Analysis and visualisations for "Reproducible research and GIScience: an evaluation using AGILE conference papers"

Daniel Nüst, Barbara Hofer

23 April, 2018

Contents

License

See the paper's "Author Contributions" section for details on the contributors of data files.

Metadata

Required libraries and runtime environment description.

```
library("pdftools")
library("stringr")
library("tidyverse")
library("knitr")
library("tidytext")
library("wordcloud")
library("RColorBrewer")
library("readr")
library("ggplot2")
library("rvest")
library("jsonlite")
library("reshape2")
library("ggthemes")
library("gridExtra")
library("grid")
library("kableExtra")
library("devtools")
library("rlang")
library("huxtable")
library("here")
library("httr")
```

```
devtools::session_info(include_base = TRUE)
```

```
## Session info -------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.4.4 (2018-03-15)
##  system   x86_64, linux-gnu
##  ui       RStudio (1.1.383)
##  language en
##  collate  en_US.UTF-8
##  tz       Europe/Berlin
##  date     2018-04-23
```

```
## Packages -----------------------------------------------------------------
```

```
##  package      * version date       source
##  assertthat     0.2.0   2017-04-11 CRAN (R 3.4.0)
##  backports      1.1.2   2017-12-13 CRAN (R 3.4.3)
##  base         * 3.4.4   2018-03-16 local
##  bindr          0.1.1   2018-03-13 CRAN (R 3.4.4)
##  bindrcpp     * 0.2.2   2018-03-29 CRAN (R 3.4.4)
##  broom          0.4.4   2018-03-29 CRAN (R 3.4.4)
##  cellranger     1.1.0   2016-07-27 CRAN (R 3.4.0)
##  cli            1.0.0   2017-11-05 CRAN (R 3.4.2)
##  colorspace     1.3-2   2016-12-14 cran (@1.3-2)
##  compiler       3.4.4   2018-03-16 local
##  crayon         1.3.4   2017-09-16 CRAN (R 3.4.1)
##  curl           3.2     2018-03-28 CRAN (R 3.4.4)
##  datasets     * 3.4.4   2018-03-16 local
##  devtools     * 1.13.5  2018-02-18 CRAN (R 3.4.3)
```

```
##  digest         0.6.15  2018-01-28 CRAN (R 3.4.3)
##  dplyr        * 0.7.4   2017-09-28 CRAN (R 3.4.2)
##  evaluate       0.10.1  2017-06-24 CRAN (R 3.4.0)
##  forcats      * 0.3.0   2018-02-19 CRAN (R 3.4.3)
##  foreign        0.8-69  2017-06-21 CRAN (R 3.4.0)
##  ggplot2      * 2.2.1   2016-12-30 CRAN (R 3.4.2)
##  ggthemes     * 3.4.2   2018-04-03 CRAN (R 3.4.4)
##  glue           1.2.0   2017-10-29 CRAN (R 3.4.2)
##  graphics     * 3.4.4   2018-03-16 local
##  grDevices    * 3.4.4   2018-03-16 local
##  grid         * 3.4.4   2018-03-16 local
##  gridExtra    * 2.3     2017-09-09 CRAN (R 3.4.1)
##  gtable         0.2.0   2016-02-26 CRAN (R 3.4.0)
##  haven          1.1.1   2018-01-18 CRAN (R 3.4.3)
##  here         * 0.1     2017-05-28 CRAN (R 3.4.4)
##  highr          0.6     2016-05-09 CRAN (R 3.4.0)
##  hms            0.4.2   2018-03-10 CRAN (R 3.4.3)
##  htmltools      0.3.6   2017-04-28 CRAN (R 3.4.0)
##  httr         * 1.3.1   2017-08-20 CRAN (R 3.4.1)
##  huxtable     * 3.0.0   2018-02-23 CRAN (R 3.4.3)
##  janeaustenr    0.1.5   2017-06-10 cran (@0.1.5)
##  jsonlite     * 1.5     2017-06-01 cran (@1.5)
##  kableExtra   * 0.8.0   2018-04-05 CRAN (R 3.4.4)
##  knitr        * 1.20    2018-02-20 CRAN (R 3.4.3)
##  labeling       0.3     2014-08-23 cran (@0.3)
##  lattice        0.20-35 2017-03-25 CRAN (R 3.3.3)
##  lazyeval       0.2.1   2017-10-29 CRAN (R 3.4.2)
##  lubridate      1.7.3   2018-02-27 CRAN (R 3.4.3)
##  magrittr       1.5     2014-11-22 CRAN (R 3.4.0)
##  Matrix         1.2-14  2018-04-09 CRAN (R 3.4.4)
##  memoise        1.1.0   2017-04-21 CRAN (R 3.4.3)
##  methods      * 3.4.4   2018-03-16 local
##  mnormt         1.5-5   2016-10-15 cran (@1.5-5)
##  modelr         0.1.1   2017-07-24 CRAN (R 3.4.1)
##  munsell        0.4.3   2016-02-13 cran (@0.4.3)
##  nlme           3.1-137 2018-04-07 CRAN (R 3.4.4)
##  parallel       3.4.4   2018-03-16 local
##  pdftools     * 1.6     2018-03-27 CRAN (R 3.4.4)
##  pillar         1.2.1   2018-02-27 CRAN (R 3.4.3)
##  pkgconfig      2.0.1   2017-03-21 cran (@2.0.1)
##  plyr           1.8.4   2016-06-08 cran (@1.8.4)
##  psych          1.8.3.3 2018-03-30 CRAN (R 3.4.4)
##  purrr        * 0.2.4   2017-10-18 CRAN (R 3.4.2)
##  R6             2.2.2   2017-06-17 CRAN (R 3.4.0)
##  RColorBrewer * 1.1-2   2014-12-07 cran (@1.1-2)
##  Rcpp           0.12.16 2018-03-13 cran (@0.12.16)
##  readr        * 1.1.1   2017-05-16 CRAN (R 3.4.0)
##  readxl         1.0.0   2017-04-18 cran (@1.0.0)
##  reshape2     * 1.4.3   2017-12-11 CRAN (R 3.4.3)
##  rlang        * 0.2.0   2018-02-20 CRAN (R 3.4.3)
##  rmarkdown      1.9     2018-03-01 CRAN (R 3.4.3)
##  rprojroot      1.3-2   2018-01-03 CRAN (R 3.4.3)
##  rstudioapi     0.7     2017-09-07 CRAN (R 3.4.1)
##  rvest        * 0.3.2   2016-06-17 CRAN (R 3.4.2)
```

```
##  scales         0.5.0   2017-08-24 CRAN (R 3.4.1)
##  selectr        0.4-1   2018-04-06 CRAN (R 3.4.4)
##  slam           0.1-42  2017-12-21 CRAN (R 3.4.3)
##  SnowballC      0.5.1   2014-08-09 cran (@0.5.1)
##  stats        * 3.4.4   2018-03-16 local
##  stringi        1.1.7   2018-03-12 CRAN (R 3.4.4)
##  stringr      * 1.3.0   2018-02-19 CRAN (R 3.4.3)
##  tibble       * 1.4.2   2018-01-22 CRAN (R 3.4.3)
##  tidyr        * 0.8.0   2018-01-29 CRAN (R 3.4.3)
##  tidyselect     0.2.4   2018-02-26 CRAN (R 3.4.3)
##  tidytext     * 0.1.8   2018-03-21 CRAN (R 3.4.4)
##  tidyverse    * 1.2.1   2017-11-14 CRAN (R 3.4.2)
##  tokenizers     0.2.1   2018-03-29 CRAN (R 3.4.4)
##  tools          3.4.4   2018-03-16 local
##  utils        * 3.4.4   2018-03-16 local
##  viridisLite    0.3.0   2018-02-01 CRAN (R 3.4.3)
##  withr          2.1.2   2018-03-15 cran (@2.1.2)
##  wordcloud    * 2.5     2014-06-13 CRAN (R 3.4.1)
##  xml2         * 1.2.0   2018-01-24 CRAN (R 3.4.3)
##  yaml           2.1.18  2018-03-08 CRAN (R 3.4.3)
```

This document is versioned in a public git repository, https://github.com/nuest/reproducible-research-and-giscience, and the current revision is 86382ef.

