Supplementary Material: Retracted articles use less free and open source software and cite it worse

David Schindler and Frank Krüger

Software Usage and Citation in Retracted Articles

This document is created as a literate data analysis where we provide descriptions, implemented code, and the results in one publication. It summarizes all data analyses concerning the scientific software landscape and software citation habits in retracted articles.

```r
library(tidyverse)
library(magrittr)
library(patchwork)
library(effectsize)

theme_set(theme_bw() +
            theme(legend.position = 'top',
                  strip.background = element_rect(fill="#E5E5E5"),
                  plot.margin = unit(c(0,2,0,0), 'mm'),
                  legend.margin=margin(0,0,0,0),
                  legend.box.margin=margin(0,0,-5,0),
                  plot.caption.position = "plot",
                  plot.caption = element_text(hjust = 0),
                  plot.tag.position = "bottomleft")
)
```

Loading Data

We load anonymized information on software in retracted and control articles. This includes identified software and additional information such as version, developer, and software type. The information was aggregated by information extraction from full-text articles. Control articles were selected by Coarsened Exact Matching. The published data which is loaded here is based on data provided by Retraction Watch and published with their permission. The original data is available from Retraction Watch. Below we describe all information sources used to create the data.

```r
df <- read_csv("software_in_retracted_and_control_articles.csv")
```

RW Article Retractions

We obtained the RW database on article retractions as of January 6th, 2022. We use information on retraction, the reason for retraction, journal, and DOI. We perform full-text analyses of these articles and can, therefore, only include articles for which plain full-text is available. This excludes a large number of articles, but the remaining sample size is still sufficient to perform a large-scale analysis, as shown by prior studies investigating article retractions, facing similar issues, and working on comparable sample sizes (Peng, Romero, and Horvát 2022).

S2ORC

The full-text data is sampled from S2ORC (Lo et al. 2020), currently the largest source of available scientific publications in plain text format. The information on software contained in articles is extracted from these full-texts. Further, metadata on scientific domain, year, and journal from S2ORC was utilized.

Summarized Retraction Reasons

There are several retraction reasons considered by Retraction Watch. An overview is available at reasons. However, these are too fine-grained to allow meaningful analyses of the given data set. Therefore, following prior work (Ribeiro and Vasconcelos 2018), we manually summarized reasons into broader categories. The top-level reasons we consider are:

1. Error: Honest errors in investigations that can occur due to multiple reasons.
2. Investigation: Investigations into publications performed by different parties, for instance, by the publisher or an institution.
3. Plagiarism: Cases in which prior work was used without correctly indicating the source
4. SelfPlagiarism: Duplication of one’s own prior work without indicating the source.
5. Misconduct: Cases in which a scientific misconduct was performed by the authors.
6. PaperMill: Articles that were automatically generated with paper mill techniques (and are not supported by actual research).
7. other: Category that matches all unmatched reasons. There are several unspecific reasons that are not related to others, e.g., “rogue editor”.

```r
reasons <- c('Error', 'Investigation', 'Plagiarism', 'SelfPlagiarism',
             'Misconduct', 'PaperMill', 'other')

read_csv("retraction_reasons.csv") %>%
  mutate(TopReason = factor(TopReason, levels = reasons)) -> reasons_print
split(reasons_print$Reason, reasons_print$TopReason)
```

```
$Error
 [1] "Error by Third Party"                "Error in Materials (General)"
 [3] "Unreliable Image"                    "Error by Journal/Publisher"
 [5] "Error in Image"                      "Error in Text"
 [7] "Original Data not Provided"          "Results Not Reproducible"
 [9] "Concerns/Issues About Image"         "Concerns/Issues About Results"
[11] "Unreliable Data"                     "Error in Methods"
[13] "Error in Analyses"                   "Error in Results and/or Conclusions"
[15] "Error in Data"                       "Unreliable Results"
[17] "Concerns/Issues About Data"

$Investigation
[1] "Investigation by ORI"
[2] "Investigation by Third Party"
[3] "Investigation by Company/Institution"
[4] "Investigation by Journal/Publisher"

$Plagiarism
[1] "Plagiarism of Image"       "Plagiarism of Data"
[3] "Plagiarism of Article"     "Plagiarism of Text"
[5] "Euphemisms for Plagiarism"

$SelfPlagiarism
[1] "Duplication of Text"        "Euphemisms for Duplication"
[3] "Duplication of Data"        "Duplication of Image"
[5] "Duplication of Article"

$Misconduct
 [1] "Misconduct by Company/Institution"
 [2] "Euphemisms for Misconduct"
 [3] "Manipulation of Results"
 [4] "Misconduct by Third Party"
 [5] "Falsification/Fabrication of Results"
 [6] "Falsification/Fabrication of Image"
 [7] "Manipulation of Images"
 [8] "Misconduct - Official Investigation/Finding"
 [9] "Falsification/Fabrication of Data"
[10] "Misconduct by Author"

$PaperMill
[1] "Hoax Paper"                 "Randomly Generated Content"
[3] "Paper Mill"

$other
 [1] "Sabotage of Materials"
 [2] "Updated to Correction"
 [3] "Complaints about Company/Institution"
 [4] "Breach of Policy by Third Party"
 [5] "No Further Action"
 [6] "Nonpayment of Fees/Refusal to Pay"
 [7] "Complaints about Third Party"
 [8] "Miscommunication by Company/Institution"
 [9] "Updated to Retraction"
[10] "Not Presented at Conference"
[11] "Miscommunication by Third Party"
[12] "Taken via Peer Review"
[13] "Contamination of Reagents"
[14] "Miscommunication by Journal/Publisher"
[15] "Salami Slicing"
[16] "Civil Proceedings"
[17] "Objections by Company/Institution"
[18] "Ethical Violations by Third Party"
[19] "Publishing Ban"
[20] "Contamination of Materials (General)"
[21] "Criminal Proceedings"
[22] "Error in Cell Lines/Tissues"
[23] "Contamination of Cell Lines/Tissues"
[24] "Bias Issues or Lack of Balance"
[25] "Complaints about Author"
[26] "Legal Reasons/Legal Threats"
[27] "Miscommunication by Author"
[28] "Doing the Right Thing"
[29] "False Affiliation"
[30] "Cites Retracted Work"
[31] "Concerns/Issues about Third Party Involvement"
[32] "Notice - Unable to Access via current resources"
[33] "Informed/Patient Consent - None/Withdrawn"
[34] "Temporary Removal"
```

```
[35] "Lack of Approval from Company/Institution"
[36] "Conflict of Interest"
[37] "Lack of Approval from Third Party"
[38] "Taken from Dissertation/Thesis"
[39] "Notice - Lack of"
[40] "Objections by Author(s)"
[41] "Withdrawn (out of date)"
[42] "Lack of Approval from Author"
[43] "Rogue Editor"
[44] "Lack of IRB/IACUC Approval"
[45] "Copyright Claims"
[46] "Withdrawn to Publish in Different Journal"
[47] "False/Forged Authorship"
[48] "Concerns/Issues about Referencing/Attributions"
[49] "Duplicate Publication through Error by Journal/Publisher"
[50] "Objections by Third Party"
[51] "Ethical Violations by Author"
[52] "Author Unresponsive"
[53] "Concerns/Issues About Authorship"
[54] "Retract and Replace"
[55] "Upgrade/Update of Prior Notice"
[56] "Fake Peer Review"
[57] "Notice - No/Limited Information"
[58] "Date of Retraction/Other Unknown"
[59] "Breach of Policy by Author"
[60] "Withdrawal"
[61] "Notice - Limited or No Information"
```

Control Articles

We select a set of control articles by Coarsened Exact Matching (CEM) (Iacus, King, and Porro 2012). Three article attributes are controlled that have a proven influence on software usage and citation habits (Schindler et al. 2022):

1. **Publication date**: coarsened to *year.* The generally observed trend is that software usage increases over time.
2. **Scientific domain**: matched *exactly.* Specific domains were observed to exhibit higher/lower software usage/citation quality. Domain order for multidisciplinary work is retained: [Computer Science, Biology] is different from [Biology, Computer Science]
3. **Journal Rank:** coarsened to *percentiles.* Higher journal rank has been associated with more formal software citations attributed to more comprehensive journal policies.

Year and domain are determined from Retraction Watch and S2ORC metadata, while the journal rank is based on the Scimago Journal Rank (SJR). Scimago offers publicly available information on journal rank on a yearly basis, which we gathered directly from the Website. The journals information for articles is added by matching Retraction Watch and S2ORC journal information with the Scimago Journal entries.

