Two main uses of R in Statistics Portugal: Estimation and confidentiality

Pedro Miguel Cardoso de Sousa[1]
(pedro.sousa@ine.pt)
Statistics Portugal, Instituto Nacional de Estatística – Delegação Porto,

Inês Rodrigues
(ines.rodrigues@ine.pt)
Statistics Portugal, Instituto Nacional de Estatística – Sede

Maria da Conceição Ferreira
(maria.ferreira@ine.pt)
Statistics Portugal, Instituto Nacional de Estatística – Delegação Porto

Pedro Campos
(pedro.campos@ine.pt)
Statistics Portugal, Instituto Nacional de Estatística – Delegação Porto

ABSTRACT

R has been used in Statistics Portugal since more than 15 years and its use is currently widespread throughout the organization. In this paper, we focus on the use of R within the Statistical Methods Unit, where there are two main areas of R usage: estimation and disclosure control.

For many of our estimation procedures, R is applied as a primary tool: we make use of packages such as RODBC for database access and Survey for data analysis on complex survey samples.

With regard to statistical disclosure control, the use of R in Statistics Portugal is intense, given the recent developments concerning packages for protecting the confidentiality of microdata and tabular data. R package sdcMicro has been a valuable tool in estimating disclosure risk concerning different intruder scenarios in a quick and friendly manner. R has also played a central role in studying and developing techniques for producing Public Use Files for the Household Budget Survey: parametric and non-parametric methods have been compared regarding their capacity to generate safe and useful synthetic data. With respect to Census data, perturbative methods for table protection have been developed, which included writing R functions to check for two priorities when analyzing usefulness: table consistency and additivity. Besides applying R at the Unit, we encourage its use in Statistics Portugal through systematic four-day courses covering some basic commands and

1. Corresponding author

more intermediate features. These allow ever more users to manage, analyze and visualize data using R.

Keywords: *R software, official statistics, estimation, confidentiality*
JEL Classification: *C80, C83, C89*

1. INTRODUCTION

Statistics Portugal is responsible for producing and disseminating, in an effective, efficient and independent manner, high-quality official statistical information relevant for the society. For this purpose, the choice of the software tools for data manipulation and analysis is a key issue in the statistical production process. Besides other specific software, R has been a valuable tool in Statistics Portugal. One of the main reasons why R used is that it offers a wide range of statistical and data handling functionality with a convenient and programmable interface (Kowarik and van der Loo, 2018). The use of R has been encouraged in Statistics Portugal for a long time now as a powerful statistical tool with some relevant aspects such as: (i) popularity and availability on the statistical community; (ii) up-to-date statistical methodologies; (iii) no license/cost of the software; (iv) ability to interact with others statistical software packages; (v) flexibility on handling data (import, export, manipulating). As a learning mechanism and with a view of spreading its use, Statistics Portugal organizes training courses for small groups of users in order to get in touch with R.

The main goal of this paper is to describe the use of R in Statistics Portugal within two specific areas: estimation and statistical disclosure control. For estimation purposes, it is important to start by referring the three main sources from which data may be collected from: (i) exhaustive surveys, in which every item of a defined population is observed; (ii) administrative sources, in which data from administrative procedures is used for statistical purposes; and (iii) sample surveys, in which data is collected from a representative sample of the population under observation. In this last case, the sampling procedure must be based on an adequate sample design. In the majority of these cases, information is obtained as estimates on specific variables and inference to the population, such as in the case of the Labour Force Survey (LFS). Package Survey (Lumley, 2017) is used in Statistics Portugal as a tool to compute LFS monthly estimates. Survey designs are specified using appropriate R functions, and sampling weights are replicated, in order to generate empirically derived standard error estimates that can then be used in the construction of confidence intervals around the sample estimate of interest. Calibration is another important step in the estimation procedure, since it allows for reducing the standard errors of the estimates produced from a survey sample by adjusting the sample's inverse-probability weights

(Kott, 2015). Finally, after calibrating the design weights, it is possible to use other functions included on the Survey package to obtain estimates like totals, means and ratios The variance is estimated by the resampling method provided by Shao and Tu (1995). We provide, later on, some details in terms of the R code regarding the several steps of the estimation procedure.

Concerning statistical disclosure control, it is important to consider that statistical agencies usually release aggregate data and under specific circumstances, microdata containing information on each individual statistical unit. Statistics Portugal and other official agencies may grant access to microdata for scientific purposes and such data are typically in the form of microdata files, therefore containing data on individual units related to persons, households, or companies. Before such dissemination takes place, there is the need to assure that no confidential information can be disclosed from the data. This requires applying statistical disclosure control (SDC) methods (e.g. data suppression or aggregation); the approach to follow – namely, which techniques to apply and aiming at which degree of confidentiality protection – largely depends on the type of product to publish. In addition, many countries have started recently to release Public Use Files (PUF). These PUFs consist of sets of records containing information on individual persons, households or business entities (microdata), but the files are prepared in such a way that individual entities cannot be identified (EUROSTAT, 2018b) We address package sdcMicro (Templ et al., 2015), that is used for the generation of scientific use files, as well as to compute various risk estimation methods. We mention the use of R in the production of microdata files for the Household Budget Survey and refer also to the assessment of post-tabular perturbative methods for protecting Census tables.

