Large-scale Assessments in Education

SOFTWARE ARTICLE Open Access

RALSA: the R analyzer for large-scale assessments

Plamen V. Mirazchiyski[1,2*]

*Correspondence:
plamen.mirazchiyski@ineri.org
[1] Educational Research Institute, Ljubljana, Slovenia
Full list of author information is available at the end of the article

Abstract

This paper presents the R Analyzer for Large-Scale Assessments (`RALSA`), a newly developed `R` package for analyzing data from studies using complex sampling and assessment designs. Such studies are, for example, the IEA's Trends in International Mathematics and Science Study and the OECD's Programme for International Student Assessment. The package covers all cycles from a broad range of studies. The paper presents the architecture of the package, the overall workflow and illustrates some basic analyses using it. The package is open-source and free of charge. Other software packages for analyzing large-scale assessment data exist, some of them are proprietary, others are open-source. However, `RALSA` is the first comprehensive `R` package, designed for the user experience and has some distinctive features. One innovation is that the package can convert SPSS data from large scale assessments into native `R` data sets. It can also do so for PISA data from cycles prior to 2015, where the data is provided in tab-delimited text files along with SPSS control syntax files. Another feature is the availability of a graphical user interface, which is also written in `R` and operates in any operating system where a full copy of `R` can be installed. The output from any analysis function is written into an MS Excel workbook with multiple sheets for the estimates, model statistics, analysis information and the calling syntax itself for reproducing the analysis in future. The flexible design of `RALSA` allows for the quick addition of new studies, analysis types and features to the existing ones.

Keywords: Analysis of large-scale assessment data, Complex sampling design, Complex assessment design, R package, Graphical user interface, Reproducible research, Open-source

Introduction

The Trends in International Mathematics and Science Study (TIMSS), Progress in International Reading Literacy Study (PIRLS) and the Programme for International Student Assessment (PISA) are among the most popular modern international large-scale assessments and surveys (ILSAs). ILSAs provide increasingly more evidence for policy interventions, to initiate or continue reforms in education (Klemencic, 2010; Wagemaker, 2014). ILSAs are conducted repeatedly, in cycles of 3 or more years to provide trends in education outcomes and, besides the data on achievement, collect information on many different variables associated with academic performance (Wagemaker, 2014).

ILSAs' data, however, do not comply with the basic statistical analyses routines researchers typically use because of the complex sampling and assessment designs that the studies employ, necessitating different analysis techniques. The objective of this article is to present a newly developed `R` package—the R Analyzer for Large-Scale Assessments (`RALSA`) for analyzing ILSAs' data, such as TIMSS, PIRLS and PISA, amongst many others. The package can handle the complex sampling and assessment designs of the studies' data automatically, without user intervention. Other software packages for analyzing ILSAs' data exist. One of these is the IDB Analyzer (IEA, 2020) developed by the IEA, a Graphical User Interface (GUI) which produces custom syntax for SPSS (IBM, 2020) or SAS (SAS, 2020) which, in turn, carry out the computations, taking into account the sampling and assessment design issues. While the IDB Analyzer is free of charge, the costs associated with the licenses for SPSS and SAS can be prohibitive for the user. Another software package is Westat's WesVar (Westat, 2008), a free of charge, standalone GUI application which computes statistics from data with complex sampling designs. However, the last release of WesVar was in 2008 and no further updates followed since then. Moreover, the user has to first import the data into WesVar and provide detailed inputs around the sampling design and method of variance estimation (Westat, 2008). The software does not accommodate the complex assessment design of ILSAs and the workflow can be rather daunting for the user.

Different `R` packages for analyzing data from ILSAs exist, free of charge and open-source, licensed under the General Public License (GPL). One of them is `BIFIEsurvey` (BIFIE et al., 2019). Other options are `intsvy` (Caro and Biecek, 2019) and `edSurvey` (Bailey et al., 2020). All of these packages can handle the complex sampling and assessment design issues. However, each one of these packages handles a different set of ILSAs, but not all. A common feature of `BIFIEsurvey` and `EdSurvey` is that they create their own specific objects when data sets are read (Bailey et al., 2020; BIFIE et al., 2019). This may sometimes create difficulties for the analyst, for example, when further data manipulation of variables is needed. `EdSurvey` has its own recoding function (`recode.sdf`) and both `BIFIEsurvey`'s and `intsvy`'s manuals (BIFIE et al., 2019; Caro and Biecek, 2019) provide examples of recoding variables using functions in the base `R` or other packages, but these can at times be overly complicated for the average `R` user. These three packages also read SPSS data files directly. `EdSurvey` also has the `readPISA` function to connect to PISA text files from cycles prior to 2015 while `BIFIEsurvey` and `intsvy` do not have this functionality. The file types `R` works best with are its native `.RData` and `.RDS` files. However, none of the aforementioned packages has the functionality to convert the original data sets into these formats. The existing packages print results directly to the console. `intsvy` is also able to write its outputs in rather basic comma-separated values (`.CSV`) files. None of these packages, however, come with a user interface for convenience of use, especially for users who lack technical skills and experience with `R`.

`RALSA` was designed and developed with the user experience in mind, acknowledging that not every researcher is a programmer. `RALSA` has easy to understand and use syntax. Even for linear and binary logistic regression analyses, the user does not need to specify the model using the traditional `R` formula syntax. `RALSA` also introduces a GUI written directly in `R` without relying on any external platform. The package can

convert original SPSS data sets (or text files in the case of PISA 2012 and earlier cycles) into native `.RData` files. While doing this, it also adds some additional attributes to the objects stored in these files and their variables for further use in data preparation and analysis. This simplifies the entire workflow and eases usage, providing a better experience. `R` supports only one type of missing value natively, `NA`, which is equivalent to system missing values in SPSS. The data conversion function adds the user-defined missing values as an attribute to each variable as found in the original data files. All analysis functions use the additional attributes attached to the data sets and variables, and apply certain routines automatically without any need for additional specification by the user. The comprehensive output system writes the analysis results into MS Excel workbooks with multiple sheets, applying cell formatting depending on the content. In addition, `RALSA` comes with informative log messages and exception handling with human-readable error messages and warnings, which are helpful for the analyst. These are among the advantages `RALSA` has over other software products. While it is not possible to cover all details of `RALSA`, this paper describes in sufficient detail all features, workflow and the architecture of the package.

Background

This section provides a rather brief and concise review to explain the technical and operational complexities and the related analysis issues in PIRLS and PISA (all other studies share more or less similar methodology). Readers interested in a more detailed and technical overview can refer to the technical documentation of the studies (see Martin et al., 2017a; OECD, 2017a).

Due to limited resources, both students and test items are sampled in ILSAs. Although Simple Random Sampling (SRS) has its merits, it is neither practical nor cost effective (Rust, 2014) and it will not permit the efficient linking of students to their teachers and schools to associate student proficiency with classroom and school factors. To maximize precision, and depending on the study, ILSAs use complex sampling techniques including multistage stratified sampling and multistage stratified cluster sampling. Stratification consists of arranging primary sampling units (i.e. schools) in groups (strata) by common characteristics they share, like location in geographic regions or school types. The first stage samples schools within these strata. The probability of selection for each school is different, where larger schools have a greater probability of being selected (probability proportional to the size of the unit, or in short—PPS sampling). The second stage in PIRLS and TIMSS involves sampling one or two intact classes (i.e. clusters) within the sampled schools (LaRoche et al., 2017). Just like in PIRLS and TIMSS, the first stage in PISA is a sample of schools, but in the second stage students within each school are sampled randomly regardless of which class they belong to (OECD, 2017b). As the goal is to provide estimates for the entire target population, these unequal probabilities are also reflected in the sampling weights.

Besides the complex sampling design, ILSAs also use a complex assessment design. This is necessary because a large number of test items is needed to assess student proficiency with a high reliability. In order to measure broad content domains, many items are necessary. To accommodate many items in a limited testing time, complex assessment designs, including matrix sampling, are used. In practice, items are grouped in

blocks by the content and cognitive domains they measure. Then the blocks are distributed across a number of test forms, so that each form contains two or more blocks. The test design is such that forms are linked through common blocks. For more information on the instrument structure, see Martin et. al. (2015) and OECD (2017e), as well as the technical documentation for studies other than PIRLS, TIMSS and PISA. This way no student takes all items. However, student achievement can still be estimated thanks to the complex scaling methodology. Specifically, ILSAs use item response theory (IRT) to estimate the item parameters which are then used with the information provided by the background and context variable in a "conditioning" process. A number of random draws (five in PIRLS and TIMSS, and 10 in later PISA cycles) are made. These are the so-called "plausible" values (PVs) which are the final scores for each tested student. See Foy and Yin (2017) and OECD (2017c), as well as the technical documentation for studies other than PIRLS and PISA. This approach, however, results in some uncertainty, or measurement error, as a result of the imputation process. This is why more than one score (PVs) is assigned to each student. For more information on the PVs and the methodology behind, see von Davier et. al. (2009).