Prerequisites

API key

An API key is needed for accessing the Springer API to automatically retrieve the number of full papers. Create a file `.Renviron` next to this document and add the following line:

```
SPRINGER_API_KEY=<your key>
```

Or set the environment variable within this notebook:

```
Sys.setenv(SPRINGER_API_KEY = "<your key>")
```

```
if (is.na(Sys.getenv("SPRINGER_API_KEY", unset = NA)))
  warning("API key is not set, please check the section \"Prerequisites\" of the Rmd file.")
```

```
## Warning: API key is not set, please check the section "Prerequisites" of
## the Rmd file.
```

```
data_path <- "paper-corpus"
```

Data

The data for the analysis is required in form of a directory with PDF files. Add the PDFs to a directory called `paper-corpus` next to this file.

You can contact the original paper authors and ask for the test dataset to reproduce the full analysis. Alternatively, you can download a selection of AGILE short papers to test the workflow using the code below which is *not* executed by default.

```
dir.create(here::here(data_path))

# harvest links to PDFs, select more years for more data,
# e.g. c(2003:2017) and increase max_files_per_year
years <- c(2017)
max_files_per_year <- 10
base_url <- "https://agile-online.org/index.php/conference/proceedings/proceedings-"
proceedings_urls <- sapply(X = as.character(years),
                           FUN = function(x) { paste0(base_url, x) }, USE.NAMES=TRUE)
proceedings_html <- lapply(X = proceedings_urls, FUN = read_html)

# papers, posters, abstracts of full papers, keynotes - we don't care as long it is pdf
# we might also catch both abstract of a poster and the poster itself
get_links <- function(page){
  all_links <- page %>%
    html_nodes(css = "a") %>%
    html_attr("href") %>%
    as.list()
  pdf_links <- tibble(links = all_links) %>%
    filter(str_detect(links, pattern = "pdf$"))
  return(pdf_links)
}

proceedings_links_any <- lapply(X = proceedings_html, FUN = get_links)

base_url <- "https://agile-online.org/"
```

```
files <- lapply(X = names(proceedings_links_any), FUN = function(x) {
  year <- x
  file_in_year <- 1
  max_files <- min(max_files_per_year, length(proceedings_links_any[[year]]$links))
  year_links <- proceedings_links_any[[year]]$links[c(1:max_files)]

  files <- lapply(X= year_links, FUN = function(x) {
    link_url <- paste0(base_url, x)
    filename <- here::here(data_path,
                           paste0(year, file_in_year, "_", basename(x)))
    if(!file.exists(filename)) {
      response <- GET(url = link_url)
      raw_content <- content(response, "raw")
      writeBin(raw_content, filename)
      #cat("Saved URL", link_url, "\t\tto file\t\t", filename, "\n")
    }
    filename
    file_in_year <<- file_in_year + 1
  })
  files
  cat("Downloaded", length(files), "files for year", year, "\n")
})
```

Code

The **text analysis** is based the R package `tidytext` from the `tidyverse` suite of packages and uses the `dplyr` grammar. Read the `tidytext` tutorial to learn about the used functions and concepts.

The **plots and tables** of survey data and evaluation use the packages `ggplot2` and `knitr::kable()`/`kableExtra`.

Reproduce paper

If you do not have the original data or do not download the data, you cannot reproduce the text analysis part of the paper, i.e. wordcloud and terms frequency analysis. **You can still reproduce the other figures**.

To create the PDF of the reproducibility package based on this document you can run the following commandsin a new R session after completing the prerequisites with the original paper corpus data.

```
require("knitr")
require("rmarkdown")
rmarkdown::render("agile-rr-paper-corpus.Rmd", output_format = "pdf_document")
```

Paper corpus: loading and cleaning

The test dataset for the analysis cannot be shared publicly due to copyrights. It comprises all nominees for the best paper award since 2008, both short papers and full papers. See the paper supplemental files for a full list of citations.

The analysis loads all files from the directory */home/daniel/git/reproducible-research-and-giscience/paper-corpus*.

```
files <- dir(path = here::here(data_path), pattern = ".pdf$", full.names = TRUE)
```

This analysis was created with the following 32 documents, 12 of which are short papers:

```
##  [1] "/paper-corpus/12010_Raubal_Winter_AGILE_winner.pdf"
##  [2] "/paper-corpus/12012_Osaragi_Hoshino_AGILE.pdf"
##  [3] "/paper-corpus/12013_Osaragi_Tsuda_AGILE.pdf"
##  [4] "/paper-corpus/12014_scheider_jones_sanchez_kessler_AGILE_winner_authorcopy.pdf"
##  [5] "/paper-corpus/12015_Kuhn_Ballatore_AGILE_winner_authorcopy.pdf"
##  [6] "/paper-corpus/12016_Almer_Perko_etal_AGILE_winner_978-3-319-33783-8_20.pdf"
##  [7] "/paper-corpus/12017_Zhu_Kyriakidis_Janowicz_AGILE_winner.pdf"
##  [8] "/paper-corpus/22010_Schaeffer_Baranski_Foerster_AGILE.pdf"
##  [9] "/paper-corpus/22012_Magalhaes_andrade_etal_AGILE.pdf"
## [10] "/paper-corpus/22013_Baglatzi_Kuhn_AGILE_authorcopy.pdf"
## [11] "/paper-corpus/22014_Groechenig_Brunauer_Rehrl_AGILE.pdf"
## [12] "/paper-corpus/22015_Mazimpaka_Timpf_AGILE_ocr.pdf"
## [13] "/paper-corpus/22016_Wiemann_AGILE_winner_978-3-319-33783-8_8.pdf"
## [14] "/paper-corpus/22017_Knoth_VocknerM_Mittlboeck_AGILE.pdf"
## [15] "/paper-corpus/32010_Körner_Hecker_etal_AGILE.pdf"
## [16] "/paper-corpus/32012_Foerster_Baranski_Borsutzky_AGILE.pdf"
## [17] "/paper-corpus/32013_shortpaper_Schwering_Li_Anacta_AGILE_winner.pdf"
## [18] "/paper-corpus/32014_Fan_Zipf_Fu_AGILE_9783319036106.pdf"
## [19] "/paper-corpus/32015_Steuer_Machl_etal_AGILE.pdf"
## [20] "/paper-corpus/32016_Juhasz_Hochmair_AGILE_978-3-319-33783-8_9.pdf"
## [21] "/paper-corpus/32017_Konkol_Kray_Ostkamp_AGILE.pdf"
## [22] "/paper-corpus/42012_shortpaper_Merki_Laube_AGILE.pdf"
## [23] "/paper-corpus/42013_shortpaper_Stein_Schlieder_AGILE.pdf"
## [24] "/paper-corpus/42014_shortpaper_Soleymani_vanLoon_Weibel_AGILE_winner.pdf"
## [25] "/paper-corpus/42015_shortpaper_Fogliaroni_Hobel_AGILE_winner.pdf"
## [26] "/paper-corpus/42016_shortpaper_Josselin_Boularouk_etal_AGILE_winner.pdf"
## [27] "/paper-corpus/42017_shortpaper_Haumann_Bucher_Jonietz_winner.pdf"
## [28] "/paper-corpus/52012_shortpaper_Kiefer_Straub_Raubal_AGILE.pdf"
## [29] "/paper-corpus/52014_shortpaper_Wiemann_Bernard_AGILE.pdf"
## [30] "/paper-corpus/52015_shortpaper_Heinz_Schlieder_AGILE.pdf"
## [31] "/paper-corpus/52016_shortpaper_Rosser_Pourabdollah_etal_AGILE.pdf"
## [32] "/paper-corpus/52017_shortpaper_Brinkhoff.pdf"
```

