TIES Measurement Report Automation Project

Michael Wu

2021-04-22

Contents

Chapter 1

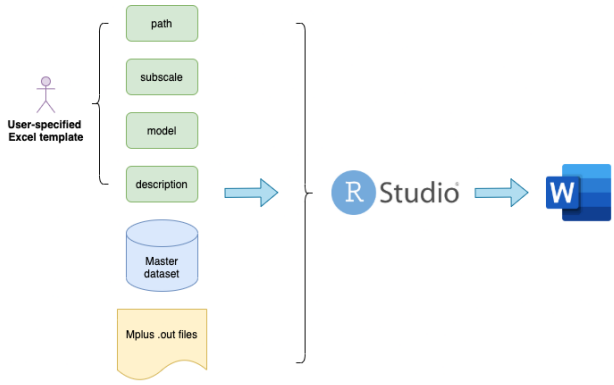
Project overview

This project is developed to generate lengthy but informative measurement reports from survey data and Mplus measurement model outputs for projects at NYU Global TIES for Children.

Typically, a research institute like TIES has the obligation to generate detailed measurement reports to better inform the funders and the cooperating agencies about its most up-to-date work. However, even with a Word template, the process from analytical results to a publishable report is **unnecessarily inefficent and prone to mistakes** even for the most careful research assistant.

Therefore, we develop an R package called `mrautomatr` to be used in conjunction with Rstudio to address this issue. Currently, the project only suits the need of NYU Global TIES, where we can impose naming rules for files and variables and most people use Mplus for measurement modeling and STATA for other analyses. Future adaptations are needed as people move their analysis to R.

The project workflow is shown below: the users specify parameters in a Microsoft Excel sheet and move several files to a destined folder, run a command in R, and (**boom!**) there is a well-formatted measurement report in Microsoft Word (powered by the `flextable` & `rmarkdown` R packages).



We chose Word over LaTex (which generates pdf files) and html (which generates web pages) simply to minimize the confusion around writing codes in R, which takes a long time to learn. After generating the reproducible parts of the report, feel free to rename it and manually edit the sections that are text-heavy. Check out an example rendered report here.

Check out the TIES R workshop series and Yihui Xie's Rmarkdown book if you'd like to learn some necessary tools to customize your reporting formats using R.

Chapter 2

Set up the package

2.1 Install the necessary softwares

You need to set up `R` and `Rstudio` on your computer before everything. `R` is the programming language that powers this project, and `Rstudio` is the interface that allows you better interact with your R code. Please follow the steps below:

- Download `R` here and install it before you install `Rstudio`.
- Download `Rstudio` here and install it.
- Open Rstudio, and click the first icon from the left on the Rstudio toolbar, and select R Markdown. Rstudio will prompt you to install several packages, just follow the instructions and install them.

2.2 Download and install the `mrautomatr` package

- Run the following lines:

```
install.packages("usethis")
install.packages("devtools")
library(usethis)
library(devtools)
```

- Set up your GitHub Personal Auth Token set following instructions on this website. This is only applicable to this package right now since it's internal and private. You may need to email Michael Wu (zw1429@nyu.edu) in order to gain access to the TIES github repository.

- Essentially, You need to set up your personal access token in a file called `.Renviron`. Install the `usethis` package and then run `usethis::create_github_token()`.
- It'll take you through the process of creating a token, and then GitHub will give you a string of characters. **Copy that text to somewhere safe for now.**
- Then, back in R, run `usethis::edit_r_environ()` and add this to the file: `GITHUB_PAT=[the token text here without the brackets]`.
- Save that file and restart your R session for changes to take effect.
- To verify that you got it to work, when R starts again, run `Sys.getenv("GITHUB_PAT")` and you should see your token, exactly how it was shown on GitHub. If you don't more troubleshooting is required.
- Run the following line:

```
devtools::install_github("nyuglobalties/mrautomatr")
library(mrautomatr)
```

- Check out the functions by running `?function_name`, e.g.:

```
?mrautomatr
```

Chapter 3

Set up the parameters

3.1 Organize your model outputs

Before you run any R codes, you need to make sure that the parameters for the report are correctly specified.