The estimates are produced using the full weight. If PVs are involved, the estimates are computed with each PV and then averaged. The standard errors are computed using the sampling weights and each one of their replicates for each PV, if used. Different studies use different methods for estimating the standard errors depending on their designs. For more information for the studies in scope of this paper see Foy and LaRoche (2017) and (OECD 2017c). For other studies, see their technical documentation.

On the other hand, PISA, as well as some other studies like the Teaching and Learning International Survey (TALIS), use the BRR method for estimating the sampling variance, with Fay's modification (OECD, 2017d). The method uses balanced half-samples. The schools are paired, based on the explicit and implicit stratifications. A total of 80 replicates of the full weight are produced where one of the schools in a zone has its weight multiplied by 1.5 and the other one by 0.5. The formulas for computing the sampling and imputation variance in PISA differ from those in PIRLS (see OECD, 2017c, 2017d) because of the partial inflation and deflation of the weights in a zone. The computational routines, however, are the same: compute with each replicate and each PV, then summarize.

For further details on sampling procedures in ILSAs and PIRLS and PISA in particular, see Rust (2014), LaRoche et. al. (2017), and OECD (2017b). For test design and test development in PIRLS and PISA see Mullis and Prendergast (2017), Martin et. al. (2015), and OECD (2017e). For a general overview and methodology of PVs see von Davier et. al. (2009), for the achievement scaling methodology in PIRLS and PISA see Foy and Yin (2017) and OECD (2017c). For estimation of the sampling and assessment variance in PIRLS and PISA, see Foy and LaRoche (2017) and OECD (2017c, 2017d). For details on other studies see the corresponding technical reports and user guides.

The rather brief description from above should have made clear by now that analyzing data from ILSAs does not comply with the basic statistical routines one may be used to. Using the built-in statistical routines in standard statistical software (SPSS, SAS, `R`, etc.) can lead to biased estimates. Rutkowski et. al. (2010) provide in-depth explanations on ILSAs' methodological and technical complexities and the consequences of ignoring the

design when performing analysis, along with the most common mistakes and misuse (or even abuse) of ILSAs' data.

The RALSA package

Presentation of the package

`RALSA` was first released in November 2020. As of its first release, `RALSA` can work with the data from all cycles of the following studies:

- Civic Education Study (CivED);
- International Civic and Citizenship Education Study (ICCS);
- International Computer and Information Literacy Study (ICILS);
- Reading Literacy Study (RLII);
- Progress in International Reading Literacy Study (PIRLS), including PIRLS Literacy and ePIRLS;
- Trends in International Mathematics and Science Study (TIMSS), including TIMSS Numeracy;
- TIMSS and PIRLS joint study (TiPi);
- TIMSS Advanced;
- Second Information Technology in Education Study (SITES);
- Teacher Education and Development Study in Mathematics (TEDS-M);
- Programme for International Student Assessment (PISA);
- Teaching and Learning International Survey (TALIS); and
- TALIS Starting Strong Survey (a.k.a. TALIS 3S).

Support for more studies will be added in the near future, also on demand. As of now, `RALSA` has the following functionality:

- Prepare data for analysis;
    - Convert data (SPSS, or text in case of PISA prior to 2015);
    - Merge study data files from different countries and/or respondents;
    - View variable properties (name, class, variable label, response categories/unique values, user-defined missing values);
    - Produce diagnostic tables to inspect the frequencies (categorical data) or descriptives (numeric data) prior to analysis to elaborate the analysis plan or hypotheses; and
    - Recode variables.
- Perform analyses
    - Percentages of respondents in certain groups and averages on variables of interest, per group;
    - Percentiles of variables within groups of respondents;
    - Percentages of respondents reaching or surpassing benchmarks of achievement;
    - Correlations (Pearson or Spearman);
    - Linear regression; and

- Binary logistic regression.

The future development of `RALSA` will add more data processing functionality and analysis types to the list from above. The flexible design and architecture of the package (see the next subsection) allow for rapid additions of both studies and functionality. In addition, further optimizations of the existing computational routines will take place (see "Further developments" section).

Architecture and workflow

The purpose of this section is to provide an overview of the architecture of the package and its internal workflow. The section also includes information on the interoperability between the data preparation and analysis functions with a set of background objects and functions used in common operations. A description of the steps the user needs to take and examples on how to use the package, along with description of specific function arguments, are provided in "Some examples of data preparation and analyses with RALSA" section. The architecture and the internal workflow of the `RALSA` package are presented in Fig. 1. The functionality in the package is divided into two major parts: (1) data preparation; and (2) analysis. The main functionality of **the first part** (data preparation) is the data conversion function (`lsa.convert.data`). The function converts the data files originally provided by the organizations conducting ILSAs into native `.RData` data files. The data files come organized in three different ways:

1. SPSS and SAS data sets per respondent type and country. Such are the data files provided for CivED, ICCS, ICILS, RLII, PIRLS (including PIRLS Literacy and ePIRLS), TIMSS (including TIMSS Numeracy), TiPi (TIMSS and PIRLS joint study), TIMSS Advanced, SITES 2006, TEDS-M, TALIS, and TALIS Starting Strong Survey. Each country in PIRLS, for example, has separate files for the different types of respondents—background files for students, their parents, teacher, and school principals. In addition to these, each country has a file with actual student answers on the achievement items, a scoring reliability file, and a student-teacher linkage file (more information on the latter one will be provided further in the paper).
2. SPSS and SAS data sets with all countries in files per respondent type. Such are the files from PISA 2015 and its later cycles. There are several respondent file types, similar to the ones from above, but the SPSS and SAS files per respondent type contain data merged from all countries together.
3. Tab-delimited ASCII text files (`.txt`) with SPSS and SAS import control syntax files (`.sps` and `.sas`). PISA 2012 and its earlier cycles' data are provided in such a way. Each respondent type has one `.txt` data file and a corresponding SPSS and SAS control syntax files to import them in the corresponding software.

Regardless of how the original data are provided, behind the scenes the `lsa.convert.data` function applies appropriate routines automatically to convert the data into native `.RData` files. In case of (3) from above, the function will import a text file, identify the corresponding `.sps` file, import it as well, extract the relevant SPSS command statements and use them to apply the properties (variable