Software Information Enrichment

Here, we load manually annotated information on software generated during information enrichment:

1. software availability: free and commercial
2. source code availability: open-source and closed-source
3. whether software is statistical software

```r
software_enrichment <- read_csv('software_enrichment.csv', na = 'na') %>%
  drop_na(free)
```

```r
n_free <- nrow(filter(software_enrichment, free==1))
n_free_and_open_source <- nrow(filter(
  software_enrichment, free==1 & source==1))
n_free_and_not_open_source <- nrow(filter(
  software_enrichment, free==1 & source==0))
n_open <- nrow(filter(software_enrichment, source==1))
paste0(
  round(n_free_and_open_source/n_free, digits=2),
  "% of free software are also open source, ",
  round(n_free_and_not_open_source/n_free, digits=2),
  "% are not open source.")
```

```
[1] "0.68% of free software are also open source, 0.32% are not open source."
```

```r
paste0(
  round(n_free_and_open_source/n_open, digits=2),
  "% of open source software are also free.")
```

```
[1] "0.99% of open source software are also free."
```

Data Corrections

We manually correct disambiguation errors that were identified during information enrichment. There are two types of cases: false positive disambiguation, where software names were linked even so they refer to different software, and false negative disambiguation where software names were not linked even so they refer to the same software. Overall, there were 12 false negative cases of software groups and 10 false positive errors of software groups. Groups refer to larger errors where names that appear multiple times are added to other groups that also appear that often. Additionally, there were 14 cases of false negatives where single occurrences were linked to a group, which we consider a small error. All individual errors are corrected here:

```
df %<>%
  mutate(Software_ID = ifelse(Software_Name == 'Image J',
                     43391,
                     Software_ID)) %>%
  # false negative matching - big error
  mutate(Software_Name = ifelse(Software_Name == 'Image J',
                       "ImageJ",
                       Software_Name)) %>%
  mutate(Software_ID = ifelse(grepl("scion", Software_String,
                                    ignore.case = TRUE),
                     43514,
                     Software_ID)) %>%
  # false postive linking - big error
  mutate(Software_Name = ifelse(grepl("scion", Software_String,
                                      ignore.case = TRUE),
                       "Scion Image",
                       Software_Name)) %>%
  mutate(Software_ID = ifelse(grepl("^limma$|^limma ",
                           Software_String, ignore.case = TRUE),
                     43771,
                     Software_ID)) %>%
  # false negative linking - big error
  mutate(Software_Name = ifelse(grepl("^limma$|^limma ",
                             Software_String, ignore.case = TRUE),
                       "limma",
                       Software_Name)) %>%
  mutate(Software_ID = ifelse(grepl("coot", Software_String,
                                    ignore.case = TRUE),
                     43895,
                     Software_ID)) %>%
  # false negative linking - big error
```

```r
    mutate(Software_Name = ifelse(grepl("coot", Software_String,
                                        ignore.case = TRUE),
                         "COOT",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("pasw", Software_String,
                                      ignore.case = TRUE),
                       43381,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("pasw", Software_String,
                                        ignore.case = TRUE),
                         "SPSS",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("fastx", Software_String,
                                      ignore.case = TRUE),
                       46206,
                       Software_ID)) %>%
    # false positive and false negative linking - big error
    mutate(Software_Name = ifelse(grepl("fastx", Software_String,
                                        ignore.case = TRUE),
                         "FASTX - Toolkit",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("tblastn", Software_String,
                                      ignore.case = TRUE),
                       45848,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("tblastn", Software_String,
                                        ignore.case = TRUE),
                         "tblastn",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("macintosh", Software_String,
                                      ignore.case = TRUE),
                       43695,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("macintosh", Software_String,
                                        ignore.case = TRUE),
                         "Mac",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("Significance Analysis of Microarrays",
                             Software_String, ignore.case = TRUE),
                       44860,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("Significance Analysis of Microarrays",
                               Software_String, ignore.case = TRUE),
                         "SAM",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("NetworkX", Software_String,
                                      ignore.case = TRUE),
                       54000,
                       Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl("NetworkX", Software_String,
                                        ignore.case = TRUE),
                         "NetworkX",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("microarray suite", Software_String,
                                      ignore.case = TRUE),
                       45171,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("microarray suite",
                               Software_String, ignore.case = TRUE),
                        "MAS",
                        Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl("Statistical Parametric Mapping",
                             Software_String, ignore.case = TRUE),
                       44213,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(grepl("Statistical Parametric Mapping",
                               Software_String, ignore.case = TRUE),
                         "SPM",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'IPA' |
                                  Software_String == 'IPA TM',
                       44597,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(Software_String == 'IPA' |
                                    Software_String == 'IPA TM',
                         "Ingenuity Pathway Analysis",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == "Ingenuity" |
                         grepl("Ingenuity", Software_String,
                               ignore.case = TRUE) &
                         (grepl("pathway", Software_String,
                                ignore.case = TRUE) |
                            grepl("ipa", Software_String,
                                  ignore.case = TRUE) |
                            grepl("system", Software_String,
                                  ignore.case = TRUE)),
                       44597,
                       Software_ID)) %>%
    # false negative linking - big error
    mutate(Software_Name = ifelse(Software_String == "Ingenuity" |
                           grepl("Ingenuity", Software_String,
                                 ignore.case = TRUE) &
                           (grepl("pathway", Software_String,
                                  ignore.case = TRUE) |
                              grepl("ipa", Software_String,
                                    ignore.case = TRUE) |
                              grepl("system", Software_String,
                                    ignore.case = TRUE)),
                         "Ingenuity Pathway Analysis",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^statistics$', Software_String,
                                      ignore.case = TRUE),
           43381,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^statistics$', Software_String,
                                        ignore.case = TRUE),
           "SPSS",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^gcos$', Software_String,
                                      ignore.case = TRUE),
           55013,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^gcos$', Software_String,
                                        ignore.case = TRUE),
           "GCOS",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^chrome$', Software_String,
                                      ignore.case = TRUE),
           45207,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^chrome$', Software_String,
                                        ignore.case = TRUE),
           "Google Chrome",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^primer ?premier$', Software_String,
                                      ignore.case = TRUE),
           55012,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^primer ?premier$', Software_String,
                                        ignore.case = TRUE),
           "Primer Premier",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^mr ?modeltest$', Software_String,
                                      ignore.case = TRUE),
           43157,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^mr ?modeltest$', Software_String,
                                        ignore.case = TRUE),
           "MrModelTest",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^unix$', Software_String,
                                      ignore.case = TRUE),
           55010,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('^unix$', Software_String,
                                        ignore.case = TRUE),
           "UNIX",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('Java ?Tree ?View', Software_String,
                                      ignore.case = TRUE),
           55008,
           Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(grepl('Java ?Tree ?View', Software_String,
                                        ignore.case = TRUE),
           "Java TreeView",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'MIRA' |
                                  Software_String == 'Mira',
                       55001,
                       Software_ID)) %>%
    # false positive linking - big error
    mutate(Software_Name = ifelse(Software_String == 'MIRA' |
                                    Software_String == 'Mira',
                         "MIRA",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^statistical$', Software_String,
                                      ignore.case = TRUE),
           43381,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('^statistical$', Software_String,
                                        ignore.case = TRUE),
           "SPSS",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('m ?- ?plus', Software_String,
                                      ignore.case = TRUE),
           43920,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('m ?- ?plus', Software_String,
                                        ignore.case = TRUE),
           "Mplus",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('treee?dit', Software_String,
                                      ignore.case = TRUE),
           55014,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('treee?dit', Software_String,
                                        ignore.case = TRUE),
           "TREEEDIT",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('redhat', Software_String,
                                      ignore.case = TRUE),
           19693,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('redhat', Software_String,
                                        ignore.case = TRUE),
           "RedHat",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^direct ?x$', Software_String,
                                      ignore.case = TRUE),
           55011,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('^direct ?x$', Software_String,
                                        ignore.case = TRUE),
           "DirectX",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String =='NET' |
                         Software_String == 'Net' |
                         Software_String == 'Net Framework',
                       55009,
                       Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(Software_String =='NET' |
                           Software_String == 'Net' |
                           Software_String == 'Net Framework',
                         "NET",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'e' |
                         Software_String == 'E - ' |
                         Software_String == 'e1071' |
                         Software_String == 'e1701' |
                         Software_String == 'E5640',
                       55007,
                       Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(Software_String == 'e' |
                           Software_String == 'E - ' |
                           Software_String == 'e1071' |
                           Software_String == 'e1701' |
                           Software_String == 'E5640',
                         "E",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('after effect', Software_String,
                                      ignore.case = TRUE),
           55006,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('after effect', Software_String,
                                        ignore.case = TRUE),
           "After Effects",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(grepl('^avid$', Software_String,
                                      ignore.case = TRUE),
           55005,
           Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(grepl('^avid$', Software_String,
                                        ignore.case = TRUE),
           "CAVID",
           Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'C50',
                       55004,
                       Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(Software_String == 'C50',
                         "C50",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'C2000',
                       55003,
                       Software_ID)) %>%
    # false positive linking - small error
    mutate(Software_Name = ifelse(Software_String == 'C2000',
                         "C2000",
                         Software_Name)) %>%
    mutate(Software_ID = ifelse(Software_String == 'C2000',
                       55003,
                       Software_ID)) %>%
    # false positive linking - small error
```

```
  mutate(Software_Name = ifelse(Software_String == 'C2000',
                       "C2000",
                       Software_Name)) %>%
  mutate(Software_ID = ifelse(Software_String == 'c',
                     43445,
                     Software_ID)) %>%
  # false positive linking - small error
  mutate(Software_Name = ifelse(Software_String == 'c',
                       "C",
                       Software_Name)) %>%
  mutate(Software_ID = ifelse(grepl('^creative suite$', Software_String,
                                    ignore.case = TRUE),
         55002,
         Software_ID)) %>%
  # false positive linking - small error
  mutate(Software_Name = ifelse(grepl('^creative suite$', Software_String,
                                      ignore.case = TRUE),
         "Creative Suite",
         Software_Name))
```

