboration instruments that are typical in the official statistics community. It is therefore worthwhile to highlight a few aspects of the way the open source community typically operates.

There are many ways in which one can choose to work together in an open source community. First of all, one can simply try or use an open source product. Each time software is used, it is in a way tested. In that sense every run contributes to its reliability. It is considered good (or nice) practice to acknowledge the author(s) of a package, for example by ‘liking’ or following the project site. Second, one can collaborate by advocating the software packages that have shown to be useful. Tools are both abundant and rapidly evolving, so even just pointing out a package on Twitter may help someone to discover the tool they need. Other typical ways to collaborate indirectly include answering questions on Q&A sites (stackoverflow is the most popular one) or mailing lists, write blog posts on applications, write a tutorial, or give talks at user meetings. A simple way to directly collaborate in an open source project is to provide feedback. One can post reports on performance, bugs in documentation or code, or discuss new features. Finally, the most direct way to collaborate is to contribute code or documentation to a project.

As for Statistics Austria and Statistics Netherlands: we are indeed using each other’s packages, in production as well as in research. Also, bug reports and documentation suggestions have gone back and forth between package authors of our offices. Besides that, both Statistics Netherlands and Statistics Austria have received bug reports, suggestions and comments from (official) statisticians worldwide. In the authors experience, such contributions significantly enhance the quality of published R packages.

These more or less informal collaboration modes stand in contrast with the often highly regulated environment that is international collaboration in official statistics. There, international collaboration projects are proposed, defined, tendered, funded, executed, deployed and finally evaluated. It also stands in contrast with the way things are standardized. Where in the official statistics community standards are often designed and chosen in a top-down
manner, in the open source community standards evolve by survival of the fittest: popular, useful ideas, standards, and software survive while others wither and die. Perhaps paradoxically, the freedom found in the open source community almost automatically induces a level of standardization. To illustrate, consider the example of IDE’s (integrated development environments) developed for R. Before RStudio was published in 2011 there were many different solutions in use: Notepad++ with NpptoR, Tinn-R, Eclipse with the StatEt plugin and more. All have their advantages and disadvantages, but once RStudio was published it took over a large chunk of users almost immediately. Something similar happened in the field of report generation. Before the ‘knitr’ package was first published in 2012, numerous developers worked on R packages allowing users to inject results of computations into a report. The useR!2012 conference even had a tutorial session and a presentation session demonstrating several solutions. Examples include Sweave (coming standard with R), odfWeave (Kuhn, 2014), and several others. Once the knitr package of Xie (2015) came out (it was first released in 2012), it quickly took over the ‘market’ for report generation because it filled a need in a way that appealed to many users. Both RStudio and knitr can currently be considered de facto open source standards and both are the result of bottom-up approaches to a particular problem.

The power of the bottom-up approach was recently recognized by the establishment of the R Consortium (2015). One of the main activities of this industry-backed group is to provide funds for project grants, where anyone from ‘the R community at large’ can write a proposal and apply for a grant. Over the last two years, eighteen projects have been funded. An example is the highly successful r-hub project (Csardi, 2015): a cloud service where package authors can run automated tests under different operating systems and build versions of R before submitting to CRAN. At the time of writing, the service has been running for two almost two years on an industry-sponsored cloud platform. Designing and building the infrastructure took an $80.000 grant and about one year time (R consortium, 2017).

In summary, the R community offers a wide range of collaboration possibilities allowing users and organizations with many levels of experience, skills, and needs to contribute. Since there are little or no formalities to take care of starting to collaborate is almost effortless.

**SUMMARY AND CONCLUSION**

In this paper, we have discussed the introduction of the open source tool R in the statistical office from the perspective of Statistics Austria and Statistics Netherlands. We have put particular emphasis on properties of R
as an open source tool which sets it apart from many tools used traditionally in statistical offices. Although moving from a Commercial off the Shelve (COTS) approach to an open source product did pose some organisational and perhaps cultural problems, our experiences have been overwhelmingly positive. The community-driven approach to collaboration and standardization that is common in the international data science community in general and in the R community in particular, is something that both authors would highly recommend to anyone working in the field of official statistics.

References
8. Edwin de Jonge, Martijn Tennekes, 2013, tabplot3d: Tabplot3d, interactive inspection of large data. R package version 0.3.3. https://CRAN.R-project.org/package=tabplot3d
15. Meindl, B., 2017, sdcTable: Methods for SDC (statistical disclosure control) in tabular data, URL http://cran.r-project.org/package= sdcTable, R package version 0.22.8.
25. Ten Bosch, O. and M. van der Loo, 2017, Awesome official statistics software. A curated list of official statistics software that is free (libre), downloadable, open source and used at statistical offices. http://www.awesomeofficialstatistics.org
34. Van der Loo, M and E de Jonge, 2017b, validate: Data Validation Infrastructure. R package version 0.2.0. https://CRAN.R-project.org/package=validate
Automation of Publications in Official Statistics using R

Guido Schulz (guido.schulz@destatis.de)
Federal Statistical Office of Germany (Destatis)

ABSTRACT
A key task of official statistical authorities is to collect and disseminate indicators periodically. Automation using a wide range of R packages bears massive potential to cut down the resources necessary for the creation of publications. Furthermore, automation in R has the potential to improve transparency, punctuality and coherence of statistical products. The dynamic reporting engine knitr in particular allows for an efficient combination of R’s functionalities of data retrieval, data manipulation and customizable plotting on the one hand, and the layout and typesetting flexibility of LaTeX or other markup languages on the other. This allows official statistical authorities to produce either ready-to-print PDFs or interactive websites while adhering to their corporate design requirements. Furthermore, dynamic reporting makes it possible to update periodic publications automatically. A work in progress example of automated statistical country profiles - a product the German Federal Statistical Office regularly publishes based on a wide range of official international sources - will be presented to illustrate both advantages and challenges in the practical use of dynamic reporting using R and knitr in particular.

Keywords: R, Automation, Dynamic Reporting, Official Statistics, knitr
JEL Classification: Y10, Y50

1. INTRODUCTION
A key task of official statistical authorities is to collect and disseminate indicators periodically. Besides the publication of statistics in public databases, one of the most common channels for dissemination is the publication of printed or online thematic reports. Notwithstanding all the differences between official statistical authorities and their respective publication requirements, it is fair to claim that all face the common challenge of ensuring the workflow of publication production functions in a reliable and resource efficient way. Traditionally, publication production workflows depend on specific colleagues scattered over a range of departments, different software (licenses) need to be employed at different stages of production and the appearance of publications has to be manually adjusted to specific corporate design requirements. Such complex workflows demand not only a high level of coordination and planning, but they are highly labour intensive. Furthermore, bottlenecks emerge during the process quite often, putting timeliness or even the quality of the product at
stake. Many publication production workflows include individual automated operations, but the complexity of workflows that mix different software - and GUI-only software in particular - quickly limits the degree of automation potentially achievable. Thus, if the goal is to make periodical production of statistical reports less labour intensive, less error-prone, more reliable and more transparent, a workflow must be established that allows for the highest possible degree of automation. While streamlining tasks and workflows should naturally be the goal of any institution, EU national statistical institutes are obliged to “systematically and regularly identify strengths and weaknesses to continuously improve process and product quality” (Eurostat, 2011), as it is set out in the principles of the European Statistics Code of Practice.

With this paper, I seek to contribute to this objective by proposing a highly customizable workflow for report production in R that addresses all of the aforementioned problems. The dynamic reporting engine knitr (Xie, 2014; 2015; 2017c) will be at the core of the proposed workflow. The paper is divided into four parts. First, I am going to argue for the introduction of knitr, explain its general purpose and main functionalities that make it fundamental for automation in R. With that knowledge, I can then propose a generic, and therefore highly flexible architecture of a knitr based workflow that incorporates a wide range of other R packages. The generic workflow will consequently be translated into a concrete real-life example, where practical strengths and difficulties become apparent. Finally, I will conclude with a short summary and name remaining challenges for a full implementation of the proposed workflow.

2. KNITR AND ITS POTENTIAL FOR AUTOMATING PUBLICATIONS

Before embarking on an appraisal of knitr’s potential for automating the production of publications, I want to briefly explain what knitr is. Knitr is an R package written and maintained by Yihui Xie, that provides a “general-purpose tool for dynamic report generation in R using literate programming techniques” (Xie 2017c). The paradigm of literate programming as formulated by (Knuth, 1984) asks programmers to “mix the source code and documentation [or narrative] together, [so that] we can either extract the source code out […] or execute the code to get the compiled results” (Xie, 2015, p.1). In our case of interest, knitr serves as a software package that compiles softcoded or dynamic source code into numeric, literal or graphical output, creating a publication ready report. Knitr’s default engine for the input source code is R, but amazingly it also accepts other programming
languages including Python, Awk, Ruby, Bash, Perl, SAS, Scala, CoffeeScript and others. In this paper, I want to stick to the simple case of using \texttt{knitr}'s default engine R. Outputted reports can take on a variety of formats - DOCX, HTML and PDF being the most important ones. Thus, \texttt{knitr} is not just a convenient dynamic reporting engine, but given the wide range of accepted input programming languages and output formats, it is also highly flexible.

Introducing \texttt{knitr} to the workflow of statistical report production and combining it with other R packages bears massive potential on various levels. Potential improvements in both production process and product quality include:

- Automated data retrieval, data processing and data presentation/visualisation together significantly reduce the labour input required.
- Replacing arduous manual hardcoding with softcoding minimizes errors.
- Report production scripts that are run automatically on specific time points foster adherence to dissemination time schedules.
- Automation enhances coherence and comparability in both design and content.
- Providing reproducible code (cf. Sandve et al., 2013) that covers the entire workflow vastly improves transparency – both inhouse and beyond.
- Simplifying report production workflows to just one or two programming languages boils down the necessary skills for practitioners.
- \texttt{Knitr} based workflows can reduce or even eliminate software license expenditures all together.

This broad list of advantages should provide enough motivation to consider incorporating \texttt{knitr} into statistical report production workflows. Now, the task is to show what exactly a technical implementation should entail for these advantages to unfold.

3. GENERIC ARCHITECTURE OF A KNITR BASED WORKFLOW

In this section I want to propose a generic architecture of a \texttt{knitr} based workflow to automate statistical report production. Alongside \texttt{knitr}, a number of other R packages will be recommended for integration into the workflow. Naturally, they may be replaced or supplemented according to the reader’s specific needs and publication requirements.
Figure 1 depicts the scheme of a generic workflow architecture. The workflow consists of a sequence of three main operations: data retrieval, data wrangling and report generation. Each step of the flow will now be explained consecutively.

3.1. Data retrieval

The raw tabular data you want to process for your report may be stored in remote or local databases or individual files. In case you want to connect to remote databases run by institutions offering official statistics, the chances are the R community will provide a handy package to do the job for you. Packages such as `wbstats` (Piburn, 2016), `eurostat` (Lahti et al., 2017), `OECD` (Persson, 2016) allow for - as the names imply - easy access to the vast databases of these intergovernmental organizations. A few national statistical institutes, such as the Dutch CBS, offer their own packages (e.g. `cbsodataR` (De Jonge, 2016)) to allow end-users to connect to their public database. For users wanting to connect to multiple data providers using the same syntax, the `rsdmx` (Blondel, 2017) package is an attractive alternative. It enables a connection to data providers that offer an embedded web-service interface to SDMX, including many international and national institutions. Connecting to local databases by contrast requires procedures from another
set of packages (RODBC (Ripley and Lapsley, 2017), RMySQL (Ooms et al., 2017), dbplyr (Wickham, 2017), etc.) from which you should pick according to the nature of your local database-management system. Importing remotely or locally stored individual rectangular data files (.csv, .tsv, .xlsx, etc.) can be conveniently done with readr. For reading remote or local data from foreign statistical packages (SPSS, Stata, SAS) the haven package provides the best functionalities.

3.2. Data wrangling

Once the data has been successfully retrieved, the next big task is to tidy, process and manipulate the data to make it fit for literal or visual representation within the report. Currently, the most powerful yet user friendly packages for data wrangling arguably are tidyr (Wickham and Henry, 2017), dplyr (Wickham, 2017) and data.table (Dowle and Srinivasan, 2017).

3.3. Report generation

Finally, in this last step knitr enters the workflow and the above promised advantages can take effect. In case you wish to produce a specifically designed report in PDF format, the source document containing a mixture of code and narrative should be an R Sweave (Rnw) file. Rnw files only accept LaTex as an input language for the narrative.

The flexibility of LaTex and its packages should allow you to tune the page layout and general typography according to your specific needs. In order to ensure coherence with the design - i.e. colours, spacings, fonts - between plots and the rest of your report, you should take advantage of the flexibility of ggplo2 (Wickham and Chang 2016, 2) theming, its extensions (ggforce (Pedersen, 2016), ggtheme (Arnold, 2017), etc.) and the packages that allow for general plot customization (gridExtra (Auguie, 2017), gtable (Wickham, 2016), extrafont (Chang, 2014), etc.). For table customization I recommend to capitalise on one of LaTex’s powerful default features for table construction and extending them with LaTex packages such as tabu (Chervet, 2010), booktabs (Els and Fear, 2016), multirow (Van Oostrum and Leichter, 2016). You may consider sourcing out all the design customization code by creating your own private (corporate) design packages in either R, LaTex or both. This not only makes your Rnw document more compact and easier to read, it also allows you to extend potential design adjustments to all of your publications effortlessly by changing only the code in your design package.

Once all the design work has been done, you should embed the R scripts that perform the previously explained data retrieval and data wrangling
steps. With the clean and processed data loaded into R, you can finally fill the report with content by creating dynamic texts, plots (ggplot2) and tables (xtable (Dahl, 2016), kableExtra (Zhu, 2017)). Interestingly, dynamic reporting seems to rely on this – compared to conventional, manual report production - reversed order of layout/design tasks and content creation.

As a final step, knitr compiles the Rnw and produces - if everything goes well - a publication-ready PDF. Since the report source document contains dynamic code performing data retrieval, data wrangling and content creation, it is possible to update the report simply by letting knitr compile the document source code anew. Even this task of updating the reports can be automated, which is particularly convenient for periodical publications. The R packages taskscheduleR (Wijffels, 2017b) for Windows and cronR (Wijffels, 2017a) for unix-alike systems allow users to schedule the compilation of an updated report at predefined points in time.

Since the report source document contains the entire workflow in code, the project becomes not only comprehensively documented, but (potentially) fully reproducible, too. Publishing the report source document e.g. on GitHub alongside the report, enhances transparency and - assuming report source document actually is fully reproducible - encourages interested readers to use the data themselves.

Looking at the generic workflow (Figure 1), we realize that we have worked our way through almost the entire scheme. The only part that remains to be explained is the case of an R markdown (Rmd) report source document, which compiles to a report in HTML. The general proceeding remains the same as for the discussed R Sweave (Rnw) source document, only the languages used for narrative and design, as well as the recommended R packages for plot and table creation change. The narrative is written in markdown or plain HTML, while design and general appearance need to be specified in CSS and HTML. In contrast to PDF reports, HTML reports allow for elements to be interactive, such as zoomable graphics, sortable tables or foldable contents. Interactivity – when applied sensibly – enhances the user experience and offers many advantages over static representations (Kirk, 2016, pp.223). This is why, for HTML reports, I recommend to using interactive report templates (e.g. rmdformats (Barnier, 2017)), interactive plots (e.g. plotly (Sievert et al., 2017), highcharter (Kunst, 2017), ggiraph (Gohel, 2017)) and

---

1. For a technical description of how knitr compiles the report source document, see Xie (2015).
2. To be precise, it is not knitr, but pandoc that carries out the very last compilation to HTML. Pandoc is a universal document converter that enables knitr to create such a wide variety of output file types from Rmd source documents (MacFarlane, 2016).
interactive tables (e.g. DT (Xie, 2016)). Most of these R packages are simple yet incredibly convenient wrappers for their respective JavaScript libraries. In addition, htmlwidgets (Vaidyanathan et al., 2017) not only helps with the embedding of widgets into R Markdown documents, it also allows users to develop their own widgets to seamlessly bridge R and JavaScript. Generally speaking, knitr is capable of combining R and JavaScript applications such as D3.js, since it is possible to “write anything to the output, including JavaScript” (Xie, 2017b).

Arguably, having to code separate report source documents for PDF and HTML reports is not very convenient. Despite the general possibility to use the very same R Markdown source document to compile both report formats, this approach will inevitably reduce customizability and coherence of design and layout.

4. PRACTICAL EXAMPLE

4.1. Extending and renewing statistical country profiles

As part of its product portfolio on international statistics, the Federal Statistical Office of Germany publishes so called statistical country profiles for the members of the G20 “based on official international statistics as published by UN organisations, World Bank, OECD, IMF, etc” (Federal Statistical Office of Germany, 2017). The country profiles published in English and German cover a wide range of topics including economy, finance, demography, health, environment and others. The data is gathered from more than 15 different sources. Until now, seven page PDF format country profiles were produced in a semi-automated, yet not fully-automatable workflow, which is only partially recordable. Updates require a considerable amount of manual work, particularly when it comes to updating plots. Four expensive proprietary software packages are used along the product in process and data has to flow manually through different graphical user interfaces.

At the Federal Statistical Office in Germany it has long been our plan to extend and renew our country profile products. Our goal is to

• broaden the set of country profiles beyond the G20, hopefully covering all 193 members states of the United Nations,
• offer a HTML web based interactive version of the country profile in addition to the static PDF, and
• provide updated versions of all profiles as soon as new data gets published.

It stands to reason that the desired goal cannot be achieved without a considerable increase in the level of automation. Evidently, adhering to
the old workflow would require absurd amounts of manual labour. Hence, we took the extension and renewal of the country profiles as a first trial to practically implement an automated \texttt{knitr} workflow. Work on this project is ongoing and still requires some tweaking, yet I want to briefly explain how I concretized the generic workflow architecture proposed above and what challenges occurred in practice.

4.2. Concrete workflow and challenges faced

To produce a PDF and HTML country profile in each English and German, four separate source documents were created. A master script calls \texttt{knitr} to compile all of these four source documents resulting in the four desired versions of the country profile. This master script itself was put in a loop that iterates through all 193 member states of the United Nations. In order to keep the source documents as simple and short as possible, R codes for data retrieval, plot and table designs and plot generation were separated out into extra R files.

All data necessary for the country profiles could conveniently be retrieved with RODBC from a regularly updated in-house local MySQL database. The already relatively clean data was processed with \texttt{dplyr}. As it was proposed above for PDF reports, plots were produced with \texttt{ggplot2} plus extending packages and tables were set using \texttt{xtable}. The HTML reports were generated on the basis an \texttt{rmdformats} template. Since project progress of the HTML version lags behind, a lot of CSS tweaking to enhance the HTML report design still needs to be done. However, interactive plots and tables could already be produced with \texttt{highcharter} and \texttt{DT}. Unfortunately, scheduled automatic report compilation has not been implemented yet. Figure 2 and Figure 3 show, for the example of France, clippings of the German language PDF and HTML prototype versions of the new country profile.\footnote{The new country profiles have not been officially published yet. Prototype versions including Rnw and Rmd report source documents are available upon request from the author (guido.schulz@destatis.de).}
Clipping of the PDF country profile

Figure 2
Unsurprisingly, many practical – mostly design-related - obstacles have paved the project so far. One central challenge was to softcode plot or table specifications flexibly enough, to cope with values of starkly differing size while adhering to corporate design requirements. The problem becomes palpable when thinking about the varying magnitudes of economic indicators for say, China and Lesotho. Satisfying all corporate design requirements, such as specific spacing in plots, guaranteeing font size coherence between plots and text or embedding a UTF8 encoded True Type font in plots required seemingly endless fiddling. The usual LaTex specific package option clashes (e.g. \texttt{xcolor} (Kern, 2016) and \texttt{pgf} (Tantau and Feuersänger, 2015)) came on top. Luckily, for all PDF report related problems more or less complicated solutions could be found, not least with the help of Stack Overflow. In contrast, a number of obstacles still need to be overcome for the HTML report. Unfortunately, to this day I could not find any decent practical example or tutorial online of how a \texttt{knitr} compiled HTML report may be properly embedded into a customized website. The recently released \texttt{blogdown} (Hill, Xie, and Amber, 2017; Xie, 2017a) package which particularly serves for website creation with R Markdown promises big advances on this front.
5. SUMMARY AND CHALLENGES AHEAD

This paper sought to show that it is both desirable and feasible to streamline the workflow of statistical publication production using the dynamic reporting engine \texttt{knitr}, potentially achieving full automation. \texttt{R} and \texttt{knitr} in particular promise labour savings, improved output quality, enhanced transparency and potentially cost reductions. A generic, thus highly customizable, architecture of a \texttt{knitr} based workflow was proposed to illustrate what a technical implementation of an automated workflow could look like. The three central operations of the workflow, namely data retrieval, data wrangling and report generation were described in detail, enabling the reader to adopt and reproduce each step according to the reader’s specific needs. Finally, I presented the concrete workflow of an on-going project of publication automation - the remaking and extension of so called statistical country profiles, a product the Federal Statistical Office of Germany regularly publishes based on a wide range of official international sources. By means of this concrete example, I pointed out a variety of practical pitfalls that commonly occur in implementation.

Admittedly, there are preconditions and limits to the workflow proposed in this paper, and they should not go unmentioned. First of all, not every variation of the generic workflow was tested, hence I might be unaware of certain problems that occur. I also acknowledge the fact that the CRAN universe is so vast and evolves so rapidly that other \texttt{R} packages might suit the reader’s needs better than ones suggested above. As with any process automation, the implementation of the \texttt{knitr} based workflow certainly depends on preliminary work. Just to name a few: all software (dependencies) need to be set up properly for all people involved in the project, data stored in databases need to be kept tidy enough to be processed automatically, a corporate design package potentially needs to be written beforehand or web content management systems need to be configured to correctly integrate the produced reports.

Though the most important preliminary task to be tackled is arguably also the biggest challenge – convince your superiors and colleagues to establish and learn a new workflow for publication production. Probably, your organisation’s actual workflow is based on a well calibrated ecosystem of (proprietary) software and a well-coordinated personal distribution of tasks and skills. Yet, I hope the impressive list of advantages of a \texttt{knitr} based workflow may serve as a good argument in favour of change. Starting to implement a new workflow for just one specific publication that also does not involve too many people and using the project to exemplify \texttt{knitr}'s
advantages might be a promising way to embark on a wider transition towards R and \texttt{knitr}. Even partial implementations of the proposed workflow may already bring significant benefits. And it is worth recalling that there are good examples to follow. Recently, the UK Government and its Office for National Statistics (ONS) started to migrate from the proprietary SAS software to R (Sellors, 2017a; 2017b) and embarked on a project to speed up the production of official statistics for publications using \texttt{knitr} (Government Digital Service, 2017). Statistics Netherlands (CBS) has already managed to largely incorporate R into their workflows, though admittedly their dependence on proprietary statistical software before the introduction of R was comparatively small (Van Der Loo, 2012).

Despite all the challenges and practical pitfalls of a \texttt{knitr} based publication production workflow, I hope this paper inspired and motivated the reader for a transition. In any case, the proposed workflow is meant a starting point for a discussion on how modern statistical report production can and should look like – an invitation to have an exchange or even cooperation on how R and \texttt{knitr} may be promoted in the distribution of official statistics.

REFERENCES

24. Persson, E., 2016, OECD: Search and Extract Data from the OECD (version 0.2.2). https://cran.r-project.org/package=OECD.


43. ***, 2016, *DT: A Wrapper of the JavaScript Library ‘DataTables’* (version 0.2). https://cran.r-project.org/package=DT.


Small Area Estimation Methodology (SAE) applied on Bogota Multipurpose Survey (EMB)

José F. Zea (josezea@usantotomas.edu.co)
Santo Tomás University, Bogota, Colombia

Felipe Ortiz (andresortiz@usantotomas.edu.co)
Santo Tomás University, Bogota, Colombia

ABSTRACT

Small Area Estimation Methodology (SAE) is widely used by statistical offices in several countries to reduce sampling errors with the help of auxiliary information. Different countries such as USA, Canada, England, Israel and European Community have within their statistical institutes offices dedicated to the application of SAE in several investigations. So far, the National Administrative Department of Statistics of Colombia (DANE), has not published official statistics that involve this methodology. The present work illustrates the advantages in the use and estimation of living conditions using SAE. Formally, the unemployment rate and the average income levels of municipalities of Cundinamarca are estimated. For this purpose, information of the Multipurpose Survey 2014 is used and is complemented with socio-demographic and economic related auxiliary information. A mixed Fay & Herriot (1979) model is used in order to get the estimates.

We use R ecosystem to develop SAE methodology. R is used for data wrangling, model adjustment, parameter estimation and finally visualization with the aid of renowned packages such as tidyr, forcats, sae, ggplot2 among others.

We will show R implementation and some remarkable results. First, a good adjustment of the model to the data; second, a reduction in the sampling errors reported by the estimation in small areas compared to the direct estimates generated by the Bogota Multipurpose Survey (EMB); and finally acceptable estimates for municipalities that were not covered by the survey.

Keywords: Small Area Estimation, Survey Sampling, Tidyvese, Household Survey, Colombia, Cundinamarca Municipalities.

JEL classification: O54, C63, C81, C83, C88
INTRODUCTION

Obtaining reliable estimates for municipalities or smaller geographical areas has been an impossible task for household surveys with traditional sampling methodologies due to financial sources and operative costs. In the present work we seek to obtain precise estimates at municipality level of average income and unemployment rate for Cundinamarca to achieve more effective public policies.

Cundinamarca is one of 32 departments of Colombia with 116 municipalities including Colombian capital Bogota. In this study we make estimations of average household income and unemployment rates in 115 municipalities (excluding Bogota due to administrative issues and for obtaining a better adjustment in estimations).