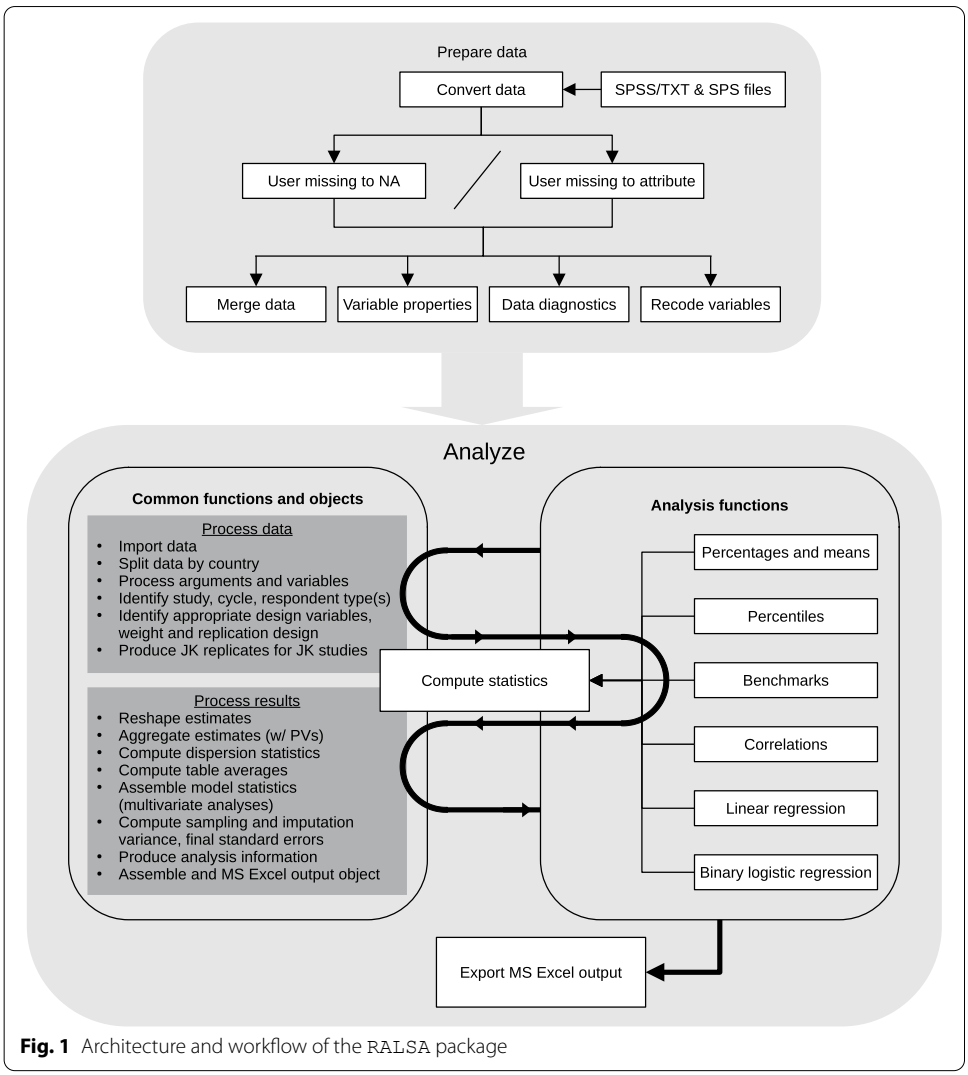


Fig. 1 Architecture and workflow of the `RALSA` package

labels, factor levels, and user-defined missing values) to the variables in the data set. Regardless of how the data files are provided, the function will export the `.RData` files with the same names as the original ones. The resulting `.RData` files contain an object with the same name as the file name. The object is of class `data.table`, an extension of `data.frame`, and now further extended to `lsa.data` class. That is, instead of creating a new object class, as other packages for analyzing ILSAs' data do, the `data.table` class is only extended with additional attributes, making all commands for working with `data.table` and `data.frame` still applicable. The `lsa.convert.data` adds the following attributes to the object:

- `study`—the name of the study;
- `cycle`—the cycle of administration of the study;
- `file.type`—the respondent type (i.e. student, teacher, school, etc.);

In addition, the `lsa.convert.data` is keyed on the country ID (`CNT` in PISA and `IDCNTRY` in all other studies) variable which is further utilized in merging and all analysis functions. R has just one missing value type (`NA`) while SPSS files have user-defined missing codes as well. The `lsa.convert.data` function can convert the data including the user-defined missing values as the default behavior. The function will assign a `missings` attribute to each variable. If not interested in the user-defined missing values, the user may add `missing.to.NA = TRUE` to the function call to set all user-defined missing values to `NA`, analogous to "system missing" in SPSS. Each variable will also have its variable label assigned to a `variable.label` attribute. This is different from the usual practice in `R data.frame`s where the variable labels are assigned to the data sets and not to the individual columns. Assigning them directly to the variables makes it possible to view, maintain and use the variable properties in merging, recoding and analyzing data without loss of information, or having redundant or irrelevant information in the data set. See "Some examples of data preparation and analyses with RALSA" section for examples on using the `lsa.convert.data` function.

Another core data preparation function is the `lsa.merge.data`, a function to merge data from different countries and different respondents within a cycle of a study. Note that this function can work with all studies, except PISA. The main reason, as noted earlier, is that all data sets in a PISA cycle are provided per respondent type, containing the data for all countries in a file. In addition, some of the components in PISA (e.g. teacher questionnaire) are optional for the countries, so different respondent types would contain different sets of countries. Also, respondent types and data file naming conventions change from one cycle to the other. This makes is extremely hard, if even possible, to handle programmatically the merging of PISA data sets consistently with each subsequent release of the study. The advantages of the merging function and the reasons why it shall be used instead of making any merges of ILSA data by hand are briefly outlined here. When merging data from different respondents in different studies, the merged files need to keep the appropriate weight variables for the combination of respondents. For example, when merging student and school data, the weight to use in the analysis must be the student weight because the school and principal characteristics become properties of the student. Merging data from different respondents will depend on the design of the study. Some merge combinations are possible in some studies and not in others because of the target populations and the content domain. For example, in PIRLS 2016 teachers do not constitute a representative sample of teachers in the population (see LaRoche et al., 2017). The sampled teachers are the ones teaching the sampled students, which constitute a representative sample of the students in an education system. In this case, the appropriate weight is the teacher weight, which is derived from the student weight after applying some adjustments (LaRoche et al., 2017). The results must be interpreted on the level of students with teachers having particular attributes. Thus, the teacher data must always be merged with the student data before analysis (Foy, 2018). The databases provide student-teacher linkage files; the information in these files is used to ensure a proper linkage in the merge process, matching the students to their teachers by IDs and linkage indicators. These student-teacher linkage files are added automatically during the merge process and the `lsa.merge.data` function takes care for the proper ID matching during the merge process. TIMSS grade 4 has the

same scenario as PIRLS. TIMSS grade 8, however, is more complicated because students have different teachers in mathematics and science and both cannot be merged at the same time. Additionally, in some countries more than one teacher teaches the students in any of the two given subjects. If the analyst tries to merge only teacher files in PIRLS or TIMSS, the merging function will stop with an error message. In ICCS and ICILS, on the other hand, teachers from the sampled schools are sampled separately from the students as a representative sample and are not necessarily teaching the students sampled from their schools. Teacher data only from different countries can be merged, but merging student and teacher data should not be done because no link between them can be established (Brese, 2018; Mikheeva and Meyer, 2020). The `lsa.merge.data` function always checks the possible merge combinations for the study data and stops with an error if a requested merge combination is not applicable. The `lsa.merge.data` function takes care to properly merge the files through all the different scenarios the studies have. It also takes care of maintaining the user-defined missing values assigned to each variable in the different data sets. When any merge is done, the function will also update the `file.type` attribute in the new merged data set (i.e. which respondent types are merged). Later, the analysis functions use the `file.type` attribute, as well as the other attributes (`study` and `cycle`), to apply the appropriate computation methods. See "Some examples of data preparation and analyses with RALSA" section for examples on how to merge data from different file types from different countries.

There are three more helper functions in the data preparation part of `RALSA`. The first one is the variable recoding function (`lsa.recode.vars`). This function can recode variables in the converted and subsequently, merged data sets. In doing this, it also takes care of the user-defined missing values, if there are any. The second helper function, `lsa.var.dict`, prints variable dictionaries. This is similar to the `DISPLAY DICTIONARY` command in SPSS. This function is helpful for planning analyses or when information on variables is desired. The function can print the variable dictionaries in the console and save them in a text file for further reference. The third helper function, `lsa.data.diag`, can be used for initial exploration of the data prior to analysis, to inspect the actual distribution of values within variables and elaborate the analysis plans. It creates frequency tables for factor variables, and descriptive statistics tables for numeric variables. These tables are always split by country ID and more splitting variables can be added, if needed. The output is exported to a multi-sheet MS Excel workbook with one sheet for each variable of interest.

The **second set** of functions in `RALSA` is the analysis functions (see "Presentation of the package" section for the full list, as of the time of writing, and "Some examples of data preparation and analyses with RALSA" section for examples of using the analysis functions). As Fig. 1 shows, all analysis functions use a set of common objects and functions that they use in the background. These are available in a separate file (`common.r`) and are not to be used on their own, but called only from the analysis functions available in the `RALSA` package. This is why the workflow, designated by the curved arrow line in the middle, goes back and forth between the "Analysis functions" and the "Common functions and objects". When a call to any analysis function is made, it initially refers to the "Process data" group of functions and objects in "Common functions and objects" in the `common.r` file. After all the data is processed (import data, split by

country, etc.), the resulting data object is returned to the analysis function. The analysis function applies all arguments passed to it and proceeds with computing the statistics. Note that the "Compute statistics" box on Fig. 1 overlaps the "Analysis functions" and the "Common functions and objects" boxes. This is because the "common.r" file contains functions and objects related to all statistics types and many of the analysis functions also contain nested functions and objects pertinent to them only. After all statistics for an analysis are computed, the results are passed to functions in the "common.r" file from within the analysis function. These common functions further process the results including functions such as reshape, aggregate, compute additional statistics, assemble the model statistics, compute the sampling and imputation variance, and compute the final standard errors. Eventually, the analysis function makes a call to a common function to create an output object to be written to the disk as an MS Excel (2007 or later) workbook with multiple sheets.

This approach, having common functions and objects shared between the analysis functions, offers significant advantages:

- A function for a certain common operation (say `produce.jk.reps.data` which produces the JK2 replicate weights for studies like PIRLS and TIMSS) located in "common.r" is developed just once, but is called and used within any analysis function;
- If any computational routine, used by all analysis functions, needs to be updated, this is done just once, but benefits all analysis functions;
- Consistency of all computations that different functions have in common;
- Avoids code repetition and minimizes the risks of mistakes and inconsistencies; and
- Makes the time for new developments much shorter.