Further, we remove 2 systematic extraction errors that were identified during information enrichment. Both are due to a specialized method being mistaken for software.

```
df %>%
  group_by(Paper_ID) %>%
  filter(any(c('BLOSUM', 'B3LYP') %in% Software_Name)) %>%
  select(Set_ID, Paper_ID, Retraction_Reason, Control_Sample_Origin,
         Year, Scientific_Domain, Journal_Rank_Percentile) ->
  paper_ids_removed

df %<>%
  filter(! Software_Name %in% c('BLOSUM', 'B3LYP'))

unique(paper_ids_removed) %>%
  filter(! Paper_ID %in% df$Paper_ID) -> paper_ids_to_add

df <- bind_rows(paper_ids_to_add, df) %>% ungroup()
```

Last, we define a second dataframe that contains the information for analyses based on retraction reasons. As we have 10 corresponding sample articles for each retracted article we can generate a separate control set for each retraction reason that is equally distributed concerning the controlled variables. We use this dataframe for extended analyses.

```r
df %>%
  select(Paper_ID, Retraction_Reason, Software_ID, Software_Name,
         Version, Developer, Citation, URL) %>%
  mutate(URL=ifelse(is.na(URL), FALSE, TRUE)) %>%
  filter(Retraction_Reason != 'non-retracted') %>%
  rename(OriginalReason=Retraction_Reason) %>%
  distinct() %>%
  mutate(set='retracted') ->
  retracted_papers

df %>%
  select(Paper_ID, Retraction_Reason, Software_ID, Software_Name,
         Version, Developer, Citation, URL, Control_Sample_Origin) %>%
  mutate(URL=ifelse(is.na(URL), FALSE, TRUE)) %>%
  inner_join(retracted_papers, by=c('Control_Sample_Origin'='Paper_ID')) %>%
  select(Paper_ID, Software_ID=Software_ID.x, Software_Name=Software_Name.x,
         Version=Version.x, Developer=Developer.x,Citation=Citation.x,
         URL=URL.x, OriginalReason) %>%
  mutate(set='non-retracted') %>%
  distinct() ->
  non_retracted_papers

df_reason_sampled <- rbind(retracted_papers, non_retracted_papers)
```

Results

Retraction Reasons

First, we are getting an overview of the reasons for article retraction and their frequency in the given data. We only look at the manually summarized, top-level reasons as there are too many different specific reasons for a meaningful analysis.

```r
df %>%
  filter(Set_ID=='retracted') %>%
  dplyr::select(Paper_ID, Retraction_Reason) %>%
  distinct() %>%
  group_by(Retraction_Reason) %>%
  count() %>%
  ungroup() %>%
  mutate(Retraction_Reason=reorder(Retraction_Reason, n)) %>%
  ggplot(aes(Retraction_Reason, n)) +
  geom_bar(stat='identity', fill='lightblue')  +
  geom_text(aes(label=n)) +
  labs(x='Reason for Retraction', y='Number of Articles',
       caption = 'Fig. S1: Number of articles corresponding to each retraction
    reasons. Articles can be retracted due to more than one
    reason.') +
  scale_fill_brewer(type='qual', palette = 6) +
  coord_flip() +
  theme(plot.caption = element_text(size=8))
```

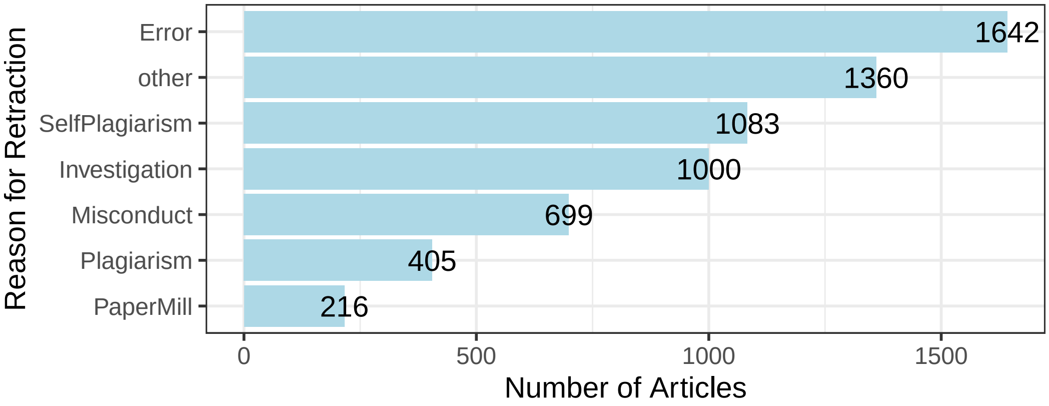


Fig. S1: Number of articles corresponding to each retraction reasons. Articles can be retracted due to more than one reason.

Software Usage in Retracted Articles

Now, we perform the analyses on the software landscape and citation styles.

Papers that Mention Software

We start with a basic analyses by looking at the relative number of articles that contain software.

Overall

We directly compare the relative numbers between sets.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID) %>%
  group_by(Set_ID, Paper_ID) %>%
  summarize(has_software=ifelse(is.na(Software_ID), 0, 1),
            .groups = "drop") %>%
  distinct() %>%
  group_by(Set_ID, has_software) %>%
  summarize(n=n()) %>%
  mutate(rel = n/sum(n)) %>%
  group_by(Set_ID) %>%
  mutate(num=n, n=sum(n)) %>%
  ungroup() %>%
  filter(has_software==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel - MoE) %>%
  select(Set_ID, rel, CIl, CIu) %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  mutate(across(where(is.numeric), round, 1))
```

```
# A tibble: 2 x 4
  Set_ID          rel   CIl   CIu
  <chr>         <dbl> <dbl> <dbl>
1 non-retracted  58.1  57.6  58.6
2 retracted      63.2  61.5  64.8
```

We further include a McNemar test for the paired, dichotomous data to test if there is a difference in the amount of articles mentioning software between retracted and control articles. The effect size is then calculated by an odds ratio between both groups.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Control_Sample_Origin) %>%
  group_by(Set_ID, Paper_ID, Control_Sample_Origin) %>%
  summarize(has_software=ifelse(is.na(Software_ID), 0, 1),
            .groups = "drop") %>%
  distinct() -> df_t

x <- split(df_t, df_t$Set_ID)

x$retracted %>%
  select(-Control_Sample_Origin) %>%
  inner_join(x$`non-retracted`, by=c("Paper_ID"="Control_Sample_Origin")) ->
  df_tmp

df_tmp %>% group_by(has_software.x, has_software.y) %>% summarize(n = n()) ->
  data

p <- matrix(
  rev(data$n),
  nrow=2,
  dimnames = list(
    "control" = c("software", "no-software"),
    "retracted" = c("software", "no-software")))

mcnemar.test(p)
```

```
    McNemar's Chi-squared test with continuity correction

data:  p
McNemar's chi-squared = 200.21, df = 1, p-value < 2.2e-16
```

```r
odds_ratio <- data$n[3] / data$n[2]
odds_ratio
```

```
[1] 1.274303
```