Read the data from PDFs and preprocess to create a tidy data structure without stop words:

```
texts <- lapply(files, pdf_text)
texts <- unlist(lapply(texts, str_c, collapse = TRUE))
infos <- lapply(files, pdf_info)

if (!is.null(texts)) {
  tidy_texts <- tibble(id = str_extract(files, "[0-9]+"),
                       file = files,
                       text = texts,
```

```
                        pages = map_chr(infos, function(info) {info$pages}))

  papers_words <- tidy_texts %>%
    select(file,
           text) %>%
    unnest_tokens(word, text)

  my_stop_words <- tibble(
    word = c(
      "et",
      "al",
      "fig",
      "e.g",
      "i.e",
      "http",
      "ing",
      "pp",
      "figure",
      "based"
    ),
    lexicon = "agile"
  )
  all_stop_words <- stop_words %>%
    bind_rows(my_stop_words)

  suppressWarnings({
    no_numbers <- papers_words %>%
      filter(is.na(as.numeric(word)))
  })
  no_stop_words <- no_numbers %>%
    anti_join(all_stop_words, by = "word") %>%
    mutate(id = str_extract(file, "[0-9]+"))
} else {
  warning("No input data provided at ", here::here(data_path))
  # create empty outputs if no input data is given
  papers_words <- tibble(word = c("no data"))
  no_stop_words <- tibble(id = c("no data"), word = c("no data"))
  tidy_texts <- tibble(id = c("no data"))
}
```

About 49 % of the words are considered stop words.

How many non-stop words does each document have?

```
kable(no_stop_words %>%
  group_by(id) %>%
  summarise(words = n()) %>%
  arrange(desc(words)))
```

id	words
12017	3735
12015	3714
12010	3606
12014	3568
32012	3441
22010	3438
12016	3428
22013	3253
32017	3148
22014	3051
32016	2997
22016	2956
22015	2870
12012	2859
32010	2851
22017	2697
32015	2590
32014	2568
42012	2540
12013	2536
42013	2356
42014	2179
42016	1929
42015	1877
22012	1850
52016	1797
52012	1786
32013	1773
42017	1747
52017	1661
52014	1540
52015	1383

Note: In the original paper corpus there was an issue with reading in one paper, which only had 15 words. Since it was not possible to copy or extract text, it was send through an OCR process (using OCRmyPDF) with the command `docker run -v $(pwd)/paper-corpus:/home/docker -it jbarlow83/ocrmypdf-tess4 --force-ocr 22015_Mazimpaka_Timpf_AGILE.pdf 22015_Mazimpaka_Timpf_AGILE_ocr.pdf` and the created file was used instead of the original.

Table: Reproducible research-related keywords in the corpus

How often do the following terms appear in each paper?

The detection matches full words using regex option `\b`.

- reproduc (“, reproducibility, reproducible, reproduce, reproduction)
- replic (`replicat.*`, i.e. replication, replicate)
- repeatab (`repeatab.*`, i.e. repeatability, repeatable)
- software
- (pseudo) code/script(s) [column name *code*]
- algorithm (`algorithm.*`, i.e. algorithms, algorithmic)
- process (`process.*`, i.e. processing, processes, preprocessing)
- data (`data.*`, i.e. dataset(s), database(s))
- result(s)
- repository(ies)

```
tidy_texts_lower <- str_to_lower(tidy_texts$text)
word_counts <- tibble(
  id = tidy_texts$id,
  `reproduc..` = str_count(tidy_texts_lower, "\\breproduc.*\\b"),
  `replic..` = str_count(tidy_texts_lower, "\\breplicat.*\\b"),
  `repeatab..` = str_count(tidy_texts_lower, "\\brepeatab.*\\b"),
  `code` = str_count(tidy_texts_lower,
    "(\\bcode\\b|\\bscript.*\\b|\\bpseudo\ code\\b)"),
  software = str_count(tidy_texts_lower, "\\bsoftware\\b"),
  `algorithm(s)` = str_count(tidy_texts_lower, "\\balgorithm.*\\b"),
  `(pre)process..` = str_count(tidy_texts_lower,
                "(\\bprocess.*\\b|\\bpreprocess.*\\b|\\bpre-process.*\\b)"),
  `data.*` = str_count(tidy_texts_lower, "\\bdata.*\\b"),
  `result(s)` = str_count(tidy_texts_lower, "\\bresults?\\b"),
  `repository/ies` = str_count(tidy_texts_lower, "\\brepositor(y|ies)\\b")
)

# https://stackoverflow.com/a/32827260/261210
sumColsInARow <- function(df, list_of_cols, new_col) {
  df %>%
    mutate_(.dots = ~Reduce(`+`, .[list_of_cols])) %>%
    setNames(c(names(df), new_col))
}

word_counts_sums <- sumColsInARow(
  word_counts,
  names(word_counts)[names(word_counts) != "id"], "all") %>%
  arrange(desc(all))

# load paper names from evaluation table
citations <- read_csv("Paper_Evaluation.csv",
                      col_types = cols_only(author = col_character(),
                                            paper = col_character()))
```

```
## Warning: Missing column names filled in: 'X12' [12], 'X14' [14]
```

```
word_counts_sums <- word_counts_sums %>%
  left_join(citations, by = c("id" = "paper")) %>%
  select(citation = author, `reproduc..`:`result(s)`, `all`)
```

```r
word_counts_sums_total <- word_counts_sums %>%
  summarise_if(is.numeric, funs(sum)) %>%
  add_column(citation = "Total", .before = 0)
word_counts_sums <- rbind(word_counts_sums, word_counts_sums_total)

# for inline testing: kable(word_counts_sums)
kable(word_counts_sums,
      caption = paste0("Reproducible research-related keywords in the corpus,",
                       " ordered by sum of matches per paper"),
      format = "latex", booktabs = TRUE) %>%
  kableExtra::landscape()
```