These two complementary views of R usage in Statistics Portugal are addressed in this paper, that it is structured as follows: in Section 1 we address estimation issues with the Survey package, and provide some details related to the estimation of monthly estimates of LFS. In Section 2, we show how we use R to deal with statistical disclosure control: the PUF for the Household Budget Survey and the approach of census data. In Section 4, we present other uses of R related to data handling and training. Finally, in Section 5, we provide some final remarks and clues for further direction of the use o R in Statistics Portugal.

2. ESTIMATION WITH SURVEY PACKAGE

Most of the information obtained by Statistics Portugal relies on surveys with probabilistic sampling designs. After having collected the data from the sample, statistical methods are used to extrapolate the information to the

population under study. For many of these procedures involving sampling, extrapolation or estimation, several statistical software packages are used across the institution but the more commonly used are SAS, SPSS, Stata and R. We use Thomas Lumley's R package survey (Lumley, 2017). This package is one the most commonly used packages on sampling, because it has a set of implemented functionalities like survey sample design, weights calibration, resample techniques and estimates calculation (Lumley et al., 2004). One implementation of this package at Statistics Portugal is on the Labour Force Survey (monthly estimates).

2.1. Labour Force Survey (LFS) monthly estimates

Labour force status is the cornerstone concept for labour market statistics, providing good quality estimates in time and change for various labour market outputs and related topics. It is a quarterly survey directed to households, designed to obtain information on the labour market and related issues through a series of personal interviews. The European Union (EU) LFS covers all citizens living in private households and excludes those in collective households, such as boarding houses, residence halls and hospitals. The definitions used are common to all EU Member States and are based on international recommendations by the International Labour Organization (ILO) (EUROSTAT, 2018b).

In 2014, Statistics Portugal initiated a monthly release of Labour Force Survey estimates for the main labour market indicators in addition to the usual quarterly estimates releases. The main goal of this initiative is to provide users to monthly updated information on labour market recent developments, allowing, at the same time, a more complete reading picture than that provided so far by Eurostat in its monthly releases of the unemployed population and unemployment rate estimates for Portugal. Based on monthly estimates of LFS it is possible to obtain monthly estimates related to successive sets of three months (moving quarters), assuring consistency with the quarterly released estimates. These estimates are centred moving quarters, where the reference month (m) corresponds to the central month of each moving quarter (INE, 2018). In fact, the reference month of each press release corresponds to the central month of the quarter composed by m-1, m and m+1 months. As a consequence, the monthly changes are calculated on values that contain common months.

In the following, we describe the several steps of the monthly estimates, using the R package Survey: survey designs, replication of sampling weights and calibration, as well as the variance computation process. Details regarding the several steps of the estimation procedure are provided with R code.

2.2. Sampling and Survey design

The LFS sample is selected from FNA (the national household sampling frame), (FNA), following a stratified/multistage scheme where the primary units (PSUs) are formed by the aggregation of INSPIRE2 grid cells of 1 km^2 (so as to contain at least 300 households) and selected with proportional probability size of the number of main residences. The strata are the geographic units at the NUTS III level (subregions). The sample is a panel type sample with a rotation scheme in which the households remain in the sample for six consecutive quarters. The total sample is divided into six subsamples (rotations) and in each quarter each sub-sample is replaced by another one after having been observed six times.

The svydesign() function creates an object describing the surveys design. In our specific example, AREA is the PSU (geographic area), PESOIN is a variable containing the initial sampling weights and basef is the dataframe that contains the actual data.

```
> desenho <- svydesign(id = ~AREA, weights = ~PESOIN, data =
basef)
> desenho
1 - level Cluster Sampling design (with replacement) With (324)
clusters. svydesign(id =~AREA, weights =~PESOIN, data = basef)
```

2.3. Replicate weights

After creating the object that describes the survey design it is possible to construct a new object containing multiple replications of the design using the command as.svrepdesign(). The parameter type defines the method used to create these replications. Considering a complex design survey or a design with non linear estimators, like in estimators based on the poststratification weights or margin adjustment, where there is not a specific formulation for the variance, multiple methods can be used for the variance calculation. In the specific case of LFS's monthly estimates the used method is the JackKnife, which creates multiple subsamples by removing one or more observations from the initial sample. The replicate weights are based on a stratified Jackknife with modifications to reduce very large weights, account for nonresponse, among other reasons (Lumley, 2004). Besides JackKnife, others methods can be specified like Bootstrap, Balanced Repeated Replicates, Fay or others.

```
> desenho_jk <- as.svrepdesign(desenho, type = "JK1")
> desenho_jk
 1 - level Cluster Sampling design (with replacement) With (324)
clusters. svydesign(id =~AREA, weights =~PESOIN, data = basef)
```

2.4. Calibration

Assuming that the population totals for X, the auxiliary variables available in the sample s, are known, let d_k be the initial sampling weights, which correspond to the inverse of the inclusion probabilities of unit k. The new calibration weights, w_k, are as close as possible (as determined by a certain distance function) to the initial weights d_k. These w_k are calibrated on the totals of the X_p variables. These new weights have to be close to the design weights d_k and satisfy the calibration equations:

$$X = \sum_{k \in s} w_k x_k, \quad [1]$$

where x_k is the value for unit k. The calibration weights are chosen to minimize a given distance measure while satisfying a set of constraints related to the auxiliary variable information. So, the calibrated weights are given by the solution of the optimization equation:

$$Min_{wk} \sum_{k \in s} d_k G(w_k/d_k), \quad [2]$$

under the constraint (1), being the distance function with $r = w_k/d_k$.