For example, all analysis functions compute the sampling and imputation variance and produce the final standard errors of the estimates. The functions for doing this are located in the "common.r" file, written once and used everywhere with no or minimal updates when necessary. The last item in the list above is especially important because the common functions and objects in the "common.r" file are used to perform routine operations, pertinent to all analysis functions. For example, the `produce.jk.reps.data` function will produce JK2 replicate weights regardless of whether the analysis function called is for computing means or logistic regression. This, and many other functions and objects in the "common.r" file facilitate then quick addition of new analysis types in `RALSA`.

To keep the package's syntax consistent, all of the analysis functions have a common set of arguments, applicable to all of them. The arguments `data.file` and `data.object` (either one of them shall be used), `split.vars`, `weight.var`, `include.missing`, `shortcut`, `output.file`, and `open.output` are common in all analysis functions. In addition, linear and binary logistic regression share the following arguments: `bckg.indep.cat.vars`, `bckg.cat.contrasts`, `bckg.ref.cats`, and `standardize`. At the same time, each function has its own arguments that are pertinent to it. To see the full list of arguments for each function and their meaning, refer to the `RALSA` reference manual (Mirazchiyski and INERI, 2021). All of the examples below

are done under MS Windows 10. RALSA, however, can work under any operating system where full installation of R is possible.

As noted earlier, RALSA comes with a GUI, which adds to its usability. The GUI is written entirely in R, using the shiny package (Chang et al., 2020) and its add-ons, i.e. without relying on any additional platform or programming language. shiny is an HTML wrapper with reactive bindings between inputs and outputs, which allows for the quick building of statistical web applications. The RALSA GUI is a web application that interacts with the data preparation and analysis functions. It runs locally, on the user's computer, in a browser. All of the functions in the RALSA package can be accessed and used through the GUI.

Some examples of data preparation and analyses with RALSA

This section provides some examples of preparing and analyzing ILSAs data using the RALSA package. Due to page limitations, the examples are not exhaustive and will use the command-line option of RALSA. All of the package functionality, however, can be used through the GUI as well. The examples here demonstrate how to use the package from the first step (converting the data) to the last one (performing an analysis). Regardless of the analysis the analyst needs they follow the same steps. Examples for each specific functionality of the package can be found at the RALSA support website (INERI, 2021). To use RALSA, an analyst needs to follow these steps:

1. Convert data. The original data is provided in SPSS files or TXT files with control syntaxes. The user first needs to convert the data into R file format (see below). This needs to be done just once when a study's data is used for the first time.
2. Merge data. ILSAs are large comparative studies collecting data from many education systems and different kinds of respondents. Depending on the study, data from different kinds of respondents can be merged together in order to answer specific research questions for a specific group.
3. View variable properties (optional). Prior to analysis, an analyst may need to inspect the properties of the variables—class (factor or numeric), response options/unique values, etc.
4. Produce data diagnostic tables (optional). Sometimes the analyst may need to produce descriptive statistics and inspect the data prior to the actual analysis to elaborate the analysis plan or hypotheses;
5. Recode variables (optional). In some cases, the analyst may need to modify the data prior to the analysis—collapse variables' response categories or reverse-reverse code variables.
6. Perform the actual analysis. Multiple analysis options are provided by RALSA (see "Presentation of the package" section).

The examples that follow until the end of this section are basic and reflect the interests of a hypothetical researcher who wants to explore the differences in reading achievement between students based on their gender, using PIRLS 2016 data from Australia, Bulgaria and Slovenia. Let's first load the package, assuming it is already installed (see the installation instructions at the RALSA support site (INERI, 2021):

```
library(RALSA)
```

Upon loading, the package will show some welcome messages. As noted earlier, the first and foremost function in the package is the data conversion function. The first example converts PIRLS 2016 SPSS data files for Australia, Bulgaria and Slovenia which will be used in the further examples:

```
lsa.convert.data(inp.folder = "C:/PIRLS_2016_SPSS_Files",
                 ISO = c("aus", "bgr", "svn"),
                 out.folder = "C:/Converted_Data_PIRLS_2016")
```

In the call to the `lsa.convert.data` from above we had to specify only the source folder (i.e. where the original SPSS files are located), the output folder (i.e. where the converted `.RData` files will be saved) and a vector of country abbreviations (fourth, fifth and sixth characters in the file names) passed to the `ISO` argument; omitting the `ISO` argument would instruct the function to convert the files for all countries in the folder. Note that the `ISO` argument is case insensitive, i.e. the country ISO codes can be either upper- or lower-case. The source files can also have their names and extensions in either upper- or lower-case. Executing the syntax from above will print the following log messages (presented here partially) to the console:

```
21  datasets found for conversion. Some datasets can be rather
    large. Please be patient.

     (  1/21)   acgausr4.sav converted in 00:00:01.489
     (  2/21)   acgbgrr4.sav converted in 00:00:01.139
     (  3/21)   acgsvnr4.sav converted in 00:00:01.160
 . . .
[output clipped]
 . . .
     ( 19/21)   atgausr4.sav converted in 00:00:01.300
     ( 20/21)   atgbgrr4.sav converted in 00:00:01.239
     ( 21/21)   atgsvnr4.sav converted in 00:00:01.180

 All 21 found files successfully converted in 00:00:35.970
```

Each of the three countries has seven files for the different respondent types. All data sets were converted in about 36 s. Now that the files for the three countries in PIRLS 2016 are converted, they can be used with all other functions in the package. The following code merges the converted student data files together:

```
lsa.merge.data(inp.folder = "C:/Converted_Data_PIRLS_2016",
               file.types = list(asg = NULL),
               out.file = "C:/Merged/Merged.RData")
```

The `file.types` argument takes a list of named vectors. The respondent type `asg` (case insensitive, can be upper- or lower-case) stands for student files. `NULL` as a value for each file type instructs the function to take all variables in these file types. Individual variables of interest can be specified as well, passing them as vectors to the file type names. Depending on the combination of respondents, the function will take only the proper design (ID, tracking, sampling and PV) variables from each of the respondents' files. The syntax from above does not specify data from which countries should be merged using the `ISO` code, so the function will merge all converted student files in the folder (three countries in this case). Executing the code from above prints the following log in the console:

```
Data files from 1 respondent types are to be merged.

Data files from 3 countries are to be merged.

   "asg" files imported in 00:00:01.029

   "asg" cases added together in 00:00:00.000

   Files from all respondents merged in 00:00:01.010

   The merged data file "Merged.RData" is saved under "C:/
       Merged" in 00:00:01.010

All operations finished in 00:00:01.349
```

The total time for all merging operations (three countries, two respondent types) took less than 1.5 s. The examples that follow until the end of this subsection use data from this merged file, containing student data from Australia, Bulgaria and Slovenia.

Executing the syntax from below computes the average student overall reading achievement by student gender (sex) in each country, a classical example, in all three countries, using the merged file from above:

```
lsa.pcts.means(data.file = "C:/Merged/Merged.RData",
               split.vars = "ASBG01",
               PV.root.avg = "ASRREA")
```

Note that the arguments in all analysis functions are case-sensitive to the variable names. The variable names are set to upper-case during the conversion process and

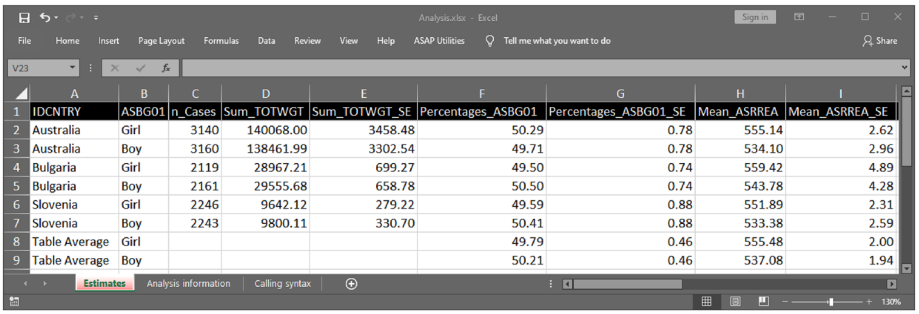
must be supplied as such. The call to the function prints log messages to the console every time data for a specific country has been processed and estimates are produced, as well as a message for adding the country achievement averages and the total time the analysis took:

```
Data file "C:/Merged/Merged.RData" imported in 00:00:01.029

Valid data from 3 countries have been found. Some computations
    can be rather intensive. Please be patient.

      ( 1/3)   Australia       processed in 00:00:01.063
      ( 2/3)   Bulgaria        processed in 00:00:01.805
      ( 3/3)   Slovenia        processed in 00:00:01.016

All 3 countries with valid data processed in 00:00:03.884
"Table Average" added to the estimates in 00:00:01.010
```

A few things have to be noted here. First, the overall reading achievement has five PVs—ASRREA01, ASRREA02, ASRREA03, ASRREA04, and ASRREA05. Only the root (ASRREA) name without the numbers is supplied to the PV.root.avg argument.[1] Behind the scenes, the function identifies all PVs with the same root, computes the averages for each one of them with the full weight and each of its replicates, and aggregates the final results, computing the standard error. Second, a weight variable was not specified using the weight.var argument. In such cases all analysis functions identify the appropriate weight to apply, depending on which respondents' data are in the file. The variable to weight by can be changed, of course, but an analysis function will check if the variable passed as a weight really belongs to the set of weighting variables for the study and cycle and, if not, will immediately exit with an error message as part of the exception handling mechanisms. Third, a full file path for saving the MS Excel output was not specified for the output.file argument, so any analysis function will save the output in the working directory (can be obtained by using the getwd() command or specified with the setwd() command). By default, all functions will open the MS Excel output once it is saved.

