gesis Leibniz Institute for the Social Sciences

Open Methodology in practice: Reproduzierbare Forschung mit R

Dr. Tobias Heycke

27.06.2019

Part 1: Introduction

Schedule

- General information about me
- Some general information about the course
- Who are you?
- Why reproducible research?
- Coding rules
- RStudio

About me

- B.Sc./M.Sc. Psychology, Radboud University Nijmegen
- In the Netherlands during ‘Stapel-Gate’
- Ph.D. Psychology, University of Cologne
- Teaching statistics and methods since 2013
- Adversarial collaborations with a number of researchers
- First registered report published in 2017
- First registered replication published in 2018

The course: schedule

Today:

- 10.00 - 18.00 o'clock
- break at 13.00 o'clock
- starting at 18.00 o'clock: wine & cheese

Tomorrow:

- 9.00 - 16.00 o'clock
- break at 12.30 o'clock

Feeback system

Red post-it: I need help Green post-it: I finished the task

Slides & Material

You can find all slides and additional materials here:

https://github.com/TobiasHeycke/reproduzierbare_forschung_pub

The course: lunch

Today: N1 Lounge, Who wants to join?

Tomorrow: Burrito Baby, Who want to join?

The course: goals

- clean programming
- markdown
- git

Important note

If you do not know how to solve a task? google it!

Who are you?

Who are you?

- Where are you from?
- Expectations?

Software we need

- R
- R Studio
- git
- MiKTeX (windows)

Reproducible analyses

Different Reproducibilities

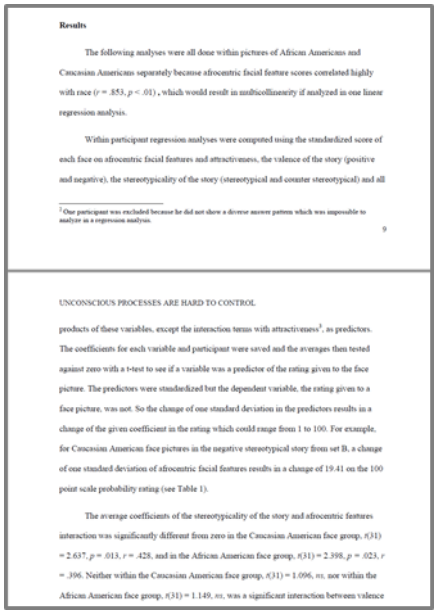
- Empirical Reproducibility
- Computational Reproducibility
- Statistical Reproducibility
- (Replication)
- Plus: machine vs. human readability

My master thesis folder

- ad_follow_upanalysis_b_weights
- analysis_01
- coefficien_analysis_for_high_low_attractiveness
- coefficient_Zattr_high
- coefficient_Zattr_low
- coefficient01
- coefficient02
- coefficient02_for_RM_analysis01
- coefficient02_for_RM_analysis02
- coefficient03
- coefficients_different0_no_between_subject_factor_split
- coefficients_dv_standadized_without_main_attr
- coefficients_dv_standadized_without_main_attr_high_attr
- coefficients_dv_standadized_without_main_attr_low_attr
- coefficients_dv_standadized_without_main_attr_story_type_split
- coefficients_dv_standadized_without_main_attr_story_type_split_for_RM_mood
- coefficients_dv_standadized_without_main_attr_story_type_split_high_attr2
- coefficients_dv_standadized_without_main_attr_story_type_split_low_attr2
- data_dv_all_cleaned
- data_dv_all_cleaned_02
- data_dv_all_cleaned_03
- data_dv_all_cleaned_04
- data_dv_all_cleaned_05
- data_dv_cub10
- data_dv_cub10_cleaned
- data_dv_cub14
- data_dv_cub14_cleaned
- data_dv_cub17
- data_dv_cub17_cleaned
- data_dv_RM_ANOVA01
- data_dv_RM_ANOVA02
- data_dv_RM_ANOVA02_1
- graph_RM_category_effect
- holm_bonferroni_correction_of_sign_t_tests
- Output_condition_on_b_weights_repeated_GLM
- Output_outlier_removed_category_effect_analysis
- output_volgorder_effect
- prepare_data_set01
- re_analyse_all_without_attr_main
- repated_measure_category_recode_variable
- repated_measure_category_recode_variable_and_ar
- repeated_for_category_with_mood_without_outliers
- repeated_measures_anova
- repeated_measures_on_b_weights
- repeated_measures_syntax_without_uotliers
- response_pattern
- RM_ANOVA_category_restructured
- volgorder_effect)syntax