Multipurpose Survey covers 32 municipalities, and Bogota which will not be taken into account in this study because it is an outlier municipality in terms of average income and has a very different economic dynamic from the other municipalities in Cundinamarca; due to its condition, as a capital of Colombia and the biggest city in terms of population, economic development, etc. More details of the survey are found in (DANE 2014).

In this study we have two main goals:

- To optimize the production of official statistics by improving the estimates of the average income and the unemployment rate to observe Cundinamarca municipalities in the 2014 Multipurpose Survey.
- To provide estimations of average income and unemployment rate for non-covered municipalities in the 2014 Multipurpose Survey.

In this work we use R ecosystem in different stages such as:

- Data wrangling of auxiliary information to municipality level and multipurpose sampling dataset (EMB).
- Direct estimates of average household income by municipality, unemployment rate and their respective variances by municipality.
- Estimates of average household income by municipality and unemployment rate by municipality using SAE in particular Fay-Herriot model.
- Visualization.

R code for reproducibility can be obtained from https://github.com/josezea/sae_emb
LITERATURE REVIEW

There are two main approaches in survey sampling theory: design-based survey sampling (Sarndal, Swensson, & Wretman 1992) and model-based survey sampling (Rao & Molina 2015).

1.1 Direct estimators for domains

In design-based approach the values of the interest variable $y$ are considered as fixed and the sample is considered as a random variable. In this approach each sample has a probability $p(s)$ to be selected. In general, $p(s)$ is not straightforward to compute. In order to estimate parameters such as total, mean, ratio and proportions is enough to have inclusion probabilities.

An unbiased direct estimator (under the survey design $p(s)$) for the population total is the Horvitz-Thompson (HT) estimator:

$$\hat{t}_{y\pi} = \frac{1}{n} \sum_{k=1}^{n} \frac{y_k}{\pi_k} = \sum_{k=1}^{n} w_k y_k$$

HT estimator is widely used to produce official statistics. HT estimator can be written in terms of expansion factor or survey weight, $w_k$, which is the inverse of inclusion probabilities $1/\pi_k$.

A direct estimator (not unbiased) for the mean is:

$$\frac{\Delta}{\bar{Y}} = \frac{\sum_{k=1}^{n} w_k y_k}{\sum_{k=1}^{n} w_k}$$

The above estimator is a particular case of a ratio estimator:

$$\hat{R} = \frac{\hat{t}_{y\pi}}{\hat{t}_{\pi\pi}}$$

where $\hat{t}_{y\pi} = \sum_{k=1}^{n} w_k y_k$ and $\hat{t}_{\pi\pi} = \sum_{k=1}^{n} w_k z_k y_k$ and $z_k$ are variables of interest observed in the survey.

An example of a ratio is the unemployment rate, which is calculated with the previous expression. The numerator is the number of unemployed people in a reference period $\hat{t}_{y\pi}$, the denominator $\hat{t}_{\pi\pi}$ is the number of economically active people in the same period. Both the numerator and denominator are computed.

An example of a mean (computed as a ratio) is mean income. Numerator and denominator of a mean are random variables and a non-linear function of total. Therefore, the mean estimator is biased; the variance of mean is increased due to randomness of the numerator and denominator.
In many occasions the interest is the estimation of parameters for subpopulations such as age groups, municipalities, combination between age groups and sex categories, etc. Those groups are called domains. Estimation of parameters in domains can be done through direct estimators and SAE methodology. The direct estimation of total for domain \(d\) is get as follows:

\[
\hat{t}_{y_{\pi,d}} = \sum_{k} \frac{\gamma_{k} z_{dk}}{\pi_{k}} = \sum_{k} w_{k} \gamma_{k} z_{dk},
\]

where \(z_{dk} = 1\), if element \(k\) is in the domain \(d\) and \(z_{dk} = 0\), in other cases.

An estimator for the mean can be computed as:

\[
\hat{Y}_{d} = \frac{\hat{t}_{y_{\pi,d}}}{\hat{N}_{d}} = \frac{\sum_{k} \gamma_{dk}}{\hat{N}_{d}} = \frac{\sum_{k} w_{k} \gamma_{dk}}{\sum_{k} w_{k} z_{dk}},
\]

where \(\gamma_{dk} = y_{k} z_{dk}\).

**FAY-HERRIOT ESTIMATOR**

Small area estimation (SAE) refers to estimation in domains (population subgroups) which have a relatively small sample size. Some examples of a small area such as counties, city administrative divisions (e.g. localities in Bogota). In this research municipalities of Cundinamarca are considered small area.

SAE is used to carry on the estimation of parameters for domains (small area) with a mixed approach between design-based and model-based estimation. SAE is based in the adjustment of mixed models which take into account within-domain variance. Auxiliary information for small area is added to decrease survey sampling error and therefore an increase in quality of the estimations. There are two main types of SAE estimators: individual-level models (Battese, Harter & Fuller 1988) and area-level models (Fay & Herriot 1979) (Rao & Molina 2015).

Fay-Herriot (FH) model links estimated average of interested variable in the area \(d\) (for \(d = 1, \ldots, D\)) with auxiliary information vector \(z_{d}\):

\[
\hat{Y}_{d} = z_{d}^{T} \beta + u_{d} + e_{d}
\]

where \(u_{d} \sim N(0, \sigma^{2}_{u}), e_{d} \sim N(0, \Sigma_{e})\), with \(\Sigma_{e} = \text{diag}(D_{1}, D_{2}, \ldots, D_{D})\). In the above equation \(z_{d}^{T} \beta + u_{d}\) is the unknown average for \(d\)th area and \(D_{d}\) is a known term, usually taken as variance of \(\hat{Y}_{d}\) (under the survey design \(p(s)\)).
The best linear unbiased predictor (BLUP) for $\bar{\theta}_d$, with $\hat{\beta}$, $\sigma_v^2$ and $D_d$ known is obtained as follows:

$$\hat{\bar{\theta}}^{BLUP}_d = \begin{cases} \hat{z}^T_d \hat{\beta} + \gamma_d \left( \hat{\bar{\theta}}_d - \hat{z}^T_d \hat{\beta} \right) & \text{if } d \in A \\ \hat{\bar{\theta}}^*_d & \text{if } d \notin A \end{cases}$$

where $\gamma_d = \frac{\sigma^2_v}{\sigma^2_v + D_d}$ and $A$ denotes the selected areas in the sample.

When $\beta$ and $\sigma^2_v$ are estimated we obtained the empirical best linear unbiased predictor (EBLUP). The computation of EBLUP is carried on as:

$$\hat{\bar{\theta}}^{FH}_d = \begin{cases} \hat{z}^T_d \hat{\beta} + \gamma_d \left( \hat{\bar{\theta}}_d - \hat{z}^T_d \hat{\beta} \right) & \text{if } d \in A \\ \hat{\bar{\theta}}^*_d & \text{if } d \notin A \end{cases}$$

EBLUP can be seen as a weighted average of direct estimation $\hat{\bar{\theta}}_d$ and indirect estimation $\hat{z}^T_d \hat{\beta}$. If $\hat{\bar{\theta}}_d$ is closed to 1, $\hat{\bar{\theta}}^{FH}_d$ is similar to $\hat{\bar{\theta}}_d$, on the other hand if $\hat{\bar{\theta}}_d$ is closed to 0, the estimator $\hat{\bar{\theta}}^{FH}_d$ tends to $\hat{z}^T_d \hat{\beta}$.

**Mean square error for Fay-Herriot estimator**

The Mean Square Error (MSE) provides an approximation of mean square error of Fay-Herriot estimator which depends on estimation method of $\beta$ and $\sigma^2_v$ (Prasad & Rao 1990). With the moments method (method developed by the same authors) and restricted maximum likelihood - REML the mean square error of estimations is obtained as:

$$MSE(\hat{\bar{\theta}}^{FH}_d) = \begin{cases} g_{1d}(\hat{\beta}^2_u) + g_{2d}(\hat{\sigma}^2_v) + 2g_{3d}(\hat{\beta}^2_u) & \text{if } d \in A \\ z^T_d (Z^T V^{-1} Z)^{-1} z^*_d + \hat{\sigma}^2_u & \text{if } d \notin A \end{cases}$$

where

$$g_{1d}(\hat{\beta}^2_u) = \frac{\hat{\beta}^2_u D_d}{\hat{\beta}^2_u + D_d}, \quad g_{2d}(\hat{\sigma}^2_v) = \frac{D_d^2}{(\hat{\beta}^2_u + D_d)^2} z^T_d (Z^T V^{-1} Z)^{-1} z_d$$

and

$$g_{3d}(\hat{\beta}^2_u) = \frac{2D_d^2}{\hat{\beta}^2_u + D_d} \left[ \hat{\beta}^2_u + 2 \hat{\beta}^2_u \sum_{i=1}^{D_d} D_i D + \sum_{i=1}^{D_d} D^2_i \right]$$

with $V = \text{diag}(\sigma^2_1, \ldots, \sigma^2_n)$.
Basic concepts of Bogota Multipurpose Survey (EMB)

The concepts related to labor force and income that we will use in this section are based on the definitions used by the Colombian Official Statistics Agency (National Administrative National Department of Statistics - DANE) and may differ slightly from the concepts used in the International Labour Organization - ILO).

In EMB survey analyzed population corresponds to houses, households and people. In this study, all economically active household members were considered, household members were grouped according to the occupational situation:

a. Working Population: consist of people who are working, performing some paid activity or who have a job or business for which they receive income. This group also includes people who worked the previous week without compensation.
   - Employees: worker, private company employee, public employee, domestic employee and hired worker.
   - Independent worker: people who performed in the previous week any activity pays for one hour or more, or perform at work in the following activities: private enterprise worker/employee, government employee, domestic employee and day laborer.
   - Unpaid Worker: people who performed the previous week some activity pays for one hour or more or perform at work in the following activities: unpaid assistant, unpaid family worker and unpaid worker.

b. Unemployed: it consists of people who are not working and not performing any paid activity or who have a job or business for which they receive income. This group excludes people who worked the previous week without compensation.

Household members can get revenues from rental income, financial aid, sales, pension, etc.

Employees get earnings mainly from salary, bonus, and payments in food, transportations, subsidies, indemnities and others. Independent workers get earnings related to any activity.

In order to compute unemployment rate it is necessary to identify employed and economically active people. In order to achieve that, DANE provides the next definitions:

- Population of working age (PWA): this group consists of people aged 12 and over in urban areas and over 10 years in rural areas.
- Economically Active Population (EAP): also called labor force and
consists of people who are in working age and are working, or are looking for a job.

- Economically Inactive Population (EIP): consists of people who are in working age who neither work nor look for a job.
- Employed (E): consists of economically active population who in the reference period were in one of the following situations: worked for at least one hour paid in cash in the reference week. Those who did not work the reference week, but had a job. Unpaid family workers who worked in the reference week for at least 1 hour
- Unemployed (U): consists of economically active population who in the reference period is not in any situation described for employed people.

DANE methodology for compute unemployed rate is estimated by the ratio the ratio \( \frac{\text{Total unemployed}}{\text{Total RAD}} \).

**METHODOLOGY**

**Data wrangling**

The datasets corresponding to occupational condition was used in order to compute income for every household of Cundinamarca municipalities and unemployment for the surveyed people living in Cundinamarca.

Data wrangling was carried out using R tidyverse packages. For data importation we used haven (Wickham & Miller 2017) for datasets in .sav format (IBM SPSS), readr (Wickham, Hester & Francois 2017) for text files and readxl (Wickham & Bryan, 2017) to read xls and xlsx files. dplyr (Wickham et al. 2017) was used in different processes such as filtering, recoding, merging (left, and inner joins) and binding different datasets and to make necessary aggregations. The script "1.DataWrangling_EMB.r" in the github repository contains all the details of the processing information.

Datasets from different sources was explored to use as auxiliary information. Tidyverse packages was used in order to conform a unified database of auxiliary variables, some of the auxiliary variables considered are:

- Rate of beneficiaries of the selection system for beneficiary social programs (SISBEN)
- Average of "Prueba Saber 11" (Standardized test similar to the American - SAT) score in the municipality.
- Area (Km\(^2\)).
- Affiliates health contributory and subsidiary regime.
- Average of cadastral appraisal in the municipality.
Direct estimations

Once the EMB and auxiliary information is depurated, direct estimations are carried on. The survey design of EMB is probabilistic, each element of the population (households) has a non-zero probability to be selected. In particular, it is a clustered and stratified design.

In order to improve the accuracy of the estimations, it is considered as strata the social stratification in Bogota (local government mechanism to assign subsidies in public services to the poorest households).

In this design, clusters correspond to a set of houses located within the same block, this group of houses is called a size measure segment. In each selected segment all houses are surveyed.

The average of income is estimated as follows:

\[ \hat{Y} = \frac{\sum_{i=1}^{n} w_k y_k}{\sum_{i=1}^{n} w_k} \]

where \( w_k \) is the survey weights (the inverse of inclusion probabilities \( \frac{1}{\pi_k} \)).

In R, We estimate the average using survey package (Lumley, 2017). The sampling design used in this exercise is complex (to increase cost efficiency), it involves stratification and cluster sampling. Unfortunately, population sizes, probabilities and weights associated to different survey sampling stages are not delivered by Colombian Statistical Office (DANE) to general public, instead they only provide the final expansion factor.

We use the fact that we can approximate inclusion probability \( \pi_k \) to \( mp_k \) (where \( p_k \) the selection probability associate to a with replacement
We estimate the average in every municipality with their respective standard errors and variance through survey package (Lumley, 2017) as follows:

```r
aprox_design <- svydesign(ids ~ 1, weights ~ SURVEY_WEIGHTS, 
data = HouseholdIncome )

df_est.income <- svyby(~INCOME, ~ID_MUNIC, aprox_design, svymean)
df_est.income$varmeansByMun <- df_est.income$se ^ 2

df_est.income$cvarmeansByMun <- 100 * cv(df_est.income)
```

The selection probability of different stages are not available in DANE open data portal, instead of this we use final sampling weight (SURVEY_WEIGHTS) which is available in EMB dataset delivered by DANE. We do not incorporate weights parameter in order to develop a with replacement survey design.

The `svydesign` function allows to define survey sampling design, `svyby` function is used to get aggregated estimations (by municipalities for example).

Estimations of means, estimated variance of mean estimator, and `cve` is done for 31 observed municipalities in Multipurpose Survey.

Other variable of interest is the unemployment rate which is computed as the ratio of total of unemployed people ($\ell_j$), and total economically active population ($\ell_j$). Those totals are computed over all surveyed households.

\[
\hat{R} = \frac{\sum_w w_k y_k}{\sum_w w_k z_k}
\]

where $w_k$ is the survey weights.

Ratio estimation is carried on for every municipality with their respective standard errors and variance through survey package (Lumley, 2017) as follows:

```r
aprox_design_Unemployment <- svydesign(ids ~ 1, 
  weights ~ SURVEY_WEIGHTS, data = UNEMPLOYMENT )

df_est.unemployment <- svyby(~Unem, by=ID_MUNIC, denominator=EAP, 
  design = aprox_design_Unemployment, svyratio)

names(df_est.unemployment)[c(1,2)] <- c("ID_MUNIC", "unempByMun")
df_est.unemployment$varunempByMun <- df_est.unemployment$'se.DS/PEA' ^ 2
df_est.unemployment$cve.unempByMun <- 100 * cv(df_est.unemployment)
```

---
In `svyby` we define numerator variable as total of Unemployed people, and denominator as Economically Active Population (EAP), `svyratio` indicates that we are estimating a ratio.

**Fay-Herriot estimations**

An unified dataset with direct estimation and auxiliary information for 31 municipalities is formed after joining respective dataframes:

```r
library(dplyr)
df <- left_join(df_EMB, AuxInfo, by = "ID_MUNIC")
```

For the estimated average of income a forward stepwise procedure is carried out to select auxiliary variables.

```r
print(formula(income_step_model))
## IncomeMeansByMun ~ CONSUMO_ENERGIA_PER_HABIT + PUNTAJE_SABER +
## AVALUOS_CATASTRALES_RURALES + NBI_2010
```

The selected variables for average income are:

- Energy use per capita in the municipality.
- Municipality unsatisfied basic needs.
- Average of “Prueba Saber 11”.
- Average of cadastral appraisal in the municipality.

A Fay-Herriot model for estimated average income is adjusted with the aid of `sae` library as follows:

```r
FH_income <- mseFH(IncomeMeansByMun ~ CONSUMO_ENERGIA_PER_HABIT +
                    PUNTAJE_SABER + AVALUOS_CATASTRALES_RURALES + NBI_2010),
               vardir = VarIncomeMeansByMun, data = df_income_model)
```

\[ \hat{Y}_U \] is adjusted by energy consumption (CONSUMO_ENERGIA_PER_HABIT), score standardized national education test (PUNTAJE_SABER) and cadastral appraisals(AVALUOS_CATASTRALES_RURALES), the directed variance (VarIncomeMeansByMun) of every area is estimated by survey design using the variance estimation methodology previously described.
RESULTS

The covered municipalities (Covered EMB) in EMB survey are presented in figure 1:

Covered and non-covered municipalities in EMB 2014

![Figure 1](image)

One of the main aspects of the Fay-Herriot estimator is that it improves the direct estimator in terms of accuracy. In figure 2 it can be noted that the mean square error of the Fay-Herriot estimator for average income is smaller than the mean square error of the direct estimator.

The same occurs for the MSE of unemployment rates estimator, the Fay-Herriot estimator has a lower MSE than the direct estimator.
As mentioned earlier, one of the most attractive aspects of small area estimation methodology is obtaining estimates for domains where no surveys were conducted (see equation [1]). Observed domains (d∈A) are predicted as a linear combination of direct estimation and the model. In unobserved domain (d∉A) the domain estimates are obtained only with model predictions.

In this case, predictions were made for 85 municipalities where no surveys were carried out. A map of average income (income in thousand Colombian pesos) by municipality is presented in figure 3.
In figure 4 average income and unemployment are shown for both 15 top and bottom municipalities of Cundinamarca in terms of income (millions of Colombian pesos) and unemployment rate. Predictions for observed and not observed municipalities in EMB are computed using FH estimator (see equation [1]).
It is interesting to note that there is a very strong linear relationship between estimates for average income and unemployment rate generated by Fay-Herriot model as it is observed in figure 5.
CONCLUSIONS

In this article we developed SAE methodology in order to minimize sampling error of direct estimations. In addition, we obtained plausible predictions for municipalities not covered by the EMB 2014.

The R software was a very useful tool to carry out the information processing (with tidyverse R packages), calculations of the point estimates with the direct estimator and the Fay-Herriot estimator and their respective variances (sae package), as well as the visualization and diffusion of the results (document, presentation).
REFERENCES

ABSTRACT

A common approach in examining data collected based on different scales is to look at their structure by means of factor analysis. This article provides a way to look not only into the overall mindfulness score as an individual characteristic, but also at how the mindfulness dimensions cluster together to provide potentially consistent individual profiles. The novelty of our contribution is two fold: we reached our goal with the help of cluster analysis, and not by the means of previous methodological approaches. Also, we applied the most popular tools used to measure mindfulness, the face facets mindfulness questionnaire, on a sample of Romanian participants which makes this research the first study on mindfulness conducted on a Romanian sample. We found that, despite the existence of some stable groups that share similar characteristics, the degree of homogeneity across individuals is pretty high. In addition, the levels of mindfulness corresponding to our participants seems to be unrelated with background variables like age, gender, and working place.

Keywords: Mixed methods, Health care, Survey research, Regression analysis

JEL Classification: C19, C38

1. INTRODUCTION

The realm of methodological tools of investigation for social sciences has largely extended in the last decades, both in terms of research designs and coverage. Beyond its extraordinary potential, the complexity and diversity of available data exposes to new challenges when choosing the most appropriate methods of analysis. With respect to scale measurement and development, the confirmatory factor analysis framework has become extremely popular in serving the objective of testing the construct validity of item sets (Gerbing & Anderson, 1988; Reise, Waller and Comrey, 2000). In the area of application
that motivates our study, mindfulness research, especially in regard to national measurement, reliability and factor structure seem to be to most targeted objectives. On the one hand, this is a positive fact due to the need of applying objective filters for the numerous questionnaires aiming to capture the essence of mindfulness (Baer, 2011). On the other hand, the risk of falling under the principles of a very standardized path can materialize in a certain level of ignorance for other properties of the data.

The current paper was originally conceived in order to build up a strong argument for putting mindfulness on the national public research agenda in Romania. This would be a natural development since the practice of mindfulness has been gradually accepted in the Western psychology (Shapiro, 2009) and medicine (Sauer et al., 2011). There is generous stream of studies assessing the clinical benefits of mindfulness (Keng, Smoski and Robins, 2011; Davis and Hayes, 2011; Baer, 2003) or in attention enhancement, present-moment awareness, and its stress reduction effects (Purser & Milillo 2015), thus creating strong premises for considering it an important tool in the contemporary mental healthcare portfolio. Moreover, the positive impact of mindfulness can be noticed also in organizational settings, in relation with decision-making (Fiol and O’Connor, 2003) and wise actions (Weick and Putnam 2006), leadership development (Goldman-Schuyler et al., 2017) and subjective wellbeing (Brown et al., 2009).

Objectives like measuring and capitalizing on the benefits of an increased mindfulness level for different populations would be much easier promoted if a national study would assess the cross-cultural validity of one of the existing questionnaires. Since our pilot study is just a preliminary step for conducting the testing on a representative sample, we took the opportunity to resort to a more unconventional statistical approach: an exploratory cluster analysis. The operational advantage is that we avoid the limitations of the minimum sample size recommended for factor analysis (Mundfrom, Shaw and Ke, 2005). Free from that constraint, the study makes two unique contributions. First, we examine the structure of the mindfulness score captured with the Five Facets Mindfulness Questionnaire (FFMQ), applied for a first time to a Romanian sample. By applying the hierarchical clustering algorithm and two linkage methods, Average and Ward, we determined a relatively high level of stability, thus homogeneity between the four clusters obtained, hinting to the fact that subjects posses similar characteristics corresponding to the five dimensions of mindfulness. Second, we consider the detected clusters and three background variables - gender, age and workplace – as potential predictors for the FFMQ score. The statistical significance of the groups and the lack of significance of all the other covariates further strengthen the
existing differences on mindfulness scores between groups, and the rather irrelevance of the background variables in explaining the level of mindfulness in our sample.

The rest of the article is structured as follows: section two consists of a brief review of the most recent approaches in examining the structure of FFMQ questionnaire and the emphasis put on factor analysis. In the empirical part we provide justification for the number of clusters we choose, we decide on the clustering algorithm and the appropriated linkage methods, and provide detailed explanations on clusters stability. Finally, we look into some potential determinants of the FFMQ score. We conclude with both methodological and content implications for the study of mindfulness, along with a careful revision of the limitations of our study.

2. MEASURES AND COMPUTATIONAL APPROACHES FOR MINDFULNESS

2.1. The Five Facets of Mindfulness Questionnaire (FFMQ)

The set of investigative tools for mindfulness ranges from one-dimensional examples, like the Mindfulness and Attention Awareness Scale (MAAS) or the Freiburg Mindfulness Inventory (FMI), to most popular multi-dimensional ones: with two factors - Toronto Mindfulness Scale (TMS), with four factors - the Kentucky Inventory of Mindfulness (KIMS) or with five factors - the Five Facets Mindfulness Questionnaire (FFMQ). As it is natural, parsimony proves more efficient in some well-defined cases (patients dealing with depression, anxiety etc), or for some groups (e.g. regular versus non-regular meditators), but at the level of the general population there is still much to discuss regarding the best and accurate coverage of mindfulness. The findings of reputed scholars in the field like Ruth Baer and her colleagues (2006) commend the higher level of sharpness captured by a multifaceted construct of mindfulness, thus offering an important choice criteria.

Bergomi, Tschacher and Kupper (2013) offer a good overview on advantages, disadvantages and particularities in implementation of these different scales. In line with their opinion that the Five Facet Mindfulness Questionnaire – FFMQ (Baer et al., 2008) is the best choice for conducting evaluations at the level of the general population, the current paper takes the tool as a reference point for exploring mindfulness on a Romanian sample.

Moreover, taking into account the objective of national comparability, the widespread use of FFMQ, strengthens the choice of the scale. Several studies have examined the psychometric properties of the scale and grassroots implications for different countries: Spain (Cebolla et al., 2012), Sweden
However, we found little or almost no evidence for calibrating this particular tool in Romania and Eastern Europe in general. Thus, the paper wants to address and minimize this gap.

The 39-item scale of FFMQ was translated into Romanian and implemented through an on-line questionnaire, sent to a small sample of participants located in Bucharest, both students and working adults.

2.2. The analytics around FFMQ

While confirmatory factor analysis probably holds the highest position for favorite approach in dealing with the FFMQ scale, there are many technical nuances to account for. The original study conducted by Ruth Baer and her colleagues (2006) was focused on the procedure of item parceling, with other papers following not only in a replication mindset (Williams et al., 2014; Deng et al., 2011; Aguado, 2015) but also in implementing slight changes. The immediate alternative was to employ individual items as indicators (Neuser, 2010), under the assumption that this manner shows a more salient proof for model fit and model specification. Tran et al. (2013) makes a strong argument about the problems associated to item parcels procedures but nonetheless there is no consensual view on the matter. From a general theoretical perspective concerning factor analysis, more recent interventions, like Zhang and Preacher (2015), suggest increase attention for the cases in which we may obtain similar rotated factor loading but very different standard errors, depending on the method chosen for factor rotation.