Over Time

We compare the numbers from the first to the last analyzed year per set.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Year) %>%
  filter(Year %in% c(2000, 2019)) %>%
  group_by(Set_ID, Paper_ID, Year) %>%
  summarize(has_software=ifelse(is.na(Software_ID), 0, 1),
            .groups = "drop") %>%
  distinct() %>%
  group_by(Set_ID, has_software, Year) %>%
  summarize(n_has_software=n(),.groups='drop') %>%
  group_by(Set_ID, Year) %>%
  mutate(n_year=sum(n_has_software)) %>%
  mutate(rel = n_has_software/n_year) %>%
  ungroup() %>%
  filter(has_software==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n_year),
         MoE = sqrt((rel * (1-rel))/n_year) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  select(Set_ID, Year, rel, CIl, CIu)  %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  mutate(across(where(is.numeric), round, 1))
```

```
# A tibble: 4 x 5
  Set_ID         Year   rel   CIl   CIu
  <chr>         <dbl> <dbl> <dbl> <dbl>
1 non-retracted  2000  35    27.6  42.4
2 non-retracted  2019  63.3  61.6  65
3 retracted      2000  18.8  -0.4  37.9
4 retracted      2019  76.7  72    81.3
```

Here, we depict the course detailed over all years in the analyses.

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Year) %>%
  group_by(Set_ID, Paper_ID, Year) %>%
  summarize(has_software=ifelse(is.na(Software_ID), 0, 1),
            .groups = "drop") %>%
  distinct() %>%
  group_by(Set_ID, has_software, Year) %>%
  summarize(n_has_software=n(),.groups='drop') %>%
  group_by(Set_ID, Year) %>%
  mutate(n_year=sum(n_has_software)) %>%
  mutate(rel = n_has_software/n_year) %>%
  ungroup() %>%
  filter(has_software==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n_year),
         MoE = sqrt((rel * (1-rel))/n_year) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  ggplot(aes(Year, rel)) +
    geom_line(aes(color=Set_ID)) +
    geom_ribbon(aes(ymin=CIl, ymax=CIu, fill=Set_ID), alpha=.2) +
    labs(x='Year', y='Relative Amount of Articles',
      caption = 'Fig. S2: Relative number of articles containing at least
    one software over time for retracted and control articles.
    95% CIs are indicated by lighter colored areas.') +
    theme(plot.caption = element_text(size=8)) +
    scale_color_manual('Type of Article',
                       values = c("#2b83ba", "#ff8585")) +
    scale_fill_manual('Type of Article',
                      values = c("#2b83ba", "#ff8585"))
```

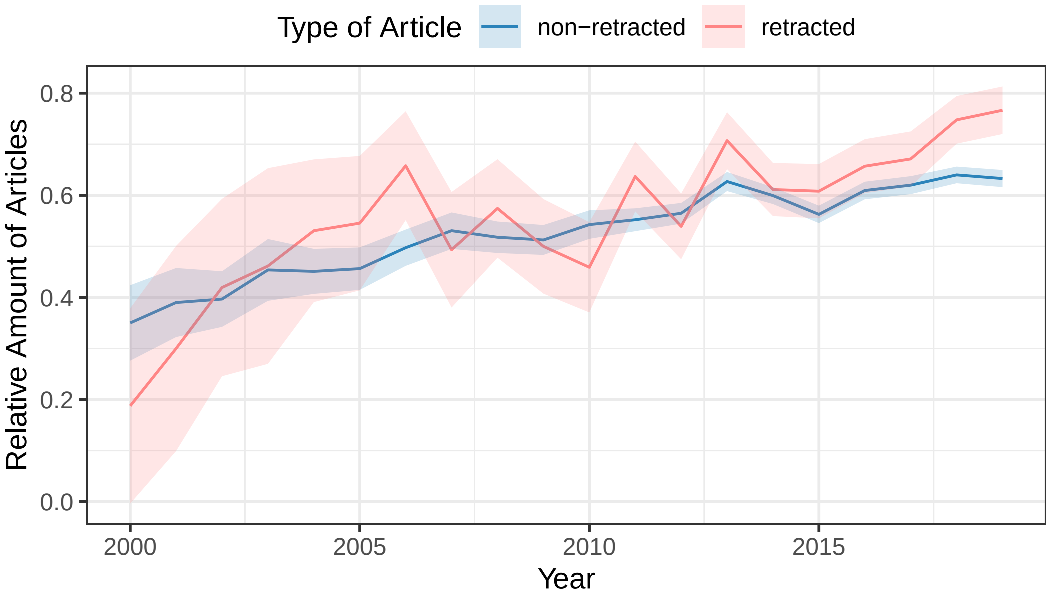


Fig. S2: Relative number of articles containing at least
one software over time for retracted and control articles.
95% CIs are indicated by lighter colored areas.

Confidence intervals are especially large for retracted articles because the overall number of samples decreases due to the year-based split (especially for earlier years, where the fewest samples are available).

Per Retraction Reason

Further, we also look at the relative number concerning specific retraction reasons. Each retraction reason has its own control set, which is created by using the 10 control samples for each article per retraction reason.

```
df_reason_sampled %>%
  group_by(Paper_ID, set, OriginalReason) %>%
  summarize(has_software=ifelse(is.na(Software_ID), 0, 1),
            .groups = "drop") %>%
  distinct() %>%
  group_by(OriginalReason, set, has_software) %>%
  summarize(n=n()) %>%
  mutate(rel = n/sum(n)) %>%
  group_by(OriginalReason) %>%
  mutate(n=sum(n)) %>%
  ungroup() %>%
  filter(has_software==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE)  %>%
  mutate(rel = rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(set=ifelse(set=="non-retracted", "Control", 'Retracted')) %>%
  select(OriginalReason, set, rel, CIu, CIl) %>% print(., n=16) %>%
  mutate(plot="Amount of Articles with Software") %>%
  mutate(OriginalReason = factor(OriginalReason, levels=reasons)) %>%
  ggplot(aes(OriginalReason, rel)) +
  geom_point(aes(color=set), position=position_dodge(width=.6)) +
  geom_errorbar(aes(ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width = .6), width=.5) +
  labs(x=element_blank(), y = "Relative Amount of Articles")  +
  scale_y_continuous(breaks = c(40, 60, 80, 100),
                     labels = c("40%", "60%", "80%", "100%")) +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  facet_wrap(~ plot, nrow=2, scales='free_y') -> p1
```

```
# A tibble: 14 x 5
   OriginalReason set         rel   CIu   CIl
   <chr>          <chr>     <dbl> <dbl> <dbl>
 1 Error          Control    62.3  63.0  61.6
 2 Error          Retracted  70.6  71.2  69.9
 3 Investigation  Control    60.5  61.4  59.6
 4 Investigation  Retracted  70.4  71.3  69.5
 5 Misconduct     Control    59.8  60.9  58.7
 6 Misconduct     Retracted  62.7  63.7  61.6
 7 PaperMill      Control    67.0  68.9  65.1
 8 PaperMill      Retracted  99.1  99.5  98.7
 9 Plagiarism     Control    49.8  51.2  48.3
10 Plagiarism     Retracted  41.7  43.2  40.3
```

```
11 SelfPlagiarism Control    62.6  63.5  61.8
12 SelfPlagiarism Retracted  72.7  73.5  71.9
13 other          Control    57.2  58.0  56.4
14 other          Retracted  61.3  62.0  60.5
```

```
p1 +
  labs(caption = 'Fig. S3: Relative amount of articles containing at least one
    software compared between retracted and control set
    divided by retraction reasons. A separate control set is
    constructed for each retraction reasons by selecting the
    ten corresponding articles for each retracted paper.') +
  theme(plot.caption = element_text(size=10))
```