My master thesis folder

- ad_follow_upanalysis_b_weights
- analysis_01
- coefficien_analysis_for_high_low_attractiveness
- coefficient_Zattr_high
- coefficient_Zattr_low
- coefficient01
- coefficient02
- coefficient02_for_RM_analysis01
- coefficient02_for_RM_analysis02
- coefficient03
- coefficients_different0_no_between_subject_factor_split
- coefficients_dv_standadized_without_main_attr
- coefficients_dv_standadized_without_main_attr_high_attr
- coefficients_dv_standadized_without_main_attr_low_attr
- coefficients_dv_standadized_without_main_attr_story_type_split
- coefficients_dv_standadized_without_main_attr_story_type_split_for_RM_mood
- coefficients_dv_standadized_without_main_attr_story_type_split_high_attr2
- coefficients_dv_standadized_without_main_attr_story_type_split_low_attr2
- data_dv_all_cleaned
- data_dv_all_cleaned_02
- data_dv_all_cleaned_03
- data_dv_all_cleaned_04
- data_dv_all_cleaned_05
- data_dv_cub10
- data_dv_cub10_cleaned
- data_dv_cub14
- data_dv_cub14_cleaned
- data_dv_cub17
- data_dv_cub17_cleaned
- data_dv_RM_ANOVA01
- data_dv_RM_ANOVA02
- data_dv_RM_ANOVA02_1
- graph_RM_category_effect
- holm_bonferroni_correction_of_sign_t_tests
- Output_condition_on_b_weights_repeated_GLM
- Output_outlier_removed_category_effect_analysis
- output_volgorder_effect
- prepare_data_set01
- re_analyse_all_without_attr_main
- repated_measure_category_recode_variable
- repated_measure_category_recode_variable_and_ar
- repeated_for_category_with_mood_without_outliers
- repeated_measures_anova
- repeated_measures_on_b_weights
- repeated_measures_syntax_without_uotliers
- response_pattern
- RM_ANOVA_category_restructured
- volgorder_effect)syntax



Results

The following analyses were all done within pictures of African Americans and Caucasian Americans separately because afrocentric facial feature scores correlated highly with race (r = .853, p < .01) , which would result in multicollinearity if analyzed in one linear regression analysis.

Within participant regression analyses were computed using the standardized score of each face on afrocentric facial features and attractiveness, the valence of the story (positive and negative), the stereotypicality of the story (stereotypical and counter stereotypical) and all

[7] One participant was excluded because he did not show a diverse answer pattern which was impossible to analyze in a regression analysis.

9

UNCONSCIOUS PROCESSES ARE HARD TO CONTROL

products of these variables, except the interaction terms with attractiveness[8], as predictors. The coefficients for each variable and participant were saved and the averages then tested against zero with a t-test to see if a variable was a predictor of the rating given to the face picture. The predictors were standardized but the dependent variable, the rating given to a face picture, was not. So the change of one standard deviation in the predictors results in a change of the given coefficient in the rating which could range from 1 to 100. For example, for Caucasian American face pictures in the negative stereotypical story from set B, a change of one standard deviation of afrocentric facial features results in a change of 19.41 on the 100 point scale probability rating (see Table 1).

The average coefficients of the stereotypicality of the story and afrocentric features interaction was significantly different from zero in the Caucasian American face group, $t(31)$ = 2.637, p = .013, r = .428, and in the African American face group, $t(31)$ = 2.398, p = .023, r = .396. Neither within the Caucasian American face group, $t(31)$ = 1.096, *ns*, nor within the African American face group, $t(31)$ = 1.149, *ns*, was a significant interaction between valence

Errors happen I

"Our main finding is that [...] high debt/GDP levels [...] are associated with notably lower growth outcomes."

"At the very minimum, this would suggest that traditional debt management issues should be at the forefront of public policy concerns."