Often, structural equation modeling (SEM) is also used to investigate the factorial structure of a questionnaire, with different preferences for the maximum likelihood minimization path (Bollen and Lennox, 1991) or for bootstrapping goodness of fit measures (Bollen and Stine, 1992). However, despite the consistent alternative, in the case of SEM analysis applied to mindfulness, the focus is more on explaining the links between different other variables, and not on the internal configuration of the concept: a structural model of procrastination, mindfulness and health relationships (Sirois and Tosti, 2012) with mindfulness as a mediator variable or a structural model of mindfulness, sleep and wellbeing (Howell, Digdon and Buro, 2010). One crucial aspect here is also the fact that those SEM studies usually employ one-dimensional scales for mindfulness (MAAS or FMI), and thus they somewhat avoid the analytical layers of the five dimensions proposed by FFMQ.
3. DATA AND EMPIRICAL ANALYSIS

3.1. Sample characteristics and several implications

The sample consists of 111 subjects, among which 47.2% are females and 52.8% are males. 43.6% of the participants in this study work in public institutions, while the rest of 56.4% work in private sector. Nearly half of the participants (46.4%) work in the IT domain, 25.5% have jobs that involve managerial decisions, while the rest of 28.1% belong to a separate category of “support jobs”, like for example being a translator, or a specialist in communication. The descriptive statistics for the five numerical dimensions of the FFMQ, as well as for the total score, are presented in Table 1 below.

### Descriptive statistics

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Min</th>
<th>Mean</th>
<th>Median</th>
<th>Max</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observing</td>
<td>12</td>
<td>24.81</td>
<td>25</td>
<td>39</td>
<td>5.26</td>
</tr>
<tr>
<td>Describing</td>
<td>17</td>
<td>30.22</td>
<td>31</td>
<td>40</td>
<td>5.74</td>
</tr>
<tr>
<td>Awareness</td>
<td>16</td>
<td>30.84</td>
<td>32</td>
<td>40</td>
<td>5.53</td>
</tr>
<tr>
<td>Non-judging</td>
<td>12</td>
<td>27.03</td>
<td>27</td>
<td>39</td>
<td>5.97</td>
</tr>
<tr>
<td>Non-reactivity</td>
<td>11</td>
<td>21.11</td>
<td>22</td>
<td>32</td>
<td>4.13</td>
</tr>
<tr>
<td>FFMQ</td>
<td>95</td>
<td>134</td>
<td>135.5</td>
<td>175</td>
<td>15.26</td>
</tr>
</tbody>
</table>

Source: Authors’ work

In each case, both the mean and the standard deviation are very similar to the results obtained for the Spanish (Cebolla et al., 2012) and Italian samples (Giovannini et al., 2014), though slightly higher compared to the Chinese study (Hou et al., 2014) in the awareness and non-judging dimension. It is noteworthy that these exact two elements are the most salient in respect to dealing with different forms of psychological distress (Hayes & Feldman, 2004).

However, there is a little bit more that can be derived from these simple characteristics of the sample. We choose to conduct a preliminary analysis of the five dimensions and final score, and investigate whether our variables are normally distributed. This fact will be essential, as we will explain later. To conduct the normality test we used the “moments” package in R, and the function jarque.test() for the Jarque – Bera Normality test (Jarque&Bera 1980), available in the “moments” package in R, test created by Frederick Novomestky (Komsta& Novomestky 2015). The null hypothesis is that the data comes from a normal distribution, while the alternative hypothesis states that there are significant differences between the distribution of the data and the normal distribution, in the sense that skewness and kurtosis are significantly
different than their default values in case of normally distributed data, namely 0 and respectively 3. Table 2 presents the results and shows that only the awareness dimension is not normally distributed.

The Jarque – Berra Normality Test

<table>
<thead>
<tr>
<th>Dimension</th>
<th>JB - statistic</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observing</td>
<td>1.37 (0.5041)</td>
<td>Fail to reject the null</td>
</tr>
<tr>
<td>Describing</td>
<td>4.4405 (0.1086)</td>
<td>Fail to reject the null</td>
</tr>
<tr>
<td>Awareness</td>
<td>7.3017 (0.026)</td>
<td>Reject the null</td>
</tr>
<tr>
<td>Non-judging</td>
<td>1.0924 (0.579)</td>
<td>Fail to reject the null</td>
</tr>
<tr>
<td>Non-reactivity</td>
<td>0.469 (0.791)</td>
<td>Fail to reject the null</td>
</tr>
<tr>
<td>FFMQ</td>
<td>0.342 (0.843)</td>
<td>Fail to reject the null</td>
</tr>
</tbody>
</table>

(p – values in parentheses)

Source: Authors’ work

One of the implications of this preliminary finding is that, according to the properties of a normal distribution, approximately 99.9% of a population for which our sample might be representative would score to each dimension except for awareness, values situated three standard deviations about the mean. Table 3 summarizes the lower and the upper values of these possible scores.

99.9% of the population would score

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Lower limit</th>
<th>Mean</th>
<th>Upper limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observing</td>
<td>9.02</td>
<td>24.81</td>
<td>40.60</td>
</tr>
<tr>
<td>Describing</td>
<td>12.99</td>
<td>30.22</td>
<td>47.45</td>
</tr>
<tr>
<td>Non-judging</td>
<td>9.11</td>
<td>27.03</td>
<td>44.94</td>
</tr>
<tr>
<td>Non-reactivity</td>
<td>8.69</td>
<td>21.11</td>
<td>33.52</td>
</tr>
<tr>
<td>FFMQ</td>
<td>88.19</td>
<td>134</td>
<td>179.81</td>
</tr>
</tbody>
</table>

Source: Authors’ work
The “awareness” dimension calls for other two separate tests that would clarify how the distribution of this variable differs from a normal distribution. In this sense we conduct two tests: the D’Agostino test for skewness (D’Agostino 1970 now available in the “moments” package in R, developed by Lukasz Komsta, 2015), and the Anscombe-Glynn test of kurtosis (Anscombe & Glynn 1983, also available in the “moments” package cited before).

The D’Agostino test is built upon the fact that the skewness of a normal distribution is 0. The real data rarely obey to this rule, in the sense that the skewness might be in fact different than zero, but not significantly different to support the idea that data are not normally distributed. This test is aimed to detect significant non-zero skewness, under the null that the data is normally distributed. In a similar manner, the Anscombe-Glynn test addresses another characteristic of a normal distribution: it is built upon the fact that a normal distribution has a kurtosis of 3, and captures the significant differences between the kurtosis of a certain data and this benchmark value. The null hypothesis of this test is that data is normally distributed, while the alternative states that it is not. Each of the two tests can be conducted as one tailed, or two tailed tests.

The skewness and kurtosis tests for Awareness (two tailed tests)

<table>
<thead>
<tr>
<th>Dimension</th>
<th>D’Agostino skewness test</th>
<th>Anscombe-Glynn test for kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Awareness</td>
<td>-0.62654 (0.008)</td>
<td>2.849 (0.95)</td>
</tr>
</tbody>
</table>

(p – values in parentheses)

Source: Authors’ work

In our case, the second column in Table 4 shows that we reject the null hypothesis of a skewness equal to 0, but the kurtosis is near enough to 3 to consider that the distribution has a similar kurtosis as a normal distribution. Moreover, the one-tailed D’Agostino test reveals that the distribution of awareness is in fact negatively skewed, suggesting that some small extreme values occur in this variable for which Figure 1 presents the histogram.
Some small extreme values occur for the awareness dimension

Figure 1

Source: Authors’ work

At the end of this descriptive subsection we can summarize that, of the five components of the mindfulness overall score, the only dimension that does not follow a normal distribution is awareness. The bias is induced by some extreme small values that would potentially suggest either some type of environmental or contextual noise at the moment of filling the questionnaire, or a low level of commitment when answering to this part of the survey for example.

3.2. An exploratory approach through cluster analysis: between the method and the empirical results

The FFMQ score has been developed and used before to rank subjects based on their level of mindfulness (Baer, 2011). The aim of this section is to explore whether within the structure of the FFMQ score there are significant patterns that group subjects into relevant categories. Let’s take for example the overall score of 148: it can be achieved from numerous combinations between its five components, like observing = 25, describing = 36, awareness = 25, non – judging = 28, and non – reactivity = 34. Our question is: among all possibilities, are some of them more likely to provide this score of 148, than others? And if the answer is yes, do these combinations create stable patterns that may predict the overall score in a consistent way? Our intention in this section is therefore to explore whether we can gain in understanding if we look into the structure of the overall score, and not only to its total value.

We have chosen to further conduct this exploration through a cluster analysis applied to a data set comprising of the five components that lead
to the final FFMQ score: observing, describing, awareness, non-judging, and non-reactivity. Cluster analysis is an unsupervised learning tool that aims at finding groups within data. The observations within groups are similar with each other, while the observations in different groups are not. This type of investigation is very common in statistical studies and has been used in various fields, though in what concerns the mindfulness studies another tool has been used widely, namely the factor analysis. To our knowledge, this is the first contribution in the area of mindfulness that uses cluster analysis to make a deeper sense of the five factors that have been derived through the preliminary investigation of the FFM questionnaire.

In the first stage we compute the Hopkins statistic, used as a measure for clustering tendency. This is a sensitive issue in the area of any type of unsupervised learning, where the analysis will tend to generate clusters even if in reality they are meaningless. To calculate the Hopkins statistic, we use the “clustertend” package in R, and the function hopkins(), and get in this case a value of 0.37. Having in mind that the critical value of the Hopkins statistic, the one that indicates that the data is uniformly distributed and there are no meaningful clusters within is 0.5, we admit that there is a certain tendency to clustering in our case, though not that high (Hopkins&Skellam 1954, Lawson&Jurs 1990).

There are currently an impressive number of clustering algorithms, but choosing among them the best one to conduct a particular analysis is not an easy task (Flynt and Dean, 2016). Moreover, the problem of determining the right number of clusters, like for example k-mean or k-medoid requires, is also a difficult choice. The performance of the final clusters is evaluated based on several characteristics (compactness, well-separation, connectedness and stability), but also the relevance of the results in terms of practical interpretation is important. To identify the right clustering method and the right number of clusters, we used in the first instance the “clValid” package. The clValid() function in this package provides validation measures for several clustering algorithms and a specified number of clusters. Among the measures available with this function, internal and stability measures are of interest for our purpose. The internal measures are based on connectivity, Silhouette Width, and Dunn Index, while the stability measures include the average proportion of non-overlap (APN), the average distance (AD), the average distance between means (ADM), and the figure of merit (FOM). A detailed description of each measure and index is available in the clValid package description (Brock et al. 2008) as well as in Datta, S. and Datta (2003, 2006). Table 5 provides the recommendations based on each measure.

As it is very common in such cases, there is no clear best algorithm or number of clusters, each of the methods providing different recommendation.
To overcome this issue, we used the “RankAggreg” package that performs aggregation of ordered lists based on the ranks, via the Cross-Entropy Monte Carlo algorithm or the Genetic Algorithm (Pihur et al. 2007).

### The recommended clustering method and number of clusters (clValid function)

<table>
<thead>
<tr>
<th>Index</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>APN</td>
<td>hierarchical - 4</td>
</tr>
<tr>
<td>AD</td>
<td>kmeans - 11</td>
</tr>
<tr>
<td>ADM</td>
<td>hierarchical - 5</td>
</tr>
<tr>
<td>FOM</td>
<td>kmeans - 11</td>
</tr>
<tr>
<td>Connectivity</td>
<td>hierarchical - 4</td>
</tr>
<tr>
<td>Dunn</td>
<td>hierarchical - 6</td>
</tr>
<tr>
<td>Silhouette</td>
<td>hierarchical - 4</td>
</tr>
</tbody>
</table>

*Source: Authors’ work*

The RankAggreg function comes out with a recommendation for hierarchical clustering with 4 clusters. Figure 2 graphically presents the final output of this preliminary investigation.

**The RankAggreg function recommends hierarchical clustering with 4 clusters**

*Source: Authors’ work*
The hierarchical agglomerative clustering is perhaps one of the most common clustering algorithms. Initially, each data point is assigned to its own singleton cluster, and then an iterative process begins. The similarity between clusters is evaluated based on a variety of pre-defined distances, the most similar clusters being merged at each step into one single cluster. Hierarchical clustering provides a dendogram, a graphical presentation of the agglomerative process, which can be cut at various levels to produce the desired number of clusters. Figure 3 will illustrate this fact.

The problem of clustering algorithm being solved, the selection of the best linkage method comes next. This is a sensible issue, as many other issues in the area of unsupervised techniques. Usually, the clustering method is selected based on the logic of the result, of its power of interpretation, or on their appropriateness for the data to be clustered. The most used methods, among the wide variety available for hierarchical clustering, are: average linkage method, complete linkage method, and Ward’s method. In our paper, we will compare the results provided by these three methods.

The Average Linkage method (Sokal&Michener 1958) is very intuitive and defines the distance between two clusters as the average distance between the data points in the first cluster, and the data points in the second cluster. As a result, at each stage of the analysis we will merge the clusters that have the smallest average linkage distance. The Average Linkage distance between two clusters G and H is defined as follows:

\[ d_{AL} = \frac{1}{N_G N_H} \sum_{x \in G} \sum_{y \in H} d_{xy} \]

Here, \( N_G \) is the size of the cluster G, and \( N_H \) is the size of the cluster H. By \( d_{xy} \) we denoted the Euclidean distance between two generic points, x from G and respectively y from H. More exactly:

\[ d_{xy} = \sum_{i=1}^{n} |x_i - y_i|, \quad n \text{ being the total number of variables to be clustered}. \]

In our particular case, this method will tend to merge groups of subjects that are, on average, very close to each other in terms of the scores achieved to each of the five components of the FFMQ score. Since taking the average in a data set has the potential to mitigate the differences between extreme values, we can expect that the clusters created based on this linkage method will be permissive and include a wider variety of individuals than other methods would include.
The Complete Linkage method creates groups for which the similarity is defined as the maximum distance between any single data point in the first cluster and any single data point in the second cluster. The Complete Linkage distance between two clusters G and H is defined as follows:

\[ d_{CL} = \max \{ d_{xy}, x \in G, y \in H \} \]

The Euclidean distance between two generic points x and y was defined above. Therefore, the method pursues to combine clusters that have the smallest complete linkage distance between them and seek for final groups that differ on the basis of the maximum distance. Based on this definition, we expect that for our data the Complete Linkage method will create clusters where the extreme values will play a significant role. More precisely, we expect that this distance will avoid merging groups close to each other in terms of average distance, because of potential outliers that are far apart.

Unlike these approaches, the Ward method (Ward 1963) is distance–free and treats the clustering process as an analysis of variance. It looks for the clusters that explain the most of the variance in the data set, and aims at maximizing the value of R–square, defined as the ratio between the variance explained by the clusters over the total variance of the initial data set.

There is a high probability that the clusterization provided by these three methods will lead to different structure of the groups. As recommended in the literature, the logic behind choosing the appropriate linkage method should be the interpretation power as well as the stability of the clusters. While there are arguments stating that choosing a linkage measure that leads to interpretable clusters if a form of psychological validation, without too much scientific value, the stability of the clusters that also have a practical meaning can be a valuable selection tool.

The stability of the clusters created via these three methods is evaluated using the Jaccard coefficient. This coefficient, originally introduced by Sneath (1957), is defined as the ratio between the size of the intersection and the size of the union of two clusters G and H as follows:

\[ J(G, H) = \frac{|G \cap H|}{|G \cup H|} \]

The clusterboot() function in the “fpc” R package provides a mean for assessing the stability of the clusters based on a bootstrap resampling scheme, and the value of the Jaccard coefficient of similarity is the main driver in reaching a conclusion. The idea behind this procedure is the following: we conduct an initial clustering and record the clusters. Then, the procedure
is repeated, this time on a different data set resulted from resampling the observations in the initial data. A comparison between the original and the new clusters will provide information regarding the most similar cluster we found after resampling, the similarity being evaluated based on the Jaccard coefficient of similarity defined above. The algorithm compares the value with 0.5: whenever the distance between the old and new cluster is less than 0.5, the cluster is considered unstable and is dissolved. We count the number of time a cluster is dissolved and decide based on the rule described in the next paragraph whether the cluster includes relevant information, or only noise.

The critical values of this coefficient are 0.5, 0.6 and 0.75: highly unstable clusters correspond values of the Jaccard coefficient less than, or at most 0.5. If the coefficient ranges between 0.6 and 0.75, we admit that there are some patterns in the data, but it is unclear which data points belong to which cluster. Whenever the Jaccard coefficient is higher than 0.75, as admit that we found stable clusters. Values of 0.85 or higher point toward highly stable results. (Hennig 2007, 2008 and Zumel 2015). In our study we run a bootstrap with 1000 iterations, and the results are recorded in Table 4.

The Complete Linkage method provides unstable groups, each of the coefficients being below the trustworthy level of 0.6. Therefore, we will not consider the Complete Linkage method as an alternative to explore our data. Unlike this case, both the Ward method and the Average Linkage lead to the conclusion that there are some patterns in the data, but it is only the Average Linkage method that provides indeed a stable group, namely the first group with a Jaccard coefficient of 0.84. Moreover, the Average Linkage method results in three groups that may be considered as relevant patterns, while the Ward method identifies only two clusters with a Jaccard coefficient higher than 0.6.

The Jaccard coefficient for each linkage method, and cluster

<table>
<thead>
<tr>
<th>Method</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ward</td>
<td>0.6636</td>
<td>0.7220</td>
<td>0.5182</td>
<td>0.5021</td>
</tr>
<tr>
<td>Average</td>
<td>0.8432</td>
<td>0.6100</td>
<td>0.3620</td>
<td>0.6027</td>
</tr>
<tr>
<td>Complete</td>
<td>0.5339</td>
<td>0.4475</td>
<td>0.4688</td>
<td>0.3504</td>
</tr>
</tbody>
</table>

Source: Authors’ work

Figure 3 graphically presents the dendograms that correspond to the three linkage methods, using hierarchical clustering and 4 clusters. The Ward method provides the most balanced groups in terms of size of the clusters, while the other two methods lead to groups with only few individuals. In what follows we will maintain the Average Linkage method and the Ward’s method to further conduct the cluster analysis.
Hierarchical clustering with 4 clusters: Ward, Average and Complete Linkage methods

Figure 3

Table 7 shows that the most consistent group created by the Average Linkage method – the one that proved to have the highest stability, according to the results presented in Table 6 – is the first group, comprising of 81.82% of the participants, namely 91 of the subjects. The least stable group, with a Jaccard coefficient of 0.362, is the third group, while the rest of the groups points toward some existing patterns, but not very stable. These remained groups include very few subjects, as follows: group 2 includes only one person; group 4 includes 16 persons, while the third group is left with 3.

Clusters characteristics – The Average Linkage Method

Table 7

<table>
<thead>
<tr>
<th>Groups</th>
<th>Observing</th>
<th>Describing</th>
<th>Awareness</th>
<th>Non-judging</th>
<th>Non-reactivity</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24.63</td>
<td>31.14</td>
<td>32.42</td>
<td>27.89</td>
<td>21.13</td>
<td>0.8182</td>
</tr>
<tr>
<td>4</td>
<td>25.25</td>
<td>23.13</td>
<td>21.50</td>
<td>20.38</td>
<td>19.94</td>
<td>0.1455</td>
</tr>
<tr>
<td>3</td>
<td>31.33</td>
<td>38.33</td>
<td>38.00</td>
<td>34.67</td>
<td>30.00</td>
<td>0.0273</td>
</tr>
<tr>
<td>2</td>
<td>14.00</td>
<td>36.00</td>
<td>16.00</td>
<td>33.00</td>
<td>11.00</td>
<td>0.009</td>
</tr>
</tbody>
</table>

Source: Authors’ work

By simply looking at the results recorded in Table 5, it is difficult to set description of the groups, based on the differences among them. The group that detaches itself from the rest is group 3, characterized by higher values than the rest for each dimension. Group two is also different in what concern the lowest values scored at observing, awareness and non-reactivity.
Compared with the remained groups, four and one, these subjects seem to be on average better at describing and have the rare quality of non–judging at nearly the similar level with the champion group, three. Group one and four seem to be more balanced in terms of scores to each dimension. While pretty similar at observing and non–reactivity (with slightly lower values in the fourth group), for the other three dimensions the first group scores up to 11 points more than group 4.

The first conclusion that derives from our result is that the sample is characterized by a high homogeneity, as long as up to 82% of the participants belong to the same groups, and this homogeneity results in pretty high scores at each dimension. We may interpret the results concerning this group as a type of “majority” profile. Although much small than group 1, group 4 includes an important percent of the subjects, and balanced scores in the sense that there are no extreme values as in group 2, for instance.

The Ward’s method produces the groups recorded in Table 8. As mentioned earlier, the size of the groups are, in this case, more balanced that in the Average Linkage case. The first two groups, which are somehow consistent, share around 36% of the subjects each, while the rest of nearly 30% are split between the two unstable groups.

clusters characteristics – The Ward’s Method

<table>
<thead>
<tr>
<th>Groups</th>
<th>Observing</th>
<th>Describing</th>
<th>Awareness</th>
<th>Non-judging</th>
<th>Non-reactivity</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27.88</td>
<td>30.50</td>
<td>31.25</td>
<td>24.95</td>
<td>20.00</td>
<td>0.3636</td>
</tr>
<tr>
<td>2</td>
<td>23.10</td>
<td>33.92</td>
<td>33.87</td>
<td>30.64</td>
<td>24.33</td>
<td>0.3545</td>
</tr>
<tr>
<td>3</td>
<td>18.41</td>
<td>24.47</td>
<td>30.24</td>
<td>30.35</td>
<td>17.65</td>
<td>0.1545</td>
</tr>
<tr>
<td>4</td>
<td>28.57</td>
<td>26.07</td>
<td>21.93</td>
<td>18.86</td>
<td>19.50</td>
<td>0.1274</td>
</tr>
</tbody>
</table>

Source: Authors’ work

As much as with the previous linkage method, the Ward’s method produces clusters that are difficult to describe by only looking at the average group values for each dimension. The first two groups seem to consist of subjects that scored pretty high value to all five dimensions. Unlike these cases, the third group achieved on average lower scores at observing and non-reactivity, whiles the fourth group scores less at non–judging and non–reactivity. However, these last groups are unstable, therefore we cannot rely on a specific profile they may point toward.

Our next step in the investigation is whether the clusters we detected are significant in explaining the final FFMQ score, and to what extend other variables like gender, age and public versus private employment may have a significant impact on the overall level of mindfulness.
3.3. Looking into some potential FFMQ determinants

In this subsection we conduct a regression analysis to explore whether the clusters we detected explain the FFMQ score, and how this score relates with other variables. While a similar analysis can be conducted based on the numerical values of each individual dimension, such an approach would fail to account for possible consistent patterns associated with similar scores of mindfulness, a phenomenon usually called cross-sectional dependence.

The model we fit based on our data is the following:

\[
FFMQ = \beta_0 + \beta_1 \text{Groups} + \beta_2 \text{Age} + \beta_3 \text{Gender} + \beta_4 \text{Working place} + \varepsilon
\]

where \(\varepsilon\) is the error term. The fact that the FFQM scores follow a normal distribution, as found in normality testing (section 3.2.) becomes a very comfortable result in this context. Table 7 provides the output of this model for each of the two clustering methods, along with the information regarding the statistical significance of the predictors.

A preliminary evaluation of the model that corresponds to the Average Linkage clustering method shows that the regressors are able to explain up to 60% of the variation in FFMQ overall score. The model is statistically significant at an overall level, although three of the background variables, gender, age and working place do not prove to be of help in explaining the variability in the overall level of mindfulness. More exactly, and according to our results, males seem to be less mindful than women, and also those subjects who work in private institutions scores less, on average, than those who work in public workplaces, but Table 7 presents the p-values that point toward differences that are not statistically significant. Age seems to be a variable that, at least in this framework, is not able to bring any contribution to the model.

However, of the four groups identified in our preliminary clustering, the first – and most stable – group has been set as reference group. Compared with this cluster, the second one has an average a lower mindfulness score, a result that also holds for the fourth cluster. Unlike these two, the third group points toward a significantly higher FFMQ score compares with the reference.

The third column in Table 9 shows the results for the Ward’s linkage method. The overall explanatory power of the model is slightly lower than in the first case, but the background variables prove to be as non-significant as before. The only notable difference is that working place and age became marginally significant, namely significant at 10% level.
The regression model for each clustering linkage method

<table>
<thead>
<tr>
<th>Model: FFMQ score</th>
<th>Estimated coefficients (Average Linkage method)</th>
<th>Estimated coefficients (Ward’s distance method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>131.04*** (&lt; 2e-16)</td>
<td>126.141*** (&lt; 2e-16)</td>
</tr>
<tr>
<td>Group:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group 1:</td>
<td>Reference</td>
<td>Reference</td>
</tr>
<tr>
<td>Group 2:</td>
<td>-26.575** (0.0097)</td>
<td>12.153*** (1.86e-06)</td>
</tr>
<tr>
<td>Group 3:</td>
<td>34.823*** (3.44e-08)</td>
<td>-12.852*** (7.32e-05)</td>
</tr>
<tr>
<td>Group 4:</td>
<td>-26.642*** (&lt; 2e-16)</td>
<td>-19.998*** (1.96e-08)</td>
</tr>
<tr>
<td>Gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Female:</td>
<td>-0.4892 (0.812)</td>
<td>Reference</td>
</tr>
<tr>
<td>Male:</td>
<td>Reference</td>
<td>-3.051 (0.168)</td>
</tr>
<tr>
<td>Working place</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public:</td>
<td>Reference</td>
<td>Reference</td>
</tr>
<tr>
<td>Private:</td>
<td>-0.129 (0.9502)</td>
<td>3.758 (0.089)</td>
</tr>
<tr>
<td>Age</td>
<td>0.213 (0.109)</td>
<td>0.250 (0.0812)</td>
</tr>
<tr>
<td>R – Squared</td>
<td>60.07%</td>
<td>55.15%</td>
</tr>
<tr>
<td>Adjusted R – Squared</td>
<td>57.75%</td>
<td>52.54%</td>
</tr>
<tr>
<td>F - statistic</td>
<td>25.83 (&lt; 2.2e-16)</td>
<td>21.11 (5.111e-16)</td>
</tr>
</tbody>
</table>

P – values in parentheses

Source: Authors’ work

The second cluster provides on average significantly higher FFMQ scores than the reference group, while groups 3 and 4 are associated with lower average scores than the reference. However, since clusters 3 and 4 are unstable, we cannot claim that we detected a particular profile that leads to lower mindfulness scores than other profiles.