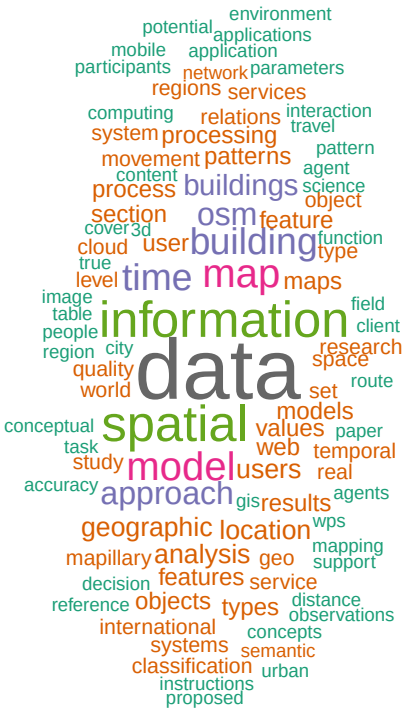
Table 2: Reproducible research-related keywords in the corpus, ordered by sum of matches per paper

citation	reproduc..	replic..	repeatab..	code	software	algorithm(s)	(pre)process..	data.*	result(s)	all
Foerster et al. (2012)	0	0	0	2	3	11	140	129	41	326
Wiemann & Bernard (2014)	0	0	0	0	0	0	20	98	3	123
Mazimpaka & Timpf (2015)	0	0	0	3	0	4	4	97	10	118
Steuer et al. (2015)	0	0	0	0	0	25	12	64	17	118
Schäffer et al. (2010)	0	0	0	0	10	1	26	65	6	108
Rosser et al. (2016)	0	0	0	0	2	1	42	51	6	105
Gröchening et al. (2014)	0	0	0	0	0	3	2	69	27	101
Almer et al. (2016)	0	0	0	1	1	1	22	53	22	100
Magalhães et al. (2012)	0	0	0	2	1	20	52	9	1	85
Juhász & Hochmair (2016)	0	0	0	0	1	1	2	55	11	70
Wiemann (2016)	0	0	0	0	3	0	8	55	1	69
Fan et al. (2014)	0	0	0	0	0	3	8	44	12	67
Merki & Laube (2012)	0	0	0	0	0	9	6	40	6	62
Zhu et al. (2017)	2	2	0	2	0	10	7	32	6	61
Kuhn & Ballatore (2015)	0	0	1	2	14	1	5	26	8	58
Soleymani et al. (2014)	1	0	0	0	0	0	4	39	9	56
Fogliaroni & Hobel (2015)	0	0	0	0	0	3	14	30	5	52
Osaragi & Hoshino (2012)	0	0	0	0	0	0	5	36	7	48
Stein & Schlieder (2013)	0	0	0	0	0	0	3	42	3	48
Körner et al. (2010)	0	0	0	0	0	6	5	30	4	45
Knoth et al. (2017)	0	0	0	3	2	1	6	25	7	44
Raubal & Winter (2010)	0	0	0	1	1	1	18	0	13	34
Konkol et al. (2017)	1	0	0	3	1	1	2	4	19	31
Kiefer et al. (2012)	1	0	0	0	2	1	9	10	8	31
Haumann et al. (2017)	0	0	0	0	0	6	8	10	2	26
Josselin et al. (2016)	0	0	0	0	2	1	9	5	8	25
Heinz & Schlieder (2015)	1	0	0	2	1	3	2	14	2	25
Osaragi & Tsuda (2013)	0	0	0	1	1	0	3	16	2	23
Baglatzi & Kuhn (2013)	1	0	0	0	0	0	6	12	3	22
Scheider et al. (2014)	0	0	0	0	1	0	0	13	4	19
Brinkhoff (2017)	0	0	0	0	1	9	2	3	2	17
Schwering et al. (2013)	0	0	0	0	0	4	2	3	5	14
Total	7	2	1	22	47	126	454	1179	280	2131

Figure: Word cloud of test corpus papers (left), and top words (right)

```
countPapersUsingWord <- function(the_word) {
  sapply(the_word, function(w) {
    no_stop_words %>%
      filter(word == w) %>%
      group_by(id) %>%
      count %>%
      nrow
  })
}

top_words <- no_stop_words %>%
  group_by(word) %>%
  tally %>%
  arrange(desc(n)) %>%
  head(20) %>%
  mutate(`# papers` = countPapersUsingWord(word)) %>%
  add_column(place = c(1:nrow(.)), .before = 0)
```

```
set.seed(1)
if (max(top_words$n) < 100) {
  minimum_occurence <- round(mean(top_words$n))
} else {
  minimum_occurence <- 100
}
cloud_words <- no_stop_words %>%
  group_by(word) %>%
  tally %>%
  filter(n >= minimum_occurence) %>% # 100 chosen manually
  arrange(desc(n))

if (nrow(cloud_words) > 0) {
  def.par <- par(no.readonly = TRUE)
  layout(matrix(c(1,0,2), 1, 3, byrow = TRUE), widths = c(lcm(12),lcm(6),lcm(6)))
  wordcloud(cloud_words$word, cloud_words$n,
            max.words = Inf,
            random.order = FALSE,
            fixed.asp = FALSE,
            rot.per = 0,
            color = brewer.pal(8,"Dark2"))
  grid.table(as.matrix(top_words),
             theme = ttheme_minimal(
               base_size = 9,
               padding = unit(c(5,5), "pt")))
  par(def.par)
} else {
  warning("No input data for wordcloud provided")
}
```



place	word	n	# papers
1	data	1058	31
2	information	589	32
3	spatial	577	30
4	map	411	25
5	model	411	25
6	building	381	24
7	time	378	30
8	approach	297	32
9	osm	292	8
10	buildings	266	15
11	geographic	249	28
12	location	239	26
13	analysis	229	28
14	users	225	19
15	results	207	30
16	web	206	21
17	models	202	20
18	values	202	23
19	patterns	196	16
20	maps	189	20

This word cloud is based on 96 unique words occuring each at least 100 times, all in all occuring 16817 times which comprises 20 % of non-stop words.

Reproduciblity assessment

```
evaldata_file <- "Paper_Evaluation.csv"
```

The following plots are based on the file Paper_Evaluation.csv, the result from the manual reproducibility assessment.

```
category_levels <- c("0", "1", "2", "3")
paper_evaluation_raw <- read_csv(evaldata_file,
    col_types = cols(
      paper = col_skip(),
      title = col_skip(),
      `Notes Reviewer` = col_skip(),
      `computational environment` = col_factor(levels = category_levels),
      `input data` = col_factor(levels = category_levels),
      `method/analysis/processing` = col_factor(levels = category_levels),
      preprocessing = col_factor(levels = category_levels),
      results = col_factor(levels = category_levels),
      X12 = col_skip(),
      X14 = col_skip(),
      `Notes Reviewer` = col_skip(),
      `Author comment` = col_skip()
      ),
    na = "NA")
categoryColumns <- c("input data",
                     "preprocessing",
                     "method/analysis/processing",
                     "computational environment",
                     "results")
```

```
options(knitr.kable.NA = '-')
kable(paper_evaluation_raw %>%
        select(-matches("reviewer")) %>%
        mutate(`short paper` = if_else(`short paper` == TRUE, "X", "")),
      format = "latex", booktabs = TRUE,
      caption = paste0("Reproducibility levels for paper corpus; ",
                       "'-' is category not available")) %>%
  kable_styling(latex_options = "scale_down")
```