Reinhard & Rogoff (2010)

Errors happen I

"We replicate Reinhart and Rogoff (2010a and 2010b) and find that coding errors, selective exclusion of available data, and unconventional weighting of summary statistics lead to serious errors that inaccurately represent the relationship between public debt and GDP growth"

Herndon, Ash, & Pollin (2014)

Reproducing own work

Source: https://twitter.com/massimo006/status/1128604521209442306

Errors in statistical reporting

- 250,000 *p*-values checked (in eight major psychology journals)

Nuijten, Hartgerink,van Assen, Epskamp, & Wicherts, 2016

Errors in statistical reporting

- 250,000 *p*-values checked (in eight major psychology journals)
- appr. 50 % of papers contained at least one inconsistency between *p*-value and test statistic

Nuijten, Hartgerink,van Assen, Epskamp, & Wicherts, 2016

Errors in statistical reporting

- 250,000 *p*-values checked (in eight major psychology journals)
- appr. 50 % of papers contained at least one inconsistency between *p*-value and test statistic
- appr. 12 % of papers contained a major inconsistency

Nuijten, Hartgerink,van Assen, Epskamp, & Wicherts, 2016

Science vs.Pseudo-Science

"An article about computational results is advertising, not scholarship. The actual scholarship is the full software environment, code and data, that produced the result."

- Claerbout & Karrenbach, 1992

Coding in general

Sources

- Weniger schlecht programmieren (Passig & Jander)
- Google style guide for R (https://google.github.io/styleguide/Rguide.xml)
- Hardley Wickham style guide (http://adv-r.had.co.nz/Style.html)

The goal of this section

- Human readability!
- You will create coding rules for yourself after my introduction

Style Guide/ Clean coding

Clean coding I

Be consistent!

Three basic rules (Kernighan & Pike, 1999):

- simplicity (short and manageable)
- clarity (easy to understand)
- generality (work well in broad range of situations)

DRY (Don't repeat yourself), Wilson et al. 2014

Clean coding II

- Use relative paths! (e.g., here, Rproj)

Clean coding II

- Use relative paths! (e.g., here, Rproj)
- Never type anything that you can obtain from a saved result

Clean coding II

- Use relative paths! (e.g., here, Rproj)
- Never type anything that you can obtain from a saved result
- One main file: call R/Rmd files from main file

Clean coding II

- Use relative paths! (e.g., here, Rproj)
- Never type anything that you can obtain from a saved result
- One main file: call R/Rmd files from main file
- Order in file: first definitions, then functions, then executable code

Clean coding III

- Language should be English (US) for everything

Clean coding III

- Language should be English (US) for everything
- Number of characters per line = 80

Clean coding III

- Language should be English (US) for everything
- Number of characters per line = 80
- Don't show your superiority/cleverness

Clean coding IV

- = vs. ->

Clean coding IV

- = vs. ->
- Spacing around operators (except :) and ->

Clean coding IV

- = vs. ->
- Spacing around operators (except :) and ->
- Indentation (try Ctrl + i in R Studio)

Clean coding V

- Curly brackets: opening curly brace should never go on its own line; a closing curly brace should always go on its own line (except else)

Clean coding V

- Curly brackets: opening curly brace should never go on its own line; a closing curly brace should always go on its own line (except else)
- Where should the comma go in a function call when using multiple lines?

Variable notation

Variable notation I

- Be consistent!

Variable notation I

- Be consistent!
- Variable names should be nouns and function names should be verbs

Variable notation I

- Be consistent!
- Variable names should be nouns and function names should be verbs
- Boolean: start name with 'is'

Variable notation II

- Variable writing: CamelCase or camelCase or pothole_case or this.case or likethis?

Variable notation II

- Variable writing: CamelCase or camelCase or pothole_case or this.case or likethis?
- Not too similar (InternationRefTemplate vs. InternationalRefTemplate)

Variable notation II

- Variable writing: CamelCase or camelCase or pothole_case or this.case or likethis?
- Not too similar (InternationRefTemplate vs. InternationalRefTemplate)
- Differences in names in the beginning of them

Variable notation III

- Don’t show your superiority/cleverness

Variable notation III

- Don't show your superiority/cleverness
- i and j for loop counters

Variable notation III

- Don't show your superiority/cleverness
- i and j for loop counters
- Don't use names already defined (e.g., data)

Comments

Comments I

- Be consistent!