4. CONCLUSIONS, DISCUSSIONS AND FURTHER RESEARCH

The Five Facets Mindfulness Questionnaire is documented as one of the valid instruments aimed at measuring the level of mindfulness. It is built upon a questionnaire of 39 questions, as a valuable index that combines five different dimensions: observing, describing, awareness, non-judging and non-reactivity. The result of this combination is an overall score that helps ranking subjects according to their level of mindfulness.

The aim of our research was to look into the structure of this score and detect potential stable patterns that may explain the variability in mindfulness level. Based on a sample of 111 subjects, we conducted first a cluster analysis based on two different linkage methods and found four groups, suggesting four types of persons built upon the scores recorded to the five dimensions mentioned above. Of these four groups created by the Average Linkage method, one was highly stable, and included the majority of the participants, suggesting
that among the subjects there is a significant degree of homogeneity. Two of the
other groups pointed toward existing patterns, but without any warranty
regarding the type of subjects belonging to these groups. One last group was
highly unstable. The results obtained in this exploratory stage suggest that,
at least in this particular sample and using the Average Linkage method, the
majority of the participants share similar characteristics in what concerns the
levels they scored to the five dimensions.

The Ward’s method identified two groups that can be associated with
consistent patterns, but the level of stability was weak: although we can admit
that these patterns may exist, it is unclear which persons belong to which
group. Since these groups account for nearly 70% of the participants, the idea
of a certain homogeneity among subjects comes up again: the results suggest
that the participants may switch the groups randomly, without creating any
significant changes in the clustering results.

In the last stage we fitted a regression model aimed at explaining the
variations in the mindfulness level, measured by the FFMQ score, based on
the groups we detected, and other three background variables: gender, age and
workplace. We found that although the background variables did not bring
any contribution, all four groups proved to be statistically significant in each
linkage case. This result suggest that the five dimensions of the FFMQ overall
score cluster the participants in our study in groups that are systematically
different in their mindfulness level, even for this particular sample some of
the groups are highly unstable. This result invites to further investigations
regarding possible relations between the patterns we found, and possible
personality profiles the mindfulness level may be associated with.

One of the limits of our study consists of the fact that the Average
Linkage method led to a group that includes the majority of the subjects. This
result prevents us from fitting and comparing any models with interactions,
models that could have clarified potential interplays between the background
variables and the clusters. If, for example, a model with interaction between
groups and working place is fitted for the Ward’s method clustering, age
becomes statistically significant, as well as the working place. That means
that it is not only age, or the working place that impact the mindfulness
overall score, but the interaction between the patterns we detected within the
mindfulness dimensions and the background variables. Having in mind the
size of the clusters in the Average Linkage case, such a model with interaction
cannot be built for comparison.

Another important limit is the convenience sampling that has been
used as the sampling method, while a replication of this investigation having
more background variables might be a good idea for further research.
Despite the evident limitations discussed above, our work contributes to the existing literature in that it provides a methodological framework that can be applied to different, and preferably richer data, both in what concerns the preliminary, exploratory analysis, but also in the second stage of assessing the significance of the clusters. The value added to the literature consists of a deeper perspective over the components of the mindfulness overall score, components treated here not only as numbers that should be added to calculate a total value, but also as potential determinants of some specific, and stable patterns that are more likely to be associated with certain values of FFMQ scores than others.

Last, but not least, to our knowledge this is the first study on mindfulness applied to Romanian subjects. The fact that we found similarities between the values recorded by Romanian participants, and subjects belonging to other countries, is a first and important step in a deeper analysis that may concern the level of mindfulness in Romania, in the sense that it invites to further interventions to increase the mindfulness level based on similar models that have been implemented worldwide.

References


Brief overview of survey sampling techniques with R

Camelia Goga (camelia.goga@univ-fcomte.fr)
Université de Bourgogne Franche-Comté

ABSTRACT

For the last decade, many R packages have been suggested for performing sample selection according to various sampling designs as well as design-based estimation of finite population interest parameters and of their variance. The goal of this article is to give a brief overview of some of these packages with application on two datasets. A special attention is given to calibration issues.

Keywords: over-calibration, penalized calibration, principal component analysis

JEL Classification: C83, C88

1 Introduction

For the last decade, we assist to a large expansion of R packages dedicated to survey sampling techniques from few packages to more than eighty nowadays. A comprehensive list of all packages dedicated to survey sampling techniques and official statistics is given by Matthias Templ at https://cran.r-project.org/web/views/OfficialStatistics.html.

Broadly speaking, there are two main R-packages dedicated to survey sampling techniques: sampling and survey. Package sampling was suggested by Alina Matei and Yves Tillé from the University of Neuchâtel and is concerned mainly by performing sample selection according to various with or without replacement sampling designs. Some estimation issues are also treated. Package survey was suggested by Thomas Lumley from the University of Auckland and is concerned with design-based estimation of finite population interest parameters and of their variance. The book Complex surveys, a guide to analysis using R of T. Lumley (Lumley (2010)) describes most of functions contained in this package. The other existing packages are dedicated to a specific issue from survey sampling. We focus mainly in this paper on sampling and survey packages.

The paper is structured as follows: we describe in section 2 the dataset used in this paper. Section 3 gives the R-functions performing the most important sampling designs and the Horvitz-Thompson estimation of finite population parameters and of their variance. Section 4 is dedicated to the calibration method which is a very popular method used in many statistical institutes. We give first the description and the R-implementation of the calibration estimator. Next, we present the penalized and the principal components calibration estimator suggested in order to robustify the calibration estimator in presence of a very large number of auxiliary data. Finally, section 5 concludes the paper.
2 Data description

We consider in this article the dataset called rec99 which contains an extract from the last French Census performed in 1999. Data is collected on $N = 554$ French towns from the south of France (the region Haute-Garonne) about:

- **POPSDC99**: the number of habitants in 1999
- **LOG**: the number of dwellings
- **LOGVAC**: the number of empty dwellings.

This dataset contains also the postal code (variable **CODE_N**) and the name of each small town (variable **COMMUNE**). Two factor variables are also available. The first one is called **BVQ** (“bassin de vie quotidien”) and it represents a code given by INSEE, the French Statistical Institute, to each small sub-regions of the region Haute-Garonne. This variable will be used as a cluster variable. The second factor variable is called **stratlog** and reflects the town population size taking four modalities: “=1” small, ..., “=4” very large. This variable will be used as a stratification variable. We give below an extract from this data-frame.

```r
> rec99=read.csv("rec99.csv")
> rec99[1:3,] ###the first 3 individuals from rec99
CODE_N COMMUNE BVQ N POPSDC99 LOG LOGVAC stratlog
1 31014 ARGUENOS 31020 57 94 1 1
2 31131 CAZAUNAUS 31020 47 56 4 1
3 31348 MONCAUP 31020 26 57 2 1
```

3 Sample selection and estimation of finite population parameters

3.1 Sample selection

Let consider a finite population $U = \{1, \ldots, N\}$ and let the sample $s$ be selected from $U$ according to a sampling design $p(\cdot)$. Denote by $\pi_k = P(k \in s)$ the first-order inclusion probabilities supposed to be positive for all $k \in U$.

Package **sampling** implements the selection of many equal and unequal sampling designs and most of them are without replacement. The outputs of functions dedicated to sample designs are $N$-dimensional vectors containing zeros for the non-selected individuals and positive values for the selected individuals. In the latter situation, this positive value represents the number of times an individual was selected and it can be equal only to one if the design is without replacement.

For example, **srswor**($n, N$) implements the selection of a simple random sample without replacement of size $n$ from a population of size $N$. The output is in case a vector of size $N = 554$ containing $n = 70$ ones and $N - n = 484$ zeros.

```r
library(sampling)
###selection of a simple random sampling without replacement of size 70
s1.rec<-srswor(70,554)
```

We can obtain the selected towns with the corresponding variables by

```r
data=rec99[which(s1.rec==1),]
```
Function \text{UPsystematic}(pik) implements the systematic sampling with unequal inclusion probabilities contained in the \(N\)-dimensional vector \(pik\). For example, if we desire selecting a systematic sampling design proportional to an auxiliary variable taking values \(x_k\) for \(k \in U\), then the first-order inclusion probabilities are \(\pi_k = n x_k / \sum_U x_k\) which can be declared in \(pik\). We select from \text{rec99} a systematic sample of size \(n = 70\) proportional to the auxiliary variable \(\text{LOG}\), total number of dwellings from each small town.

```r
###selection of a proportional to size systematic sample of size n=70
###total of the auxiliary variable
attach(rec99)
tLOG=sum(LOG)
tLOG
[1] 197314
###inclusion probabilities
pk=LOG/tLOG###
pik=70*pk
###selection of the sample
sys.rec=UPsystematic(pik)
data=rec[which(sys.rec==1),]
detach(rec99)
```

The usual systematic sampling is obtained in the particular case \(\pi_k = n / N\), namely \(\text{pik} = \text{rep}(n/N, N)\).

Function \text{UPpoisson}(pik) implements the Poisson sampling with given unequal inclusion probabilities \(\pi_k\) contained in \(pik\). For \(\pi_k = \pi\) for all \(k \in U\), we obtain the Bernoulli sampling. The balanced sampling, the Brewer, Sampford, Tillé and many other unequal sampling designs are also implemented in \text{sampling}. We mention also the very recent package \text{balancedsampling} of Anton Grafstrom implementing the balanced sampling in a very fast way thanks to the \text{Rcpp} package. A spatially balanced sample is obtained by using the local pivot and implemented by functions \text{lpm1} and \text{lpm2} already used for the national seashore inventory in Sweden in 2015.

A stratified sample may be selected by using the function \text{strata} (rather slow) and cluster or multi-stage samples with functions \text{cluster} and \text{mstage} from package \text{sampling}.

We give below the procedure for selecting a stratified sample from our study population. The population is divided into four strata built with respect to the population town. There are four strata and the stratum number is contained in the \text{stratlog} variable. To select a stratified sample, one needs to give the sample sizes to select within each strata. If these sample sizes are chosen according to a proportional or \(X\)-optimal allocation, they need to be computed before the sample selection. We will use the \(X\)-optimal allocation where the auxiliary variable is \(\text{LOG}\). Data needs to be ordered with respect to the stratification variable before considering the stratified sampling.

```r
###selection of a stratified sample with LOG-optimal allocation
###computing the LOG-optimal allocation
sdlog=tapply(LOG,stratlog,sd)
n=70; N=554
mech=round(n*N*sdlog/sum(N*sdlog))
###1 2 3 4
###2 4 12 52
###selection of the sample
rec99.order=rec99[order(rec99$stratlog),]
```
stsi.rec=strata(rec99.order,"stratlog",size=nech,method="srswor")

Information about the selected stratified sample is obtained with getdata:

###getting the stratified sample data
stsi.data=getdata(rec, stsi.rec)

We mention also the package \texttt{stratification} to construct strata according to the Lavallée-Hidiroglou method and package \texttt{SamplingStrata} to determine the optimal stratification of sampling frames for multipurpose sampling surveys.

Consider now the cluster sampling. Our study population is divided into $N_c = 32$ clusters called BVQ ("bassin de vie quotidien") and constructed according to a cut-off of the entire region Haute-Garone into disjointed regions according to some proximity criteria. The variable BVQ\_N from \texttt{rec99} gives the cluster number. We select $n_c = 4$ clusters by simple random sampling without replacement.

###selection of a cluster sample
grap.rec=cluster(rec99, clustername="BVQ\_N", size=4, method="srswor")

<table>
<thead>
<tr>
<th>ID_unit</th>
<th>BVQ_N</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31239</td>
<td>276 0.125</td>
</tr>
<tr>
<td>2</td>
<td>31239</td>
<td>273 0.125</td>
</tr>
<tr>
<td>3</td>
<td>31239</td>
<td>274 0.125</td>
</tr>
<tr>
<td>4</td>
<td>31239</td>
<td>275 0.125</td>
</tr>
<tr>
<td>5</td>
<td>31239</td>
<td>262 0.125</td>
</tr>
</tbody>
</table>

Information about the selected clusters is obtained with getdata:

###getting the cluster sample data
grap.data=getdata(rec99, grap.rec)

3.2 Estimation of finite population totals

Let $Y$ be the interest variable and we want to estimate the finite population total of $Y$:

$$t_Y = \sum_{k \in U} y_k.$$

Supposing that the first inclusion probabilities $\pi_k$ are positive for all $k \in U$. Then, we can construct the unbiased Horvitz-Thompson estimator of $t_Y$ given by:

$$\hat{t}_d = \sum_{k \in s} d_k y_k,$$

where $d_k = 1/\pi_k$ are the sampling weights. Supposing that all $\pi_k > 0$, the variance of the Horvitz-Thompson estimator may be estimated unbiasedly by the Horvitz-Thompson variance estimator given by

$$\hat{V}_{HT}(\hat{t}_d) = \sum_{k \in U} \sum_{l \in U} \pi_{kl} - \pi_k \pi_l \frac{y_k y_l}{\pi_k \pi_l}.$$

For without replacement designs of equal size, the variance may be estimated unbiasedly also by the Yates-Grundy-Sen variance estimator:

$$\hat{V}_{YGS}(\hat{t}_d) = \frac{1}{2} \sum_{k \in U} \sum_{l \in U} \pi_{kl} - \pi_k \pi_l \left( \frac{y_k}{\pi_k} - \frac{y_l}{\pi_l} \right)^2.$$
Remark that these two variance estimators may be different. Both packages `sampling` and `survey` provide functions for computing these two variance estimators. Besides, variance estimators based on replicates weights are implemented in package `survey`.

We detail now the computation of the Horvitz-Thompson estimator $\hat{t}$ and of its variance by using package `survey`. The creation of a design object is necessary:

```
svydesign(id, probs=NULL, strata = NULL, fpc=NULL, data = NULL, weights=NULL)
```

- `id`: the label unity, always asked; (id0 or id1 a faire tilde means no cluster)
- `strata`: stratification variable;
- `weights`: formula or vector of inclusion probabilities of size equal to sample size;
- `fpc`: formula or vector with the finite population correction (same size as `weights`); if `fpc` not specified, then the sample is with replacement;
- `data`: the sample data;
- `weights`, `fpc`, `data` are optional.

For example, in case of simple random sampling without replacement of size $n = 70$, the weights are equal to $554/70$ and we obtain:

```r
library(survey)
technique <- svydesign(id=~CODE_N, weights=rep(554/70,70), fpc=rep(70/554,70), data=rec99[which(si.rec==1),])
```

The sample data `ech.si` is an object and information, in particular inclusion probabilities and variable values, contained within may be obtained with `attributes` and `summary`:

```r
attributes(ech.si)
```

```r
$names
[1] "cluster"  "strata"  "has.strata"  "prob"
 "allprob"  "call"  "variables"
[8] "fpc"  "pps"
$class
[1] "survey.design2"  "survey.design"
```

```r
$allprob
```

```r
$variables
```

```r
summary(ech.si)
```

Independent Sampling design

```
svydesign(id = ~CODE_N, weights = rep(554/70, 70), fpc = rep(70/554, 70), data = rec99[which(si.rec == 1),])
```

Probabilities:
The finite population total is estimated with `svytotal` and the mean with `svymean`. Output of these functions give at the same time standard-deviation estimation:

```r
# the estimation of the total and standard-deviation for the SI sample
svytotal(~LOGVAC, ech.si)
```

<table>
<thead>
<tr>
<th>total</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>13613</td>
<td>2488.4</td>
</tr>
</tbody>
</table>

We give below the finite estimation total of `LOGVAC` with stratified and cluster samplings.

### stratified simple random sampling
#### without replacement (STSI)

```r
# stratified simple random sampling without replacement (STSI)
# creation of the stsi design object
N1=221;N2=169;N3=110;N4=54
n1=nech[1],n2=nech[2],n3=nech[3],n4=nech[4]
poids_stsi=c(rep(N1/n1,n1),rep(N2/n2,n2),rep(N3/n3,n3),rep(N4/n4,n4))
fpcst=c(rep(n1/N1,n1),rep(n2/N2,n2),rep(n3/N3,n3),rep(n4/N4,n4))
ech.stsi=svydesign(id=~CODE_N,strata=~stratlog,weights=poids_stsi,fpc~fpcst, data=stsi.data)
```

<table>
<thead>
<tr>
<th>total</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10512</td>
<td>3058.1</td>
</tr>
</tbody>
</table>

To compare different sampling designs, simulation must be performed but this is beyond the scope of this paper.
3.3 Estimation of non-linear interest parameters

If we want to estimate more complex parameters of interest \( \theta \), then most of the times \( \theta \) maybe obtained as the unique solution (explicite or implicite) of a population estimation equation:

\[
\mathcal{S}(\theta) = \sum_{k \in U} S_k(\theta) = 0.
\]

Coefficients of linear and logistic regression, ratio and calibration estimators maybe obtained in this way. For implicit parameters, the solution is approached by numerical algorithms. The estimator of \( \theta \) is \( \hat{\theta}_d \), the solution of

\[
\hat{\mathcal{S}}(\hat{\theta}_d) = \sum_{k \in s} S_k(\hat{\theta}_d) = 0,
\]

and, by Taylor linearization, the variance estimator is given by (Binder (1983)):

\[
\hat{\mathcal{V}}(\hat{\theta}_d) = A^{-1} \cdot \hat{\mathcal{V}}_{HT}(\hat{\mathcal{S}}(\theta)) \cdot (A')^{-1},
\]

where \( A = \hat{\mathcal{S}}(\hat{\theta}_d) \). Package \texttt{survey} implements this Taylor variance estimator for many non-linear estimators such as the ratio or calibration estimator (see also section 4). Design-based estimation of quantiles is treated by function \texttt{svyquantile} from \texttt{survey}. Multivariate quantile with survey data is proposed by \texttt{Gmedian}.

Packages \texttt{laeken} and \texttt{convey} are dedicated to the estimation of income inequality indicators such as Gini or Theil index.

4 Improving the Horvitz-Thompson estimator for the estimation of finite population totals

It is well-known that the Horvitz-Thompson estimator may be improved by using auxiliary information and the calibration, as suggested by Deville and Särndal (1992), is one of the most popular method used in many statistical institutes.

Consider that \( p \) auxiliary variables \( X_1, \ldots, X_p \) are available and let \( x_k = (X_{1k}, \ldots, X_{pk})' \) pour \( k \in U \). We suppose that the total of auxiliary variables are known, namely \( t_x = \sum_{k \in U} x_k \) is known.

The calibration method consists in finding new weights \( w_k^{cal} = (w_k^{cal})_{k \in s} \) such that they are as close as possible in the sense of a distance \( \Upsilon_s \) to the sampling designs \( d = (d_k)_{k \in s} : \)

\[
w_k^{cal} = \arg\min_d \Upsilon_s(w, d)
\]

and subject to the calibration constraints

\[
\sum_{k \in s} w_k^{cal} x_k = t_x.
\]

The calibrated estimator is \( \hat{t}_d^{cal} = \sum_{k \in s} w_k^{cal} d_k \). Several distance functions \( \Upsilon_s(w, d) \) have been considered in Deville and Särndal (1992) such as the chi-squared, raking, logit distances and they proved that the calibration estimator is asymptotically equivalent to the estimator obtained for the chi-squared distance, \( \Upsilon_s(w, d) = \sum_{k \in s} (w_k - d_k)^2/d_k \). For the chi-squared distance, the calibration weights are given by

\[
w_k^{cal} = d_k - d_k x_k' \left( \sum_{k \in s} d_k x_k x_k' \right)^{-1} (t_{xd} - t_x), \quad k \in s,
\]
where $\hat{t}_{xd} = \sum_{k \in s} d_k x_k$. The calibration estimator is

$$
\hat{t}_{y}^{\text{cal}} = \sum_{k \in s} w_{k}^{\text{cal}} y_k = \sum_{k \in s} d_k y_k - (\hat{t}_{xd} - t_x) \hat{\beta}_X;
$$

where $\hat{\beta}_X = (\sum_{k \in s} d_k x_k x_k' )^{-1} \sum_{k \in s} d_k x_k y_k$. The variance estimator of $\hat{t}_{y}^{\text{cal}}$ is given by

$$
\hat{V}(\hat{t}_{y}^{\text{cal}}) = \sum_{k \in s} \sum_{k \in s} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_k} y_k - \frac{x_k' \hat{\beta}_X y_k - x_k' \hat{\beta}_X}{\pi_k}.
$$

In R, several packages are concerned with calibration. Functions calib, gencalib in package sampling compute the calibration weights and the $g$-weights. Packages laeken (calibWeights function) and simPop (calibSample function) suggest faster implementations (depending on the example) of parts of calib. Function calibrate from package survey compute the calibration weights. Weights for computing the ratio estimator and respectively, the poststratified estimator are obtained with functions svyratio and poststratified. We mention also the very recent package icarus.

We focus now on the package survey. The calibration weights are obtained with the calibrate function:

```r
calibrate(design, formula, population, variance=NULL,
         bounds=c(-Inf,Inf), calfun=c("linear","raking","logit"),...)
```

- *design*: the survey design objet;
- *formula*: model formula for calibration model
- *population*: vectors of population totals
- *calfun*: distance functions used in calibration
- *bounds*: bounds for the calibration weights
- *variance*: if not NULL, then the calibration variance is proportional to the linear combination of those columns of the model matrix

We use this function in order to estimate the population total of LOGVAC by taking into account the auxiliary information given by the intercept and the variable LOG whose total is 197314:

```r
ech.cal<-calibrate(ech.si,~LOG,c(554,197314))
```

The value of this function is an object, we can extract the calibration weights with 1/ech.cal$prob and the values of the interest variable with ech.cal$variables. The calibration estimator for the estimation of the total of LOGVAC and its estimated standard-deviation (obtained by Taylor linearization) is obtained as follows:

```r
svytotal(~LOGVAC, ech.cal)
```

<table>
<thead>
<tr>
<th>total</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGVAC</td>
<td>9853.5</td>
</tr>
</tbody>
</table>

The ratio estimator may be obtained with the function svyratio. Since the ratio estimator is a particular case of the calibration estimator, it can be also obtained with calibrate by excluding the intercept and variance proportional to the single auxiliary variable (here the variable LOG) with the option variance=1.
echo.cal2<-calibrate(echo.si,"LOG-1,197314, variance=1")
svytotal("LOGVAC,echo.cal2")

total SE
LOGVAC 9461.9 882.2

The very recent package Icarus, created by Antoine Rebecq (ex Insee, now Ubisoft, Montreal), is dedicated only to calibration method. It was implemented for users from the INSEE and this is why, it is inspired from the very popular macro SAS CALMAR ("calage sur marges"). Nonresponse can be handled and also penalized calibration, but the variance estimation is not provided, like CALMAR. The estimated total or mean with the so obtained calibration weights may be obtained with functions weightedTotal or weightedMean. A detailed description (in French) of this package is given in Rebecq (2016).

The calibration function is:
calibration(data, marginMatrix, colWeights, method = "linear", bounds = NULL, ...)  
• data: survey data;  
• marginMatrix: population totals of auxiliary variables;  
• colWeights: the sampling weights;  
• method: the method used to calibrate ("linear", "raking", "logit", "truncated");  
• bounds: vector of lower and upper bounds for the calibration weights;

We use this function to estimate again the population total of LOGVAC by taking into account the auxiliary information given by the intercept and the variable LOG.

#####population totals
m_1=c("taillepop",0,554)
m_LOG=c("LOG",0,197314)
marges_rec=rbind(m_1,m_LOG)

########################
poids=rep(554/70,70)
taillepop=rep(1,70)

#####???of the intercept
essai=cbind(rec99[which(si.rec==1), ], taillepop, poids)

#####calibration weights
poids_cal=calibration(data=essai,marginMatrix=marges_rec,colWeights="poids", bounds=c(0.4, 2.1), description=TRUE)

#####calibration estimation of the total of LOGVAC
logvac_cal=weightedTotal(essai$LOGVAC,poids_cal)
#[1] 9853.461

We remark that we obtain the same estimation as with the calibrate function from the survey package.
4.1 Estimation with over-calibration

Nowadays, we are at the age of big-data and large datasets may be recorded thanks to connected smart objects such as smartphones, computers and smart meters. In this situation, a very large number of auxiliary variables may be obtained.

Performing calibration with a very large number of auxiliary information, called also over-calibration by Guggemos and Tillé (2010), can result in negative, very large and unstable calibration weights $w_k$. Moreover, in this situation, the predefined benchmarks on the weight ratio $w_k/d_k$ may not be satisfied if for example, the chi-square distance is used. With many auxiliary variables, collinearity problems may arise and the matrix $\sum_{k\in s} d_k x_k^T x_k$ may not be invertible and so, the calibration weights cannot be computed. And lastly, it can be remarked that the variance of the calibration estimator increases (Silva and Skinner (1997), Cardot et al. (2017), Chauvet and Goga (2017)).

Several solutions have been suggested to overcome these drawbacks. We may choose the most important variables but with multipurpose surveys, this choice may be difficult to put into practice. We may use instead all the variables and perform penalized calibration via the ridge regression (Bardsley and Chambers (1984); Chambers (1996); Rao and Singh (2009); Guggemos and Tillé (2010); Beaumont and Bocci (2008)). Very recently, Cardot et al. (2017) suggested compressing the information contained in the auxiliary variables by performing principal component calibration.