Table 3: Reproducibility levels for paper corpus; '-' is category not available

author	short paper	input data	preprocessing	method/analysis/processing	computational environment	results
Zhu et al. (2017)		0	1	1	1	1
Knoth et al. (2017)		0	-	0	1	1
Konkol et al. (2017)		2	2	1	1	1
Haumann et al. (2017)	X	0	1	1	0	1
Brinkhoff (2017)	X	0	-	1	0	0
Almer et al. (2016)		0	-	1	1	1
Wiemann (2016)		2	-	1	1	1
Juhász & Hochmair (2016)		0	1	1	0	0
Josselin et al. (2016)	X	1	-	0	0	1
Rosser et al. (2016)	X	0	-	1	0	0
Kuhn & Ballatore (2015)		-	-	-	-	-
Mazimpaka & Timpf (2015)		2	1	1	1	1
Steuer et al. (2015)		2	0	1	1	1
Fogliaroni & Hobel (2015)	X	-	-	-	-	-
Heinz & Schlieder (2015)	X	0	0	1	1	1
Scheider et al. (2014)		1	1	2	1	1
Gröchening et al. (2014)		2	0	1	0	1
Fan et al. (2014)		0	1	1	0	1
Soleymani et al. (2014)	X	0	0	1	0	0
Wiemann & Bernard (2014)	X	0	0	1	0	0
Osaragi & Tsuda (2013)		0	1	1	0	1
Baglatzi & Kuhn (2013)		-	-	-	-	-
Schwering et al. (2013)	X	0	0	1	-	1
Stein & Schlieder (2013)	X	0	-	1	0	1
Osaragi & Hoshino (2012)		0	0	1	0	1
Magalhães et al. (2012)		0	0	1	0	0
Foerster et al. (2012)		1	-	1	1	1
Merki & Laube (2012)	X	0	-	1	1	1
Kiefer et al. (2012)	X	0	1	1	0	1
Raubal & Winter (2010)		-	-	-	-	-
Schäffer et al. (2010)		0	0	1	1	1
Körner et al. (2010)		-	-	-	-	-

Conceptual papers

```
paper_evaluation <- paper_evaluation_raw %>%
  # add year column
  mutate(year = as.numeric(str_extract(author, "[0-9]+"))) %>%
  # create new attribute for conceptual papers
  mutate(conceptual = is.na(`input data`)
         & is.na(preprocessing)
         & is.na(`method/analysis/processing`)
         & is.na(`computational environment`)
         & is.na(results))

count_conceptual <- nrow(paper_evaluation %>%
                           filter(conceptual))
count_mixed <- nrow(paper_evaluation %>%
                      filter(is.na(`input data`)
                             | is.na(preprocessing)
                             | is.na(`method/analysis/processing`)
                             | is.na(`computational environment`)
                             | is.na(results)))
```

5 papers are purely conceptual (all categories have value `NA`). These are not included in the following statistics.

15 papers are partically conceptual (at least one category has a value of `NA`). These are evaluated.

14 papers are not applicable for preprocessing criterion.

Overall conference contributions

How many conference contributions were made at AGILE conferences over the years?

We need to scrape data from the AGILE website for short papers and posters.

```
base_url <- "https://agile-online.org/index.php/conference/proceedings/proceedings-"
proceedings_urls <- sapply(X = as.character(c(2003:2017)),
                           FUN = function(x) { paste0(base_url, x)},
                           USE.NAMES = TRUE)
proceedings_html <- lapply(X = proceedings_urls, FUN = read_html)

get_paper_links <- function(page){
  links <- page %>%
    html_nodes(css = "a") %>%
    html_attr("href") %>%
    as.list() %>%
    tibble(links = .) %>%
    filter(str_detect(links,
                      pattern = "(ShortPapers|papers|proceedings|papers/Paper_)/[^pP]"))
  return(links)
}

# papers, posters, abstracts of full papers - we don't care as long it is pdf
get_all_links <- function(page){
  all_links <- page %>%
    html_nodes(css = "a") %>%
    html_attr("href") %>%
    as.list()

  pdf_links <- tibble(links = all_links) %>%
    filter(str_detect(links, pattern = "pdf$")) %>%
    # keep only one of poster abstract and poster PDF:
    filter(!str_detect(links, pattern = "Poster_in_PDF.pdf")) %>%
    # some keynotes are also available for Download (at least one in 2012), remove them:
    filter(!str_detect(links, pattern = "(keynotes|Keynote)"))

  return(pdf_links)
}

get_non_full_papers_links <- function(page){
  get_all_links(page) %>%
    # 2017 includes full paper abstracts in the PDFs, remove them:
    filter(!str_detect(links, pattern = "FullPaperAbstract"))
}

proceedings_links_short_and_full_papers <- lapply(X = proceedings_html,
                                                  FUN = get_non_full_papers_links)
```

Get the ISBNs of AGILE proceedings via harvesting AGILE and Springer websites. Then query Springer API for number of chapters in each book to get the full paper count.

```
if(is.na(Sys.getenv("SPRINGER_API_KEY", unset = NA))) {
  # no API key provided, add some dummy data for the document to render
  all_contributions <- NA
```

```
  full_papers <- NA
  paper_counts <- tibble(year = c(NA))
  sample_full_papers <- NA
  sample_short_papers <- NA
} else {
  base_url_lngc <- "https://agile-online.org/index.php/conference/springer-series"
  # 2007 and 2017 are missing on the AGILE website
  lngc_2007 <- "https://link.springer.com/book/10.1007%2F978-3-540-72385-1"
  lngc_2017 <- "https://link.springer.com/book/10.1007/978-3-319-56759-4"

  springer_api_key <- paste0("&api_key=", Sys.getenv("SPRINGER_API_KEY"))
  springer_api_base <- "http://api.springer.com/metadata/json?"

  lngc_html <- read_html(base_url_lngc)

  lngc_books_urls <- lngc_html %>%
    html_nodes(css = "a") %>%
    html_attr("href") %>%
    tibble(links = .) %>%
    filter(str_detect(links, pattern = "/book/")) %>%
    add_row(links = lngc_2007) %>%
    add_row(links = lngc_2017)

  get_full_paper_count <- function(link) {
    # extract id for book
    isbn <- read_html(link) %>%
      html_nodes("span[id=print-isbn], dd[itemprop=isbn]") %>%
      html_text()
    year <- read_html(link) %>%
      html_nodes("span[id=copyright-info], div[class=copyright]") %>%
      html_text() %>%
      gsub("[^0-9]", "", .) %>%
      as.numeric(.)

    url <- str_c(springer_api_base, "q=isbn:", isbn, springer_api_key)

    #cat("Query with isbn ", isbn, " for year ", year, ": ", url, "... ")
    metadata <- fromJSON(url)
    total <- as.numeric(metadata$result$total)
    #cat("Result: ", total, "\n")
    return(tibble(year = year, `full paper` = total))
  }

  lngc_full_paper_counts <- bind_rows(lapply(lngc_books_urls$links, get_full_paper_count))

  counts_any <- sapply(proceedings_links_short_and_full_papers,
                       function(x) { length(x[["links"]]) })
  non_full_paper_counts <- tibble(
    year = as.numeric(names(counts_any)),
    `short paper/poster` = counts_any)

  paper_counts <- full_join(lngc_full_paper_counts, non_full_paper_counts, by = "year") %>%
    arrange(desc(year))
```

```
  all_contributions <-
    sum(paper_counts$"full paper", na.rm = TRUE) +
    sum(paper_counts$"short paper/poster", na.rm = TRUE)
  full_papers <- sum(paper_counts$"full paper", na.rm = TRUE)

  sample_full_papers <- paper_evaluation %>%
    filter(`short paper` == FALSE) %>%
    count() %>%
    .$n
  sample_short_papers <- paper_evaluation %>%
    filter(`short paper` == TRUE) %>%
    count() %>%
    .$n

  kable(paper_counts)
}
```

year	full paper	short paper/poster
2017	20	125
2016	23	65
2015	20	61
2014	22	68
2013	24	57
2012	23	74
2011	27	53
2010	21	66
2009	22	71
2008	23	41
2007	28	75
2006	-	57
2005	-	77
2004	-	96
2003	-	91

Overall **1330 conference contributions** (including posters and short papers), of which **253 are full papers**, in the years 2003 to 2017.