First, copy and paste all **currently available** final Mplus models (only the `.out` files) into one folder (e.g. a folder called `Measurement report/Models` somewhere on Box). This includes:

- EFA models
- CFA models
- Longitudinal invariance models
- Treatment invariance models
- Age invariance models
- Gender invariance models

3.2 Fill in the Excel template

Second, fill in the Excel sheet template that we provide. This Excel file is downloaded along with the `mrautomatr` package. You can find its file path on your computer by running `system.file("templates", "input_template.xlsx", package = "mrautomatr")` in R. Copy and paste it somewhere in your computer (e.g. a folder called `Measurement report/template`).

Alternatively, you can also find this template (`input_template.xlsx`) located in `inst/templates/` in the package GitHub repository. Simply hit `Download` to download the template and store somewhere in your computer.

In this template, you need to manually type in the following parameters. For any parameters that are not available temporarily, you can leave blank and still be able to generate the report (with error messages shown in the Word document telling you that you need to specify more parameters/fix certain things to have a full report).

3.2.1 Tab 1: `path`

A shorthand to get file path on Mac: go to the path/file and hit `command + option + C`.

If you are using Box, we recommend typing in an R function from `mrautomatr` – `box_path("...")` in the `path` tab in the excel sheet. The `...` part is whatever sub-folders on your computer under the general Box folder. Examples are `box_path("Box 3EA Team Folder/Data Management")`, or `box_path("Peru/Data/Full")`. This is because different people can have different access to Box folders at different time (e.g. you only had access to `Data Management` in March, but gained access to `Box 3EA Team Folder` in April), and if you only specify the local path on your computer, you will need to re-specify it in the excel sheet whenever your level of access changes. `box_path(...)` prevents this from happening.

- `year` will show up in the first line of your document (not the title).
- `measure` will show up in the first line of your document.
- `data_file_path` should be wherever the final master data is located. It will be used to calculate summary statistics and bivariate correlations. Our tool currently takes the following data formats: `.csv`, `.xlsx`, `.dta`.
- `fs_data_file_path` refers to the file path where the tabular data of the Mplus-generated factor scores is saved. Because Mplus **does not** generate a spreadsheet, you will need to:
  - **(1) copy and paste the factor scores into an Excel sheet, and**
  - **(2) insert the first row and name the variables exactly the same as they are in your master dataset and in your other Mplus models.**
  - **(3) save the sheet either as a .csv or an .xlsx file.**
- `model_file_path` leads you to all the Mplus outputs.

3.2.2 Tab 2: `subscale`

- The first row should contain the subscale/factor names. They should be the same as the ones in your Mplus models.

- For each subscale/factor, list the items. The rows can be of unequal length (i.e. you can leave blanks for subscales with smaller number of items).
- These are specified to generate reliability estimates from the master dataset.
- **Always** add an underscore and a wave tag (`_#`) in the `subscale` section to both the construct name and the item names in your excel sheet (e.g. AANXDEP_1). Whatever numbers comes after "_" will be shown in the table as the wave tag. And the wave tag also needs to be added for the variables in the master dataset. Not sure if this is easy to do in Mplus, but you can certainly export another master dataset after running things like `names(dat)[1:10] <- paste(names(dat)[1:10], "_1", sep = "")` on the variables you want to modify.

If you have already run quite many models and generated multiple reports, I'd suggest using the `get.omega.bywave()` in the`mrautomatr` and manually fill in the omegas. E.g.:

```
get.omega.bywave(model = "xxx.out",
                 path = "/Users/xxx")
```

3.2.3 Tab 3: `model`

- This specifies all necessary Mplus model names (i.e. `xxx.out`).
- List all available models in the order of waves (e.g. wave 1 before wave 2).
- There is no restrictions on the file names, but please follow the naming rules for reproducibility purposes.