Figure 2 shows the partial output from this analysis. The MS Excel workbook has three sheets. The first sheet contains the estimates per country and the average estimates from all countries used in the analysis. The first two columns represent the variables we split the analysis by. Note that the calling syntax from above specified only student sex (ASBG01) as a splitting variable, but all analysis functions add the country ID as the first one automatically, so all analyses are performed by country. These two columns are followed by the number of sampled cases with valid data and their population estimates,

[1] In studies like PISA, ICCS and ICILS the PVs have a different naming convention. For example, in PISA the names of the set of 10 PVs are PV1MATH, PV2MATH, PV3MATH,...and PV10MATH. The root PV name has to be specified in PV.root.avg as "PV#MATH". For more details, see RALSA's reference manual.



	A	B	C	D	E	F	G	H	I
1	IDCNTRY	ASBG01	n_Cases	Sum_TOTWGT	Sum_TOTWGT_SE	Percentages_ASBG01	Percentages_ASBG01_SE	Mean_ASRREA	Mean_ASRREA_SE
2	Australia	Girl	3140	140068.00	3458.48	50.29	0.78	555.14	2.62
3	Australia	Boy	3160	138461.99	3302.54	49.71	0.78	534.10	2.96
4	Bulgaria	Girl	2119	28967.21	699.27	49.50	0.74	559.42	4.89
5	Bulgaria	Boy	2161	29555.68	658.78	50.50	0.74	543.78	4.28
6	Slovenia	Girl	2246	9642.12	279.22	49.59	0.88	551.89	2.31
7	Slovenia	Boy	2243	9800.11	330.70	50.41	0.88	533.38	2.59
8	Table Average	Girl				49.79	0.46	555.48	2.00
9	Table Average	Boy				50.21	0.46	537.08	1.94

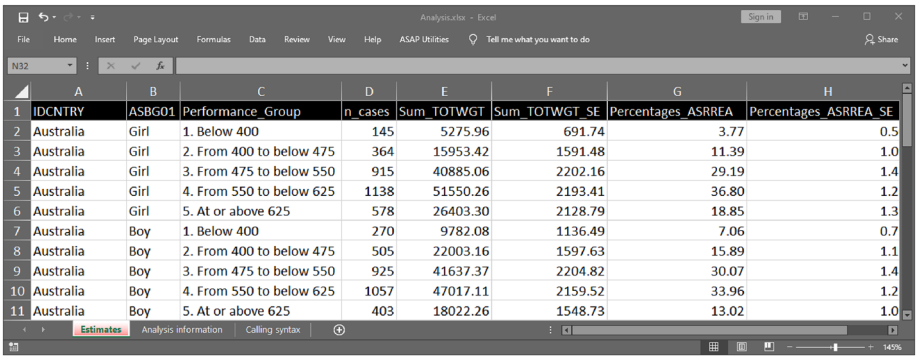
Fig. 2 Partial output from the `lsa.pcts.means` function

and their standard errors. The last two columns in Fig. 2 represent the average overall achievement scores estimated using the PVs for female and male students per country and the standard errors of the estimates. The rest of the columns (not displayed in the figure) contain the variance (sampling and measurement) terms, dispersion and percentage of missing values by groups specified by the splitting variables. The second sheet contains some information about the analysis: which data file was used, which countries it contains, which study and cycle, which weight variable and replication method, how many replicates, start, end time and duration per country. The last sheet contains the calling syntax which was executed to perform the analysis. This is useful if later the analysis needs to be repeated, for example, when the data was updated or some recoding was made.

The next analysis computes the percentage of students reaching all different performance levels (benchmarks) by student sex for the overall reading achievement. This is the calling syntax:

```
lsa.bench(data.file = "C:/Merged/Merged.RData",
          split.vars = "ASBG01",
          PV.root.bench = "ASRREA")
```

Note that the benchmark levels were not specified in the syntax above using the `bench.vals` argument, so defaults (400, 475, 550 and 625) are used automatically. Different benchmark values are applied as default, depending on the study. In the case of PISA and ICCS the defaults will also depend on the specific cycle, because different cycles have different definitions of proficiency levels. In addition, in PISA the different achievement scales also have different proficiency levels and the default values will depend on the set of PVs used in the analysis. The benchmark values can, of course, be changed by adding, for example, `bench.vals = c(475, 550)` to the syntax from above. Other functions also have default values for different arguments, and any of the defaults for any of the functions can be changed.

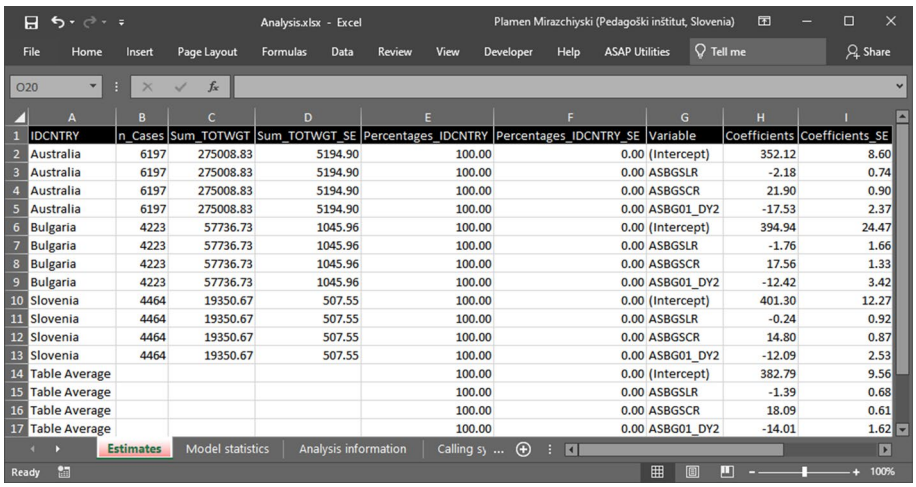


	A	B	C	D	E	F	G	H
1	IDCNTRY	ASBG01	Performance_Group	n_cases	Sum_TOTWGT	Sum_TOTWGT_SE	Percentages_ASRREA	Percentages_ASRREA_SE
2	Australia	Girl	1. Below 400	145	5275.96	691.74	3.77	0.5
3	Australia	Girl	2. From 400 to below 475	364	15953.42	1591.48	11.39	1.0
4	Australia	Girl	3. From 475 to below 550	915	40885.06	2202.16	29.19	1.4
5	Australia	Girl	4. From 550 to below 625	1138	51550.26	2193.41	36.80	1.2
6	Australia	Girl	5. At or above 625	578	26403.30	2128.79	18.85	1.3
7	Australia	Boy	1. Below 400	270	9782.08	1136.49	7.06	0.7
8	Australia	Boy	2. From 400 to below 475	505	22003.16	1597.63	15.89	1.1
9	Australia	Boy	3. From 475 to below 550	925	41637.37	2204.82	30.07	1.4
10	Australia	Boy	4. From 550 to below 625	1057	47017.11	2159.52	33.96	1.2
11	Australia	Boy	5. At or above 625	403	18022.26	1548.73	13.02	1.0

Fig. 3 Partial output from the `lsa.bench` function

The partial output from executing the code from above is shown on Fig. 3. Just as with the previous analysis, the first two columns contain the groups defined by the splitting variables—country ID (added automatically) and student sex. The next column contains the benchmark levels for each group. These are followed by the number of sampled cases with valid data and the population estimates for the number of students with their standard errors. The next column presents the percentages of female and male students performing within the ranges of score points defined by the benchmark values and their standard errors. The other two sheets contain the same information as the ones in the output from the previous analysis—information about the analysis and the calling syntax.