4.2 Penalized calibration

The calibration equations are “released”, namely they are only approximatively satisfied and the error between $\hat{t}_w$ and $t_x$ is controlled by means of a penalty. More exactly, we search for the weights $w^{pen}_k(\lambda) = (w^{pen}_k(\lambda))_{k\in s}$ satisfying the following penalized optimization problem:

$$w^{pen}_k(\lambda) = \arg\min_w \sum_{k\in s} (w_k - d_k)^2 + \lambda \sum_{j=1}^p C_j (t_{w, X_j} - t_{X_j})^2,$$

where $C_j, j = 1, \ldots, p$, is a user-specified cost associated with not satisfying the $j$th calibration equation. If $\lambda C_j = 0$, then the constrain upon $t_{X_j}$ is not considered and if $\lambda C_j$ is very large, then the constraint upon $t_{X_j}$ is exactly satisfied.

The penalized weights are given by:

$$w^{pen}_k(\lambda) = d_k - d_k X_k \left( \sum_{k \in s} d_k x_k x_k^T + \lambda^{-1} C^{-1} \right)^{-1} (\hat{t}_{w} - t_x), \quad k \in s$$

and the penalized calibration estimator is a GREG-type estimator with a ridge coefficient of regression:

$$\hat{t}^{pen}_{yw}(\lambda) = \sum_{k \in s} w^{pen}_k(\lambda) y_k = \sum_{k \in s} d_k y_k - \left( \sum_{k \in s} d_k x_k - \sum_{k \in U} x_k \right)^T \hat{\beta}_X(\lambda),$$

where $\hat{\beta}_X(\lambda) = \left( \sum_{k \in s} d_k x_k x_k^T + \lambda^{-1} C^{-1} \right)^{-1} \sum_{k \in s} d_k x_k y_k$. The variance estimator of $\hat{t}^{pen}_{yw}(\lambda)$ is

$$\hat{V}(\hat{t}^{pen}_{yw}(\lambda)) = \sum_{k \in s} \sum_{k' \in s} \frac{\pi_{kl} - \pi_k \pi_l}{\pi_{kl}} y_k - x_k' \hat{\beta}_X(\lambda) y_l - x_l' \hat{\beta}_X(\lambda) \frac{\pi_{kl}}{\pi_l}.$$
The method has already been used at Statistics Canada and at INSEE. We can perform penalized calibration by using the package icarus and function calibration. We give below again this function with the options related to the costs $C_j$ and the tuning parameter $\lambda$ needed for performing penalized calibration:

\[
\text{calibration}(\text{data, marginMatrix, colWeights, method = "linear",}
\text{ bounds = NULL, q = NULL, costs = NULL, gap = NULL, lambda = NULL, ...})
\]

- **costs**: vector of $C_j$ costs, they must be given;
- **gap**: $\max(w_k/d_k) - \min(w_k/d_k)$
- **lambda**: the tuning parameter $\lambda$ used in penalized calibration; by default, chosen automatically by the bisection algorithm.

No need to consider bounded distances this is why only the chi-square and the raking distance are considered. Different costs may be used, a very large $C_j$ meaning that the constraint must be exactly satisfied.

### 4.3 Calibration on principal components

The alternative to penalized calibration is to “compress” the information contained in the $X$-matrix by considering principal components analysis. Consider the principal components $Z_1, \ldots, Z_p$,

\[
Z_j = Xv_j, \quad j = 1, \ldots, p.
\]

where $v_j$ is the $j$th eigenvector associated to the $j$th eigenvalue $\lambda_j$ of $N^{-1}X'X$. We search for weights $w_{pc} = (w_{ks})_{k \in s}$ satisfying:

\[
w_{pc} = \arg\min_{w} \sum_{k \in s} \frac{(w_k - d_k)^2}{d_k}
\]

subject to

\[
i_w Z_j = tZ_j \quad \text{for} \quad j = 1, \ldots, r \quad \text{with} \quad r << p,
\]

where $tZ_j$ is the total of the $j$th principal component $Z_j$. The principal component calibration estimator is a GREG-type estimator with a principal component coefficient of regression:

\[
\hat{\beta}_{X}^{pc}(r) = \left( \sum_{k \in s} d_k x_k \right)^{-1} \sum_{k \in s} d_k x_k y_k.
\]

The variance estimator of $\hat{\beta}_{X}^{pc}(r)$ is

\[
AV(\hat{\beta}_{X}^{pc}(r)) = \sum_{k \in s} \sum_{k' \in s} \frac{\pi_{kl} - \pi_{k'} \pi_{l}}{\pi_{kl}} y_k - x_k \hat{\beta}_{X}^{pc}(r) y_l - x_l \hat{\beta}_{X}^{pc}(r).
\]

If no complete auxiliary information is available, then the PC variables $Z_j$ may be estimated by $\hat{Z}_j = X\hat{v}_j$ and consider calibration on the zero-totals of $\hat{Z}_j$ (Cardot, Goga and Shehzad, 2017).

With R, function `svyprcomp` of package `survey` performs principal components analysis with survey data and allows obtaining the principal component estimates when data is not complete:
svyprcomp(formula, design, center = TRUE, scale = FALSE, scores = FALSE, ...)

- **formula**: model formula describing variables to be used
- **design**: the survey design object
- **center = TRUE**: data is centered by default
- **scale = FALSE**: data is not scaled by default
- **scores = FALSE**: scores on each component needed for biplot are not returned.

The value is an object similar to `prcomp` but with survey design information. Using `svyprcomp`, we can compute the estimated $\hat{Z}_j$ which can be used next as calibration variables in `calibrate` or `calibration`.

5 Conclusion

We presented a brief overview of the most used packages dedicated to survey sampling techniques with implementation on a real dataset. Nonresponse is often in survey sampling and many packages on missing data exist but most of them do not take into account sampling weights. There are still many fields in survey sampling theory which are not implemented in R such as nonparametric estimation with survey data, robust estimation for totals, indirect sampling.

REFERENCES

BayesRandomForest: An R implementation of Bayesian Random Forest for Regression Analysis of High-dimensional Data

Oyebayo Ridwan Olaniran (rid4stat@yahoo.com)
Universiti Tun Hussein Onn Malaysia

Mohd Asrul Affendi Bin Abdullah (afendi@uthm.edu.my)
Universiti Tun Hussein Onn Malaysia

ABSTRACT

Random Forest (RF) is a popular method for regression analysis of low or high-dimensional data. RF is often used with the later because it relaxes dimensionality assumption. RF major weakness lies in the fact that it is not governed by a statistical model, hence probabilistic interpretation of its prediction is not possible. RF major strengths are distribution free property and wide applicability to most real life problems. Bayesian Additive Regression Trees (BART) implemented in R via package BayesTree or bartMachine offers a bayesian interpretation to random forest but it suffers from high computational time as well as low efficiency when compared to RF in some specific situation. In this paper, we propose a new probabilistic interpretation to random forest called Bayesian Random Forest (BRF) for regression analysis of high-dimensional data. In addition, we present BRF implementation in R called BayesRandomForest. We also demonstrate the applicability of BRF using simulated dataset of varying dimensions. Results from the simulation experiment shows that BRF has improved efficiency over its competitors.

Keywords: Random Forest, Bayesian Additive Regression Trees, High-dimensional, R

JEL Classification: C11, C39

1 INTRODUCTION

Ensemble of trees based methods have become popular choices for predicting qualitative response (classification) and quantitative response (regression) [1]. Ensemble methods train multiple learners to construct one learner from training data. Random Forest (RF, [2]) and Gradient Boosting Machine (GBM, [3]) are the two well established ensemble based method. The two methods focused on improving the unstable prediction problem in Classification and Regression Trees (CART, [2]). Apart from RF and GBM,
other recently developed ensemble based tree methods are Bayesian Additive Regression Trees (BART, [4]), dynamic trees [5], Bayesian Forest and Empirical Bayesian Forest (BF; EBF, [6]) and Bayesian Additive Regression Trees using Bayesian Model Averaging (BART-BMA, [7]). [7] conducted a simulation study using [8] data originally motivated by Multivariate Adaptive Regression Splines (MARS) to compare the predictive performance of RF, BART and BART-BMA. The simulation results revealed that BART is only better than RF in terms of computation time.

RF procedure requires selection of bootstrapped sample of training data \((n \times p)\) and subsample of feature set \(p\) to create splits used in the split selection stage. The number of subsampled feature is often held fixed as \(\approx \sqrt{p}\) for classification or \(\approx p/3\) for regression. This subsample size does not take into account the number of relevant features in the entire feature set, thus the chance of selecting relevant features increases with increased \(p\). Therefore, using same data configuration, the predictive performance of RF reduces with increasing \(p\). However, if the feature space is well populated with relevant features, RF predictive performance will spontaneously increase due to increase in the hypergeometric probability.

In this paper, we extend RF by introducing Bayesian weighted sampling and splitting and called it Bayesian Random Forest (BRF). The weighted splitting is achieved via Bayesian inference of the two sampling procedures involved in RF. We developed a fully Bayesian ensemble of tree procedure that is similar in spirit to RF. We also implemented the procedure in an ongoing R package “BayesRandomForest”.

2 OVERVIEW OF BAYESIAN RANDOM FOREST

Bayesian Random Forest (BRF) is a Bayesian implementation of the nonparametric function estimates obtainable from regression trees. Regression trees rely on recursive binary partitioning of predictor space into a set of hyper rectangles in order to approximate some unknown function \(f\) [1]. The main advantages of Tree-based regression models are attributed to their success in modelling of linear, non-linear and interaction effects. The main weakness of single tree method is the instability in prediction due to the top down approach involved in recursive partitioning. This suggests the need to average many trees to reduce variance. BRF is an adaptive version of RF but guided with Bayesian reasoning. That’s every stage of estimation in RF is mimicked but estimation fully relies on Bayesian paradigm.

Formally, given training dataset \([Y, x_{i1}, x_{i2}, \ldots, x_{ip}, i = 1,2,\ldots,n]\), where \(Y\) assumes continuous values and \(x_i\) is the vector of features, BRF can be describe as:
where $\hat{\beta}_m$ is an estimate of $Y$ in region $m$, $I$ is the number of trees in the forest, $\mathcal{S}_i(\beta_m; x \in R_m)$ is a single regression tree, $\epsilon$ is the random noise that occurs in estimating $\hat{\beta}_m$ and its assumed to be independent and identically Gaussian distributed with mean zero and variance $\sigma^2$ over all trees. BRF model is very much similar to BART [4] but differs in terms of prior specification and posterior estimation approach.

### 2.1 Priors and Posterior Specification for Bayesian Random Forest

BRF has three major parameters which can be attributed to its ensemble nature. The first parameter is attributed to model uncertainty, which in this case is tree uncertainty. Here we propose a uniform prior $\mathcal{S}_0 \sim U(0, 1)$ such that $P(\mathcal{S}_0) = 1$ for any candidate tree. We used this prior specification to retain the average weighing procedure of RF so that each tree $\mathcal{S}$ has equal right. The advantage of this prior is to retain the strength of RF in terms of correcting overfitting problem. The second form of prior is prior on terminal node parameter $\beta_m$, here we propose Gaussian prior $N(\mu, \sigma^2)$. We adapted the bootstrap prior technique [9 - 11] to obtain the prior hyperparameters $\mu$ and $\sigma^2$ for each tree. The major advantage of bootstrap prior is that it guarantees an unbiased estimate of $\beta_m$ for each tree. For the parameter $\sigma^2$, we propose the standard gamma default prior $G(\alpha, \theta)$ [12] with $\alpha = \theta$ such that $E[\sigma^2|G(\alpha, \theta)] = 1$. The complete prior specification for BRF is thus:

$$
P(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m) = \prod_{j=1}^{I} P(\mathcal{S}_j, \beta_m) P(\sigma^2) 
$$

(2)

$$
P(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m) = \prod_{j=1}^{I} P(\beta_m) P(\sigma^2) 
$$

(3) follows from (2) since $P(\mathcal{S}_0) = 1$. The posterior distribution using $L(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m|y, x)$ and (3) is then obtain via Metropolis Hasting (MCMC) algorithm [13].

Now to mimic RF completely, we also specified some procedural priors similar in spirit to bootstrapping and features subsampling in RF. For the two procedures, we proposed Bayesian simple random sampling with replacement and Bayesian simple random sampling without replacement with posterior densities given in (4) and (5):
where \( \pi \) is the probability of selecting any \( i \in \mathcal{N} \) in each \( j \) step, \( \Gamma(d) \) is the gamma function evaluated at \( d \), \( \alpha \) is the prior expected number of times any \( i \in \mathcal{N} \) could be selected, \( b \) is its complement, \( V \) is the number of relevant features whose posterior is sought, \( h \) is the sample realization of relevant features, \( p \) is the total number of features, \( l \) is the number of subsampled features as in RF, \( S \) is the prior number of relevant features and \( T \) is the prior number of irrelevant features.

If we denote the posterior density in (4) and (5) as \( \omega \) and \( \delta \), we then obtain a weighted Bayesian regression tree at each \( j \) step by weighing the data by \( \omega \) and then weighing the impurity at each split by \( \delta \). For a Sum Squares Error (SSE) impurity, we propose a weighted impurity using:

\[
\text{SSE}(\delta) = (1 - \delta) \left[ \sum_{i=1}^{m} (y_i - \hat{\beta}_m)^2 \right]
\]

where \( \hat{\beta}_m \) is the posterior mean of \( \beta_m \) at each node \( m \). The variable with weight \( \delta \to 1 \), will correspond to variable with minimal unweighted \( \text{SSE}(\delta) \) and therefore useful for further splitting step. If on the other hand \( \delta \to 0 \), implies the variable is not useful and therefore expected to yield a maximal unweighted \( \text{SSE}(\delta) \). In this case, the proposed weighted \( \text{SSE}(\delta) \) returns the unweighted \( \text{SSE}(\delta) \) so that the variable is dropped at the splitting stage. The idea behind this is to control the mixture behavior of hypergeometric distribution. The dominant category determines the estimates of categories probability. RF fails to balance this gap by specifying \( l = \sqrt{p} \); for example if \( p = 10000 \); \( l = 100 \), which implies taking a random sample of 100 features to be used in each split. Suppose there are 5 relevant features as in Friedman (1991), the hypergeometric probability of selecting any relevant features is approximately 0.049. This implies that at each splitting step, there is about 95% chance of selecting irrelevant feature. This high probability can be attributed to fewer number of relevant features in relation to large number of features \( p \). Thus RF assumes that the entire feature space \( p \) is reasonably populated with relevant features. The dilemma with RF is to think of increasing \( l \), yes this will increase the hypergeometric probability but at the expense of increasing the correlation between trees. This is indeed the situation where RF irretrievably breaks down [14].
3 FRIEDMAN FIVE DIMENSIONAL DATA

Following (4, 6), [8] simulated data was used to compare the results of BRF, BF, RF, BART and GBM. The simulated datasets where \( x_1, \ldots, x_p \) are \( \text{iid} \sim U(0.1) \) random variables and \( \varepsilon \sim N(0.1) \) were formulated as;

\[
y = 10\sin(x_1 x_2) + 20(x_2 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon
\]

The five methods were compared over various dataset sizes with five relevant features \( [x_1, \ldots, x_5] \) and complement of \( p = [100, 500, 1000, 5000, 10000] \) as the irrelevant features. The associated hypergeometric probabilities with \( p \) are \( P(V|l = \sqrt{p}) = 0.416, 0.202, 0.150, 0.060, 0.050 \). The predictive performance of the methods was assessed with 10 folds cross validation of Root Mean Squared Error (RMSE) at sample size \( n = 50, 100 \). All analyses were carried out using R with newly developed function “BayesRandomForest” accessible in [15] for BRF, “bartMachine” [1] for BART, “gbm” [16] for GBM, “randomForest” [17] for RF and “ranger” [18] for BF.

Table1 shows the 95% credible interval for Bayesian based methods (BRF, BF, BART) and confidence interval for frequentist based methods (GBM and RF) of RMSE. The intervals are computed using the bake-off 10 folds cross-validation of \( 10 \times n \). The results show that 95% width for BRF is the lowest when compared to other methods. The least performing method is BART with the maximum width. The next in category in terms of stability of RMSE is RF. The poor performance of BART with increasing \( p \) in Fig1. and Fig2. is attributed to the nature of tree growth which involves using all features for each tree.

<table>
<thead>
<tr>
<th>Method</th>
<th>( n = 50 )</th>
<th>( n = 100 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.5%</td>
<td>97.5%</td>
</tr>
<tr>
<td>BRF</td>
<td>15.32</td>
<td>25.26</td>
</tr>
<tr>
<td>BF</td>
<td>28.50</td>
<td>33.03</td>
</tr>
<tr>
<td>RF</td>
<td>24.77</td>
<td>31.44</td>
</tr>
<tr>
<td>BART</td>
<td>23.18</td>
<td>32.53</td>
</tr>
<tr>
<td>GBM</td>
<td>19.92</td>
<td>36.70</td>
</tr>
</tbody>
</table>

This lead to overfitting and eventual poor performance in out of sample validation. Furthermore, the existing robust method to large \( p \) with low relevant feature is GBM because of its internally embedded feature
selection procedure [15]. BRF results challenged this claim with stable and better results at sample size $n = 50, 100$.

Figure 1

RMSE with increasing $p$ at sample size $n = 50$, as expected increasing $p$ reduces the hypergeometric probability which increases RMSE of RF (purple dotted lines). Similarly, GBM, BRF and BART are affected with the increase but the effect is minimal on BRF. BF tends to be stable over but the RMSE is on the high side.

Figure 2
RMSE with increasing $p$ at sample size $n = 100$, as expected increasing $p$ reduces the hypergeometric probability which increases RMSE of RF (*purple dotted lines*). The increase in sample size tends to balance the methods RMSE but BART, GBM, BF and RF are still affected with the increase in $p$.

## 4 CONCLUSION

In this paper, we presented the theoretical framework for Bayesian Random Forest (BRF) for regression analysis of high-dimensional data. By way of example, We consider its application on simulated Friedman data set with large $P$ and fewer number of relevant features. We also compared the predictive performance of the method with some existing methods using RMSE via 10 folds cross-validation. The results observed from the simulation study shows that BRF is highly robust to large $p$ small relevant feature issue at a reasonable sample size $n$ when compared with its competitors.

### Funding

This work was supported by Universiti Tun Hussein Onn, Malaysia [grant numbers Vot, U607].

### References


Bayesian Method with Clustering Algorithm for Credit Card Transaction Fraud Detection

Luis Jose S. Santos (ljsxnx@gmail.com)
De La Salle University, Philippines

Shirlee R. Ocampo (shirlee.ocampo@dlsu.edu.ph)
De La Salle University, Manila, Philippines

ABSTRACT
Card transaction fraud is prevalent anywhere. Even with current preventive measures like the Europay, MasterCard and Visa (EMV) chips, possible weaknesses or loopholes can be exploited by fraudsters. This paper explores Naïve Bayes Classifier and clustering algorithms to detect fraud in credit card transactions. Data on fraud labels and arrival times of transactions were simulated by the Markov Modulated Poisson Process. Amounts of transactions for genuine and fraud transactions were simulated based on two Gaussian distributions. Kinds of spenders and types of fraudsters serve as the bases for the parameters used in the simulation of the data. Using the simulated data, EM clustering algorithm with three different initializations and K-means were applied to cluster transaction amounts into high, medium and low. The Naïve Bayes classifier algorithm was then applied to classify the transactions as good or fraud for the simulated data of 9 types of fraudsters across all clustering algorithms. Simulations and analyses were done using R software. Results include comparisons of true positive rates, false positive rates, and detection accuracies among the nine types of fraudsters across all clustering algorithms. For 3 clusters, (high, medium, low transaction amounts), the Naïve Bayes Method with clustering algorithms resulted to an average of 76% true positive (TP) detection, 18% false positive (FP) detection, with an overall accuracy of 81%. The same averages of TP, FP, and overall accuracy were obtained using 2 clusters (high, and low). EM clustering algorithm generated TP, FP, and overall accuracy of 80%, 16%, and 83% respectively.

Keywords: credit card transaction, fraud detection, Naïve Bayes Classifier, clustering algorithms, Markov Modulated Poisson Process data simulation

JEL Classification: C11, C39
1. INTRODUCTION

E-commerce is basically any form of transaction between two parties electronically without the need for physical contact. A popular medium of payment for such is the use of credit cards. The steady growth of the use of credit cards looks favorable, but there are possible adverse effects when it comes to the security of such cards. Although credit cards possess numerous measures of security to prevent fraudulent use, they are not completely immune to attacks. These attacks can be categorized as either a physical attack on the card or card-holder, or a virtual attack such as phishing. The numerous ways of committing fraud are alarming because large amounts of money were lost by banks. According to Panigrahi, et al (2009), card fraud is still an ongoing problem despite all the security measures available for each card. Fraudsters are adaptive, and given time, will eventually be able to avoid further security, and hence, having a detective measure for fraud is highly recommended to bypass financial losses.

Methods for card fraud detection are mostly based on neural networks, machine learning, and data-mining. For instance, a detection method based on Bayesian Networks (BN) was proposed by Mukhanov (2008) using Naïve Bayesian Classifier. The Naïve Bayes is a form of BN which uses conditional independence of attributes which is better in terms of predictive power as compared to constraint-based and score-based algorithms which focuses on the structure of the Bayesian Network.

A fusion approach including Bayesian learning was proposed as a novel approach to fraud detection (Panigrahi, et al, 2009). This recently devised method uses Dempster-Shafer theory in conjunction with the Bayesian Learner for determining whether a new transaction is genuine or not. The data used was simulated, and the results produced more accurate detections of fraud. Hidden Markov Model (HMM) was used by Srivastava et al (2005). Using only transaction amounts, this method produced accuracy of over 80%.

This paper explores Naïve Bayes Classifier and clustering algorithms to detect fraud in credit card transactions. It aims to introduce a simple fraud detection system that can be used by credit card companies as an initial measure of fraud detection.

Within the fraud detection system (FDS), the objectives include the comparison of the Expectation-Maximization (EM) and K-means clustering algorithms. K-means clustering has been the common choice in profiling the spending behavior of credit card holders. In this study, the EM Algorithm is explored because of its status as model-based clustering algorithm.
2. DATA AND METHODOLOGY

2.1 Data Simulation

Data simulation procedure was adapted from Panigrahi et al (2009). Data simulation is composed of three components: Markov Modulated Poisson Process (MMPP), Genuine Gaussian Distribution Module (GGDM), and Fraud Gaussian Distribution Module (FGDM).

MMPP is a Poisson arrival process that has its parameter $\lambda$ controlled by an underlying Markov process. It has two states, genuine (g) and fraud (f), with transaction arrival times $\lambda_g$ and $\lambda_f$ respectively. Amounts of transactions for genuine and fraud transactions were simulated based on two Gaussian distributions. GGDM was used to generate transaction amounts for different genuine customers by varying mean $\mu$ and standard deviation $\sigma$. This is a Gaussian process with mean $\mu_g$ and standard deviation $\sigma_g$. FDGM was used to generate transaction amounts for fraudsters. This Gaussian process can simulate different categories of fraudsters by varying mean $\mu_f$ and standard deviation $\sigma_f$. The basic flow of this algorithm is shown in Figure 1. All genuine transactions are initially recorded under a single database but will be classified later based on their fraud labels.

![Data Simulation](Figure 1)

(Panigrahi, et al, 2009)
Kinds of spenders and types of fraudsters serve as the bases for the simulation of different sets of data using R software. Srivastava, et al (2005) generalized spenders into the usual range of their spending i.e. high spender, low spender. For fraudsters, expert knowledge suggests two kinds of fraudsters. The first kind maximizes their benefit by making several low value purchases with common items. The second kind would do the opposite by making one high value purchase, perhaps with electronics. Although cited in Panigrahi et al (2009), the simulator settings were not described specifically. However, these settings produced varied results among each other. Table 1 summarizes these settings and the parameters are defined accordingly: $\lambda_g$ and $\lambda_f$ are Poisson rates for genuine and fraudulent distributions respectively; and $q_{gf}$ and $q_{fg}$ are the transition probabilities from good to fraud and fraud to good respectively. The values of these parameters are to be inputted for the MMPP, implemented using the *HiddenMarkov* R package to simulate data. The means of the two Gaussian distributions, denoted by $\mu_g$ and $\mu_f$, are to be used following the MMPP, and are defined to be percentages of an unknown credit limit.

Data Simulator Parameter Settings

<table>
<thead>
<tr>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_g$</td>
</tr>
<tr>
<td>SS1</td>
</tr>
<tr>
<td>SS2</td>
</tr>
<tr>
<td>SS3</td>
</tr>
<tr>
<td>SS4</td>
</tr>
<tr>
<td>SS5</td>
</tr>
<tr>
<td>SS6</td>
</tr>
<tr>
<td>SS7</td>
</tr>
<tr>
<td>SS8</td>
</tr>
<tr>
<td>SS9</td>
</tr>
</tbody>
</table>

(Panigrahi, et al, 2009)

2.2 Clustering Algorithms

Using the simulated data, Expectation-Maximization (EM) with three initializations and K-means algorithms were applied to cluster transaction amounts into three clusters (high, medium and low) in comparison with two clusters (high and low).

2.2.1. Expectation-Maximization Clustering

Expectation-Maximization (EM) clustering is a type of probabilistic model-based clustering algorithm as described below.
Expectation (E-Step): Construct a conditional distribution $Q_i$ of $z_i$ over $x_i$, that is,

$$Q_i = p(z_i \mid x_i; \theta) \quad (1)$$

where $x_i$ are $m$ observed variables from a data set with $i = 1, 2, 3, \ldots, m'$, $z_i$ are $m$ latent random variables obtained from the observed variables $x_i$, and $\theta$ is the parameter for the conditional distribution $Q_i$.