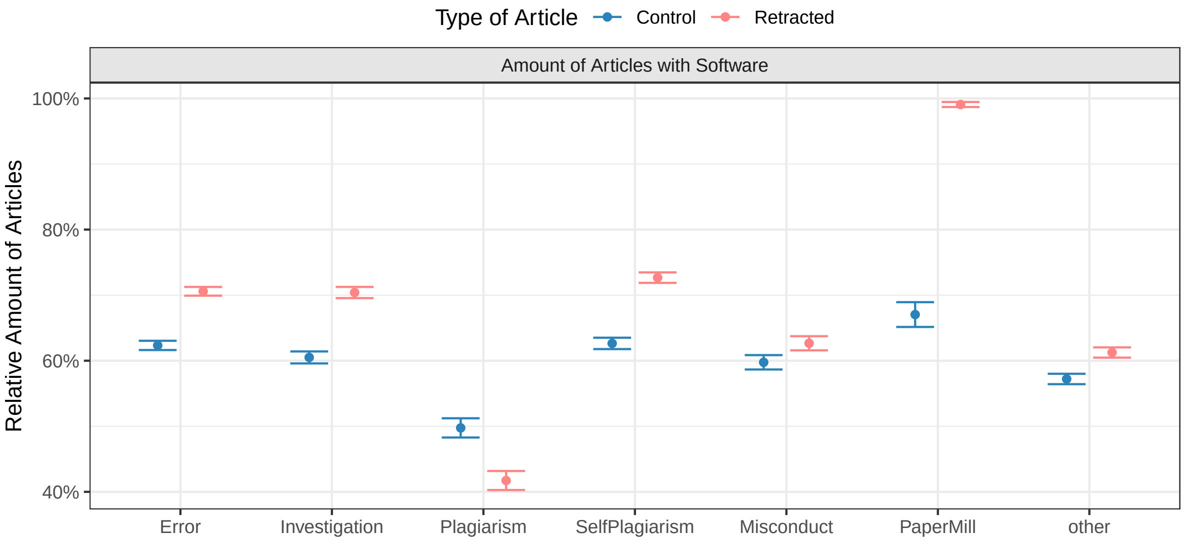


Fig. S3: Relative amount of articles containing at least one software compared between retracted and control set divided by retraction reasons. A separate control set is constructed for each retraction reasons by selecting the ten corresponding articles for each retracted paper.

```
p1 <- p1 + theme(legend.position='top',
        axis.text.x=element_blank())
```

Number of Different Software

Next, we look at the average number of different software that is mentioned within articles that contain software.

Overall

First, the basic compare between sets.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID) %>%
  group_by(Set_ID, Paper_ID) %>%
  summarize(n=ifelse(is.na(Software_ID), 0, n_distinct(Software_ID))) %>%
  filter(n > 0) %>%
  ungroup() %>%
  distinct() %>%
  group_by(Set_ID) %>%
  summarize(m=mean(n), sd=sd(n), num=n(), ) %>%
  mutate(CIl=m-(qt(p=.975, df=num-1)*(sd/sqrt(num))),
         CIu = m+(qt(p=.975, df=num-1)*(sd/sqrt(num))))
```

```
# A tibble: 2 x 6
  Set_ID            m    sd   num   CIl   CIu
  <chr>         <dbl> <dbl> <int> <dbl> <dbl>
1 non-retracted  3.32  3.69 19008  3.27  3.38
2 retracted      2.92  3.05  2067  2.79  3.05
```

We further include a two-sample t-test to test if there is a difference in the number of software provided between retracted and control articles. An unpaired t-test is selected as data is not exactly paired because articles without software are removed for this test and we are considering a quantitative variable with the number of software. The effect size is calculated by using Cohen's d.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Control_Sample_Origin) %>%
  group_by(Set_ID, Paper_ID, Control_Sample_Origin) %>%
  summarize(n=ifelse(
    is.na(Software_ID),
    0,
    n_distinct(Software_ID)),
    .groups = 'drop') %>%
  filter(n > 0) %>%
  distinct() -> df_t

x <- split(df_t, df_t$Set_ID)

t.test(
  x = x$`non-retracted`$n,
  y = x$retracted$n,
  alternative = "two.sided",
  paired = F)
```

```

    Welch Two Sample t-test

data:  x$`non-retracted`$n and x$retracted$n
t = 5.5737, df = 2765.4, p-value = 2.734e-08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2612017 0.5447248
sample estimates:
mean of x mean of y 
 3.322654  2.919690 
```

```r
cohens_d(x$`non-retracted`$n, x$retracted$n)
```

```
Cohen's d |       95% CI
------------------------
0.11      | [0.07, 0.16]

- Estimated using pooled SD.
```

Per Year

Then, as before, a year based comparison.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Year) %>%
  group_by(Set_ID, Paper_ID, Year) %>%
  summarize(n=ifelse(is.na(Software_ID), 0, n_distinct(Software_ID)),
            .groups='drop') %>%
  filter(n > 0) %>%
  distinct() %>%
  group_by(Year, Set_ID) %>%
  summarize(m=mean(n), sd=sd(n), num=n()) %>%
  mutate(CIl=m-(qt(p=.975, df=num-1)*(sd/sqrt(num))),
         CIu = m+(qt(p=.975, df=num-1)*(sd/sqrt(num)))) %>%
  ggplot(aes(Year, m)) +
  geom_line(aes(color=Set_ID)) +
  geom_ribbon(aes(ymin=CIl,ymax=CIu, fill=Set_ID), alpha=.3) +
  labs(x='Year', y='Number of Distinct Software',
    caption = 'Fig. S4: Mean number of distinct software mentioned in articles
    that contain at least one software, depicted over time for
    retracted and control articles. 95% CIs are indicated by lighter
    colored areas.') +
  theme(plot.caption = element_text(size=8)) +
    scale_color_manual('Type of Article',
                      values = c("#2b83ba", "#ff8585")) +
    scale_fill_manual('Type of Article',
                      values = c("#2b83ba", "#ff8585"))
```

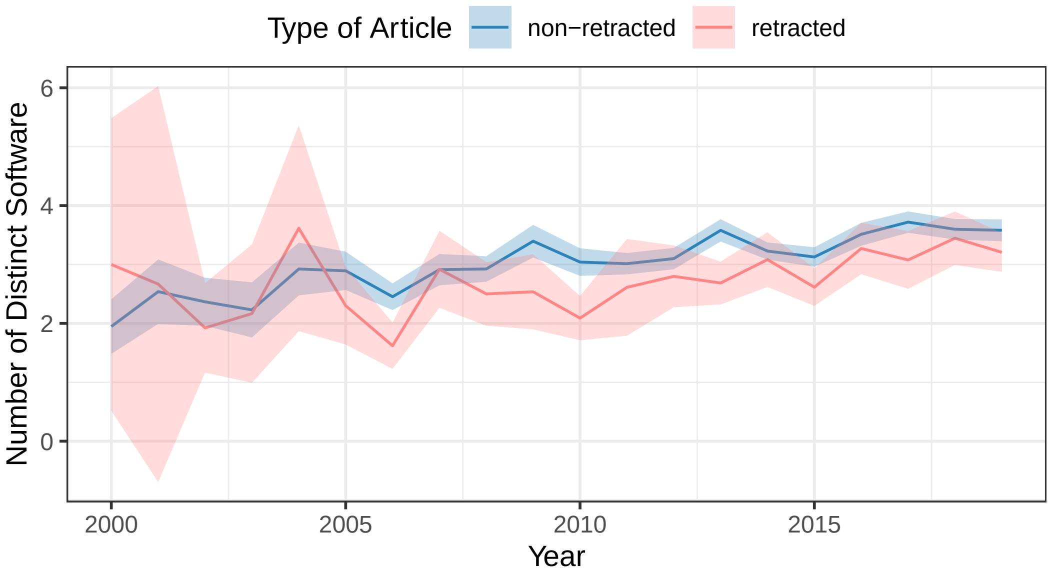


Fig. S4: Mean number of distinct software mentioned in articles that contain at least one software, depicted over time for retracted and control articles. 95% CIs are indicated by lighter colored areas.

Similarly, the CIs are quite large due to the reduced sample size, especially for the retracted set and in early years.

Per Reason

Again, we view the results per retraction reason. Here, we also combine the two generated plots for a better illustration of the results.

```r
df_reason_sampled %>%
  group_by(set, OriginalReason, Paper_ID) %>%
  drop_na() %>%
  summarize(n=ifelse(is.na(Software_ID), 0, n_distinct(Software_ID))) %>%
  filter(n > 0) %>%
  ungroup() %>%
  distinct() %>%
  group_by(set, OriginalReason) %>%
  summarize(m=mean(n), sd=sd(n), num=n(), min=min(n),
            max=max(n), median=median(n)) %>%
  mutate(CIl=m-(qt(p=.975, df=num-1)*(sd/sqrt(num))),
         CIu = m+(qt(p=.975, df=num-1)*(sd/sqrt(num)))) %>%
  mutate(set=ifelse(set=='non-retracted', "Control", "Retracted")) %>%
  select(OriginalReason, set, m, CIu, CIl) %>% print(., n=16) %>%
  mutate(plot="Number of Distinct Software") %>%
  rename(rel=m) %>%
  mutate(OriginalReason = factor(OriginalReason, levels=reasons)) %>%
  ggplot(aes(OriginalReason, rel)) +
  geom_point(aes(color=set), position=position_dodge(width=.6)) +
  geom_errorbar(aes(ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width = .6), width=.5) +
  labs(x=element_blank(), y = "Distinct Software")  +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  ylim(1, 3.75) +
  theme(legend.position='none',
        axis.text.x = element_text(angle=0)) +
  facet_wrap(~ plot, nrow=2, scales='free_y') -> p2
```