We use the *logit* method (Deville and Särndal, 1992) which provides lower limits and upper limits on the weight ratios, using the following distance function:

$$\begin{cases} G(r) = (r-L)\log\left(\frac{r-L}{1-L}\right) + (U-r)\log\left(\frac{U-r}{U-1}\right), & L < r < U, \\ \infty, & Otherwise \end{cases} \quad [3]$$

where L is the lower limit and U the upper limit. Using this method, the total of a given variable of interest is estimated using the calibration estimator for the total:

$$\hat{\tau}_y = \sum_{h=1}^{H} \sum_{k \in s_h} w_k y_k, \quad [4]$$

In this specific case the function calibrate() is used on the previous object containing the 324 sample replicates (desenho_jk) and adjusts the weights according to the population total margins for these disaggregation variables:

E1 - NUTS2, sex and 5-years age groups;

E2 - NUTS3 (or groups of NUTS3) by six age groups;

E3 - NUTS3 (or groups of NUTS3) by sex.

```
> calibra_jk <- calibrate(desenho_jk, make.formula(c(E1, E2,
E3)), est_estr, aggregate.index=~ALOJ, bounds=c(0.25, 4), calfun
= "logit", épsilon = 1e-9)
> calibra_jk
Call: calibrate(desenho_jk, make.formula(c(E1, E2, E3)), est_
estr, aggregate.index=~ALOJ, bounds = c(0.25,4), calfun="logit",
epsilon=1e-9) Unstratified cluster jacknife (JK1) with 324
replicates.
```

The parameters est_estr are the "known" population estimates per stratum and ALOJ are the households IDs.

2.5. Weights after calibration

The function weights() allows us to visualize the result the calibration process. PESOIN are the sample initial weights (d_k) and PESOFIM are the weights after calibration (w_k). d_k / w_k assumes values between 0.25 and 4 as expected and as previously defined by the bounds parameter on the calibrate() command.

```
> wk <- weights(calibra_jk, type = "sampling")
```

On table 1 we can visualize the initial weights and their correspondent adjusted weights.

Example of the Calibration process

Table 1

NUTS2	AREA	ALOJ	SEXO	Q_ETARIOS	G_ETARIOS	AGREG_N3	IDADEE	dk	wk	dk/wk
1	2018	2018_0147	1	1	1	1	2	298,953	538,055	1,800
1	2041	2041_0144	2	12	5	1	2	353,493	380,617	1,077
1	2318	2318_0492	1	3	1	5	2	279,081	236,067	0,846
2	2641	2641_0206	1	14	5	6	2	368,747	324,778	0,881
2	2473	2473_0538	1	8	3	7	2	354,955	720,957	2,031
2	2537	2537_0425	2	11	5	9	2	317,138	312,688	0,986
3	2745	2745_0591	1	16	6	13	2	392,416	302,046	0,770
3	2894	2894_1433	2	2	1	13	2	442,975	426,179	0,962
3	2894	2894_1433	2	10	4	13	2	442,975	426,179	0,962
3	2894	2894_1597	2	9	4	13	2	442,975	492,916	1,113
4	2918	2918_0039	1	1	1	14	2	141,125	223,426	1,583
4	2930	2930_0213	1	3	1	14	2	141,125	124,039	0,879
4	2919	2919_0516	2	14	5	14	2	111,226	61,093	0,549
5	2325	2325_0521	2	4	2	19	2	109,424	133,470	1,220
5	2325	2325_0141	2	7	3	19	2	109,424	198,410	1,813
6	3095	3095_0309	2	6	2	20	1	72,923	53,174	0,729
6	3109	3109_0017	1	1	1	20	2	80,346	75,876	0,944
7	3221	3221_0380	1	17	6	21	2	78,986	62,041	0,785
7	3246	3246_1019	2	3	1	21	2	78,744	70,080	0,890

2.5. Analysis of the variables estimates

After calibrating the design weights it is possible to use several functions included on the survey package to obtain estimates like totals (svytotals()), means(svymeans()) or ratios (svyratio()). 3 The variance of $\hat{\tau}_y$ was estimated by the resampling method (Shao and Tu, 1995).

```
> svyby(POP_ACT, REG_COD_F + SEXO, calibra_jk, svytotal, vartype
= c("var", "cvpct"))
```

Variance and coefficient of variation of the total active population by NUTS II and sex

Table 2

NUTS2	SEXO	POP_ACT	VAR	CV(%)
1	1	940006,27	64224866,07	0,85
1	2	887776,39	91145381,88	1,08
2	1	600436,45	51376995,54	1,19
2	2	548571,36	59188803,81	1,40
3	1	692965,68	44375526,61	0,96
3	2	723397,40	54725716,17	1,02
4	1	184209,38	4663305,22	1,17
4	2	163538,13	7626837,24	1,69
5	1	109588,99	1705448,76	1,19
5	2	110648,40	2211930,20	1,34
6	1	66573,32	1299742,85	1,71
6	2	55847,93	2377161,95	2,76
7	1	66909,97	1787817,69	2,00
7	2	66356,72	2409876,77	2,34

3. STATISTICAL DISCLOSURE CONTROL

3.1. Statistical Disclosure Control (SDC) of microdata

Statistics Portugal may grant access to confidential data for scientific purposes (under specific conditions, which include recognition of the research entity and approval of the research proposal). Such data are typically in the form of microdata files, therefore containing data on individual units. Statistical disclosure control (SDC) methods consist of restricting the amount of, or modifying, the original data, in order to reduce the risk of disclosing information on the statistical units.