Maximization (M-Step): Maximize the expression from the E-Step with respect to its parameters, that is,

$$\theta = \arg \max_\theta \sum_i \sum_{z_i} Q_i \log \frac{p(x_i, z_i; \theta)}{Q_i} \quad (2)$$

Repeat E and M step until convergence.

Used as a clustering algorithm, EM assumes a finite mixture of Gaussian distributions. This assumption is also used in the \textit{EMCluster} R package used in this research. In this case, $Q_i$ is of the form

$$f(x \mid \theta) = \sum_{k=1}^{K} \pi_k \cdot \varphi(x \mid \mu_k, \Sigma_k) \quad (3)$$

where $x$ is a $p$-dimensional observation,

$$\theta = \{\pi_1, \pi_2, \ldots, \pi_{K-1}, \mu_1, \mu_2, \ldots, \mu_K, \Sigma_1, \Sigma_2, \ldots, \Sigma_K\}.$$ 

$\pi_1, \pi_2, \ldots, \pi_K$ are the mixing proportions for the $K$ clusters such that $\sum_{k=1}^{K} \pi_k = 1$, and $0 < \pi_k < 1$ for $k = 1, 2, \ldots, K$, and

$$\varphi(x \mid \mu_k, \Sigma_k)$$

are multivariate Gaussian distributions with mean vector $\mu_k$ and variance-covariance matrix $\Sigma_k$.

The data is then partitioned into $K$ clusters by the maximum posterior

$$\arg \max_k \hat{p}_k \cdot \varphi(x_i \mid \hat{\mu}_k, \hat{\Sigma}_k) \quad (4)$$

Although the EM clustering algorithm is efficient, its effectiveness is largely dependent on its starting point; especially at a high number of clusters (Hu, 2015). This paper explored 3 initialization methods available in the R program, namely, \texttt{RndEM}, \texttt{emEM}, and \texttt{svd}.

The three initializations found in the \textit{EMCluster} package guide of the R program are briefly described as follows. The first initialization, \texttt{RndEM}, selects $K$ centers from the data, and the other data are grouped to the closest
center based on Euclidean distance. The initial with the highest log likelihood is chosen as the initial value for the EM algorithm until convergence. The second initialization, \( emEM \), also starts by selecting \( K \) centers, but this time directly applying a shorter version of the EM algorithm. Once the algorithm converges to a specified value, that value is used as the start of the regular EM algorithm. The third initialization singular value decomposition \( svd \) selects centers from a major component space. The data are then grouped around the centers by K-means, which also generates the initial. This initial is then used for the EM algorithm.

### 2.2.2. K-means Clustering

K-means clustering is a type of partition-based clustering, which aims to partition the points into \( k \) groups such that the sum of squares from points to the assigned cluster centers are minimized. The Within Cluster Sum of Squares (WCSS) used in \( kmeans() \) function in R is given by

\[
WCSS = \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \overline{x_{kj}})^2
\]

where \( x_{ij} \) is the \( ij \)th object,
\( n \) is the number of objects, \( i = 1, 2, \ldots, n \),
\( p \) is the number of variables from \( j = 1, 2, \ldots, p \), and
\( \overline{x_{kj}} \) is the mean variable \( j \) of all elements in group \( k \).

The algorithm of this K-means clustering is as follows:
1. Allocate objects to clusters with the nearest cluster mean.
2. Search for the \( K \)-partition with locally optimal WCSS by moving objects from one cluster to another.
3. Repeat step 2 until convergence.

Each clustering algorithm was scored using the silhouette coefficient (SC) given by

\[
s = \frac{b - a}{\max(a, b)}
\]

where \( a \) is a mean distance between a sample and all other points in the same class, and
\( b \) is a mean distance between a sample and all other points in the next nearest cluster.
High values of SC (closer to 1) indicate highly dense clustering. Values less than zero would indicate either overlapping clusters or incorrect clustering. All outcomes of these clustering algorithms were inputted to Naïve Bayes algorithms.

2.3 Naïve Bayes Classifier

The Naïve Bayes classifier is a set of learning algorithms based on applying the Bayes’ rule given by

\[
P(B_r|A) = \frac{P(B_r \cap A)}{P(A)}
\]

(7)

\[
= \frac{P(B_r)P(A|B_r)}{P(A)}
\]

(8)

where \(A\) and \(B_i\) are events from a sample space, \(i = 1, 2, ..., r, ..., k\) and \(P(A) = \sum_{i=1}^{k} P(B_i)P(A|B_i)\), following the rule of total probability.

The Naïve Bayes' Classifier has the ‘naïve’ assumption of independence between pair of attributes and can be derived as follows:

Let \(A\) be any event, and \(B_i\) independent attributes of event \(A\). Construct a conditional likelihood table as shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>(B_1)</th>
<th>(B_2)</th>
<th>...</th>
<th>(B_i)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>(P(B_1</td>
<td>A))</td>
<td>(P(\neg B_1</td>
<td>A))</td>
<td>(P(B_2</td>
</tr>
<tr>
<td>(\neg A)</td>
<td>(P(B_1</td>
<td>\neg A))</td>
<td>(P(\neg B_1</td>
<td>\neg A))</td>
<td>(P(B_2</td>
</tr>
</tbody>
</table>

Consider computing for \(P(A|B_1 \cap \neg B_2 \cap \neg B_3 \cap B_i)\). Applying Bayes’ rule from eqn [5],

\[
P(A|B_1 \cap \neg B_2 \cap \neg B_3 \cap B_i) = \frac{P(B_1 \cap \neg B_2 \cap \neg B_3 \cap B_i|A)P(A)}{P(B_1 \cap \neg B_2 \cap \neg B_3 \cap B_i)}
\]

(9)

This becomes more and more computationally difficult as the number of \(B_i\)’s increase. Therefore, the naïve independence assumption of Naïve Bayes is applied resulting to
This can be generalized into

$$P(A_l | B_1, ..., B_N) = \frac{1}{Z} P(A_l) \prod_{i=1}^{n} P(B_i | A_l)$$  \hspace{1cm} (12)$$

where $A_l$ is a class $A$ at level $L$, $B_i$, are the $N$ features of the classes, $i = 1, 2, ..., N$, and $Z = \prod_{i=1}^{n} P(B_i)$ is the scaling factor.

The distribution of $P(B_i | A_l)$ is not specified, and as such can take on different distributions. For this paper, $P(B_i | A_l)$ assumes a multinomial distribution since the features of the Naïve Bayes Classifier will have two levels (for the dummy binary variables, and clustering for $K = 2$), and three levels (for clustering $K = 3$. This incarnation of the Naïve Bayes classifier is called the Multinomial Naïve Bayes.

Note that the conditional likelihood table is in the form of a probability mass function (PMF), and the expression $\prod_{i=1}^{n} P(B_i | A_l)$ from eqn [11], the product of all conditional probabilities dependent on $A_l$, can be treated as a likelihood function. Likewise, the expression $\frac{1}{Z} P(A_l)$ also from eqn [11] can be treated as a prior probability. Multiplying those two expressions results to a posterior probability following the Bayesian Method.

**2.4. Proposed Fraud Detection System**

The typical flow of transaction fraud detection systems (FDS) starts with the classification of transaction amounts. This is to acquire the spending behavior of the credit card holder. After which, the data acquired from the clustering is incorporated into whatever model is proposed in order to detect fraud. The proposed FDS is derived from Srivastava et al (2008), Mukahnov (2008), and Panigrahi et al (2009). It is described as follows.

1. Record past transactions classified separately as genuine or fraud along with transaction amounts ($m$), and arrival time of transaction ($a$). This is called the initial transaction history.
2. Classify transactions separately into either genuine or fraud.
3. Compute the interarrival times ($i$) from the arrival times for all genuine transactions using $i = a_r - a_{r-1}$
4. Create four dummy binary variables ($T_1, T_2, T_3, T_4$) based on the following rules:
5. Cluster transaction amounts \( m \) into \( K \) clusters \( (K = 2 \text{ or } K = 3) \).
6. Repeat steps 3 to 5 for fraudulent transactions.
7. Include the created variables into the initial transaction history and call this the secondary transaction history.
8. Apply the Naïve Bayes algorithm to the secondary transaction history to classify new transactions as either genuine or fraudulent.

The created dummy variables \( T_{ij} \) and a set of clustered transaction amounts \( m_i \) were used along with the fraud labels to train the Naïve Bayes, implemented by the \textit{bnlearn} package. Note that \( T_{ij} \) are the created dummy binary variables in the proposed FDS, \( i = 1, 2, 3, 4, j = 0, 1 \), and \( m_i \) refers to the clustered transaction amounts, \( i = 1, 2, 3 \) for 3 clusters, \( i = 1, 2 \) for 2 clusters. Naïve Bayes conditional likelihood is of the form shown in Tables 3 and 4 for clusters 3 and 4, respectively.

\[
T_1 = \begin{cases} 
1 & \text{if } i < 0.333 \\
0 & \text{otherwise}
\end{cases}, \quad T_2 = \begin{cases} 
1 & \text{if } 0.333 < i < 0.667 \\
0 & \text{otherwise}
\end{cases}, \quad T_3 = \begin{cases} 
1 & \text{if } 0.667 < i < 1 \\
0 & \text{otherwise}
\end{cases}, \quad T_4 = \begin{cases} 
1 & \text{if } i > 1 \\
0 & \text{otherwise}
\end{cases}
\]

Naïve Bayes Likelihood Table for \( K = 3 \) Clusters

\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline
 & \( T_{1j} \) & \( T_{2j} \) & \( T_{3j} \) & \( T_{4j} \) & \( m_j \) \\
\hline
\hline
Fraud (F) & \( P(T_{1j}|F) \) & \( P(T_{2j}|F) \) & \( P(T_{3j}|F) \) & \( P(T_{4j}|F) \) & \( P(m_j|F) \) \\
\hline
Genuine (¬F) & \( P(T_{1j}|¬F) \) & \( P(T_{2j}|¬F) \) & \( P(T_{3j}|¬F) \) & \( P(T_{4j}|¬F) \) & \( P(m_j|¬F) \) \\
\hline
\end{tabular}

Naïve Bayes Likelihood Table for \( K = 2 \) Clusters

\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
 & \( T_{1j} \) & \( T_{2j} \) & \( T_{3j} \) & \( T_{4j} \) & \( m_j \) \\
\hline
\hline
Fraud (F) & \( P(T_{1j}|F) \) & \( P(T_{2j}|F) \) & \( P(T_{3j}|F) \) & \( P(T_{4j}|F) \) & \( P(m_j|F) \) \\
\hline
Genuine (¬F) & \( P(T_{1j}|¬F) \) & \( P(T_{2j}|¬F) \) & \( P(T_{3j}|¬F) \) & \( P(T_{4j}|¬F) \) & \( P(m_j|¬F) \) \\
\hline
\end{tabular}
For evaluating the performance of the FDS, three common metrics were taken. The true positive rate (TPR), false positive rate (FPR), and the overall accuracy are often used in fraud detection studies, and these were used by Panigrahi et al (2009), Srivastava et al (2008), and Maes et al (1993). A good fraud detection system should typically have high TPR and overall accuracy, and low FPR. They are computed as follows:

$$TPR = \frac{Number \ of \ True \ Positive \ (TP) \ Cases}{Total \ Number \ of \ Positive \ Cases}$$ (12)

$$FPR = \frac{Number \ of \ False \ Positive \ (FP) \ Cases}{Total \ Number \ of \ Negative \ Cases}$$ (13)

$$Overall \ Accuracy = \frac{Number \ of \ TP \ Cases + Number \ of \ FP \ Cases}{Total \ Number \ of \ Cases}$$ (14)

Note that in this case, positive will refer to fraud transactions, and negative will refer to genuine transactions.

3. RESULTS AND DISCUSSION

Nine initial transaction histories were generated, each containing 300 transactions with different ratios of fraud. Of the nine data sets generated, the highest amount of fraudulent transactions amounted to over ninety percent of transactions (SS6). The lowest prevalence of fraud is from SS1 followed by SS2. A summary of the simulated data sets is shown in Table 5.

<table>
<thead>
<tr>
<th>Summary of Simulated Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>SS1</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>Genuine</td>
</tr>
<tr>
<td>282</td>
</tr>
<tr>
<td>94.0%</td>
</tr>
<tr>
<td>Fraud</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>6.0%</td>
</tr>
</tbody>
</table>

The EM clustering algorithm and its initializations produced results close to each other, while the K-means algorithm result was somewhat different. Distributions of cases falling in both 3 clusters and 2 clusters for all simulated data sets were generated along with corresponding Silhouette Coefficients (SC). On the average, $K = 2$ clusters exhibited denser clusters compared to $K = 3$ clusters. Different clustering algorithms produced different results across the nine parameter settings. The highest SC belongs to initialization $emEM$ of the EM algorithm at both $K = 3$ and $K = 2$, with values...
of 0.830851 and 0.508284, respectively. K-means consistently produced low values of SC, all with values below 0.3. A summary of these coefficients is presented in Tables 6 and 7.

### Silhouette Coefficient for K = 3 Clusters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SS1</th>
<th>SS2</th>
<th>SS3</th>
<th>SS4</th>
<th>SS5</th>
<th>SS6</th>
<th>SS7</th>
<th>SS8</th>
<th>SS9</th>
</tr>
</thead>
<tbody>
<tr>
<td>RndEM</td>
<td>-0.0322</td>
<td>0.0721</td>
<td>0.3394</td>
<td>-0.0337</td>
<td>0.0074</td>
<td>0.1052</td>
<td>0.4822</td>
<td>0.1146</td>
<td>0.2834</td>
</tr>
<tr>
<td>emEM</td>
<td>-0.0162</td>
<td>0.0935</td>
<td>-0.084</td>
<td>-0.0726</td>
<td>0.2602</td>
<td>0.4524</td>
<td>-0.0277</td>
<td>0.5082</td>
<td>-0.0304</td>
</tr>
<tr>
<td>svd</td>
<td>-0.1409</td>
<td>0.7778</td>
<td>0.2569</td>
<td>0.214</td>
<td>0.1646</td>
<td>0.1613</td>
<td>0.3681</td>
<td>0.132</td>
<td>0.2755</td>
</tr>
<tr>
<td>K-means</td>
<td>0.0622</td>
<td>0.0283</td>
<td>-0.0479</td>
<td>-0.0320</td>
<td>-0.0029</td>
<td>0.1604</td>
<td>0.01665</td>
<td>0.2247</td>
<td>0.0875</td>
</tr>
<tr>
<td>Average</td>
<td>-0.0318</td>
<td>0.2429</td>
<td>0.116</td>
<td>0.0189</td>
<td>0.1073</td>
<td>0.2199</td>
<td>0.2085</td>
<td>0.2449</td>
<td>0.1539</td>
</tr>
</tbody>
</table>

### Silhouette Coefficient for K = 2 Clusters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SS1</th>
<th>SS2</th>
<th>SS3</th>
<th>SS4</th>
<th>SS5</th>
<th>SS6</th>
<th>SS7</th>
<th>SS8</th>
<th>SS9</th>
</tr>
</thead>
<tbody>
<tr>
<td>RndEM</td>
<td>0.2222</td>
<td>0.6422</td>
<td>0.2222</td>
<td>0.5586</td>
<td>0.2420</td>
<td>0.1724</td>
<td>0.3184</td>
<td>0.6323</td>
<td>0.0039</td>
</tr>
<tr>
<td>emEM</td>
<td>0.2122</td>
<td>0.6422</td>
<td>0.8308</td>
<td>0.7714</td>
<td>0.1374</td>
<td>0.4125</td>
<td>0.3115</td>
<td>0.6376</td>
<td>0.0039</td>
</tr>
<tr>
<td>svd</td>
<td>0.2301</td>
<td>0.6298</td>
<td>0.1990</td>
<td>0.3103</td>
<td>0.7262</td>
<td>0.5662</td>
<td>0.7514</td>
<td>0.6443</td>
<td>0.1659</td>
</tr>
<tr>
<td>K-means</td>
<td>0.1386</td>
<td>0.0663</td>
<td>0.0759</td>
<td>0.0444</td>
<td>0.0761</td>
<td>0.3037</td>
<td>0.0604</td>
<td>0.5534</td>
<td>0.0413</td>
</tr>
<tr>
<td>Average</td>
<td>0.2008</td>
<td>0.4951</td>
<td>0.3317</td>
<td>0.4212</td>
<td>0.2954</td>
<td>0.3637</td>
<td>0.3604</td>
<td>0.6169</td>
<td>0.0538</td>
</tr>
</tbody>
</table>

Naïve Bayes was then applied to the different simulated data sets with the integrated clusters from EM and K-means algorithms. The performance metrics TPR, FPR and overall accuracy were computed. The overall accuracies of the nine simulated data sets with 3 clusters in Table 8 show above 90% in SS1 and SS6. With accuracies of above 80% are those of SS2 and SS3, while the rest register an accuracy of above 70%.

### Naïve Bayes Overall Accuracy for K = 3 Clusters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SS1</th>
<th>SS2</th>
<th>SS3</th>
<th>SS4</th>
<th>SS5</th>
<th>SS6</th>
<th>SS7</th>
<th>SS8</th>
<th>SS9</th>
</tr>
</thead>
<tbody>
<tr>
<td>emEM</td>
<td>95.3%</td>
<td>89.0%</td>
<td>75.7%</td>
<td>64.0%</td>
<td>92.3%</td>
<td>96.3%</td>
<td>58.7%</td>
<td>65.0%</td>
<td>88.7%</td>
</tr>
<tr>
<td>RndEM</td>
<td>94.7%</td>
<td>88.0%</td>
<td>96.0%</td>
<td>64.0%</td>
<td>61.0%</td>
<td>94.0%</td>
<td>93.7%</td>
<td>65.0%</td>
<td>71.7%</td>
</tr>
<tr>
<td>svd</td>
<td>94.3%</td>
<td>98.0%</td>
<td>88.3%</td>
<td>88.3%</td>
<td>79.0%</td>
<td>93.3%</td>
<td>90.3%</td>
<td>88.0%</td>
<td>90.7%</td>
</tr>
<tr>
<td>K-means</td>
<td>91.7%</td>
<td>78.0%</td>
<td>74.7%</td>
<td>65.3%</td>
<td>66.0%</td>
<td>90.7%</td>
<td>54.3%</td>
<td>67.7%</td>
<td>83.3%</td>
</tr>
<tr>
<td>Average</td>
<td>94.0%</td>
<td>88.3%</td>
<td>83.7%</td>
<td>70.2%</td>
<td>74.6%</td>
<td>93.6%</td>
<td>75.5%</td>
<td>71.4%</td>
<td>78.6%</td>
</tr>
</tbody>
</table>

For Naïve Bayes with K = 2 clusters, the results do not differ from the accuracies of the case K = 3 clusters. SS4 has a significant increase in the average accuracy of 8%, with its accuracy registering at 78%. Parameter settings SS5, SS7, and SS8 had the opposite outcome and produced lower overall accuracy than its counterpart. Settings 7 and 8 had over 7% decrease in accuracy,
while setting 5 had a 5% decrease in accuracy. Table 9 contains the summary of the overall accuracies of the nine simulated data sets with 2 clusters.

**Naïve Bayes Overall Accuracy for** \(K = 2\) **Clusters**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SS1</th>
<th>SS2</th>
<th>SS3</th>
<th>SS4</th>
<th>SS5</th>
<th>SS6</th>
<th>SS7</th>
<th>SS8</th>
<th>SS9</th>
</tr>
</thead>
<tbody>
<tr>
<td>emEM</td>
<td>94.7%</td>
<td>92.7%</td>
<td>96.3%</td>
<td>94.7%</td>
<td>61.0%</td>
<td>93.7%</td>
<td>60.7%</td>
<td>63.7%</td>
<td>83.7%</td>
</tr>
<tr>
<td>RndEM</td>
<td>95.0%</td>
<td>92.7%</td>
<td>75.7%</td>
<td>82.3%</td>
<td>62.0%</td>
<td>91.0%</td>
<td>61.3%</td>
<td>64.7%</td>
<td>80.1%</td>
</tr>
<tr>
<td>Svd</td>
<td>94.3%</td>
<td>92.3%</td>
<td>76.7%</td>
<td>71.0%</td>
<td>91.3%</td>
<td>92.0%</td>
<td>93.3%</td>
<td>64.7%</td>
<td>84.0%</td>
</tr>
<tr>
<td>K-means</td>
<td>92.7%</td>
<td>78.0%</td>
<td>75.7%</td>
<td>64.0%</td>
<td>60.7%</td>
<td>89.3%</td>
<td>57.3%</td>
<td>64.7%</td>
<td>71.7%</td>
</tr>
<tr>
<td>Average</td>
<td>94.2%</td>
<td>88.9%</td>
<td>81.1%</td>
<td>78.0%</td>
<td>68.8%</td>
<td>91.5%</td>
<td>68.2%</td>
<td>64.4%</td>
<td>79.9%</td>
</tr>
</tbody>
</table>

While results across the simulated data sets are varied, most of the overall accuracies of the fraud detection system are higher than 70%. For \(K = 3\) clusters, two initializations of the EM algorithm registered an accuracy of 80% (emEM and RndEM), with another initialization capping at 90% (svd). At \(K = 2\), initialization RndEM from EM clustering, and K-means made virtually similar results, while the highest accuracies belonged to emEM and svd. The results of the three initializations of EM Algorithm across the settings are more alike.

Both the true positive rate (TPR) and the false positive rate (FPR) also registered higher at the EM algorithm initializations, with RndEM scoring both the highest TPR and the lowest FPR among all the clustering algorithms. K-means clustering algorithm scored 72.96% at the overall accuracy with TPR and FPR of 64% and 26%, respectively. The case for high and low clusters produced different results, with emEM having the highest TPR. The two other initializations of EM obtained the lowest FPR. Furthermore, the averages of the initializations indicate that EM is superior to K-means in accuracy, as well as TPR and FPR for 2 and 3 clusters. Table 10 summarizes the TPR and FPR of the fraud detection system across all the clustering algorithms and initializations used. On the average, EM clustering algorithm has an overall accuracy of 83.83% with TPR of 80.14% and FPR of 15.34%. Figures 2 and 3 are visual supplements to Tables 10 and 11.

**Algorithm and Initialization Accuracy for** \(K = 3\) **Clusters**

<table>
<thead>
<tr>
<th></th>
<th>emEM</th>
<th>RndEM</th>
<th>svd</th>
<th>K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR</td>
<td>73.80%</td>
<td>85.71%</td>
<td>80.93%</td>
<td>64.52%</td>
</tr>
<tr>
<td>FPR</td>
<td>18.35%</td>
<td>12.10%</td>
<td>15.56%</td>
<td>26.34%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>80.56%</td>
<td>80.89%</td>
<td>90.04%</td>
<td>72.96%</td>
</tr>
</tbody>
</table>
Algorithm and Initialization Accuracy for $K = 2$ Clusters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>emEM</th>
<th>RndEM</th>
<th>svd</th>
<th>K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR</td>
<td>81.07%</td>
<td>75.91%</td>
<td>79.35%</td>
<td>65.75%</td>
</tr>
<tr>
<td>FPR</td>
<td>19.36%</td>
<td>15.46%</td>
<td>15.12%</td>
<td>23.64%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>83.70%</td>
<td>80.07%</td>
<td>83.96%</td>
<td>71.74%</td>
</tr>
</tbody>
</table>

Over-all TPR-FPR across the Clustering Algorithms for $K = 3$ Clusters

The spread of TPR and FPR with overall accuracy across the 9 simulated data sets are shown in Figures 4 and 5. SS6 got the highest TPR, and
over-all accuracy, while SS3 has the lowest TPR for $K = 3$. At $K = 2$, SS6 had the highest overall accuracy, and SS7 seems to have performed worst with both the highest FPR, as well as the second lowest TPR.

**5. CONCLUSIONS AND RECOMMENDATIONS**

The proposed Naïve Bayes method with clustering is generally effective with an overall accuracy of 81% with 3 clusters. At 2 clusters, the overall accuracy does not deviate much with an average accuracy of 79%. The true positive rate of this method averages at 80% and can be further improved. Both cases of clusters produced the same averages. Results also show that the
accuracy is dependent on the type of fraudsters where fraudsters with large transactions are easier to detect with high TPR and overall accuracy.

Among the three performance metrics, the EM clustering algorithm is better than the K-means algorithm. Increasing the number of clusters is recommended that may help determine the best initialization for the EM clustering algorithm, and profile the spending patterns of both credit card holders and fraudsters. More specifically, use $K = 5$ for, very low, low, medium, high, and very high transaction amounts. It is also recommended to test this method with other clustering algorithms to achieve better silhouette coefficients.

For the simulator, it is suggested that the mean of the Gaussian Distributions should be turned into an actual amount rather than a percentage of a credit limit. Since the credit limit should be unknown to the fraudster, it is less realistic to include the credit limit in the simulator. Furthermore, the proposed FDS algorithm could also be revised by combining it with other Bayesian Network methods. Integration with other fraud detection methods can also be explored. It is highly recommended to apply the proposed method to real-life data if available.

