Meta-analysis data management with the \{metamanager\} package

Tamas Nagy, PhD
Eotvos Lorand University
Budapest, Hungary
We are living in a word where facts can be regarded as subjective therefore we need more evidence to support claims.
Pyramid of scientific evidence

- Idea, opinion, anecdote
- Case study, case series study
- Cross-sectional studies
- Case-control studies
- Longitudinal and cohort studies
- Experiment, RCT
- Meta-analysis, systematic review

Stronger evidence
What is the point of a meta-analysis

- Helps to draw conclusions for a whole field
- Synthesize research findings in an area across several independent studies
- May help to show the “true” effect
- It is possible to test new hypotheses if data are available from the articles
The process of doing a meta-analysis

1. Search literature
2. Screen articles for eligibility
3. Select the studies of interest
4. Extract data from published articles
5. Calculate effect sizes and aggregate across studies
6. Statistical inference
Shortcomings of meta-analysis

Garbage in - garbage out

Publication bias

Questionable research practices
There are standards that regulate how to do a meta-analysis properly

- **Protocol** defines the aims and methods
- Decisions about article screening and data extraction should be made at least by **two independent researchers**
- Assessing the quality of the included studies
- But usually the amount of work requires more collaborators
Doing a meta-analysis is not easy

- Domain specific knowledge
- Methodological knowledge
- Teamwork
- Data management and organization
Data management usually organized in Excel
There should be a package for this!

- Reduce project management overhead
- File management should be automated and made reproducible
- Not straightforward to coordinate human and machine generated outputs
- Integrate with a GUI that is easy to use
- Currently, there is only one other package `{metagear}` for meta-analysis workflow management, but it is not using the tidyverse principles, and it has a different philosophy and focus
The \texttt{metamananger} is here to help!

- Automate file creation for manual coding (via Google drive)
- Aggregate, evaluate, validate human coded data
- Clear and reproducible outputs
- Embracing the tidyverse principles (piping, readability, simplicity of functions, seamless integration with tidy packages)
Setting up a meta-analysis project

`init_gdrive("research/meta-analysis")`: Creates a folder structure on google drive. Can be self specified or a default structure.

`literature_search/`
`screening/`
`fulltext/`
`screening_consensus/`
`extraction/`
`extraction_consensus/`  

`init_team()`: Creates a google sheet with team member data. Accepts a tibble or opens a google sheet with header

`init_meta("research/meta-analysis")`: Does both, using defaults
Identify duplicates

```r
merge_sources(workaholism_psychinfo,
              workaholism_pubmed,
              workaholism_scoporus,
              .renames = c("journal" = "publication"),
              .tidy_names = TRUE,
              .all_char = TRUE) %>%
flag_duplicate_title(keys = c("psyid", "pmid", "doi", "eid", "sid")) %>%
make_id(c("psyid", "pmid", "doi", "eid", "sid")) %>%
flag_duplicate_id(keys = c("psyid", "pmid", "doi", "eid", "sid")) %>%
filter(duplicate_by_title != 1 & duplicate_by_id != 1)
```
Create article screening files

```r
merged_records %>%
  assign_articles(team_df = team_df, effort_col = "screening_effort", seed = 42)
%>%
  add_columns(c("decision", "reason"))
%>%
  save_locally(local_path = "temp/screening/",
               nesting = "reviewer",
               postfix = "screening",
               overwrite = TRUE)
```

# A tibble: 4,390 x 15

<table>
<thead>
<tr>
<th>decision</th>
<th>reason</th>
<th>identifier</th>
<th>id</th>
<th>source</th>
<th>title</th>
<th>journal</th>
<th>authors</th>
<th>year</th>
<th>abstract</th>
<th>type</th>
<th>level</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>pmid</td>
<td>24357~</td>
<td>pubmed</td>
<td>&quot;Res~</td>
<td>Journa~</td>
<td>Silk, ~</td>
<td>2013</td>
<td>&quot;Whilst n~</td>
<td>NA</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>pq_id</td>
<td>33710~</td>
<td>proqu~</td>
<td>&quot;U G~</td>
<td>NA</td>
<td>Young,~</td>
<td>2007</td>
<td>Backgroun~</td>
<td>THES</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>doi</td>
<td>10.10~</td>
<td>Scopus</td>
<td>&quot;B~</td>
<td>Accide~</td>
<td>Mammar~</td>
<td>2013</td>
<td>Two studi~</td>
<td>Arti~</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>doi</td>
<td>10.11~</td>
<td>Scopus</td>
<td>&quot;B~</td>
<td>Entrep~</td>
<td>Wolfe ~</td>
<td>2015</td>
<td>In this s~</td>
<td>Arti~</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>doi</td>
<td>10.10~</td>
<td>Scopus</td>
<td>&quot;C~</td>
<td>Femini~</td>
<td>De E.N.</td>
<td>2012</td>
<td>&quot;This art~</td>
<td>Arti~</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
<tr>
<td>&quot;NA&quot;</td>
<td>&quot;NA&quot;</td>
<td>doi</td>
<td>10.10~</td>
<td>Scopus</td>
<td>&quot;G~</td>
<td>Comput~</td>
<td>Charsk~</td>
<td>2011</td>
<td>Does usin~</td>
<td>Arti~</td>
<td>NA</td>
<td>reviewe~</td>
</tr>
</tbody>
</table>