`RALSA` can perform not just univariate statistical analyses with ILSA data. As of the time of writing, correlation, linear regression and binary logistic regression are available as well. The code below computes linear regression coefficients for a model where the student overall reading achievement `ASRREA` is a function of the student sex (`ASBG01`), how much Students Like Reading scale (`ASBGSLR`) and Students Confident in Reading scale (`ASBGSCR`). That is, the analyst will test if the differences between male and female students are statistically significant when controlling for these scales. The student sex is a dichotomous factor (i.e. categorical) variable. The two other variables are complex scales constructed using multiple student background variables (for more details see Martin et al., 2017b). The syntax for this analysis looks like this:



Analysis.xlsx - Excel — Plamen Mirazchiyski (Pedagoški inštitut, Slovenia)

File Home Insert Page Layout Formulas Data Review View Developer Help ASAP Utilities Tell me Share

O20

	A	B	C	D	E	F	G	H	I
1	IDCNTRY	n_Cases	Sum_TOTWGT	Sum_TOTWGT_SE	Percentages_IDCNTRY	Percentages_IDCNTRY_SE	Variable	Coefficients	Coefficients_SE
2	Australia	6197	275008.83	5194.90	100.00	0.00	(Intercept)	352.12	8.60
3	Australia	6197	275008.83	5194.90	100.00	0.00	ASBGSLR	-2.18	0.74
4	Australia	6197	275008.83	5194.90	100.00	0.00	ASBGSCR	21.90	0.90
5	Australia	6197	275008.83	5194.90	100.00	0.00	ASBG01_DY2	-17.53	2.37
6	Bulgaria	4223	57736.73	1045.96	100.00	0.00	(Intercept)	394.94	24.47
7	Bulgaria	4223	57736.73	1045.96	100.00	0.00	ASBGSLR	-1.76	1.66
8	Bulgaria	4223	57736.73	1045.96	100.00	0.00	ASBGSCR	17.56	1.33
9	Bulgaria	4223	57736.73	1045.96	100.00	0.00	ASBG01_DY2	-12.42	3.42
10	Slovenia	4464	19350.67	507.55	100.00	0.00	(Intercept)	401.30	12.27
11	Slovenia	4464	19350.67	507.55	100.00	0.00	ASBGSLR	-0.24	0.92
12	Slovenia	4464	19350.67	507.55	100.00	0.00	ASBGSCR	14.80	0.87
13	Slovenia	4464	19350.67	507.55	100.00	0.00	ASBG01_DY2	-12.09	2.53
14	Table Average				100.00	0.00	(Intercept)	382.79	9.56
15	Table Average				100.00	0.00	ASBGSLR	-1.39	0.68
16	Table Average				100.00	0.00	ASBGSCR	18.09	0.61
17	Table Average				100.00	0.00	ASBG01_DY2	-14.01	1.62

Estimates | Model statistics | Analysis information | Calling sy ...

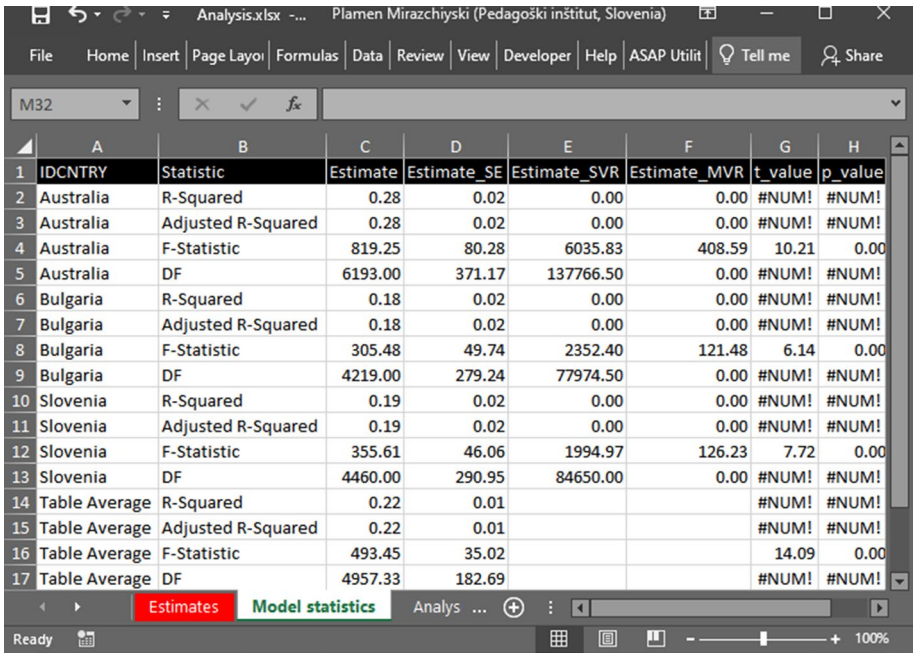
Ready 100%

Fig. 4 Partial output from the `lsa.lin.reg` function

```
lsa.lin.reg(data.file = "C:/Merged/Merged.RData",
            PV.root.dep = "ASRREA",
            bckg.indep.cat.vars = "ASBG01",
            bckg.indep.cont.vars = c("ASBGSLR", "ASBGSCR"))
```

Note how the independent variables are specified. The student sex is a categorical variable and is passed to `bckg.indep.cat.vars`. For variables passed as categorical, the function uses contrast coding. For now, `RALSA` supports `dummy`, `deviation` and `simple` contrast coding schemes. When `dummy` is used, the intercept is the average on the dependent variable for the respondents choosing the reference category and the slopes are the differences between intercept and the average of respondents on the dependent variable choosing every other category.[2] The syntax from above does not specify any value for the `bckg.cat.contrasts` argument (`dummy`, `deviation`, or `simple`). Thus, `dummy` is used by default. The syntax also does not use the `bckg.ref.cats` to specify which category in `bckg.ref.cats` shall be used as a reference one. Thus, the first category in `ASBG01` ["Girl"] is used by default. The function automatically identifies the number of distinct levels in the variables passed to

[2] See the `RALSA` reference manual for description of the other contrast coding schemes.



	A	B	C	D	E	F	G	H
1	IDCNTRY	Statistic	Estimate	Estimate_SE	Estimate_SVR	Estimate_MVR	t_value	p_value
2	Australia	R-Squared	0.28	0.02	0.00	0.00	#NUM!	#NUM!
3	Australia	Adjusted R-Squared	0.28	0.02	0.00	0.00	#NUM!	#NUM!
4	Australia	F-Statistic	819.25	80.28	6035.83	408.59	10.21	0.00
5	Australia	DF	6193.00	371.17	137766.50	0.00	#NUM!	#NUM!
6	Bulgaria	R-Squared	0.18	0.02	0.00	0.00	#NUM!	#NUM!
7	Bulgaria	Adjusted R-Squared	0.18	0.02	0.00	0.00	#NUM!	#NUM!
8	Bulgaria	F-Statistic	305.48	49.74	2352.40	121.48	6.14	0.00
9	Bulgaria	DF	4219.00	279.24	77974.50	0.00	#NUM!	#NUM!
10	Slovenia	R-Squared	0.19	0.02	0.00	0.00	#NUM!	#NUM!
11	Slovenia	Adjusted R-Squared	0.19	0.02	0.00	0.00	#NUM!	#NUM!
12	Slovenia	F-Statistic	355.61	46.06	1994.97	126.23	7.72	0.00
13	Slovenia	DF	4460.00	290.95	84650.00	0.00	#NUM!	#NUM!
14	Table Average	R-Squared	0.22	0.01			#NUM!	#NUM!
15	Table Average	Adjusted R-Squared	0.22	0.01			#NUM!	#NUM!
16	Table Average	F-Statistic	493.45	35.02			14.09	0.00
17	Table Average	DF	4957.33	182.69			#NUM!	#NUM!

Fig. 5 Model statistic output sheet from the `lsa.lin.reg` function

`bckg.indep.cat.vars`, and creates the contrasts internally. The output from this analysis is presented partially in Fig. 4. The first few columns are the same as with the previous analyses. These are followed by the columns for the coefficients and their standard errors. The other two columns present the sampling and imputation variance terms. The last two columns (not visible in Fig. 4) contain the t-values and p-values. The sheets containing the analysis information and the calling syntax are also present in this output. However, the linear (as well as the binary logistic) regression function also adds one more sheet with the model statistic (see Fig. 5): R^2, adjusted R^2, F-statistic, and degrees of freedom. The coefficients in the output are unstandardized, this is the default behavior. Standardized coefficients are included adding `standardize = TRUE` to the arguments.