The used **sample** contains 20 full papers (7.91 %) and 12 short papers (percentage respectively full number of short papers not available because not distinguishable from poster abstracts for some years).

Table: Statistics of reproducibility levels per criterion

```
evaldata_numeric <- paper_evaluation %>%
  # must convert factors to numbers to calculate the mean and median
  mutate_if(is.factor, funs(as.integer(as.character(.))))

summary(evaldata_numeric[,categoryColumns])
```

```
##    input data     preprocessing    method/analysis/processing
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.000
##  Median :0.0000   Median :0.5000   Median :1.000
##  Mean   :0.4815   Mean   :0.5556   Mean   :0.963
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :2.0000   Max.   :2.0000   Max.   :2.000
##  NA's   :5        NA's   :14       NA's   :5
##  computational environment    results
##  Min.   :0.0000            Min.   :0.0000
##  1st Qu.:0.0000            1st Qu.:1.0000
##  Median :0.0000            Median :1.0000
##  Mean   :0.4615            Mean   :0.7778
##  3rd Qu.:1.0000            3rd Qu.:1.0000
##  Max.   :1.0000            Max.   :1.0000
##  NA's   :6                 NA's   :5
```

```
# apply summary independently to format as table
summaries <- sapply(evaldata_numeric[,categoryColumns], summary)
exclude_values_summary <- c("1st Qu.", "3rd Qu.")
kable(subset(summaries, !(rownames(summaries) %in% exclude_values_summary)),
      digits = 2,
      col.names = c("input data", "preproc.", "method/analysis/proc.",
                    "comp. env.", "results"),
      caption = paste0("\\label{tab:levels_statistics}Statistics of ",
                       "reproducibility levels per criterion"))
```

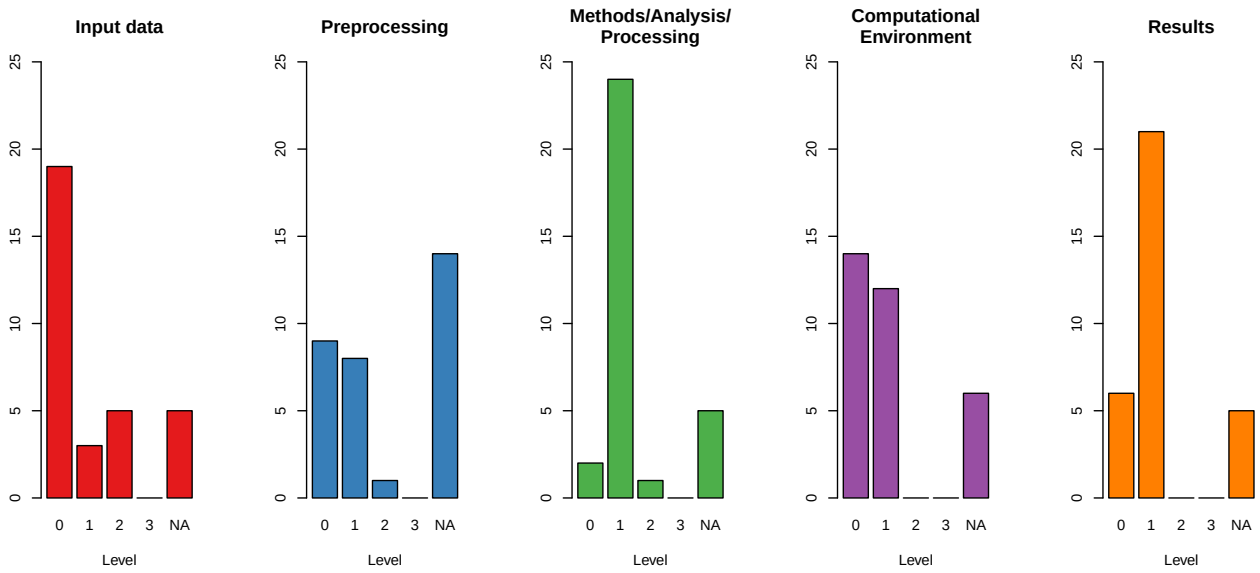
Table 5: Statistics of reproducibility levels per criterion

	input data	preproc.	method/analysis/proc.	comp. env.	results
Min.	0.00	0.00	0.00	0.00	0.00
Median	0.00	0.50	1.00	0.00	1.00
Mean	0.48	0.56	0.96	0.46	0.78
Max.	2.00	2.00	2.00	1.00	1.00
NA's	5.00	14.00	5.00	6.00	5.00

The preprocessing has 18 values, with `0` and `1` around the "middle" resulting in a fraction as the median.

Figure: Results of the evaluation of the corpus of 32 papers

```
# match the colours to time series plot below
colours <- RColorBrewer::brewer.pal(length(categoryColumns), "Set1")
level_names <- c("0", "1", "2", "3", "NA")

criteriaBarplot = function(data, main, colour) {
  barplot(table(data, useNA = "always"),
          main = main,
          xlab = "Level",
          ylim = c(0,25),
          names.arg = level_names,col = colours[colour])
}

par(mfrow = c(1,length(categoryColumns)))
criteriaBarplot(paper_evaluation$`input data`,
                main = "Input data", colour = 1)
criteriaBarplot(paper_evaluation$`preprocessing`,
                main = "Preprocessing", colour = 2)
criteriaBarplot(paper_evaluation$`method/analysis/processing`,
                main = "Methods/Analysis/\nProcessing", colour = 3)
criteriaBarplot(paper_evaluation$`computational environment`,
                main = "Computational\nEnvironment", colour = 4)
criteriaBarplot(paper_evaluation$results,
                main = "Results", colour = 5)
```




```
data_level_zero <- paper_evaluation %>%
  filter(`input data` == 0) %>%
  count() %>% .$n

data_level_two <- paper_evaluation %>%
  filter(`input data` == 2) %>%
  count() %>% .$n

preprocessing_included <- paper_evaluation %>%
  filter(!is.na(preprocessing)) %>%
```

```
  count() %>% .$n

methods_and_results_eq_one <- evaldata_numeric %>%
  filter(`method/analysis/processing` == 1 & results == 1) %>%
  count() %>% .$n
```

19 papers have level 0 and 5 have level 2 in the data criterion.

18 papers include some kind of preprocessing.

18 papers have level 1 in both methods and results criterion.