3.2.4 Tab 4: `description`

- This is specified to have a description of the items at the beginning of the report.
- You can format this tab in any ways that you like, but the caveat is that (1) the first row will be taken as the header and set to bold, and (2) you cannot merge cells.

Variable name	Description
`year`	Study site and year
`measure`	Measure name
`data_file_path`	Local file path to the master dataset on your own computer
`fs_data_file_path`	Local file path to the factor score dataset on your own computer

Variable name	Description
`model_file_path`	Local file path to all the Mplus .out files
`subscale`	Subscales and their corresponding items
`model_efa`	EFA models
`model_cfa`	CFA models
`model_inv_tx`	Treatment invariance models
`model_inv_gender`	Gender invariance models
`model_inv_age`	Age invariance models
`model_inv_lg`	Longitudinal invariance model
`description`	Detailed item descriptions

Chapter 4

Generate the report

After carefully setting your parameters, you can now generate your report!

There are three ways to generate/knit reports:

1. Generate one report for one measure using the default settings `render_report()`
2. Generate one report for one measure using customized settings by the users `render_report_manual()`
3. Generate multiple separate reports for multiple measures using default settings `render_report_multiple()`

After generating the report, make sure to rename it and manually edit the sections that are text-heavy. The renaming is **necessary** because you may accidentally overwrite your manual edits if you regenerate the report in R.

Rmarkdown is not powerful yet to allow back-translation from Word to R codes, so your manual changes in Word will **NOT** be reflected in the R codes when you regenerate the report for some reasons (e.g. wrong file names). So we recommend finalizing the tables and plots before you write texts in the Word document (or you can just store the texts in another and move them over to the master report whenever you feel ready). If you've already written extensively in a knitted report and want to fix certain small sections/add some numbers, you can run the individual functions in R and manually make the changes.

Note. Error messages are shown in the knitted document in their corresponding section. They are usually about certain parameters not being specified. Other error messages should also be pretty understandable.

Note. Warning messages are not shown in the knitted document. They are usually okay to ignore as they often come from the `MplusAutomation` package failing to read certain bits of the Mplus output that are not quite important

to generating the reports. They may also come from that WRMR is reported instead of SRMR due to Mplus version conflict. If it's telling you an error related to `xxx_fscores.csv`, simply ignore it (see this GitHub issue for an explanation). You can also set `printwarning = TRUE` in rendering the reports (see the functions below) to have warnings printed in the documents too, or run `warnings()` in R to get all warning messages.

4.1 `render_report()`

This function renders one report for the specified measure.

Run `?render_report()` to see what each argument represents. `parameters` allows you to specify a list of parameters to control the `params` section in the Rmarkdown template, you can omit this argument to use the default settings. See the bottom of this page for explanations on these parameters.

Example:

```
render_report(output_dir = "/Users/michaelfive/Google Drive/NYU/3EA/test",
              output_file = "Report_lebanon_cs.docx",
              parameters = list(
                       # set R code print options
                       printcode = FALSE,
                       printwarning = FALSE,
                       storecache = FALSE,

                       # set report overall parameters
                template = "/Users/michaelfive/Google Drive/NYU/3EA/test/input_template_
                       set_title = "Lebanon Year 1 (2016-2017)",
                       set_author = "Jane Doe",

                       # select report sections
                       item = TRUE,
                       descriptive = TRUE,
                       ds_plot = TRUE,
                       correlation_matrix_lg = TRUE,
                       correlation_matrix_bivar = TRUE,
                correlation_matrix_item = FALSE, # BE CAREFUL! This might crash the docume
                       efa_screeplot = TRUE,
                       cfa_model_fit = TRUE,
                       cfa_model_plot = TRUE,
                       cfa_model_parameters = TRUE,
                       cfa_r2 = TRUE,
                       internal_reliability = TRUE,
                       summary_item_statistics = TRUE,
```

```
                        item_total_statistics = TRUE,
                        inv_tx = TRUE,
                        inv_gender = TRUE,
                        inv_age = TRUE,
                        inv_lg = TRUE
                        ))
```

4.2 render_report_manual()

Run `?render_report_manual()` to see what each argument represents.