References
Appendix

R codes:
#Simulation:
#package Hidden Markov, xlsx
Q <- matrix(c(-.2, .2, .8, -.8), byrow=TRUE, nrow=2)
X <- mmpp(NULL, Q, delta=c(.99, 0.01), lambda=c(8, 2))
y <- simulate(x, nsim=299, seed = 10)
at <- y$tau #arrival times
lab <- y$ys
m <- list()
for (i in 1:length(lab)){
  ml[i] = (lab[i] - 1)
}
label <- unlist(m) #fraud labels (0, 1)
set.seed(123)
not <- list()
for (i in 1:length(label)){
  if (label[i] == 0){not[i] = rnorm(1, 10, )}
  else if (label[i] == 1){ n[i] = rnorm(1, 50, )}
  else {NULL}
}
ch <- ml[ml$fraud.label == 0, ]
ch$interarrival.times <- iat.ch
ch <- ch[, c(1, 2, 5, 3, 4)]
d1 <- list()
d2 <- list()
d3 <- list()
d4 <- list()
for (i in 1:nrow (ch)){
  if((iat.ch[i] > .33)&(iat.ch[i]<=.66)){
    d2[i] = 1
  } else {d2[i] = 0}
}
for (i in 1:nrow (ch)){
  if(iat.ch[i] >= .66& (iat.ch[i]<=1)){
    d3[i] = 1
  } else {d3[i] = 0}
}
for (i in 1:nrow (ch)){
  if(iat.ch[i] > 1){
    d4[i] = 1
  } else {d4[i] = 0}
}
ch$cl1 <- cl1
ch$cl2 <- cl2
ch$cl3 <- cl3
ch$cl4 <- cl4
ch$NA. <- NULL
fh <- ml[ml$fraud.label == 1, ]
at2 <- fh$arrival.times
l2 <- list()
for (i in 1:nrow (fh)){
  if(iat.fh[i] <=33){
    d1[i] = 1
  } else {d1[i] = 0}
}
for (i in 1:nrow (fh)){
  if(iat.fh[i] >.33& (iat.fh[i]<=.66)){
    d2[i] = 1
  } else {d2[i] = 0}
}
for (i in 1:nrow (fh)){
  if(iat.fh[i] > 1){
    d4[i] = 1
  } else {d4[i] = 0}
}
fh$cl1 <- cl1
fh$cl2 <- cl2
fh$cl3 <- cl3
fh$cl4 <- cl4
fh$NA. <- NULL
d2 <- list()
d3 <- list()
d4 <- list()
for (i in 1:nrow(fh)) {
  if (iat.fh[i] <= .33) {
    d1[i] = 1
  } else {
    d1[i] = 0
  }
}
for (i in 1:nrow(fh)) {
  if ((iat.fh[i] > .33) & (iat.fh[i] <= .66)) {
    d2[i] = 1
  } else {
    d2[i] = 0
  }
}
for (i in 1:nrow(fh)) {
  if ((iat.fh[i] > .66) & (iat.fh[i] <= 1)) {
    d3[i] = 1
  } else {
    d3[i] = 0
  }
}
for (i in 1:nrow(fh)) {
  if (iat.fh[i] > 1) {
    d4[i] = 1
  } else {
    d4[i] = 0
  }
}
D1 <- unlist(d1)
D2 <- unlist(d2)
D3 <- unlist(d3)
D4 <- unlist(d4)
fh$D1 <- D1
fh$D2 <- D2
fh$D3 <- D3
fh$D4 <- D4
m2 <- data.frame(fh$amount)

amt.rnd <- init.EM(m2, nclass = 3, method = "em.EM")
amt.em <- init.EM(m2, nclass = 3, method = "Rnd.EM", EMC = EMC.Rnd)
amt.svd <- emgroup(m2, nclass = 3)
amt.kmn <- kmeans(m2, nclass = 3, iter.max = 10, nstart = 1)
c1 <- factor(amt.rnd$cl)
c2 <- factor(amt.em$cl)
c3 <- factor(amt.svd$cl)
c4 <- factor(amt.kmn$cl)
fh$cl1 <- c1
fh$cl2 <- c2
fh$cl3 <- c3
fh$cl4 <- c4
fh$NA. <- NULL

new <- rbind(oh, fh)
mlnew <- new[order(new$arrival.times),]
write.xlsx(mlnew, "SS1_sim.xlsx", sheetName = "CH", append = TRUE, row.names = FALSE)
vars3 <- c("D1", "D2", "D3", "D4", "cl3")
vars4 <- c("D1", "D2", "D3", "D4", "cl4")
nb1 <- naive.bayes(mat1, "fraud.label",
vars1)
nb2 <- naive.bayes(mat2, "fraud.label",
vars2)
nb3 <- naive.bayes(mat3, "fraud.label",
vars3)
nb4 <- naive.bayes(mat4, "fraud.label",
vars4)
pred1 <- predict(nb1, mat1, prob = TRUE)
pred2 <- predict(nb2, mat2, prob = TRUE)
pred3 <- predict(nb3, mat3, prob = TRUE)
pred4 <- predict(nb4, mat4, prob = TRUE)
pscore1 <- t(data.frame(attr(pred1, 'prob')))
rownames(pscore1) <- NULL
pscore1 <- unlist(pscore1)  # pscore1[, 1]
score of 0, pscore1[, 2] score of 1
pscore2 <- t(data.frame(attr(pred2, 'prob')))
rownames(pscore2) <- NULL
pscore2 <- unlist(pscore2)
pscore3 <- t(data.frame(attr(pred3, 'prob')))
rownames(pscore3) <- NULL
pscore3 <- unlist(pscore3)
pscore4 <- t(data.frame(attr(pred4, 'prob')))
rownames(pscore4) <- NULL
pscore4 <- unlist(pscore4)
write.xlsx(mat1, "RndBN.xlsx", row.names = FALSE)
write.xlsx(mat2, "emBN.xlsx", row.names = FALSE)
write.xlsx(mat3, "svdBN.xlsx", row.names = FALSE)
write.xlsx(mat4, "kmnBN.xlsx", row.names = FALSE)
#BN Accuracy and Performance
pred1 <- read.xlsx("RndBN.xlsx", sheetName = "Sheet1", header = TRUE)
pred2 <- read.xlsx("emBN.xlsx", sheetName = "Sheet1", header = TRUE)
pred3 <- read.xlsx("svdBN.xlsx", sheetName = "Sheet1", header = TRUE)
pred4 <- read.xlsx("kmnBN.xlsx", sheetName = "Sheet1", header = TRUE)
mod1 <- prediction(pred1$pscore1,
pred1$fraud.label)
mod2 <- prediction(pred2$pscore1,
pred2$fraud.label)
mod3 <- prediction(pred3$pscore1,
pred3$fraud.label)
mod4 <- prediction(pred4$pscore1, 
pred4$fraud.label)
jpeg('ROCRnd.jpeg')
plot(performance(mod1, 'tpr', 'fpr'))
dev.off()

jpeg('ROCem.jpeg')
plot(performance(mod2, 'tpr', 'fpr'))
dev.off()

jpeg('ROCsvd.jpeg')
plot(performance(mod3, 'tpr', 'fpr'))
dev.off()

jpeg('ROCKmn.jpeg')
plot(performance(mod4, 'tpr', 'fpr'))
dev.off()
The performances of R GPU implementations of the GMRES method

Bogdan Oancea (bogdan.oancea@faa.unibuc.ro)
University of Bucharest

Richard Pospisil (richard.pospisil@upol.cz)
Palacky University of Olomouc

ABSTRACT

Although the performance of commodity computers has improved drastically with the introduction of multicore processors and GPU computing, the standard R distribution is still based on single-threaded model of computation, using only a small fraction of the computational power available now for most desktops and laptops. Modern statistical software packages rely on high performance implementations of the linear algebra routines there are at the core of several important leading edge statistical methods. In this paper we present a GPU implementation of the GMRES iterative method for solving linear systems. We compare the performance of this implementation with a pure single threaded version of the CPU. We also investigate the performance of our implementation using different GPU packages available now for R such as gmatrix, gputools or gpuR which are based on CUDA or OpenCL frameworks.

Keywords: R; GPU; statistical software; GMRES

JEL Classification: C6, C8

1. INTRODUCTION

Since the appearance of the first supercomputers in early ‘70s the nature of parallel computing has changed drastically. Besides the advancements in technology that allow building processors with a very high clock rate and a low specific dimension, today the commodity computers are based on multicore architecture that can run multiple tasks in parallel. Besides the classical multiheaded programming, another parallel computing paradigm called GP-GPU computing (general-purpose computing on graphics processing units), first experimented around 2000-2001 (see for example Larsen and McAllister, 2001), is widely used nowadays to speed up numerical intensive applications.

While numerical intensive parts of applications are handled
traditionally by the CPUs, now GPUs have FLOP rates much higher than CPUs because GPUs are specialized for highly parallel intensive computations and they are designed with much more transistors allocated to data processing rather than control flow or data caching. This is shown in figure 1 where it can be noticed that the total surface of the device dedicated to ALUs is much higher in case of GPU than CPU.

The theoretical FLOP rates for GPUs and CPUs are presented in figure 2 and the memory bandwidth in figure 3. Current GPUs have FLOP rates at least 100 times greater than CPUs and the memory bandwidth is at least 10 times greater for GPU memories than for the main memory of the computer. These figures indicate that GPUs are ideal candidates to dispatch the numerical parts of applications.

A comparison between CPU and GPU structure

Statistical software relies heavily on numerical computations and can take advantage of the new paradigm of GPU computing. In this paper we will evaluate the performances of an R implementation of a numerical algorithm widely used in linear algebra libraries, namely the GMRES method to solve linear systems (Saad and Schultz, 1986). The rest of the paper is structured like this: in section 2 we briefly present the GPU computing frameworks currently available for scientific computing, in section 3 we present the GMRES algorithm, in section 4 we present some R packages that implements the GPU-GPU computing model and discuss three R implementations that uses GPU computing, comparing their performances with the serial implementations. Section 5 concludes our paper.

A comparison of the theoretical FLOP rates of typical CPUs and GPUs

Figure 2

A comparison of the memory bandwidth for CPU and GPU

Figure 3

2. GPU COMPUTING FRAMEWORKS

GPUs are designed to solve general problems that can be formulated as data-parallel computations: the same stream of instructions is executed in parallel on many (different) data elements with a high ratio between arithmetic operations and memory accesses, like the SIMD approach of the parallel computers taxonomy. Currently there are several frameworks that implement GP-GPU model:

- CUDA (Compute Unified Device Architecture) – first introduced in 2006 by NVIDIA, is a general purpose parallel programming architecture that uses the parallel compute engines in NVIDIA GPUs to solve numerical intensive problems in a more efficient way than a CPU does (NVIDIA Corporation, 2007);

- OpenCL (Open Computing Language) - is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, digital signal processors (DSPs), field-programmable gate arrays (FPGAs) and other hardware accelerators (OpenCL Specification Version 1.0, 2008);

- OpenACC - (Open Accelerators) is a programming standard for parallel computing developed by Cray, CAPS, NVIDIA and PGI. The standard is designed to simplify parallel programming of heterogeneous CPU/GPU systems – it works on NVIDIA, AMD and Intel accelerators (OpenAcc-Standard.org, 2017);

In this paper we used CUDA framework and the R packages that allows access to the CUDA API. CUDA parallel programming framework works with three important abstractions (Oancea et al., 2012a):

• a hierarchy of thread groups;
• shared memories;
• a barrier synchronization accessible by the programmer;

CUDA parallel programming paradigm requires programmers to (NVIDIA Corporation, 2017):

• partition the problem into coarse tasks that can be run in parallel by blocks of threads;
• divide each task into finer groups of instructions that can be executed cooperatively in parallel by the threads within a block;

The main programming language for CUDA is C but nowadays there are bindings to other languages too: Java, Fortran, Common Lisp, Python, R. While C libraries that uses CUDA for numerical computing are mature now,
the bindings to other languages are only at the beginning. In (Oancea et al., 2012b) we investigated the performance of a Java library that uses CUDA for numerical computations and found that one can achieve an important speedup compared to the classical (serial) version of the respective algorithm while in (Oancea and Andrei, 2013) we investigated a hybrid approach CPU-GPU. In this paper we will test if a linear algebra algorithm implemented in R is capable to obtain a significant speedup compared with a serial R implementation.

CUDA paradigm of GPU computing

![CUDA paradigm of GPU computing](http://docs.nvidia.com/cuda/cuda-c-programming-guide/)

3. GMRES METHOD

One question may arise: why GMRES? When using GPU for numerical computations the biggest speedups are obtained by algorithms that use level 3 BLAS operations (such as matrix-matrix multiplication). GMRES

---

1 http://docs.nvidia.com/cuda/cuda-c-programming-guide/
uses only level 1 and level 2 operations (vector updates and matrix-vector products) and we wanted to test if the available GPU R packages can obtain an important advantage (speedup) over classical numerical algorithms even for methods that are not easily parallelizable.

The GMRES (Generalized Minimum RESidual) was first proposed in 1986 as a Krylov subspace method for nonsymmetric systems (Saad and Schultz, 1986). This method approximates the solution by a vector in a Krylov subspace with minimal residual – this vector is build using the so-called Arnoldi iteration.

The problem to be solved is $Ax = b$. The n-th Krylov subspace for this problem is: $K_n = K_n(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{n-1}b\}$. GMRES approximates the exact solution of the linear system with by the vector $x_n \in K_n$ that minimizes the Euclidean norm of $r_n = Ax_n - b$. The restarted version of the GMRES algorithm that we implemented is presented below (Kelley, 1995):

1. choose $x_0$ and compute $r_0 = b - Ax_0$ and $v_1 = r_0 / \|r_0\|
2. for j = 1, 2, ..., m do
3. \[ h_{ij} = (Av_i, v_j) \text{ for } i = 1, 2, ..., j \]
4. \[ \tilde{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij} v_i \]
5. \[ h_{j+1,j} = \|\tilde{v}_{j+1}\| \]
6. \[ v_{j+1} = \tilde{v}_{j+1} / h_{j+1,j} \]
7. endfor
8. compute $x_m = x_0 + V_m y_m$ where $y_m$ minimizes $\|\beta e_1 - \tilde{H}_m y\|$ for $y \in \mathbb{R}^m$
9. Restart – compute $r_m = b - Ax_m$
10. if $r_m < \varepsilon$ stop
11. else $x_0 = x_m$, $v_1 = r_m / \|r_m\|$ and go to line 2 (for) (restart iterations)

This algorithm contains a level 2 BLAS operation, namely the matrix-vector product $Av_i$ that has to be evaluated at each iteration and level 1 BLAS operations – vector updates inside the for loop. The least squares problem (8) can be solved maintaining a QR factorization of H and it requires $O(mN)$ floating point operations (Kelley, 1995). There are several versions of the GMRES algorithms presented in the literature that work at a block level to improve the efficiency or use techniques to parallelize the computations (Chronopoulos, 1986), (Chronopoulos and Kim, 1992), (Chronopoulos and Swanson, 1996), (Chronopoulos and Kucherov, 2010), (Ranjan et al. 2016). In this paper we focused on using GPU
CUDA framework to improve the speed of execution of the GMRES algorithm. Obtaining an increase of the speed of execution through parallelization using GPU is a demanding task since the overhead of memory transfers between main memory and device memory is high and one cannot allocate the entire matrices and vectors needed to obtain the solution only on device memory due to the limited amount of memories currently available on GPU cards. A tradeoff between computations done by the CPU and GPU should be achieved to obtain a good speedup.

4. R SUPPORT FOR GPU COMPUTING

Currently there are several R packages that support GPU/CUDA framework: gmatrix (Moris, 2016), gpuR (Determan, 2017), gputools (Buckner and Wilson, 2013), cudaBayesreg (da Silva, 2011), HiPLARb (Szeremi, 2012), HiPLARM (Nash and Szeremi, 2012) Rcula (Morris, 2013). Some of these packages work only with CUDA while others also support OpenCL or OpenACC. While some packages are designed to solve a single type of problems, others are general purpose packages that allow users to implement different types of algorithms. From this latter category we chose gmatrix, gpuR and gputools and implemented the GMRES iterative algorithm for solving linear systems to evaluate the performances of the current R packages that uses the GPU computing model.

gmatrix is a general-purpose package that works with only NVIDIA CUDA toolkit. It has gmatrix() and gvector() methods that allow users to create matrices and vectors on GPU device and all the computations with these matrices/vectors are performed by default on GPU. Data can be transferred between the host memory (RAM) and device memory (GPU) using two simple methods: h() and g(). This package implements several numerical operations with matrices and vectors on the GPU such as matrix multiplication, addition, subtraction, the Kronecker product, the outer product, comparison operators, logical operators, trigonometric functions, indexing, sorting, random number generation, etc. A matrix or a vector on device memory can be created as simple as:

```r
> A <- gmatrix(1:400, 20, 20)
> g_seq <- gseq(1:20)
> b <- as.gvector(g_seq)
```

Performing algebraic operations with such objects is very simple since most of the arithmetic and logical operators (+, -, *, /, %%, ^, ==, !=, < , >, <=, >= & , |) are overloaded:

```r
> c <- A %*% b
```
In our implementation of the GMRES algorithm using gmatrix package we performed only the matrix-vector product on GPU while the rest of the operations are performed by the CPU. The reason why we chose to send to the GPU only the matrix-vector product is that the level 1 operations start to have a speedup > 1 only for very large vectors (N>5e5) according to (Morris, 2016) and due to GPU memory limitations, we used N between 1e3 and 1e4 which is considerably lower than the threshold where the speedup is greater than one.

gputools is another general-purpose package that works with CUDA toolkit. Matrices and vectors are created on the host memory using the normal methods and then they are transferred to the device memory where computations took place. After the matrices and vectors are created, the matrix operations dispatched to the GPU were performed using gpuMatMult(). Again, we implemented only level 2 operations on GPU because level 1 operations would not be efficient to send to the GPU due to the high overhead incurred by the memory transfers.

Below is a simple example of how a matrix-vector product can be send for execution to the GPU:

```r
N <- 200
da <- matrix(rnorm(N * N), nrow = N)
db <- matrix(rnorm(N), nrow = N)
dc <- gpuMatMult(da, db)
```

gpuR is the third package that we tested, and it works both with CUDA and OpenCL backends. It has gpuMatrix() and gpuVector() methods that allow users to create objects on the host memory but the computations with these objects are performed on GPU in a transparent way. There are also two other methods vclMatrix() and vclVector() that creates the respective objects directly into the device memory and work asynchronously. By using the asynchronous mode, R will immediately return to the CPU after calling any operation that involves vcl objects. When one tries to access data computed using vcl objects, but the actual computations done by the GPU are not ready, R will wait until the computations are finished and the data become available.

Below are examples of how some basic matrix operations look like using gpuR:

```r
N <- 200
da <- matrix(rnorm(N * N), nrow = N)
db <- matrix(rnorm(N), nrow = N)
dagpu <- gpuMatrix(da, type = "double")
dbgpu <- gpuMatrix(db, type = "double")
dc <- dagpu %*% dbgpu
```
The most commonly used operators are overloaded: %*%, +, -, *, /, crossprod, tcrossprod, etc. to work with GPU objects. For GMRES we implemented all numerical operations on GPU using vcl objects and methods: this approach speeds up the computation but put a limit through the available GPU memory.

The experimental setup consists of a computer with:
- Intel core i7 4710HQ processor at 2.5GHz;
- 6 MB level 3 cache, 1 MB level 2 cache and 256 KB level 1 cache;
- 16 GB DDR3 RAM;
- NVIDIA GeForce 840M graphic card (it has a Maxwell architecture);
- 2 GB video RAM with a bandwidth of 16 GB/s;
- 384 shader units clocked at 1029 MHz;
- Ubuntu ver. 16.04;
- CUDA framework ver. 8.0.61;
- R ver. 3.2.3;
- gmatrix ver. 0.3, gputools ver. 1.1, gpuR ver. 1.2.1;

The following table shows the speedup of the GPU implementations compared with a serial algorithm (gmres() from pracma package). We used matrices with dimensions between 1000 and 10000. The size of the problem was limited by the available amount of the graphics card memory. It can be observed that the speedups are rather modest, only gpuR showing a speedup between 3 and 4. Increasing the problem size could lead to a better speedup but the limited amount of memory on the graphics card precluded us to use bigger matrices.

**Running times for different implementations and different size of the problem**

<table>
<thead>
<tr>
<th>N</th>
<th>gmatrix implementation</th>
<th>gputools implementation</th>
<th>gpuR implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.06</td>
<td>0.75</td>
<td>0.99</td>
</tr>
<tr>
<td>2000</td>
<td>1.28</td>
<td>0.77</td>
<td>1.11</td>
</tr>
<tr>
<td>3000</td>
<td>1.33</td>
<td>0.83</td>
<td>1.25</td>
</tr>
<tr>
<td>4000</td>
<td>1.33</td>
<td>0.96</td>
<td>1.67</td>
</tr>
<tr>
<td>5000</td>
<td>1.36</td>
<td>1.04</td>
<td>2.33</td>
</tr>
<tr>
<td>6000</td>
<td>1.46</td>
<td>1.17</td>
<td>2.90</td>
</tr>
<tr>
<td>7000</td>
<td>1.71</td>
<td>1.25</td>
<td>3.21</td>
</tr>
<tr>
<td>8000</td>
<td>2.25</td>
<td>1.30</td>
<td>3.75</td>
</tr>
<tr>
<td>9000</td>
<td>2.45</td>
<td>1.41</td>
<td>4.10</td>
</tr>
<tr>
<td>10000</td>
<td>2.95</td>
<td>1.58</td>
<td>4.25</td>
</tr>
</tbody>
</table>
5. CONCLUSIONS

GPU computing becomes an important solution for numerical computing. While C libraries are mature and deliver high speedups, R packages that implements GPU computing are at the beginning. For the three packages that we tested we obtained moderate speedups (~3…4) compared with the serial version of the algorithm. These low speedups are due to the limited amount of device memory that precluded us to use bigger matrices. It is shown in (Morris, 2016) that for level 1 BLAS operations the size of the vector should be greater than 5E5 to obtain speedups greater than 1, but such dimensions are too high for our version of GMRES to fit into the device memory. For commodity computers, where the available amount of graphics card memory is limited the speedups are low and they are comparable with speedups obtained by using a tuned linear algebra library (Oancea et al., 2015). Nevertheless, using GP-GPU together with specialized R packages could be a solution for the future.
References


Economic Interdependence Between The GDP and External Trade in Romania - A VEC Method Analyse

PhD student Artur - Emilian SIMION
The Romanian Academy / National Institute of Statistics Romania

PhD Student Florentina Viorica GHEORGHE
The Romanian Academy/National Institute of Statistics-Romania

ABSTRACT

Trading transactions between nations have a very long history, but in recent years international trade has become increasingly important with a larger share of GDP devoted to exports and imports and it is considered to be a major component of sustainable economic growth. The correlation between external trade flows and gross domestic product (GDP) have been analyzed in many specialized economic papers. The developed econometric models have demonstrated the strong connection between these macroeconomic indicators.

This paper once again demonstrates the long-term and short-term relationship between these variables using the VEC econometric model on quarterly GDP, Export, Import and GFCF data of Romania from 1995 to 2015. The VECM analysis was performed using R statistical software and is based on data extracted from the Eurostat, European Union Statistical Office website and are expressed in millions of euro.

Keywords: GDP, export, import, GFCF, vector correction error, heteroscedasticity, cointegration, stationarity

JEL classification: F14, F17, Q11

1. INTRODUCTION

The relationship between GDP and external trade flows, export and import, has long been one of the most debated subject of international economic development, with particular attention to research. Fundamental economic theories show the contribution of exports to economic growth through the so called ‘multiplying effect of external trade’ (Tekin, 2012). At the same time, the growing exports generate higher degree of economic
GDP

\[ \text{GDP} = \text{FC} + \text{GFCF} + \Delta S + (\text{E-I}) \]  

[1]

According to the expenditure computation method:

\begin{itemize}
  \item GDP, GFCF and trade flows, had oscillatory evolution but with similar trends (increase or decrease) over the selected period, as Figure 1 illustrates.
  
  \section*{2. METHODOLOGICAL CONSIDERATIONS}
  
  \textbf{The research method selection}
  
  In order to analyse the connection between GDP and external trade, the present paper aims to test a multivariate VEC model. Thus, the variables used in the analysis are GDP, GFCF, import and export. For generating the regression equation, a VEC model was used, after testing the stationarity of the selected series (Dickey-Fuller Augmented unit test), cointegration (Johansen cointegration test) and the causality link between the variables (Granger test and Wald test).
  
  \textbf{Description of variables}
  
  In the study of the economic interdependence between GDP, export, import and GFCF, quarterly data from 1995-2015, were considered, GDP as a dependent variable and export, import and GFCF as independent variables. Values are expressed in millions of euro.

GDP is a macroeconomic indicator reflecting the monetary value of final goods and services – that are bought by the final user – produced in a country, in a given period of time (quarter or year).

According to the expenditure computation method:

\[ \text{GDP} = \text{FC} + \text{GFCF} + \Delta S + (\text{E-I}) \]  

[1]

Gross fixed capital formation represents the value of the durable goods acquired by resident units for the purpose of subsequent use in the production process.

Exports and imports of goods and services consist of selling or purchasing transactions in goods and services (sales, barter, donations, etc.) between residents and non-residents.

**Theoretical aspects of the proposed analysis**

R statistical software was used for analysis, by its package named vars(Pfaff, 2008a), which is specific for the VAR, SVAR and SVEC analyzes. R is by far the most used open source statistical software among scientific communities, since it envisages the advantages of an open source system: low costs related only with the training of users, technical support provided by a large community of users, continuous upgrade and linkage with the way statisticians think and work.

In R software, the Augmented Dickey-Fuller test is implemented within the function ur.df(), from the package urca(Pfaff, 2008b). Johansen cointegration test was performed using ca.jo() function. Moreover, Granger test for causality was computed by using grangertest() function and also Wald test with its specific wald.test() function and Arch test with arch.test() function.

For the four data series (GDP, exports of goods and services, import of goods and services and GFCF), we tested:
- stationarity – root unit test – Dickey-Fuller Augmented,
- cointegration – Johansen and cointegration test and
- causal link between variables (Granger test and Wald test).

For the regression equation, generated with the VEC model, the following residue tests were performed:
- the normality test of the residue (Jaque-Bera – Hitogram distribution test),
- Residue correlation test (Breusch-Godfrey test) and
- heteroscedasticity (Breusch-Pagan-Godfrey test and ARCH test).