Table: Mean levels per criterion for full and short papers

```
summaries_short_paper <- sapply(evaldata_numeric %>%
                                  filter(`short paper` == TRUE) %>%
                                  select(categoryColumns), summary)
means_short_paper <- subset(summaries_short_paper, rownames(summaries) %in% c("Mean"))
rownames(means_short_paper) <- c("Short papers")
summaries_full_paper <- sapply(evaldata_numeric %>% filter(`short paper` == FALSE) %>%
                                 select(categoryColumns), summary)
means_full_paper <- subset(summaries_full_paper, rownames(summaries) %in% c("Mean"))
rownames(means_full_paper) <- c("Full papers")
```

```
kable(rbind(means_full_paper, means_short_paper),
      digits = 2,
      col.names = c("input data", "preproc.", "method/analysis/proc.", "comp. env.", "results"),
      caption = paste0("\\label{tab:mean_full_vs_short}",
                       "Mean levels per criterion for full and short papers"))
```

Table 6: Mean levels per criterion for full and short papers

	input data	preproc.	method/analysis/proc.	comp. env.	results
Full papers	0.75	0.67	1.00	0.62	0.88
Short papers	0.09	0.33	0.91	0.20	0.64

Extra table: Mean levels averaged across criteria over time

```
means_years <- evaldata_numeric %>%
  filter(conceptual == FALSE) %>%
  group_by(year) %>%
  summarise(mean = mean(c(`input data`,
                          preprocessing,
                          `method/analysis/processing`,
                          `computational environment`,
                          `results`),
                        na.rm = TRUE),
            `paper count` = n())

means_years_table <- means_years %>%
        mutate(mean = round(mean, 2),
               `paper count` = as.character(`paper count`)) %>%
        mutate(labels = str_c(year, " (n = ", `paper count`, ")")) %>%
        #column_to_rownames("labels") %>%
        select(mean) %>%
        t()
```

```
kable(means_years_table,
      caption = "Summarised mean values over all criteria over time")
```

Table 7: Summarised mean values over all criteria over time

mean	0.6	0.57	0.54	0.6	0.93	0.62	0.74

Figure: Reproducibility levels over time

```
evaldata_years <- evaldata_numeric %>%
  filter(conceptual == FALSE) %>%
  filter(year != 2011) %>%
  group_by(year) %>%
  summarise(input = mean(`input data`, na.rm = TRUE),
         preprocessing = mean(preprocessing, na.rm = TRUE),
            method = mean(`method/analysis/processing`, na.rm = TRUE),
            environment = mean(`computational environment`, na.rm = TRUE),
            results = mean(results, na.rm = TRUE))
paper_count_years <- evaldata_numeric %>%
  filter(conceptual == FALSE) %>%
  filter(year != 2011) %>%
  group_by(year) %>%
  summarise(`paper count` = n())

evaldata_years_long <- melt(evaldata_years, id.vars = c("year"))
ggplot(evaldata_years_long, aes(year, value)) +
  geom_bar(aes(fill = variable), position = "dodge", stat = "identity") +
  ylab("mean value of criterion level") +
  scale_x_continuous(breaks = evaldata_years$year,
                     labels = paste0(paper_count_years$year,
                                     " (n=",
                                     paper_count_years$`paper count`,
                                     ")")) +
  scale_fill_brewer(palette = "Set1", name = "Category") +
  theme_tufte(base_size = 18) +
  theme(legend.position = c(0.15,0.75),
        legend.text = element_text(size = 14)) +
  ylim(0, 3) +
  stat_summary(fun.y = mean, fun.ymin = mean, fun.ymax = mean, shape = "-", size = 2) +
  stat_summary(fun.y = mean, geom = "line", linetype = "dotted", mapping = aes(group = 1))
```

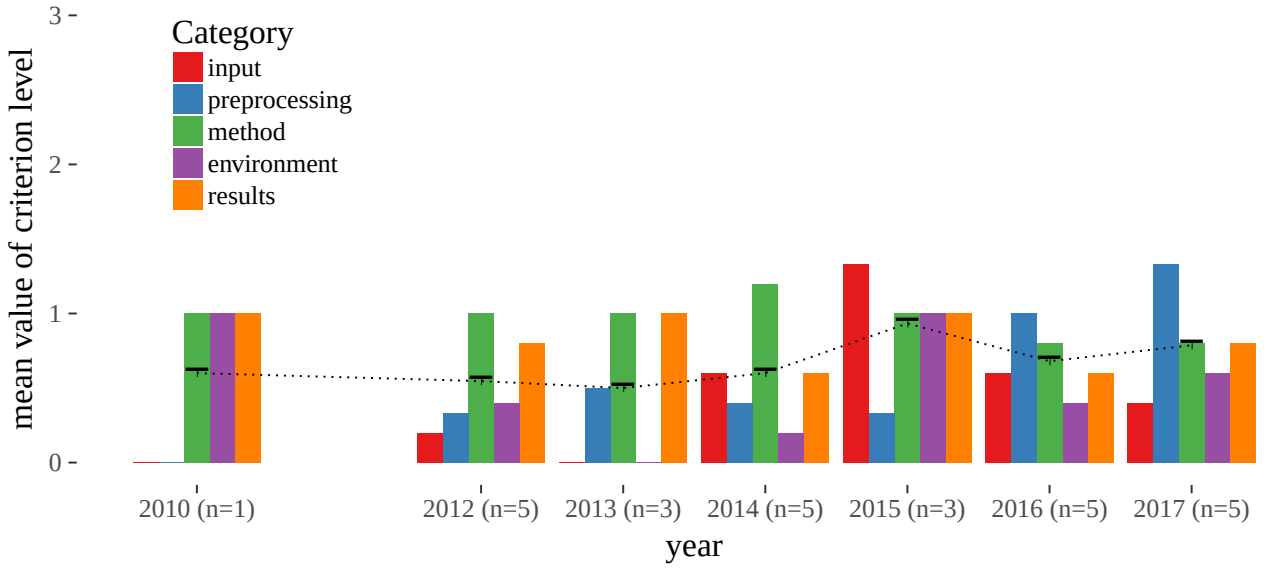
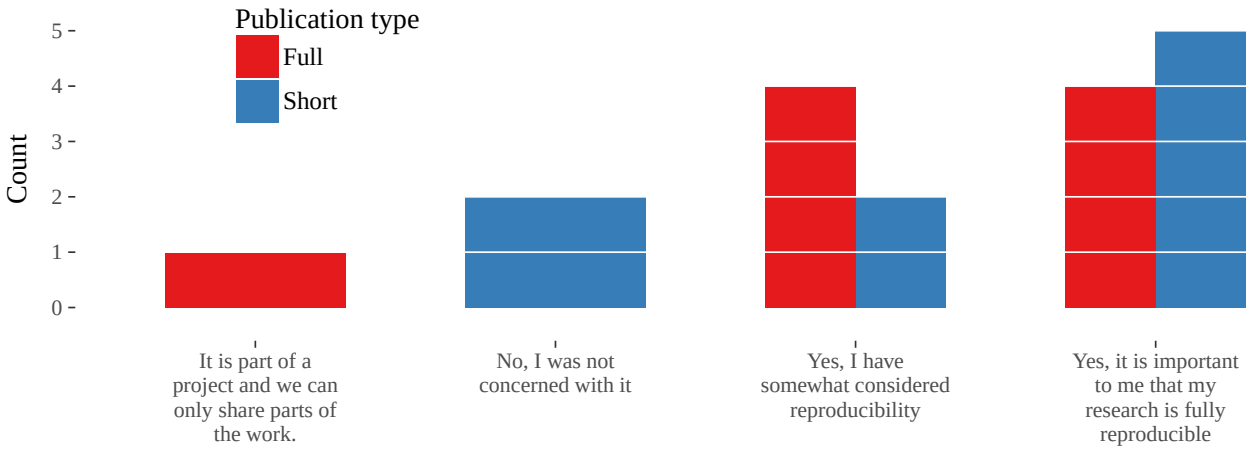