Example:

```
render_report_manual(output_file = "Report_lebanon_cs.docx",
              output_dir = "/Users/michaelfive/Google Drive/NYU/3EA/test")
```

This function opens a Shiny web page where you can click/unclick sections you'd like to include/exclude in the report (see descriptions below). It also renders one report for the specified measure.

4.3 render_report_multiple()

Run `?render_report_multiple()` to see what each argument represents. This function renders multiple reports at the same time with parameters globally set for all reports.

Example:

```
render_report_multiple(input_dir = "/Users/michaelfive/Google Drive/NYU/3EA/test",
                    templates = c("input_template_lebanon_cs.xlsx",
                                  "input_template_niger_psra.xlsx"),
                output_dir = "/Users/michaelfive/Google Drive/NYU/3EA/test",
                       # set parameters globally for all documents
                       parameters = list(
                       # set R code print options
                       printcode = FALSE,
                       printwarning = FALSE,
                       storecache = FALSE,

                       # set report overall parameters
                       set_author = "Jane Doe",
                # report title comes from the `year` tab in each excel template

                       # select report sections
                       item = TRUE,
```

```
        descriptive = TRUE,
        ds_plot = TRUE,
        correlation_matrix_lg = TRUE,
        correlation_matrix_bivar = TRUE,
correlation_matrix_item = FALSE, # BE CAREFUL! This might crash the documen
        efa_screeplot = TRUE,
        cfa_model_fit = TRUE,
        cfa_model_plot = TRUE,
        cfa_model_parameters = TRUE,
        cfa_r2 = TRUE,
        internal_reliability = TRUE,
        summary_item_statistics = TRUE,
        item_total_statistics = TRUE,
        inv_tx = TRUE,
        inv_gender = TRUE,
        inv_age = TRUE,
        inv_lg = TRUE
        ))
```

Parameters	Description
`printcode`	whether you'd like R codes to be printed in your document
`printwarning`	whether you'd like to print warnings in running the codes
`storecache`	whether you'd like to store `knitr` cache (only for programming purposes, see here)
`set_title`	title
`set_author`	author
`template`	parameter template file path
`item`	print item descriptions
`descriptive`	print descriptive statistics table
`ds_plot`	print descriptive statistics histograms
`correlation_matrix_lg`	print factor-level correlation matrix from longitudinal invariance model
`correlation_matrix_bivar`	print factor-level correlation matrix from master dataset
`correlation_matrix_item`	print item-level correlation matrix from master dataset (set to `FALSE` because correlations among dozens of items may be unnecessary)
`efa_screeplot`	print EFA screeplot at all waves
`cfa_model_fit`	print CFA model fits at all waves
`cfa_model_plot`	print CFA model path diagram (for the first specified CFA model; i.e. Time 1; assuming factor structure does not change)
`cfa_model_parameters`	print CFA model parameters at all waves (factor loadings and thresholds)
`cfa_r2`	print CFA model R-squared at all wave

Parameters	Description
`internal_reliability`	print estimates of internal reliability (Cronbach's Alpha and McDonald's Omega, descriptions of the other indices can be found here)
`summary_item_statistics`	print summary item statistics (descriptions of the other indices can be found here)
`item_total_statistics`	print total item statistics (descriptions of the other indices can be found here)
`inv_tx`	print model fits for treatment invariance models at all waves
`inv_gender`	print model fits for gender invariance models at all waves
`inv_age`	print model fits for age invariance models at all waves
`inv_lg`	print model fit for the longitudinal invariance model

Chapter 5

Individual functions

If you are an R user who wishes to run individual functions in this package to get results in R instead of Word, you can check the help pages of those functions by running `?mrautomatr`.