# ... with 4,380 more rows, and 2 more variables: reviewer <chr>, name <chr>
Uploading and downloading gdrive folders

#This also converts files to a google sheets that can be edited online
upload_to_gdrive(local_path = "temp/screening/",
gdrive_path = "research/meta-analysis/screening",
overwrite = TRUE)

#Downloads all files from a google folder as a nested dataframe
screening <- get_from_gdrive(gdrive_path = "research/meta-analysis/screening",
all_char = FALSE)
Why Google drive?

- There are built-in functions in the package to convert data files to Google sheets
- Google sheets provide **collaborative editing** with **version control**
- **User friendly** GUI and API
- **Not obligatory**: All files are also saved locally, other cloud solutions can be used too, like github or dropbox
- It would be possible to create a shiny app for front end (e.g. [http://dev.open-meta.org/](http://dev.open-meta.org/))
Evaluate screening performance

```r
screening %>%
  unnest() %>%
  calculate_agreement()
```

<table>
<thead>
<tr>
<th>name_pair</th>
<th>include_both</th>
<th>exclude_both</th>
<th>no_agreement</th>
<th>invalid_decision</th>
<th>all_items</th>
<th>kr_level</th>
<th>kr_alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom_Jerry</td>
<td>0</td>
<td>34</td>
<td>20</td>
<td>0</td>
<td>54</td>
<td>nominal</td>
<td>-0.216</td>
</tr>
<tr>
<td>Tom_Rose</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>nominal</td>
<td>-0.343</td>
</tr>
<tr>
<td>Jerry_Rose</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>10</td>
<td>nominal</td>
<td>0.0952</td>
</tr>
</tbody>
</table>
Human errors during coding of the studies

Valid reasons for exclusion

- publication type
- study design
- sample characteristics
- manipulation
- outcome measures
- time frame
- duplicated
- unavailable
- unspecified reason

What we see in the file

- publication type
- pub type
- publication
- study design
- study d
- sample characteristics
- sample
- samp
- ...

Correcting and highlighting human errors

```r
# Correct typos in the reasons for excluded studies
correct_terms <- read_lines("valid_exclude_reasons.txt")
screening %>%
  unnest() %>%
  mutate(exclude_reason_corr = correct_categories(exclude_reason, correct_terms))
```
Humans tend to ignore variable types

```r
# Safely convert col_types with human readable errors

articles %>%
    conversion_errors(c("year", "measurement_n"), as.integer)

# A tibble: 2 x 3
file  fun            conversion_errors observations
<chr> <chr>            <dbl> <list>
1 Ali_extract.csv as.integer      25 list(43, 88, 89, 123, ...)  
2 Tamas_extract.csv as.integer     3  list(43, 288, 589)
```
Create data extraction files

```r
variables_to_add <- c("subsample_id", "subsample_age", "subsample_female%", "measurement_point", "measurement_n", "measurement_mean", "measurement_sd", "study_design", "study_quality")

get_from_gdrive(screening_path) %>%
  unnest() %>%
  filter(is.na(exclude_reason)) %>%
  assign_articles(team_df = team_df, effort_col = "extraction_effort", seed = 42) %>%
  add_columns(variables_to_add, before = FALSE) %>%
  save_locally(local_path = "temp/extraction/", nesting = "reviewer", postfix = "screening", overwrite = TRUE)

upload_to_gdrive(local_path = "temp/extraction/",
gdrive_path = "research/meta-analysis/extraction", overwrite = TRUE)
```

Twitter: @nagyt
https://github.com/nthun/metamanager
Future plans

- Machine learning augmented screening process
- Automatic extraction of citations from included articles (pdf or wos)
- More data visualization (flow charts, keywords in abstracts, ...)
- Handle article pdfs (download, categorize, distribute, etc.)
- Tidyeval
- Fancy package stuff: more vignettes, unit tests, pkgdown site, hex sticker
How to get / contribute?

Please star, fork, open issues, PR on github!  
https://github.com/nthun/metamanager

Installation:

devtools::install_github("nthun/metamanager")

Thank you!

Twitter: @nagyt