Figure: Author survey results on the importance of reproducibility

```
Reproducibility_Survey <- read_delim(file = "Reproducibility_Survey.csv",
    delim = ";",
    escape_double = FALSE,
    col_types = cols(`Short/Full Paper` = col_factor(levels = c("Full",
        "Short")),
        Timestamp = col_datetime(format = "%m/%d/%Y %H:%M:%S"),
        X15 = col_skip()),
    trim_ws = TRUE) %>%
  rename(`considered reproducibility` =
 `Have you considered the reproducibility of research published in your nominated paper?`)

considered_reproducibility <- Reproducibility_Survey %>%
  group_by(`Short/Full Paper`,
           `considered reproducibility`) %>%
  filter(!is.na(`considered reproducibility`)) %>%
  count()

responses_full <- considered_reproducibility %>%
  filter(`Short/Full Paper` == "Full") %>%
  .$n %>% sum()
responses_short <- considered_reproducibility %>%
  filter(`Short/Full Paper` == "Short") %>%
  .$n %>% sum()

responses_for_papers_count <- length(
  # substract 1 for "The author has not agreed"
  unique(Reproducibility_Survey$`Please select your nominated AGILE Best Paper.`)) - 1

anonymous_responses_count <- Reproducibility_Survey %>%
  filter(is.na(`considered reproducibility`)) %>%
  count()

ggplot(data = Reproducibility_Survey %>%
  filter(!is.na(`considered reproducibility`)),
  aes(x = `considered reproducibility`,
           fill = `Short/Full Paper`)) +
  geom_bar(width = 0.6, position = "dodge") +
  scale_fill_brewer(palette = "Set1", name = "Publication type") +
  scale_x_discrete(label = function(x) str_wrap(x, width = 20),
                   name = paste0("Have you considered the reproducibility of ",
                                 "research published in your nominated paper? (n = ",
                                 sum(considered_reproducibility$n), ")")) +
  scale_y_discrete(name = "Count", limits = c(0:12)) +
  theme_tufte(base_size = 18) +
  theme(legend.position = c(0.2,0.8),
        legend.text = element_text(size = 16),
        legend.key.size = unit(1, "cm")) +
  geom_hline(yintercept = seq(1:10), col = "white", lwd = 0.5)
```



Have you considered the reproducibility of research published in your nominated paper? (n = 18)

Of the 18 responses the plot is based on, 9 are short and 9 full papers.

The 24 responses cover 14 papers and include 6 responses without consent to use the data.

Table: Hindering circumstances for reproducibility for each survey response

```
hindering_circumstances <- Reproducibility_Survey %>%
  select(starts_with('Please rate')) %>%
  drop_na() %>% # remove responses with no answers
  # order the levels of the factors:
  mutate_all(factor, levels = c("Not at all",
                                "Slightly hindered",
                                "Moderately hindered",
                                "Strongly hindered",
                                "Main reason"), ordered = TRUE)

names(hindering_circumstances) <- sapply(names(hindering_circumstances), function(name) {
  if (grepl(".*legal.*", name, ignore.case = TRUE))
    return("Legal restrictions")
  else if (grepl(pattern = ".*time.*", x = name, ignore.case = TRUE))
    return("Lack of time")
  else if (grepl(pattern = ".*tools.*", x = name, ignore.case = TRUE))
    return("Lack of tools")
  else if (grepl(pattern = ".*motivation*", x = name, ignore.case = TRUE))
    return("Lack of incentive")
  else if (grepl(pattern = ".knowledge.*", x = name, ignore.case = TRUE))
    return("Lack of knowledge")
  else return(NA)
})

# count the occurences of "main reason" for each question
hindering_circumstances %>%
  summarise_all(funs(sum(grepl(pattern = "Main reason", x = .))))
```

```
## # A tibble: 1 x 5
##   `Lack of time` `Lack of knowledge` `Lack of tools` `Lack of incentive`
##            <int>               <int>           <int>                <int>
## 1              3                   0               2                    0
## # ... with 1 more variable: `Legal restrictions` <int>
```

```
main_reason_counts <- as.data.frame(t(hindering_circumstances %>%
          summarise_all(
            funs(sum(grepl(pattern = "Main reason", x = .))))) %>%
  rename(count = V1) %>%
  rownames_to_column(var = "circumstance") %>%
  arrange(desc(count))

# sort the columns (circumstances) by the number of "main reason" answers
hindering_circumstances <- hindering_circumstances %>%
  select(main_reason_counts$circumstance) %>%
  # sort the rows by the colum with most "main reason" answers
  arrange(desc(!! rlang::sym(main_reason_counts$circumstance[[1]])))

crcmstncs_ht <- as_hux(hindering_circumstances)
# configure font size and cell padding
font_size(crcmstncs_ht) <- 8

bg_colors <- brewer.pal(n = 5, name = "GnBu")
```

```
crcmstncs_ht <- crcmstncs_ht %>%
  # set background colors for cells
  set_background_color(where(crcmstncs_ht == "Main reason"), bg_colors[[5]]) %>%
  set_background_color(where(crcmstncs_ht == "Strongly hindered"), bg_colors[[4]]) %>%
  set_background_color(where(crcmstncs_ht == "Moderately hindered"), bg_colors[[3]]) %>%
  set_background_color(where(crcmstncs_ht == "Slightly hindered"), bg_colors[[2]]) %>%
  set_background_color(where(crcmstncs_ht == "Not at all"), bg_colors[[1]]) %>%
  add_colnames() %>%
  # format column names:
  set_bold(row = 1, col = 1:length(crcmstncs_ht), TRUE) %>%
  set_bottom_border(row = 1, col = 1:length(crcmstncs_ht), 1) %>%
  set_font_size(row = 1, col = 1:length(crcmstncs_ht), value = 10) %>%
  # add label, caption, and float:
  set_label("tab:hindering_circumstances") %>%
  set_latex_float("ht") %>%
  set_width(1) %>%
  set_caption(paste0(
    "Hindering circumstances for reproducibility for each survey response ",
    #"with columns sorted by the respective count of 'main reason' ",
    #"and rows sorted by the answer categories in descending order"
    "(n = ", nrow(hindering_circumstances),
    "); background colour corresponds to cell text."))

crcmstncs_ht
```

Table 8: Hindering circumstances for reproducibility for each survey response (n = 17); background colour corresponds to cell text.

Legal restrictions	Lack of time	Lack of tools	Lack of knowledge	Lack of incentive
Main reason	Strongly hindered	Not at all	Not at all	Strongly hindered
Main reason	Not at all	Not at all	Not at all	Moderately hindered
Main reason	Slightly hindered	Strongly hindered	Moderately hindered	Strongly hindered
Main reason	Not at all	Slightly hindered	Not at all	Not at all
Strongly hindered	Strongly hindered	Strongly hindered	Moderately hindered	Strongly hindered
Moderately hindered	Main reason	Not at all	Not at all	Not at all
Slightly hindered	Moderately hindered	Slightly hindered	Slightly hindered	Moderately hindered
Slightly hindered	Not at all	Main reason	Strongly hindered	Not at all
Not at all	Moderately hindered	Not at all	Moderately hindered	Not at all
Not at all	Strongly hindered	Strongly hindered	Strongly hindered	Slightly hindered
Not at all	Moderately hindered	Not at all	Not at all	Not at all
Not at all	Slightly hindered	Main reason	Not at all	Strongly hindered
Not at all	Main reason	Not at all	Not at all	Not at all
Not at all	Main reason	Not at all	Not at all	Not at all
Not at all	Moderately hindered	Moderately hindered	Not at all	Strongly hindered
Not at all	Not at all	Not at all	Not at all	Not at all
Not at all	Slightly hindered	Not at all	Slightly hindered	Not at all