```
# A tibble: 14 x 5
# Groups:   set [2]
   OriginalReason set           m   CIu   CIl
   <chr>          <chr>     <dbl> <dbl> <dbl>
 1 Error          Control    3.57  3.65  3.50
 2 Investigation  Control    3.38  3.47  3.29
 3 Misconduct     Control    3.51  3.62  3.40
 4 PaperMill      Control    3.40  3.59  3.22
 5 Plagiarism     Control    2.89  3.05  2.73
 6 SelfPlagiarism Control    3.50  3.59  3.41
 7 other          Control    3.19  3.27  3.11
 8 Error          Retracted  2.95  3.10  2.80
 9 Investigation  Retracted  2.73  2.88  2.58
```

```
10 Misconduct     Retracted  2.68  2.89  2.47
11 PaperMill      Retracted  3.00  3.20  2.81
12 Plagiarism     Retracted  2.60  3.22  1.99
13 SelfPlagiarism Retracted  2.81  2.97  2.66
14 other          Retracted  3.08  3.33  2.83
```

```
p_out <- p1 / p2 + plot_layout(heights = c(2,1))
ggsave('software_amount.jpg', p_out, width=8, height=5)
p_out  +
  labs(caption = 'Fig S5 (Article Fig. 1.): Software mentions in scholarly articles per
    retraction reason separated by retracted and corresponding con-
    trol articles. The sets of control papers are constructed by selecting
    the ten corresponding articles for each retracted article. Top: pro-
    portion of articles that contain at least one software mention. Bottom:
    average number of software mentions per article with at least one soft-
    ware mention. Error bars indicate 95% CIs.') +
  theme(plot.caption = element_text(size=14))
```

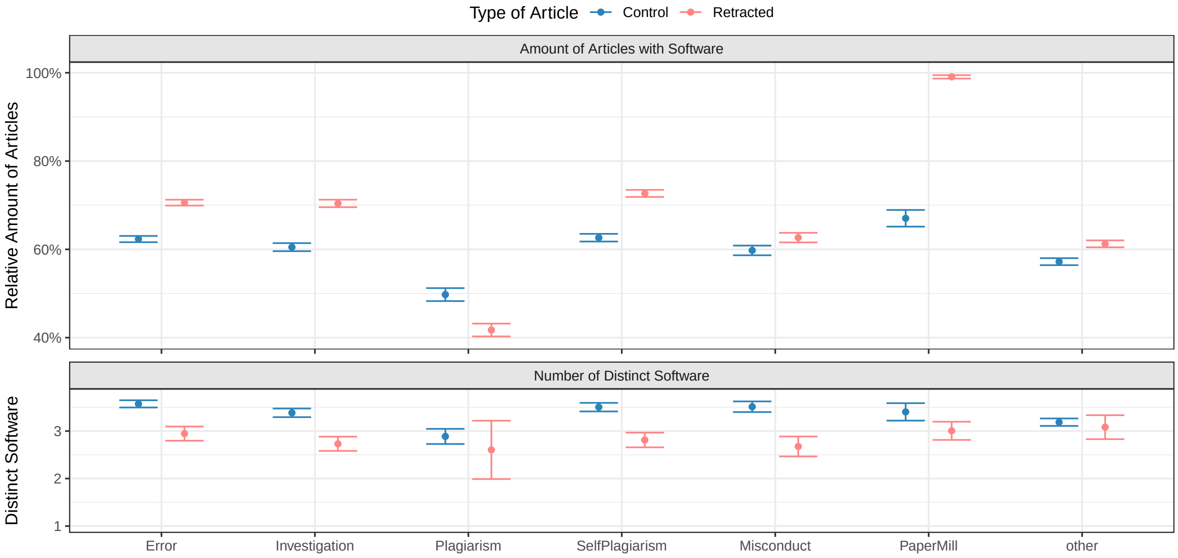


Fig S5 (Article Fig. 1.): Software mentions in scholarly articles per retraction reason separated by retracted and corresponding con−trol articles. The sets of control papers are constructed by selecting the ten corresponding articles for each retracted article. Top: pro−portion of articles that contain at least one software mention. Bottom: average number of software mentions per article with at least one soft−ware mention. Error bars indicate 95% CIs.

Software Based Analysis

So far, we have looked at general differences in software usage between retracted and non-retracted articles. Now, we look at differences in usage of specific software. First, how often specific software is used between sets.

Overall

We analyze in which percentage of articles individual software is used (within all articles that mention software), and how the distributions vary between sets.

```
df %>%
  filter(Software_Type != "OperatingSystem") %>%
  dplyr::select(Set_ID, Paper_ID, Software_Name, Software_ID) %>%
  mutate(Software_Name=str_replace_all(Software_Name, " - ","-")) %>%
  drop_na() %>%
  distinct() %>%
  group_by(Set_ID) %>%
  mutate(n_articles = n_distinct(Paper_ID)) %>%
  group_by(Set_ID, Software_Name, n_articles, Software_ID) %>%
  count() %>%
  ungroup() %>%
  group_by(Set_ID) %>%
  mutate(rel=n/n_articles) %>%
  mutate(SE = sqrt((rel*(1-rel))/n_articles)) %>%
  mutate(CIl = rel - (1.96*SE), CIu = rel + (1.96*SE)) %>%
  ungroup() %>%
  group_by(Software_ID) %>%
  mutate(s = sum(rel)) %>%
  ungroup() %>%
  slice_max(order_by = s, n = 40) %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  mutate(Software_Name=reorder(Software_Name, s)) %>%
  select(Set_ID, Software_Name, n, rel, CIl, CIu) %>% print(., n=40) %>%
  mutate(rel=ifelse(Set_ID=='non-retracted', -rel,rel)) %>%
  mutate(CIl=ifelse(Set_ID=='non-retracted', -CIl,CIl)) %>%
  mutate(CIu=ifelse(Set_ID=='non-retracted', -CIu,CIu)) %>%
  mutate(Set_ID=ifelse(Set_ID=='non-retracted',
                       "Control",
                       "Retracted")) %>%
  ggplot(aes(rel, Software_Name)) +
  geom_col(aes(fill=Set_ID)) +
  geom_errorbar(aes(y=Software_Name, xmin=CIl, xmax=CIu), width=0.8) +
  geom_text(aes(label=paste0(format(abs(rel),digits=1,nsmall=1), "%"),
                x = ifelse(abs(CIl)>5, sign(rel)*2.2, CIu + sign(rel)*1.8)),
            size=3) +
  labs(x='Relative Number of Articles',
       y='Disambiguated Software') +
  scale_fill_manual('Type of Article',
                    values = c("#2b83ba", "#ff8585")) +
  scale_x_continuous(breaks=c(-.2,-.1,0,.1,.2,.3,.4)*100,
                     labels=paste0(c(.2,.1,0,.1,.2,.3,.4)*100, "%")) ->
  p_software
```

```
# A tibble: 40 x 6
   Set_ID        Software_Name      n   rel   CIl   CIu
   <chr>         <fct>          <int> <dbl> <dbl> <dbl>
 1 non-retracted SPSS            3859 20.3  19.7  20.9 
 2 retracted     SPSS             740 35.8  33.8  37.9 
 3 non-retracted Prism           1865  9.82  9.39 10.2 
 4 retracted     Prism            274 13.3  11.8  14.7 
 5 non-retracted ImageJ          1587  8.35  7.96  8.75
 6 retracted     ImageJ           262 12.7  11.3  14.1 
 7 non-retracted R               1713  9.02  8.61  9.43
 8 retracted     R                 96  4.65  3.74  5.56
 9 non-retracted SAS             1170  6.16  5.82  6.50
10 retracted     SAS               69  3.34  2.57  4.12
11 non-retracted TargetScan       195  1.03  0.883 1.17
12 retracted     TargetScan       164  7.94  6.78  9.11
13 non-retracted BLAST           1014  5.34  5.02  5.66
14 retracted     BLAST             72  3.49  2.70  4.28
15 non-retracted Excel            905  4.76  4.46  5.07
16 retracted     Excel             66  3.20  2.44  3.95
17 non-retracted MATLAB           785  4.13  3.85  4.42
18 retracted     MATLAB            54  2.62  1.93  3.30
19 non-retracted Stata            737  3.88  3.61  4.15
20 retracted     Stata             52  2.52  1.84  3.19
21 non-retracted CellQuest        248  1.31  1.14  1.47
22 retracted     CellQuest        105  5.08  4.14  6.03
23 non-retracted Image-Pro Plus   238  1.25  1.09  1.41
24 retracted     Image-Pro Plus    91  4.41  3.52  5.29
25 non-retracted FlowJo           361  1.90  1.71  2.09
26 retracted     FlowJo            60  2.91  2.18  3.63
27 non-retracted Photoshop        387  2.04  1.84  2.24
```