Preparing a microdata file for research use requires analysing/ measuring the (re-) identification risk of statistical units and applying SDC

methods to reduce such risk to an appropriate level. (Re-)identification risk is estimated for each unit and considering different 'disclosure scenarios' (cross-tabulations of key/indirect identifying variables – variables for which an intruder may possible have data and based on which he/she would attempt to (re-) identify units in the microdata file). Risk is estimated using functions freqCalc (to compute/estimate sample and population frequency counts) and indivRisk (to estimate the risk for each observation) from R package sdcMicro (Templ et al., 2015); the maximum individual risk, as well as the (absolute and relative) number of records whose individual risk is above a given threshold, are registered for each scenario. We currently use this approach to decide on the adequate level of risk/utility of the protected microdata: several disclosure scenarios are read from an excel file into R, some of which differ only in the degree of detail provided by recoding the same variable in different ways (e.g. age in 5- or 10-year groups).

R is also used for applying several SDC methods (as suppression, global or top/bottom recoding; in particular, package sdcMicro is used for applying microaggregation, in which records are grouped and every record in each group receives the same value (usually the group average), to perturb continuous identifying variables (e.g. turnover) in some business surveys.

3.2. Public Use Files (PUF) for the Household Budget Survey

Public Use Files (PUF) include data on individual statistical units and are to be of public access; therefore, these files are anonymised "in such a way that the statistical unit cannot be identified, either directly or indirectly, when account is taken of all relevant means that might reasonably be used by a third party" (Regulation (EC) No 223/2009 of the European Parliament and of the Council, of 11 March 2009). PUF are intended to be used for education or test purposes (e.g. by researchers when developing their application to access microdata files for research use); no inferences should be drawn from them (valid inferences can only be made from scientific use files).

In order to adequately reduce the risk of (re-)identification, strict SDC methods should be applied (full anonymisation); an option is to generate synthetic data (data simulated from models estimated based on the original data): since synthetic data are not observed, disclosure risk is very low; on the other hand, usefulness can be maintained, namely by keeping the structure/ characteristics of the SUF and its variables and by using models that capture the main relations between the variables in the original data.

A methodology for producing PUF for the Household Budget Survey is being developed by Statistics Portugal, which also consists of generating synthetic data; one synthetic data file is produced by simulating all variables

from its estimated distributions. Our approach is to generate the data based on its sample distribution and, as a final step, re-calibrate sample weights in order to preserve the main population distributions; we opt to only generate values for the synthetic sample, without generating the synthetic population first (which would then need to be sampled to produce the PUF); this was a reason for us not to use package simPop (Templ et al., 2017), which is most adequate to generate synthetic populations, as the main tool to generate the data. Some variables were generated by drawing from their univariate or conditional (conditioned on a factor variable) sample distribution (e.g. NUTS II region and household size); for a main set of variables, values were generated so that their main multivariate relationships were kept. Therefore, two types of models were compared in view of simulating identifying variables, as well as income and expenditure totals, based on their sequential conditional distributions (each variable is estimated by using all previous generated variables as covariates): parametric (multinomial logistic and log-linear regressions) and non-parametric (Classification and Regression Trees – CART) models. Parametric models were estimated by using functions as multinom from package nnet (Venables and Ripley, 2002) (or polr from MASS (Venables and Ripley, 2002) in case of an ordered factor response) and data were generated based on function rMultinom from package Hmisc (Harrell, 2018); CART were fitted by using package rpart (Therneau and Atkinson, 2018) and values were simulated from the trees with partykit (Hothorn and Zeileis, 2015). Since design variables were altered by simulating data, sample weights need to be adjusted; this is performed by using function calibSample from package simPop (Temp et al., 2017). Simple descriptive graphs and main indicators show good results from both methods:

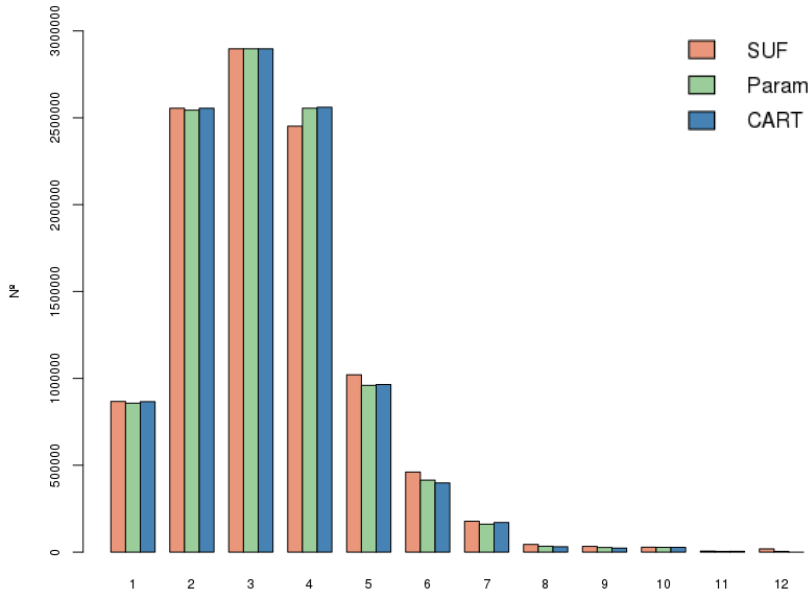
Main indicators computed based on real and synthetic data (HBS 2010/2011)

Table 3

	Median equivalised disposable income (€)	Mean equivalised disposable income (€)	At-risk-of-poverty threshold (€)	At-risk-of-poverty rate after social transfers (%)	Gini coefficient for equivalised disposable income (%)	Income quintile share ratio (S80/S20 (N.º)	Mean annual household total expense (€)
SUF (real)	11 000	13 750	6 600	14.8	33.2	5.2	20 391
Parametric	11 140	13 100	6 684	19.2	31.7	5.1	19 942
CART	10 800	13 279	6 480	15.5	32.6	5.1	19 661

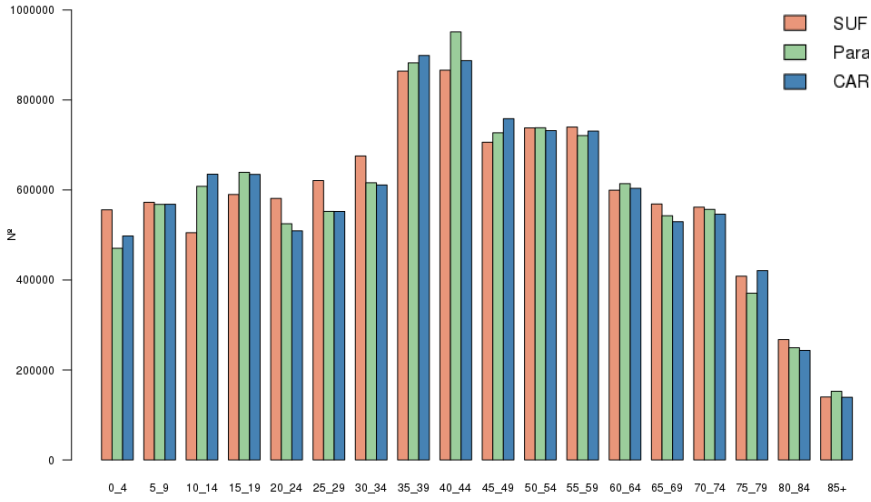
Distribution of household size (nº of persons) from SUF (real) and synthetic data (HBS 2010/2011)

Figure 1



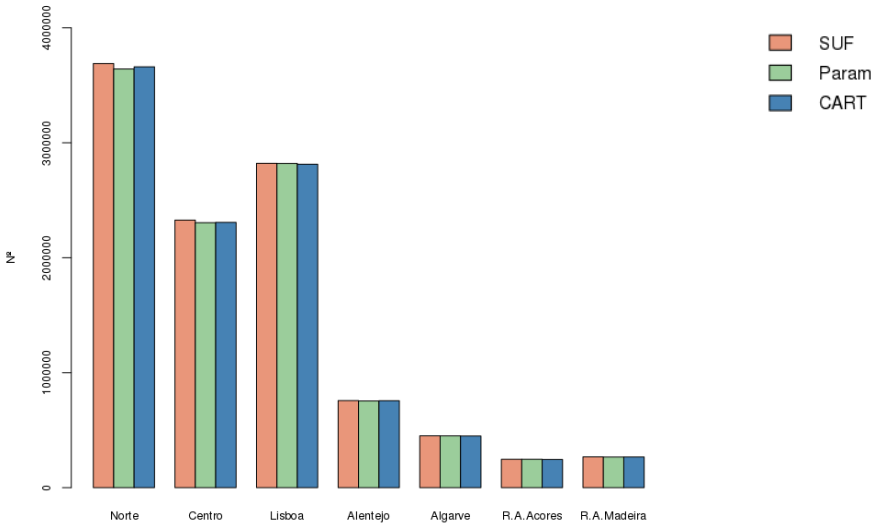
Age group distribution from real and synthetic data (HBS 2010/2011)

Figure 2



Region (NUTS II) distribution from real and synthetic data (HBS 2010/2011)

Figure 3



3.3. SDC for Census data

Confidentiality protection of Census data, both at national and European level, implies some challenges: Census results consist mostly of tabular data whose structure is established beforehand and where several tables have common cells (linked tables); in particular, for Census 2021, another problem is the management of differencing risk between hypercube and grid level data (two classifications whose cells may eventually be subtracted one from the other, resulting in small counts). Such factors make the implementation of non-perturbative methods (methods that reduce the information in the original data through its aggregation or suppression) commonly used to protect tabular data (as cell suppression or recoding) non feasible; as an alternative, perturbative methods (methods where data is modified by purposely including an element of error), as swapping records in the microdata from which tables are obtained or adding random noise to cells, are considered.