As with the previous analyses, executing the syntax from above prints log messages to the console:

```
Data file "C:/Merged/Merged.RData" imported in 00:00:01.019

Valid data from 3 countries have been found. Some computations
    can be rather intensive. Please be patient.

     ( 1/3)   Australia              processed in 00:00:18.990
     ( 2/3)   Bulgaria               processed in 00:00:17.069
     ( 3/3)   Slovenia               processed in 00:00:17.250

All 3 countries with valid data processed in 00:00:56.736
"Table Average" added to the estimates in 00:00:01.040

Model statistics table assembled in 00:00:01.050

Warning messages:
1: Independent categorical background variable(s) were passed
    to "bckg.indep.cat.vars", but no reference categories were
    provided for the "bckg.ref.cats" argument. Default reference
     categories were set: the minimum value(s) available in the
    data for categorical independent variable(s).
2: Independent categorical background variable(s) were passed
    to "bckg.indep.cat.vars", but no contrast coding schemes
    were provided for the "bckg.cat.contrasts" argument. "dummy"
     coding was set as default for all variables passed to "
    bckg.indep.cat.vars".
```

Note the warning messages at the end of the log. These notify that the default contrast coding scheme (`dummy`) and default reference category (`1`) were applied automatically because they were not specified by the user. This is an example of the custom warning messages `RALSA` introduces to notify the analyst about default operations taking place. Other warning and error messages are printed to the console as part of the exception handling routines of `RALSA`, but their description goes beyond the scope of this paper.

All of the features (data preparation and analysis) `RALSA` has are fully accessible through the GUI. To start the GUI, execute the following command in the console:

```
ralsaGUI()
```

The graphical user interface will start in the default web browser installed in the operating system. Follow the instructions to prepare data and perform analyses.

Further developments

`RALSA` is a new R package for analyzing ILSAs' data released for the first time in November 2020. The current version of the package contains a set of functions (see "Presentation of the package" section) to prepare data and perform analyses. The development of the package, however, will continue in the future. **Firstly**, it will bring more features to the existing analysis types:

1. Add median and mode as measures of central tendency to the `lsa.pcts.means` function (it currently supports only arithmetic average);
2. Interaction terms to the linear regression analysis type; and
3. Interaction terms to the binary logistic regression analysis type.

These additional features will extend the existing analysis functions and provide the analysts more options, allowing them to explore relationships (as in the linear and logistic regression analysis types) in more depth. **Secondly**, it will add new analysis types:

1. Cross-tabulations with chi-square test for independence;
2. ANOVA;
3. Quantile regression;
4. Nominal logistic regression;
5. Ordinal logistic regression;
6. Multilevel modeling; and
7. Analysis of process data (also known as log data analysis).

The necessity for these analysis types stems from the nature of ILSAs' data. Most of the variables in ILSAs' data are categorical. So far all packages (including `RALSA`) have only binary logistic regression where the dependent variable is categorical. However, the dependent variable in this kind of analysis can be only dichotomous. ILSAs' data, on the other hand, have many variables with more than two categories. This is why nominal logistic regression (using nominal dependent variable), ordinal logistic regression (ordinal outcome variable) and ANOVA can be very useful. Also, quantile regression can be the solution when the assumptions of the linear regression models (i.e., linearity, homoscedasticity, independence, or normality) are not met. Similarly, cross-tabulation can be used to test the relationship between categorical variables. In addition, ILSAs' data are nested—students are nested in classes, classes are nested in schools; this presents a good opportunity for applying multilevel models, which have much more power than single-level regression models. Lastly, most of the studies change the test and questionnaire delivery mode from paper and pencil to computer-based assessment (CBA). Along with the data they collect through the tests and questionnaires, the studies also collect process data which can provide insights into cognitive and non-cognitive processes and constructs (Provasnik, 2021). The analysis of these data, however, is not a straightforward task. The current state of affairs, along with the challenges are described in a recent paper from Provasnik (2021) who also raised the concern that even the concepts and terminology are not defined yet to allow us to move forward with strategies and approaches.

Thirdly, future versions of RALSA will bring some improvements on the programming level to improve the performance:

1. Add parallel processing to make all computations faster;
2. Optimize the data conversion function for faster conversion of SPSS (or TXT in case of PISA pre-2015 cycles) data files; and
3. Make the GUI run as a background job in RStudio, i.e. without blocking the R/RStudio console.

The lists above reflect the main plans for the near future. Even more functionality will be added in the long term. Suggestions and requests from RALSA users are also welcome. Such requests can be submitted using the contact form at the RALSA dedicated website (INERI, 2021). The flexible and open architecture of the package will allow these further developments to come to life faster. RALSA receives regular updates. Since its first appearance in November 2020, it has received four updates which not only came with bug fixes, but also with many improvements and additions of new functionality.

Summary and discussion

ILSAs provide valuable information to researchers and policy makers worldwide. Secondary analyses of the data can reveal how different aspects of the curricula, organization of instruction, background characteristics, and the general context of education can impact the educational outcomes. Analysis of these data serves as a tool for making new or changing existing policies, starting and/or supporting ongoing reforms (Klemenčič and Mirazchiyski, 2018; Wiseman, 2010). Given the importance of the decisions based on the results, ILSAs require special tools for analyzing their data due to the methodological and statistical complexities accompanying them. Different software solutions accommodating these complexities exist, although they have many limitations:

- One of them uses expensive software products to perform the computations;
- Some of them can work with data from limited number of studies;
- Most of them create their own object types and in general are very difficult to work with;
- None of the R packages can convert the provided data into native `.RData` sets;
- Just one of them can read the text files from PISA prior to 2015 cycle;
- None of the R packages have a comprehensive output system to export the results; and
- None of the R packages has a graphical user interface which makes them very difficult to work with for the non-technical users.

The overall aim of RALSA is to bring functionality to analyze data with complex sampling and assessment design, adding more studies and analysis types, providing a user-friendly experience for analysts having minimal knowledge and/or skills in using R and analyzing ILSAs' data. This is facilitated by the easy-to-comprehend structure and functions' arguments, as well as the automatic recognition of the study, cycle and respondent types in the used data to apply the appropriate computation routines. In addition, the GUI helps the less experienced R user to use all the functionality. RALSA has flexible architecture which allows it to easily add new studies and functions to analyze ILSAs' data. These are amongst the main

advantages of the package compared with other tools for analyzing ILSAs' data. The future developments of RALSA, will include more analysis types, addition of more features to the existing ones, as well as optimizations and improvements on the software level like the speed of the computations and convenience. RALSA was first released in November 2020 and since then, it has received multiple updates, bringing many improvements and new features every time.

Besides RALSA and the software tools for analyzing ILSAs' data discussed in this paper, it is worth mentioning the R package lsasim (Matta et al., 2018). lsasim was not discussed in this paper because it does not represent software for analyzing ILSAs' data. However, it is still an R package which relates to large-scale assessments and deserves attention because it provides an entire framework for simulating ILSAs' data. These simulations can aid setting the methodology of new ILSAs' (or new ILSA cycles) themselves, by providing the possibility to plan and test their design in advance (for more details see Matta et al., 2018) and, thus, improve the implementation of ILSAs in future.

RALSA is a free and open-source package. Just as with any other open-source projects, users are invited to suggest improvements, functionality and studies to be supported. R was chosen to build RALSA as it is free and open-source itself. Other free and open-source platforms exist as well. Python (Python Software Foundation, 2021), for example, is a programming language which is quite mature and has many packages built for data science with a large user base. Despite its merits, in its essence Python is a general-purpose programming language. Many other programming languages with statistical extension libraries can be given as examples here. R, on the other hand, is a programming language for statistical computing and statistical software at the same time. R is an object-oriented functional programming language, where classes and methods can be extended through inheritance mechanisms (Matloff, 2011), *lingua franca* of statistical computing because of its power and lack of restrictions (Everitt and Hothorn, 2010). That is, different from other general-purpose programming languages, R is specialized in statistics with more built-in statistical procedures than any other programming language. In addition, users can submit and share their own packages (just as with RALSA) adding more functionality to R. All this made R the platform of choice to build RALSA and guarantee its further growth in the future.

For more information on RALSA, detailed step-by-step tutorials, and other information, please visit the package's support website (INERI, 2021). The website provides easy-to-follow guides from the package installation to performing analyses even for analysts who have never used R so far. The tutorials cover both the command-line and GUI use.

Acknowledgements
The author acknowledges Eva Klemenčič Mirazchiyski for her support, as well as the International Educational Research and Evaluation Institute (INERI) for funding the RALSA development.

Authors' contributions
PVM prepared the entirety of the article without collaboration with or contributions from other authors. The author read and approved the final manuscript.