```
28 retracted     Photoshop        45  2.18   1.55   2.81
29 non-retracted Quantity One    225  1.18   1.03   1.34
30 retracted     Quantity One     59  2.86   2.14   3.58
31 non-retracted MEGA            513  2.70   2.47   2.93
32 retracted     MEGA             27  1.31   0.818  1.80
33 non-retracted ClustalW        502  2.64   2.41   2.87
34 retracted     ClustalW         22  1.07   0.623  1.51
35 non-retracted miRanda          73  0.384  0.296  0.472
36 retracted     miRanda          51  2.47   1.80   3.14
37 non-retracted Primer          250  1.32   1.15   1.48
38 retracted     Primer           28  1.36   0.857  1.85
39 non-retracted DAVID           181  0.953  0.815  1.09
40 retracted     DAVID            23  1.11   0.661  1.57
```

```r
ggsave("software_differences.jpg", p_software, width=8, height = 5)
p_software +
  labs(caption = 'Fig S6 (Article Fig. 2.): Proportion of retracted and control
    articles mentioning software out of the top 20 most used
    software. Error bars indicate 95% CIs.') +
  theme(plot.caption = element_text(size=12))
```

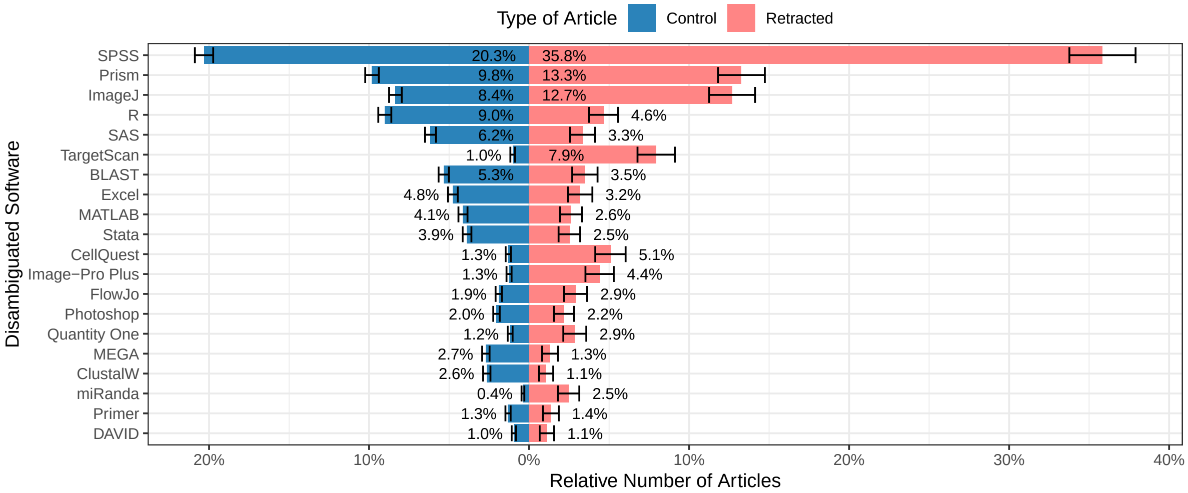


Fig S6 (Article Fig. 2.): Proportion of retracted and control articles mentioning software out of the top 20 most used software. Error bars indicate 95% CIs.

Statistics software

We perform the same analyses limited to the most frequently used statistical software because it is the most common software group.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_Name, Software_ID) %>%
  drop_na() %>%
  distinct() %>%
  group_by(Set_ID) %>%
  mutate(n_articles = n_distinct(Paper_ID)) %>%
  group_by(Set_ID, Software_Name, n_articles, Software_ID) %>%
  count() %>%
  ungroup() %>%
  group_by(Set_ID) %>%
  mutate(rel=n/n_articles) %>%
  mutate(SE = sqrt((rel*(1-rel))/n_articles)) %>%
  mutate(CIl = rel - (1.96*SE), CIu = rel + (1.96*SE)) %>%
  ungroup() %>%
  group_by(Software_ID) %>%
  mutate(s = sum(rel)) %>%
  ungroup() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  filter(type == "Stat") %>%
  slice_max(order_by = s, n = 30) %>%
  mutate(rel=ifelse(Set_ID=='non-retracted', -rel,rel)) %>%
  mutate(CIl=ifelse(Set_ID=='non-retracted', -CIl,CIl)) %>%
  mutate(CIu=ifelse(Set_ID=='non-retracted', -CIu,CIu)) %>%
  mutate(Set_ID=ifelse(Set_ID=='non-retracted',
                       "Control",
                       "Retracted")) %>%
  mutate(Software_Name=reorder(Software_Name, s)) %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  ggplot(aes(rel, Software_Name)) +
  geom_col(aes(fill=Set_ID)) +
  geom_errorbar(aes(y=Software_Name, xmin=CIl, xmax=CIu), width=0.8) +
  geom_text(aes(label=paste0(format(abs(rel),digits=1,nsmall=1), "%"),
                x = ifelse(abs(CIl)>5, sign(rel)*2.2, CIu + sign(rel)*1.8)),
            size=3) +
  labs(x='Relative Number of Articles',
       y='Disambiguated Software',
       caption = 'Fig S7: Proportion of retracted and control
    articles mentioning software out of the top 15 most used
    statistical software. Error bars indicate 95% CIs.') +
  scale_fill_manual('Type of Article',
                    values = c("#2b83ba", "#ff8585")) +
  scale_x_continuous(breaks=c(-.2,-.1,0,.1,.2,.3,.4)*100,
                     labels=paste0(c(.2,.1,0,.1,.2,.3,.4)*100, "%")) +
  theme(plot.caption = element_text(size=12))
```

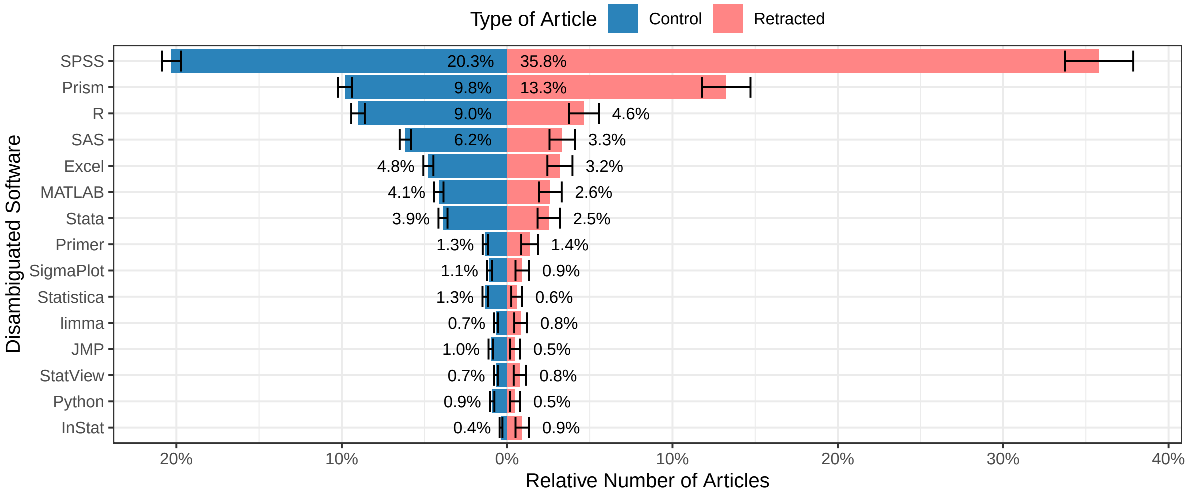


Fig S7: Proportion of retracted and control articles mentioning software out of the top 15 most used statistical software. Error bars indicate 95% CIs.