Perturbative methods that act on the cells of Census tables (instead of acting on the microdata), called post-tabular methods, can protect against disclosure, but can adversely affect table consistency (the same cell occurring in different tables have the same perturbed value) and/or additivity (marginal totals equal the sum of the corresponding inner cells). With a view of studying methods aiming at protecting Census data, we wrote functions that enable to

easily check table consistency and additivity, in particular for a given set of linked tables (hypercube) (see Annex 1).

4. OTHER USE CASES OF R AT STATISTICS PORTUGAL

4.1. Data handling

Most of the tasks at Statistics Portugal are done with data stored on a Data Warehouse or other Oracle/SQLServer database. The most commonly used tools are included on packages like RODBC or ROracle. When the purpose is to establish a connection to big Oracle databases then ROracle is highly recommended because it is an optimized interface with better performance than the generic RODBC.

```
> library(RODBC)
> con <- odbcConnect("ORACLE-PROD", uid = "user.id", pwd =
"passwd")
> result <- sqlQuery(con, "select * from UNIV_ACT Where YEAR =
2018")
> odbcClose(con)
```

The odbcConnect() function establishes a connection with the ODBC connection called "ORACLEPROD" using the specified credentials. Using the sqlQuery() function it is possible to extract data using SQL commands on a specific connection. In order to avoid unnecessary open connections to a database server the `odbcClose`() function can be used (Ripley and Lapsley, 2017).

```
> library ("ROracle")
> drv <- dbDriver("Oracle")
> connect.string <- "(DESCRIPTION = (ADDRESS = (PROTOCOL = tcp)
(HOST = "ORACLE-PROD") (PORT=1521))(CONNECT_DATA = (SID = dw)))"
> con <- dbConnect(drv, username = "user.id", password = "passwd",
dbname = connect.string)
> result <- dbSendQuery(con, "select * from UNIV_ACT Where YEAR
= 2018")
```

In this case, some extra parameters must be configured but with faster performance (Mukhin et al., 2016).

In situations where data manipulation is needed on bigger datasets and the amount of available RAM is limited, the usage of packages such as data.table (Dowle et al., 2018) or dplyr/dbplyr (Wickham et al., 2016) are recommended. Due the simplicity of the dplyr's pipeline syntax, this second

option is more frequently used. One other very useful feature on the dplyr/dbplyr package is being able to do some data manipulation directly on the data base server using dplyr syntax. This allows us to execute multiple operations on big datasets and transfer to our local computer only the result of these operations, saving local RAM usage.

```
> mtr <- tbl(con, "METERS")
> mtr_result <- mtr
%>% group_by(ID,DATA)
%>% summarise(CONST = sum(CONS,na.rm = T))
> mtr_result
# source: lazy query [?? x 3]
# Database: OraConnection
# Groups: ID
ID DATA CONST
<chr> <chr> <dbl>
1 1 2017-03-01 1428
2 1 2017-03-02 1476
3 1 2017-03-03 1428
4 1 2017-03-04 1060
5 1 2017-03-05 1068
6 1 2017-03-06 1728
7 1 2017-03-07 1744
8 1 2017-03-08 1664
9 1 2017-03-09 1476
10 1 2017-03-10 1668
# ... with more rows
```

The tbl function creates a link to the table "METERS" called mtr using the connection to the database defined by con. This link is an object, commonly called tibble, that allow several types of operations. In this case we are creating a new object called mtr_result containing variable (CONST) with the sum of CONS aggregated by `ID` and `DATE`. This aggregation is being done one directly on the database server using R code that will be translated into SQL commands (visible with function show_query()). By typing mtr_result on R we can view a glimpse of the object. It is possible to transfer all data into R using the function collect().

Occasionally some procedures require importing data included on stand-alone files. If these files are text/csv format normally base package functions like read.csv / read.table do the task, but if files are from third party software like Microsoft Excel, SAS, SPSS or STATA, other packages will have to be used and functions import and export from the package rio are very useful and simple to use (hong Chan et al., 2018).

```
> library(rio)
> data<-import(file="file.sas7bdat")
> export(data, file="output.sav")
> export(data, file="output.tmp", format="SPSS").
```

These functions allow us to import a SAS file, and export the object data into an SPSS file. The extension of the file defines the format to be used (.sav for SPSS) and the parameter format defines the output format when the file extension unrecognized.

4.2. "r" R courses

Statistics Portugal has made an effort setting R a preferred tool among users, by providing two levels of four days courses to their collaborators. The purpose of these courses is to show the potential/usefulness of R in their regular tasks and see R as a versatile tool with no licensing or cost needed. The first level course is a basic approach to R, mainly covering four subjects: R essentials (R interface - the adopted IDE is RStudio), some basic functionalities, basic commands, syntax rules, principal operators and how to get Help in R; objects (object types (mode and attributes), operations with arrays, sequences, indexing, order/sort functions); import/export data (how to import and export several types of files into R, editing data, connection to databases and using RODBC package for importing files); and data analysis (some basic descriptive statistics, graphics and statistical inference). On the second level of the R courses, some of the previous points are debated on a more advanced manner with additional focus on these three points: database access (connecting to databases using packages such a RODBC or ROracle); data visualization (Working with plot, ggplot2 package (Wickham, 2016) and some samples with Shiny Dashboard); advanced data analysis (statistical Inference, variance analysis, linear regression, decision trees and multivariate analysis).