Funding
The development of the RALSA package and the preparation of this manuscript was funded by the International Educational Research and Evaluation Institute (INERI).

Availability of data and materials
The RALSA package is publicly available on Comprehensive R Archive Network, CRAN: https://cran.r-project.org/package=RALSA. The data used in the examples are publicly available can be found on IEA's (https://www.iea.nl/data-tools/repository) and OECD's (http://www.oecd.org/pisa/data/) websites.

Declarations

Ethics approval and consent to participate

In this manuscript the officially published PIRLS and PISA data sets were used in the examples. These data sets were downloaded as public use files from the IEA's (https://www.iea.nl/data-tools/repository) and OECD's (http://www.oecd.org/pisa/data/) websites . Therefore, neither consent to participate or consent for publication nor ethics approval were required for the sample analyses.

Consent for publication

I provide my consent to publish this manuscript upon publication in the Springer open journal LSA.

Competing interests

The author declares that he has no competing interests.

Author details

[1]Educational Research Institute, Ljubljana, Slovenia. [2]International Educational Research and Evaluation Institute, Ljubljana, Slovenia.

Received: 15 January 2021 Accepted: 13 September 2021
Published online: 04 October 2021

References

Bailey, P., Emad, A., Huo, H., Lee, M., Liao, Y., Lishinski, A., Nguyen, T., Xie, Q., Yu, J., & Zhang, T. (2020). *EdSurvey: Analysis of NCES education survey and assessment data [computer software manual]*. Retrieved from https://cran.r-project.org/package=EdSurvey (R package version 2.6.1).

BIFIE, Robitzsch, A., & Oberwimmer, K. (2019). *BIFIEsurvey: Tools for survey statistics in educational assessment [computer software manual]*. Retrieved from https://cran.r-project.org/package=BIFIEsurvey (R package version 3.3-12).

Brese, F. (2018). Analyzing the ICCS 2016 data using the IEA IDB analyzer. In H. Köhler, S. Weber, F. Brese, W. Schulz, & R. Carstens (Eds.), *ICCS 2016 user guide for the international database* (pp. 33–36). IEA.

Caro, D., & Biecek, P. (2019). *intsvy: International assessment data manager [computer software manual]*. Retrieved from https://cran.r-project.org/package=intsvy (R package version 2.4).

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2020). *shiny: Web application framework for R [computer software manual]*. Retrieved from https://cran.r-project.org/package=shiny (R package version 1.5.0).

Everitt, B. S., & Hothorn, T. (2010). *A handbook of statistical analyses using R* (2nd ed.). Taylor and Francis Group, LLC.

Foy, P. (2018). *PIRLS 2016 user guide for the international database*. TIMSS & PIRLS Study Center.

Foy, P., & LaRoche, S. (2017). Estimating standard errors in the PIRLS 2016 results. In M. Martin, I. Mullis, & M. Hooper (Eds.), *Methods and procedures in PIRLS 2016* (pp. 41–422). Lynch School of Education, Boston College.

Foy, P., & Yin, L. (2017). Scaling the PIRLS 2016 achievement data. In M. Martin, I. Mullis, & M. Hooper (Eds.), *Methods and procedures in PIRLS 2016* (pp. 12.1–12.38). TIMSS & PIRLS International Study Center.

IBM. (2020). *IBM SPSS [computer software]*. Retrieved from https://www.ibm.com/analytics/spss-statistics-software (version 27.0).

IEA. (2020). *IDB analyzer [computer software manual]*. Retrieved from https://www.iea.nl/data-tools/tools#section-308 (version 4.0.39).

INERI. (2021). RALSA: R Analyzer for Large-Scale Assessments. http://ralsa.ineri.org/. Accessed 10 Sept 2021.

Institute, S. A. S. (2020). *SAS/STAT software [computer software]*. Retrieved from https://www.sas.com/en_us/software/stat.html (version 9.4M7).

Klemencic, E. (2010). The impact of international achievement studies on national education policymaking: the case of Slovenia—How many watches do we need? In A. W. Wiseman (Ed.), *The impact of international achievement studies on national education policymaking, (International perspectives on education and society)* (vol. 13, pp. 239–266). International perspectives on education and society.

Klemenčič, E., & Mirazchiyski, P. (2018). League tables in educational evidence-based policy-making: can we stop the horse race, please? *Comparative Education*, *54*(3), 309–324.

LaRoche, S., Joncas, M., & Foy, P. (2017). Sample design in PIRLS 2016. In M. Martin, I. Mullis, & M. Hooper (Eds.), *Methods and procedures in PIRLS 2016* (pp. 3.1-3.34). Lynch School of Education, Boston College.

Martin, M., Mullis, I., & Foy, P. (2015). Assessment design for PIRLS, PIRLS literacy, and ePIRLS in 2016. In I. V. S. Mullis & M. Martin (Eds.), *PIRLS 2016 assessment framework* (2nd ed., pp. 55–69). Lynch School of Education, Boston College.

Martin, M., Mullis, I., & Hooper, M. (Eds.). (2017). *Methods and procedures in PIRLS 2016*. Chestnut Hill: Lynch School of Education, Boston College.

Martin, M., Mullis, I. V. S., Hooper, M., Yin, L., Foy, P., Fishbein, B., & Liu, J. (2017). Creating and interpreting the PIRLS 2016 context questionnaire scales. In M. O. Martin, I. V. S. Mullis, & M. Hooper (Eds.), *Methods and procedures in PIRLS 2016* (pp. 14.1-14.106). Lynch School of Education, Boston College.

Matloff, N. (2011). *The art of R programming*. No Starch Press.

Matta, T. H., Rutkowski, L., Rutkowski, D., & Liaw, Y.-L. (2018). lsasim: An r package for simulating large-scale assessment data. *Large-scale Assessments in Education, 6*(1), 15.

Mikheeva, E., & Meyer, S. (2020). Analyzing the ICILS 2018 data using the IEA IDB analyzer. In E. Mikheeva & S. Meyer (Eds.), *IEA international computer and information literacy study 2018: User guide for the international database* (pp. 39–76). IEA.

Mirazchiyski, P., & INERI. (2021). RALSA: R Analyzer for Large-Scale Assessments [computer software manual]. Retrieved from https://CRAN.R-project.org/package=RALSA (R package version 1.0.1).

Mullis, I., & Prendergast, C. (2017). Assessment Design for PIRLS, PIRLS Literacy, and ePIRLS in 2016. In I. V. S. Mullis & M. Martin (Eds.), *PIRLS 2016 assessment framework* (2nd ed., pp. 55–69). Lynch School of Education, Boston College.

OECD. (2017a). *PISA 2015 technical report*. OECD.
OECD. (2017b). Sample design. *PISA 2015 technical report* (pp. 65–87). OECD.
OECD. (2017c). Scaling PISA data. *PISA 2015 technical report* (pp. 127–186). OECD.
OECD. (2017d). Survey weighting and the calculation of sampling variance. *PISA 2015 technical report* (pp. 115–126). OECD.
OECD. (2017e). Test design and test development. *PISA 2015 technical report* (pp. 29–55). OECD.
Provasnik, S. (2021). Process data, the new frontier for assessment development: Rich new soil or a quixotic quest? *Large-scale Assessments in Education, 9*(1), 1.
Rust, K. (2014). Sampling, weighting, and variance estimation in international large-scale assessments. In L. Rutkowski, M. von Davier, & D. Rutkowski (Eds.), *Handbook of international large-scale assessments: Background, technical issues, and methods of data analysis* (pp. 117–154). CRC Press.
Rutkowski, L., Gonzalez, E., Joncas, M., & von Davier, M. (2010). International large-scale assessment data: Issues in secondary analysis and reporting. *Educational Researcher, 39*(2), 142–151.
The Python Software Foundation. (2021). The Python Software Foundation. https://www.python.org/psf-landing/. Accessed 10 Jan 2021.
von Davier, M., Gonzalez, E., & Mislevy, R. (2009). What are plausible values and why are they useful? *IERI Monograph Series, 2*(1), 9–36.
Wagemaker, H. (2014). International large-scale assessments: From research to policy. In L. Rutkowski, M. von Davier, & D. Rutkowski (Eds.), *Handbook of international large-scale assessments: Background, technical issues, and methods of data analysis* (pp. 11–36). CRC Press.
Westat. (2008). *WesVar [computer software manual]*. Retrieved from https://www.westat.com/capability/information-technology/wesvar (version 5.1.17).
Wiseman, A. (Ed.). (2010). *The impact of international achievement studies on national education policymaking*. International Perspectives on Education and Society, Emerald Group Publishing Limited.

Publisher's Note