Comments I

- Be consistent!
- Don't repeat the code in different words (code should be written well enough already)

Comments I

- Be consistent!
- Don't repeat the code in different words (code should be written well enough already)
- Explain the goal & context for the code

Comments II

- Don’t forget to change comments when you change the code

Comments II

- Don't forget to change comments when you change the code
- Use TODO or todo to mark spots that still need your attention (e.g., when commenting out parts of the code)

Information at top of document

Additional information I

- date

Additional information I

- date
- name of file

Additional information I

- date
- name of file
- corresponding author and contact

Additional information II

- List of software necessary to run code (including OS)

Additional information II

- List of software necessary to run code (including OS)
- List of necessary packages (and version numbers)

Additional information III

The following could also be included in an additional read me file

- Folder structure necessary

Additional information III

The following could also be included in an additional read me file

- Folder structure necessary
- What data is needed (where)

Additional information III

The following could also be included in an additional read me file

- Folder structure necessary
- What data is needed (where)
- Context/description (e.g., journal article)

Folder structure

Folder structure I

A uniform folder structure can help you find items in older projects.

As with most previous points, you should choose whatever works best for you.

- Be consistent!

Folder structure I

A uniform folder structure can help you find items in older projects.

As with most previous points, you should choose whatever works best for you.

- Be consistent!
- File and folder naming: use a-z, 0-9 and underscore

Folder structure I

A uniform folder structure can help you find items in older projects.

As with most previous points, you should choose whatever works best for you.

- Be consistent!
- File and folder naming: use a-z, 0-9 and underscore
- No blank spaces in file or folder names

Folder structure I

A uniform folder structure can help you find items in older projects.

As with most previous points, you should choose whatever works best for you.

- Be consistent!
- File and folder naming: use a-z, 0-9 and underscore
- No blank spaces in file or folder names
- Names: simple, as short as possible, as long as necessary

Folder structure II

A possible structure could be one of these:

- e.g., folders: data, fig, lit, org, pub, src

Folder structure II

A possible structure could be one of these:

- e.g., folders: data, fig, lit, org, pub, src
- e.g., folders: code, documentation, inputs, outputs

Folder structure II

A possible structure could be one of these:

- e.g., folders: data, fig, lit, org, pub, src
- e.g., folders: code, documentation, inputs, outputs
- e.g., folders: data, experiments, material, paper, presentations

Task I

Create your own style guide!

See my_coding_style.docx in the handout folder.

Send the guide to: `Claudia.ODonovan-Bellante@gesis.org`

General reproducibility tips

General reproducibility tips

- R Studio
- Loading packages
- Loading data

R Studio

IDE for R, not necessary but helpful

The following should also be installed:

- MixTeX (full version, win only, instructions: https://tobiasheycke.github.io/pages/fullmiktex.html)
- Add-in: citr
- packages (rmarkdown, devtools, papaja)

Set up R Studio

- save utf-8 (Tools > Global Options > Code > Saving)
- set up terminal (Shift+Alt+T, Tools > Global Options > Terminal > Shell > Git Bash)
- set up color scheme (optional, Tools > Global Options > Appearance)
- set up 80 character vertical line (optional, Tools > Global Options > Code > Display)
- R projects

Loading packages

All packages should to be loaded in the beginning of the script (first chunk)

Helpful function: pacman

```
all_packages <- c("papaja", "afex", "tidyr")
if(!require("pacman")) stop("Please install the 'pacman'
pacman::p_load(all_packages)
```

Loading data I

- Ideally script should use raw data files
- If preparing data takes too long: cache processed data
- Either use chunk options (see later)
- Or save processed data as csv/Rdata and have Boolean in beginning of script to decide whether to use raw or processed data

Loading data II

Manual caching example:

```
load_processed_data <- TRUE
if(load_processed_data){
  load("processed_data/visibility_data")
} else {
  visibility_data <- read_gitdata("path_to_file"
                                  , "vis_data")
  save(visibility_data
       , file = "processed_data/visibility_data")
}
```

Loading data III

If you use personally distributed data sets (e.g., large surveys):

- use md5 checksums to test if data sets are identical

Task II

Write a function that tests if a data set has the same md5 checksum as a provided sum

Tip: `tools::md5sum()`

You can test the function using the file: `example_data.csv`

End of part I