Many of our users already use other software packages in some of their tasks and this allows them to compare the different types of approach. One key feature that make most users a bit skeptical is the command-line user interface rather than point and click menus like other software packages.

5. SUMMARY AND FINAL REMARKS

Over the last two decades R has become on of the most important software used by methodologists and data scientists worldwide. In official statistics, there is a growing active worldwide community of users, where there is wide support from the industry (uROS2018, 2018). Indeed, slowly

but certainly, statistical agencies are introducing R as a valid tool for statistical production as well. This is the case, for example, of the Austrian and Dutch offices, that are amongst the first national statistical institutes to approve R as a tool for production, (Kowarik and vand der Loo, 2018). The authors claim that R is used in their offices for statistical disclosure control for micro and tabular data, for the visualization and imputation of missing data, for analysis of time series in R, and for data editing, among other purposes.In Statistics Portugal R is used mostly for two specific purposes: estimation and statistical disclosure control. We address these two areas with some details regarding several specific applications: for the estimation case, we use package Survey and its current implementation to produce monthly estimates of the on the Labour Force Survey. In terms of statistical disclosure control, we illustrate the use of the package sdcMicro to produce scientific use files, the use of several R packages to produce PUF for the Household Budget Survey and refer also to the assessment of post-tabular perturbative methods for protecting Census tables.

It is difficult to predict how the use of R will be in the near future in Statistics Portugal, but given the amount of people interested in taking the training courses in the last few years, the popularity of R, and the increasing quantity of functions available in different domains of the statistical activity, we would guess that there will be a growing number of procedures and areas where R, for sure, will become essential in Statistics Portugal.

References

1. **Ripley, R.**, and **Lapsley, M.**, 2017, *"RODBC: ODBC Database Access"*. ***https://CRAN.R-project.org/package=RODBC***. R package version 1.3-15.
2. **Chan, Chung hong, Chan, G., Leeper, T.**, and **Jason Becker**, 2018, *"Rio: A Swiss-army knife for data file I/O"*. R package version 0.5.10.
3. **Mukhin, D., James, D. A.**, and **Jake Luciani**, 2016, *"ROracle: OCI Based Oracle Database Interface for R"*. ***https://CRAN.R-project.org/package=ROracle***. R package version 1.3-1.
4. **Deville, J.C.**, and **Carl-Erik Särndal**, 1992, *"Calibration estimators in survey sampling"*. Journal of the American Statistical Association, 87(2):376–382, 7.
5. **EUROSTAT**, 2018a, *CROS*, retrieved from: ***https://ec.europa.eu/eurostat/cros/content/puf-public-use-files_en*** in 2018, September 7th
6. **EUROSTAT**, 2018b, *Statistics Explained*, retrieved from: ***https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Labour_force_survey_(LFS)*** in 2018, September 7th
7. **Wickham, H.**, 2016, *"ggplot2: Elegant Graphics for Data Analysis"*. Springer-Verlag New York,. ISBN 978-3-319-24277-4. ***http://ggplot2.org***.
8. **Wickham, H., Francois, R., L Henry,** and **K Müller**, 2016, *"The dplyr package"*. R Core Team.
9. **Harrell Jr, F. E.**, 2018, *Hmisc: Harrell Miscellaneous*. R package version 4.1-1. ***https://CRAN.R-project.org/package=Hmisc***
10. **Hothorn, T., Zeileis, A.**, 2015, *"partykit: A Modular Toolkit for Recursive Partytioning in R"*. Journal of Machine Learning Research, 16, 3905-3909.

11. **INE (Statistics Portugal)**, 2018, *Press Release, Monthly Employment and Unemployment Estimates*, July 2018
12. **Kowarik, A.**, and **vand der Loo, M.**, 2018, *Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria*, Romanian Statistical Review, nr.1, 2018, pp. 15-29
13. **Kott, P. S.**, 2016, *Calibration weighting in survey sampling, Computational Statistics*, Volume 8, Issue 1, January/February 2016, Pages 39-53
14. **Lumley, T. et al**, 2004, *"Analysis of complex survey samples"*. Journal of Statistical Software, 9(1): 1–19.
15. **Lumley, T**., 2017, *"survey: analysis of complex survey samples"*. R package version 3.32.
16. **Dowle, M.**, and **Srinivasan, A.**, 2018, *data.table: Extension of `data.frame`*. R package version 1.11.4. ***https://CRAN.R-project.org/package=data.table***
17. **Shao, J.**, and **Tu, D.**, 1995, *"The jackknife and bootstrap"*, Springer
18. **Templ, M., Kowarik, A., Meindl, B.**, 2015, *"Statistical disclosure control for microdata using the R package sdcMicro".* Journal of Statistical Software, 67(1), 1-36.
19. **Templ, M., Meindl, B., Kowarik, A., Dupriez, O.,** 2017, *"Simulation of synthetic complex data: The R-package simPop"*. Journal of Statistical Software, 79(i10).
20. **Therneau, T., Atkinson, B.**, 2018, *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13. ***https://CRAN.R-project.org/package=rpart***
21. **uROS2018**, 2018, *The Use of R in Official Statistics*, retrieved from ***https://www.aanmelder.nl/uros2018#.W5J2rM5Kjcs*** in September, 7th, 2018
22. **Venables, W. N., Ripley, B. D.**, 2002 *Modern Applied Statistics with S. Fourth Edition*. Springer, New York. ISBN 0-387-95457-0