**3. THE VEC ECONOMETRIC MODEL ON QUARTERLY GDP, EXPORT, IMPORT AND GFCF FOR ROMANIA**

**a. Basic statistics**

The table below shows the Summary Statistics: average, median, maximum, minimum, standard deviation values for each of the variables:
Summary Statistics

<table>
<thead>
<tr>
<th></th>
<th>GDP</th>
<th>GFCF</th>
<th>EXPORT</th>
<th>IMPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>21697.72</td>
<td>5704.752</td>
<td>7359.042</td>
<td>8743.940</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>18949.60</td>
<td>4433.300</td>
<td>6665.900</td>
<td>8687.250</td>
</tr>
<tr>
<td><strong>Maximum</strong></td>
<td>47575.60</td>
<td>19176.20</td>
<td>16927.50</td>
<td>17653.10</td>
</tr>
<tr>
<td><strong>Minimum</strong></td>
<td>5821.300</td>
<td>880.9000</td>
<td>1546.600</td>
<td>1855.300</td>
</tr>
<tr>
<td><strong>Std. Dev.</strong></td>
<td>12534.30</td>
<td>4351.000</td>
<td>4708.647</td>
<td>5137.808</td>
</tr>
<tr>
<td><strong>Skewness</strong></td>
<td>0.335075</td>
<td>0.878157</td>
<td>0.489688</td>
<td>0.115374</td>
</tr>
<tr>
<td><strong>Kurtosis</strong></td>
<td>1.688394</td>
<td>2.912817</td>
<td>1.992009</td>
<td>1.458280</td>
</tr>
<tr>
<td><strong>Jarque-Bera</strong></td>
<td>7.592937</td>
<td>10.82285</td>
<td>6.913274</td>
<td>8.505509</td>
</tr>
<tr>
<td><strong>Probability</strong></td>
<td>0.022450</td>
<td>0.004465</td>
<td>0.031536</td>
<td>0.014225</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td>1822609.0</td>
<td>479199.2</td>
<td>618159.5</td>
<td>734491.0</td>
</tr>
<tr>
<td><strong>Sum Sq. Dev.</strong></td>
<td>1.30E+10</td>
<td>1.57E+09</td>
<td>1.84E+09</td>
<td>2.19E+09</td>
</tr>
<tr>
<td><strong>Observations</strong></td>
<td>84</td>
<td>84</td>
<td>84</td>
<td>84</td>
</tr>
</tbody>
</table>

Source: R output on Summary statistics

On the basis of these statistics, we can establish that the GDP is between 5,821.3 million euro in the first quarter of 1997 and 47,575.6 million euro in the fourth quarter of 2015. The average value of this indicator for the period 1995-2015 is 21,697 EUR million. The values of the Skewness and Kurtosis tests show that the distribution considered is not a symmetrical one; the majority of the values are between the minimum and the average of the data series (the median of the series is less than the average of the series).

Similarly, the GFCF is between 880.9 million euro in the first quarter of 1997 and 19,176.2 million euro in the third quarter of 2008. The average value of this indicator for the period 1995-2015 is 5,704.8 million euro. The values of the Skewness and Kurtosis tests allow us to assert that the considered distribution is not a symmetrical one; the majority of the values are between the minimum and the average of the series (the median of the series is less than the average of the series).

For export, the values are between 1,546.6 million euro in the first quarter of 1995 and 16,927.5 million euro in the 3rd quarter of 2015. The average value of this indicator for the period 1995-2015 is 7,359.0 million euro. Also, the values of the Skewness and Kurtosis tests allow us to assert that the considered distribution is not a symmetrical one; the majority of the values are between the minimum and the average of the series (the median of the series is less than the average of the series).

Imports have values ranging from 1,855.3 million euro in the first quarter of 1995 and 17,653.1 million euro in the fourth quarter of 2015. The average value of this indicator for the period 1995-2015 is 8,743.9 million
As in the case of exports, the values of the Skewness and Kurtosis tests allow us to assert that the considered distribution is not a perfectly symmetrical one; the majority of the values are between the minimum and the average of the series (the median of the series is less than the average of the series).

GDP as against Export, Import and GFCF

The coefficients in the correlation diagram show that there are strong links between the four selected variables.

The correlation coefficients

<table>
<thead>
<tr>
<th></th>
<th>GDP</th>
<th>GFCF</th>
<th>EXPORT</th>
<th>IMPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDP</td>
<td>1</td>
<td>0.93</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>GFCF</td>
<td>0.93</td>
<td>1</td>
<td>0.78</td>
<td>0.88</td>
</tr>
<tr>
<td>EXPORT</td>
<td>0.93</td>
<td>0.78</td>
<td>1</td>
<td>0.96</td>
</tr>
<tr>
<td>IMPORT</td>
<td>0.97</td>
<td>0.88</td>
<td>0.96</td>
<td>1</td>
</tr>
</tbody>
</table>

Source: R output on correlation coefficient
b. Stationarity and cointegration tests of the series

Stationarity test of the series (Unit root test – Dickey-Fuller Augmented)

For each series, the stationary test (Unit root test – Dickey-Fuller Augmented) was performed on both the initial series and the differentiated series (order 1 and order 2). The tests indicated the following: the GDP, GFCF import and export data series are not stationary. They become stationary after the 2nd order differentiation.

<table>
<thead>
<tr>
<th>Time series</th>
<th>Type:</th>
<th>VEC model</th>
<th>1st order differentiated</th>
<th>2nd order differentiated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>The raw data series</td>
<td>t-Statistic</td>
<td>Prob.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GDP</td>
<td>Constant</td>
<td>-0.461219</td>
<td>0.8923</td>
<td>-2.955982</td>
</tr>
<tr>
<td></td>
<td>Constant, Linear Trend</td>
<td>-2.719189</td>
<td>0.2320</td>
<td>-2.945832</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>1.227984</td>
<td>0.9428</td>
<td>-2.332456</td>
</tr>
<tr>
<td>GFCF</td>
<td>Constant</td>
<td>-0.846903</td>
<td>0.7996</td>
<td>-3.795122</td>
</tr>
<tr>
<td></td>
<td>Constant, Linear Trend</td>
<td>-2.926148</td>
<td>0.1603</td>
<td>-3.760478</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.426570</td>
<td>0.8033</td>
<td>-3.642607</td>
</tr>
<tr>
<td>Import</td>
<td>Constant</td>
<td>-0.285973</td>
<td>0.9213</td>
<td>-4.495050</td>
</tr>
<tr>
<td></td>
<td>Constant, Linear Trend</td>
<td>-2.629423</td>
<td>0.2688</td>
<td>-4.480220</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>1.621892</td>
<td>0.9737</td>
<td>-3.867286</td>
</tr>
<tr>
<td>Export</td>
<td>Constant</td>
<td>1.001181</td>
<td>0.9963</td>
<td>-8.355813</td>
</tr>
<tr>
<td></td>
<td>Constant, Linear Trend</td>
<td>-2.073932</td>
<td>0.5523</td>
<td>-8.537736</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>3.695023</td>
<td>0.9999</td>
<td>-7.329138</td>
</tr>
</tbody>
</table>

Source: R output on ADF tests

Cointegration test of the series (Johansen Test)

The cointegration test of the GDP, Export, Import and GFCF series (Johansen Test) indicates the presence of:

- 4 cointegration equations (Trace Test), respectively
- 4 cointegration equations (Maximum Eigenvalue Test).

<table>
<thead>
<tr>
<th>Test Type</th>
<th>No Intercept</th>
<th>Intercept</th>
<th>Intercept</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Trend</td>
<td>No Trend</td>
<td>No Trend</td>
<td>Trend</td>
</tr>
<tr>
<td>Trace</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Max-Eig</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Source: R output on Johansen tests
Selection of the delay order (Schwarz criterion – SC)
The selection of the delay order is based on the Schwarz criterion: the lowest SC coefficient indicates the delay order – in this case the selected delay order is 2.

Schwarz criterion for lag selection

<table>
<thead>
<tr>
<th>Lag</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>67.35190</td>
</tr>
<tr>
<td>1</td>
<td>66.49417</td>
</tr>
<tr>
<td>2</td>
<td>64.97728*</td>
</tr>
</tbody>
</table>

Table 5

Source: R output on Schwarz criterion

Based on the above-mentioned analyzes we decided to apply the VEC model on the 2nd order differentiated data, the 2nd order delay and one cointegration equation.

c. Estimation of the model parameters

The VEC model was performed by using the facilities offered by the function SVEC(), which allows the user to set some parameters, i.e. the cointegration rank, maximum number of iteration, convergence value of algorithm, number of bootstrap replications.

The resulted regression equation is the following:

$$
\Delta^2 \text{GDP} = C_{(1)} + (\Delta \text{GDP}_{(-1)} - 0.7277 \times \Delta \text{GFCF}_{(-1)} - 0.0284 \times \Delta \text{EXPORT}_{(-1)} - 0.4791 \times \Delta \text{IMPORT}_{(-1)} - 260.0107) 
$$

[2]

Short-term Causality Granger under the VEC environment

The short-term Causality Granger under the VEC environment shows that GDP is significantly influenced by imports and GFCF and less significant by exports;

Short-term Causality Granger under the VEC environment

<table>
<thead>
<tr>
<th>Dependent variable: $\Delta^2\text{GDP}$</th>
<th>Excluded</th>
<th>Chi-sq</th>
<th>df</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta^2\text{GFCF}$</td>
<td></td>
<td>61.94723</td>
<td>2</td>
<td>0.0000</td>
</tr>
<tr>
<td>$\Delta^2\text{EXPORT}$</td>
<td></td>
<td>2.379645</td>
<td>2</td>
<td>0.3043</td>
</tr>
<tr>
<td>$\Delta^2\text{IMPORT}$</td>
<td></td>
<td>97.81950</td>
<td>2</td>
<td>0.0000</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>152.1148</td>
<td>6</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 6

Source: R output on Short-term Causality Granger
The empirical results of VEC model

The ANOVA test indicates the validity of the selected model (F-statistic = 203.1143 and Prob (F-statistic) = 0.000000 < 0.05). We noticed that the adjusted R2 is about 0.96 for the first equation, meaning that exports, imports and GFCF determine 96% of GDP and are very close to 1, suggesting that there is a strong link between variables.

Most of the coefficients are significant, indicating that exports, imports and GFCFs influence GDP, both in the short and long term (coefficient of cointegration equation is negative and significant).

The coefficient of the VEC model

<table>
<thead>
<tr>
<th></th>
<th>Coefficient</th>
<th>Std. Error</th>
<th>t-Statistic</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(1)</td>
<td>-4.2806</td>
<td>0.3423</td>
<td>-12.5061</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(2)</td>
<td>2.0418</td>
<td>0.2825</td>
<td>7.2271</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(3)</td>
<td>1.1432</td>
<td>0.1435</td>
<td>7.9648</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(4)</td>
<td>-1.2673</td>
<td>0.3446</td>
<td>-3.6788</td>
<td>0.0005</td>
</tr>
<tr>
<td>C(5)</td>
<td>-1.8065</td>
<td>0.2295</td>
<td>-7.8698</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(6)</td>
<td>0.5773</td>
<td>0.5913</td>
<td>0.9764</td>
<td>0.3322</td>
</tr>
<tr>
<td>C(7)</td>
<td>-0.1905</td>
<td>0.5955</td>
<td>-0.3199</td>
<td>0.7500</td>
</tr>
<tr>
<td>C(8)</td>
<td>-1.8822</td>
<td>0.3773</td>
<td>-4.9885</td>
<td>0.0000</td>
</tr>
<tr>
<td>C(9)</td>
<td>0.7414</td>
<td>0.4060</td>
<td>1.8259</td>
<td>0.0721</td>
</tr>
<tr>
<td>C(10)</td>
<td>-38.3786</td>
<td>199.0471</td>
<td>-0.1928</td>
<td>0.8477</td>
</tr>
</tbody>
</table>

Source: R output on VEC model

The high value of the free term indicates a significant influence of the factors that determine GDP and were not included in the model.

Tests diagnosis for residues
- Autocorrelation of errors
  The Residue Correlation Test (Breusch-Godfrey test) shows that errors are not correlated (does not reject the null hypothesis: there is no correlation of errors – Chi-Square Probability = 0.06 > 0.05)

- Homoscedasticity of random errors
  The Breusch-Pagan-Godfrey and ARCH tests show that the errors are not heteroscedastic (do not reject the null hypothesis: errors are homoscedastic):
  o Breusch-Pagan-Godfrey test (Prob. Chi-Square = 0.07 > 0.05)
  o ARCH test (Prob. Chi-Square(1)=0.97 > 0.05)

- Random errors have normal distribution
  The Jarque-Bera test and the histogram indicate a normal distribution of residues (Jarque-Bera = 1.87 and probability = 0.39 > 0.05 – does not reject the null hypothesis that residues are normally distributed)
The previously generated regression function (GDP function of export, import, and GFCF) has the following characteristics:
- the R-adjusted is close to 1 (0.96), which means that exports, imports and GFCFs determine for 96% GDP and are very close to 1; it results that there is a strong link between variables.
- The ANOVA test indicates the validity of the model (F-statistic = 203.1143 and Prob (F-statistic) = 0.000000 <0.05).
- Most of the coefficients are significant, indicating that exports, imports and GFCFs influence GDP both in the short and long term (coefficient of cointegration is negative and significant).
- Residues are not autocorrelated, have a normal distribution and are homoscedastic

Moreover, the CUSUM test indicates the stability of the regression coefficients:
In conclusion, the previously VEC model can be considered representative to describe autoregressive links between GDP as the dependent variable and export, import and GFCFs as independent variables.

The dispersion of forecasting errors for GDP is explained in the long term by 48-50% GDP, 28-29% of GFCF, 6-7% exports and imports in proportion of 12-16%.

The Variance Decomposition diagram of ΔGDP

<table>
<thead>
<tr>
<th>Period</th>
<th>S.E.</th>
<th>ΔGDP</th>
<th>ΔGFCF</th>
<th>ΔEXPORT</th>
<th>ΔIMPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1777.006</td>
<td>100.0000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>2</td>
<td>2245.774</td>
<td>64.86742</td>
<td>33.14135</td>
<td>1.85387</td>
<td>0.135835</td>
</tr>
<tr>
<td>3</td>
<td>2472.162</td>
<td>53.53104</td>
<td>27.63598</td>
<td>3.951347</td>
<td>14.88163</td>
</tr>
<tr>
<td>4</td>
<td>2483.612</td>
<td>53.32394</td>
<td>27.40623</td>
<td>4.273487</td>
<td>14.99635</td>
</tr>
<tr>
<td>5</td>
<td>3016.434</td>
<td>67.15655</td>
<td>18.68581</td>
<td>3.892076</td>
<td>10.26557</td>
</tr>
<tr>
<td>6</td>
<td>3299.282</td>
<td>56.48911</td>
<td>30.21683</td>
<td>4.704797</td>
<td>8.589270</td>
</tr>
<tr>
<td>7</td>
<td>3461.774</td>
<td>51.31616</td>
<td>27.56136</td>
<td>5.878433</td>
<td>15.24406</td>
</tr>
<tr>
<td>8</td>
<td>3473.028</td>
<td>51.21318</td>
<td>27.42494</td>
<td>5.92749</td>
<td>15.43913</td>
</tr>
<tr>
<td>9</td>
<td>3829.084</td>
<td>59.20562</td>
<td>22.66376</td>
<td>5.362978</td>
<td>12.76764</td>
</tr>
<tr>
<td>10</td>
<td>4043.297</td>
<td>53.15111</td>
<td>29.75909</td>
<td>5.639006</td>
<td>11.45080</td>
</tr>
<tr>
<td>11</td>
<td>4169.178</td>
<td>49.90329</td>
<td>28.03509</td>
<td>6.395765</td>
<td>15.57585</td>
</tr>
<tr>
<td>12</td>
<td>4182.219</td>
<td>49.85006</td>
<td>27.91057</td>
<td>6.443036</td>
<td>15.79633</td>
</tr>
<tr>
<td>13</td>
<td>4458.535</td>
<td>55.37470</td>
<td>24.65510</td>
<td>6.021675</td>
<td>13.94833</td>
</tr>
<tr>
<td>14</td>
<td>4632.615</td>
<td>51.29337</td>
<td>29.66734</td>
<td>6.119223</td>
<td>12.92007</td>
</tr>
<tr>
<td>15</td>
<td>4738.316</td>
<td>49.03117</td>
<td>28.37191</td>
<td>6.731419</td>
<td>15.86550</td>
</tr>
<tr>
<td>16</td>
<td>4751.386</td>
<td>48.87560</td>
<td>28.26495</td>
<td>6.770001</td>
<td>16.08946</td>
</tr>
</tbody>
</table>

Source: R output on Variance Decomposition diagram
The importance of GFCF in GDP is relatively constant over time around 30%. Imports of goods and services in the first quarter of the forecast have a small contribution to GDP (under 1%), increasing very much from the second quarter of the forecast to approx. 15%, which remains relatively constant over time.

4. CONCLUSIONS

The stationary tests of the series (Unit root test – Dickey-Fuller Augmented) indicated that the series are not stationary and become stationary after the first differentiation (export, import and GFCF), respectively, after the second differentiation (GDP). The cointegration test of the series (Johansen Test) indicates the presence of 4 cointegration equations. The Short-term Causality Granger, under VEC environment, shows that there are bi-directional relations of influence between GFCF and GDP, imports and GDP and imports and GFCF. Unidirectional relations exist between: export and import (export influences import and not vice versa).

The VEC model – GDP based on export, import and GFCF – can be considered representative to describe autoregressive links between GDP, as the dependent variable, and export, import and GFCFs as independent variables because: the R adjusted index is close to 1 (0.96) which indicates that there is a strong link between the variables; the ANOVA test indicates the validity of the model (F-statistic = 203.1143 and Prob (F-statistic) = 0.000000 <0.05); most of the coefficients are significant, indicating that exports, imports and GFCFs influence GDP both on a short and long term (coefficient of cointegration equation is negative and significant); the residues are not autocorrelated, have a normal distribution and are homoscedastic; the CUSUM test indicates the stability of the regression coefficients. The coefficients of cointegration equations are negative and significant for all four regression equations, which means that in the long term there are causal bidirectional relations between the four selected variables.

In the long term, exports, imports and GFCF have an effect on GDP growth, with the largest impact on GFCF, followed by imports. In the short term, exports have a positive impact on GDP, and imports and GFCF have a slightly negative impact. All this suggests that Romania must continue to support exports in order to stimulate economic growth. However, it should be remembered that exports of primary products, products based on natural resources and low technology products (pauperization exports) do not lead to sustainable economic growth.
Evolution of Romania’s GDP, GFCF, export and import in the period 1996-2015 (value in million Euro)


The long-term effect of import on GDP growth can be explained by the fact that, through imports, it is allowed faster access to high technology, which is an important positive factor for sustainable economic growth. Another explanation for the long-term effect of imports on GDP growth may be the increase in consumption from import, which does not support sustainable economic growth.

The major long-term impact of gross fixed capital formation on GDP suggests the need to increase GFCF by facilitating investment.

Bibliography


<table>
<thead>
<tr>
<th>PERIOD</th>
<th>GDP</th>
<th>GFCF</th>
<th>Export</th>
<th>Import</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995Q1</td>
<td>6,418.0</td>
<td>1,505.2</td>
<td>1,546.6</td>
<td>1,855.3</td>
</tr>
<tr>
<td>1995Q2</td>
<td>6,811.3</td>
<td>1,367.5</td>
<td>1,735.1</td>
<td>2,138.4</td>
</tr>
<tr>
<td>1995Q3</td>
<td>8,327.6</td>
<td>1,713.7</td>
<td>1,883.3</td>
<td>2,121.4</td>
</tr>
<tr>
<td>1995Q4</td>
<td>7,082.8</td>
<td>1,560.6</td>
<td>2,069.7</td>
<td>2,539.5</td>
</tr>
<tr>
<td>1996Q1</td>
<td>5,875.7</td>
<td>1,120.9</td>
<td>1,755.7</td>
<td>2,127.1</td>
</tr>
<tr>
<td>1996Q2</td>
<td>6,541.7</td>
<td>1,354.1</td>
<td>1,906.6</td>
<td>2,408.0</td>
</tr>
<tr>
<td>1996Q3</td>
<td>8,688.6</td>
<td>1,860.4</td>
<td>1,892.7</td>
<td>2,306.2</td>
</tr>
<tr>
<td>1996Q4</td>
<td>7,866.4</td>
<td>2,288.1</td>
<td>2,095.0</td>
<td>3,027.0</td>
</tr>
<tr>
<td>1997Q1</td>
<td>5,821.3</td>
<td>880.9</td>
<td>1,996.0</td>
<td>2,174.7</td>
</tr>
<tr>
<td>1997Q2</td>
<td>6,666.8</td>
<td>1,249.3</td>
<td>2,247.2</td>
<td>2,711.5</td>
</tr>
<tr>
<td>1997Q3</td>
<td>9,613.6</td>
<td>1,962.9</td>
<td>2,388.3</td>
<td>2,768.6</td>
</tr>
<tr>
<td>1997Q4</td>
<td>9,317.6</td>
<td>2,734.0</td>
<td>2,221.7</td>
<td>3,027.0</td>
</tr>
<tr>
<td>1998Q1</td>
<td>7,704.5</td>
<td>1,092.0</td>
<td>2,135.4</td>
<td>2,387.2</td>
</tr>
<tr>
<td>1998Q2</td>
<td>8,774.1</td>
<td>1,591.5</td>
<td>2,186.3</td>
<td>2,974.1</td>
</tr>
<tr>
<td>1998Q3</td>
<td>11,008.5</td>
<td>2,140.2</td>
<td>2,142.1</td>
<td>2,916.7</td>
</tr>
<tr>
<td>1998Q4</td>
<td>9,658.5</td>
<td>2,115.2</td>
<td>2,094.3</td>
<td>3,089.5</td>
</tr>
<tr>
<td>1999Q1</td>
<td>7,847.1</td>
<td>1,434.2</td>
<td>2,176.7</td>
<td>2,649.7</td>
</tr>
<tr>
<td>1999Q2</td>
<td>9,371.9</td>
<td>1,787.5</td>
<td>2,560.1</td>
<td>2,916.7</td>
</tr>
<tr>
<td>1999Q3</td>
<td>9,592.0</td>
<td>1,913.5</td>
<td>3,640.3</td>
<td>3,313.0</td>
</tr>
<tr>
<td>2000Q1</td>
<td>7,948.1</td>
<td>1,128.4</td>
<td>2,762.2</td>
<td>2,968.4</td>
</tr>
<tr>
<td>2000Q2</td>
<td>9,533.9</td>
<td>1,801.2</td>
<td>3,119.6</td>
<td>3,780.4</td>
</tr>
<tr>
<td>2000Q3</td>
<td>11,218.7</td>
<td>2,365.6</td>
<td>3,591.3</td>
<td>3,885.6</td>
</tr>
<tr>
<td>2000Q4</td>
<td>11,734.8</td>
<td>2,446.3</td>
<td>3,774.8</td>
<td>4,178.0</td>
</tr>
<tr>
<td>2001Q1</td>
<td>8,717.7</td>
<td>1,370.3</td>
<td>3,585.4</td>
<td>4,214.6</td>
</tr>
<tr>
<td>2001Q2</td>
<td>10,871.2</td>
<td>2,553.3</td>
<td>3,816.9</td>
<td>4,896.7</td>
</tr>
<tr>
<td>2001Q3</td>
<td>12,484.6</td>
<td>2,846.3</td>
<td>3,995.3</td>
<td>4,282.5</td>
</tr>
<tr>
<td>2001Q4</td>
<td>13,128.7</td>
<td>2,941.3</td>
<td>3,602.0</td>
<td>5,025.0</td>
</tr>
<tr>
<td>2002Q1</td>
<td>9,728.3</td>
<td>1,555.6</td>
<td>3,812.9</td>
<td>4,350.3</td>
</tr>
<tr>
<td>2002Q2</td>
<td>11,571.2</td>
<td>2,369.7</td>
<td>4,200.7</td>
<td>4,963.6</td>
</tr>
<tr>
<td>2002Q3</td>
<td>13,048.6</td>
<td>3,143.7</td>
<td>4,543.5</td>
<td>5,048.0</td>
</tr>
<tr>
<td>2002Q4</td>
<td>14,106.4</td>
<td>3,379.2</td>
<td>4,578.6</td>
<td>5,489.8</td>
</tr>
<tr>
<td>2003Q1</td>
<td>10,138.8</td>
<td>1,629.2</td>
<td>4,369.3</td>
<td>4,773.6</td>
</tr>
<tr>
<td>2003Q2</td>
<td>12,093.8</td>
<td>2,604.3</td>
<td>4,306.9</td>
<td>5,485.1</td>
</tr>
<tr>
<td>2003Q3</td>
<td>14,760.6</td>
<td>3,656.8</td>
<td>4,837.8</td>
<td>5,590.8</td>
</tr>
<tr>
<td>2003Q4</td>
<td>15,649.3</td>
<td>3,790.3</td>
<td>4,752.9</td>
<td>6,286.9</td>
</tr>
<tr>
<td>2004Q1</td>
<td>11,153.0</td>
<td>1,788.5</td>
<td>4,962.2</td>
<td>5,675.5</td>
</tr>
<tr>
<td>2004Q2</td>
<td>13,849.0</td>
<td>3,105.9</td>
<td>5,376.7</td>
<td>6,749.3</td>
</tr>
<tr>
<td>2004Q3</td>
<td>16,885.4</td>
<td>4,368.0</td>
<td>5,746.5</td>
<td>6,982.5</td>
</tr>
<tr>
<td>2004Q4</td>
<td>19,575.5</td>
<td>4,498.6</td>
<td>5,799.2</td>
<td>7,982.3</td>
</tr>
</tbody>
</table>