Functions to check for table additivity and consistency

Annex 1

```
# ---------------------------------- #
# Group 1 hypercubes
# ---------------------------------- #
# Variables: GEO.N. SEX. AGE.H. LMS.H. HST.H. FST.H.
var_HC1.1 <- list(age.h = age.h, lms.h = lms.h, sex = sex)
var_HC1.2 <- list(age.h = age.h, hst.h = hst.h, sex = sex)
var_HC1.3 <- list(age.h = age.h, fst.h = fst.h, sex = sex)
var_HC1.4 <- list(age.h = age.h, lms.h = lms.h, hst.h = hst.h)

# generate hypercubes
# ---------------------------------- #
ncub <- 4
grp <- 1
HC <- list()
for(h in 1:ncub){
  HC[[h]] <- as.data.frame(addmargins(table(get(paste0("var_HC",
grp, ".", h)))))
}

# ---------------------------------- #
# Consistency
# ---------------------------------- #
# Freq.p are the perturbed frequencies

consist <- function(HC, grp = 1, Freq.p = "Freq.p"){
  for(i in 1:ncub){
     assign(paste("v", i, sep = ""), names(get(paste0("var_HC",
grp, ".", i))))
  }
  dif <- apply(combn(1:ncub, 2), 2, function(x){
me   <-   merge(HC[[x[1]]][HC[[x[1]]][,get(paste0("v",   x[1]))
[!(get(paste0("v", x[1])) %in% get(paste0("v", x[2])))]] %in% "Sum",],
HC[[x[2]]][HC[[x[2]]][,get(paste0("v",  x[2]))[!(get(paste0("v",
x[2])) %in% get(paste0("v", x[1])))]] %in% "Sum",],
by   =   get(paste0("v",   x[1]))[get(paste0("v",   x[1]))   %in%
get(paste0("v", x[2]))])

       me[,paste0(Freq.p, ".x")] - me[,paste0(Freq.p, ".y")]
  })
  if(all(unlist(dif) == 0)){
    consist.t[r] <<- c("OK!: Consistency") } else {
      consist.t[r] <<- c("Warning: No Consistency !")}
}

# ---------------------------------- #
# Additivity
# ---------------------------------- #
# checks for additivity of a table, w.r.t. the global total and
each dimension subtotal
# HC is a list (of tables); Freq.p are the perturbed frequencies
```

```
adit <- function(HC, grp = 1, Freq.p = "Freq.p"){

   tb <- sapply(HC[,names(get(paste0("var_HC", grp, ".", h)))],
function(x){x == "Sum"})
  tab <- HC[apply(tb, 1, function(x){length(x[x]) %in% 2:3}),]
   tab$grupo <- apply(tab[, names(get(paste0("var_HC", grp, ".",
h)))], 1, function(x){
               as.numeric(paste0(which(as.vector(unlist(x))    !=
as.vector(unlist(x))[1]), collapse = ""))})
  tab$grupo[is.na(tab$grupo)] <- 0

  somas <- aggregate(tab[, Freq.p], by = list(tab$grupo), sum)
  if(any(somas[, "x"] != somas[, "x"][1])) {
    adit.t.m[r,h] <<- c(paste0("Warning: No additivity (total) !
- ", "HC1.", h))} else {
       adit.t.m[r,h] <<- c(paste0("HC1.", h, " OK!: Additivity
(total)"))}

   tb2 <- sapply(HC[,names(get(paste0("var_HC", grp, ".", h)))],
function(x){x == "Sum"})
  tab0 <- HC[apply(tb2, 1, function(x){length(x[x]) == 0}),]
  tab2 <- HC[apply(tb2, 1, function(x){length(x[x]) == 2}),]

  adit.v <- sapply(1:length(get(paste0("var_HC", grp, ".", h))),
                   function(y){sapply(unique(get(paste0("var_HC",
grp, ".", h))[[y]]),
                                      function(x) {sum(tab0[tab0[,y]
%in% x, Freq.p]) -
                                                   sum(tab2[tab2[,y] %in%
x, Freq.p])})})

  if(any(lapply(adit.v, sum) != 0)) {
    adit.v.m[r,h] <<- c(paste0("Warning: No additivity (dimension
subtotals) ! - ", "HC1.", h,
                       ": variable ", paste0(names(get(paste0("var_
HC", grp, ".", h)))[which(lapply(adit.v, sum) != 0)],
                                             collapse = ",
"))))} else {
                                                adit.v.m[r,h] <<-
c(paste0("HC1.", h, " OK!: Additivity (dimension subtotals)"))}
}

for(h in 1:ncub){
  adit(HC = HC[[h]], grp = 1, Freq.p = "Freq.p")
}
```