Per Reason

Next, we look at individual software split per retraction reason.

```
df_reason_sampled %>%
  select(set, Paper_ID, Software_Name, Software_ID, OriginalReason) %>%
  drop_na() %>%
  distinct() %>%
  group_by(set, OriginalReason) %>%
  mutate(n_articles = n_distinct(Paper_ID)) %>%
  group_by(set, OriginalReason, Software_Name, n_articles, Software_ID) %>%
  count() %>%
  group_by(set, OriginalReason) %>%
  mutate(rel=n/n_articles) %>%
  mutate(SE = sqrt((rel*(1-rel))/n_articles)) %>%
  mutate(CIl = rel - (1.96*SE), CIu = rel + (1.96*SE)) %>%
  ungroup() %>%
  group_by(Software_ID, OriginalReason) %>%
```

```r
  mutate(s = sum(rel)) %>%
  ungroup() %>%
  group_by(OriginalReason) %>%
  slice_max(order_by = s, n = 20) %>%
  ungroup() %>%
  mutate(rel=ifelse(set=='non-retracted', -rel,rel)) %>%
  mutate(CIl=ifelse(set=='non-retracted', -CIl,CIl)) %>%
  mutate(CIu=ifelse(set=='non-retracted', -CIu,CIu)) %>%
  mutate(Software_Name=reorder(Software_Name, s)) %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  mutate(OriginalReason = factor(OriginalReason, levels=reasons)) ->
  tmp_df

tmp_df %>%
  filter(Software_Name == 'TargetScan', OriginalReason == 'PaperMill') %>%
  select(set, OriginalReason, Software_Name, rel, CIl, CIu) %>%
  mutate(across(where(is.numeric), round, 1))%>%
  mutate(across(where(is.numeric), abs))
```

```
# A tibble: 2 x 6
  set           OriginalReason Software_Name   rel   CIl   CIu
  <chr>         <fct>          <fct>         <dbl> <dbl> <dbl>
1 non-retracted PaperMill      TargetScan      3.7   2.8   4.7
2 retracted     PaperMill      TargetScan     39.7  33.2  46.3
```

```r
tmp_df %>%
  filter(Software_Name == 'SPSS', OriginalReason == 'PaperMill') %>%
  select(set, OriginalReason, Software_Name, rel, CIl, CIu) %>%
  mutate(across(where(is.numeric), round, 1))%>%
  mutate(across(where(is.numeric), abs))
```

```
# A tibble: 2 x 6
  set           OriginalReason Software_Name   rel   CIl   CIu
  <chr>         <fct>          <fct>         <dbl> <dbl> <dbl>
1 non-retracted PaperMill      SPSS           32.9  30.5  35.4
2 retracted     PaperMill      SPSS           72    65.9  78
```

```r
tmp_df %>%
  ggplot(aes(rel,Software_Name)) +
  geom_col(aes(fill=set)) +
  geom_errorbar(aes(y=Software_Name, xmin=CIl, xmax=CIu)) +
  labs(x='Relative Number of Mentions',
       y='Disambiguated Software',
       caption = 'Fig S8: Proportion of retracted and control articles mentioning
    software out of the top 10 most used software per retraction
    reason. A separate control set is constructed for each retrac-
    tion reasons by selecting the ten corresponding articles for
    each retracted paper.Error bars indicate 95% CIs.') +
  scale_fill_manual('Type of Article',
                    values = c("#2b83ba", "#ff8585")) +
  theme(legend.position = 'top',
        plot.caption = element_text(size=14)) +
  scale_x_continuous(breaks=c(-.2,0,.2,.4,.6,.8)*100,
                     labels=paste0(c(.2,0,.2,.4,.6,.8)*100, "%")) +
  facet_wrap(scales="free_y", ~ OriginalReason)
```

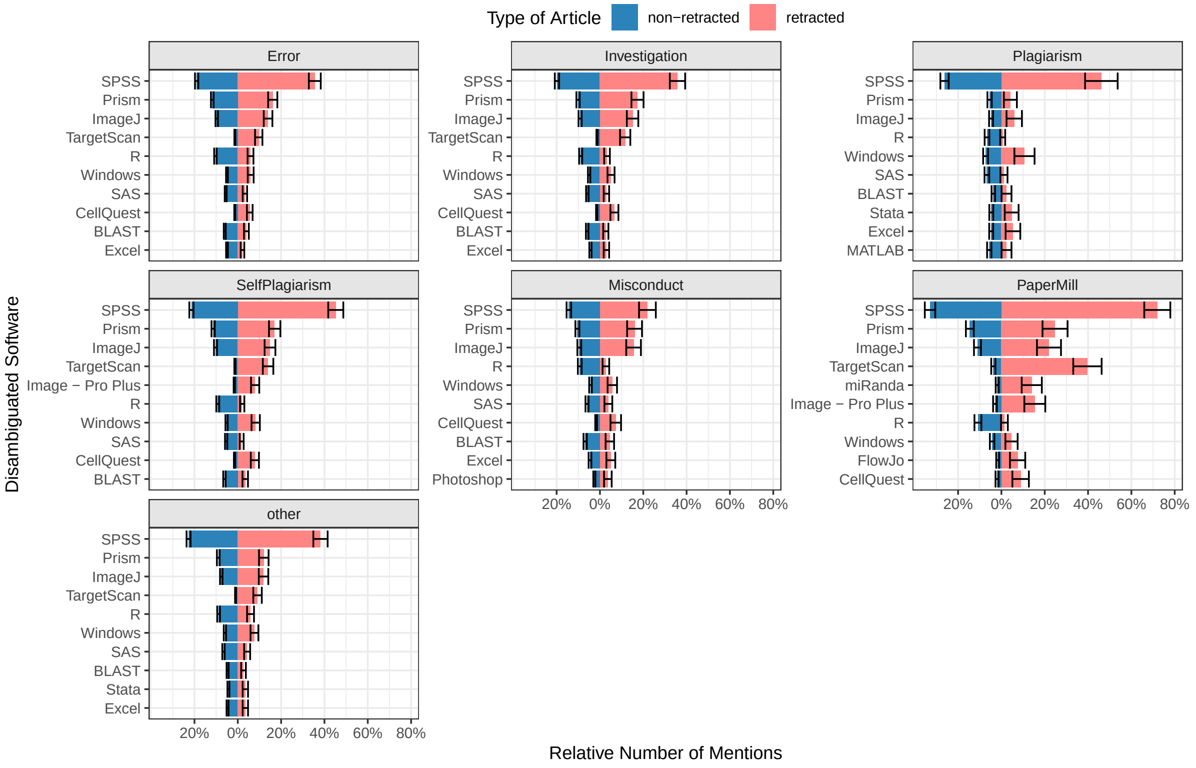


Fig S8: Proportion of retracted and control articles mentioning software out of the top 10 most used software per retraction reason. A separate control set is constructed for each retrac−tion reasons by selecting the ten corresponding articles for each retracted paper.Error bars indicate 95% CIs.

Software Distribution

We look at how software is distributed within articles by analyzing in what proportion of articles any of the top n software appears. This gives us an estimate of how diverse the used software is.

```r
get_nums <- function(df, num) {
  df %>%
    select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
    drop_na(Software_ID) %>%
    filter(Set_ID == 'non-retracted') %>%
    distinct() %>%
    group_by(Software_ID, Software_Name) %>%
    summarize(n=n(), .groups = 'drop') %>%
    arrange(desc(n)) %>%
    slice_head(n = num) -> top_n_control

  df %>%
    select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
    drop_na(Software_ID) %>%
    filter(Set_ID == 'retracted') %>%
    distinct() %>%
    group_by(Software_ID, Software_Name) %>%
    summarize(n=n(), .groups = 'drop') %>%
    arrange(desc(n)) %>%
    slice_head(n = num) -> top_n_retracted

  df %>%
    select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
    drop_na(Software_ID) %>%
    filter(Set_ID == 'non-retracted') %>%
    group_by(Set_ID) %>%
    mutate(o=n_distinct(Paper_ID)) %>%
    distinct() %>%
    filter(Software_ID %in% top_n_control$Software_ID) %>%
    group_by(Set_ID, o) %>%
    summarize(n = n_distinct(Paper_ID), .groups = 'drop_last') %>%
    mutate(rel = n / o) -> res1

  df %>%
    select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
    drop_na(Software_ID) %>%
    filter(Set_ID == 'retracted') %>%
    group_by(Set_ID) %>%
    mutate(o=n_distinct(Paper_ID)) %>%
    distinct() %>%
    filter(Software_ID %in% top_n_retracted$Software_ID) %>%
    group_by(Set_ID, o) %>%
    summarize(n = n_distinct(Paper_ID), .groups = 'drop_last') %>%
    mutate(rel = n / o) -> res2

  rbind(res1, res2) %>%
    mutate(num_id=num)
}

lapply(1:76, function(i){get_nums(df, i)}) %>% bind_rows() -> out_df

out_df %>%
  ggplot(aes(x=num_id, y=rel, group=Set_ID, color=Set_ID)) +
  geom_line() +
  geom_point() +
  labs(x='Top n software',
       y='Amount of Articles',
       caption = 'Fig S9: Relative amount of articles mentioning at least one of
       the top n software out of all articles that mention software.') +
  scale_y_continuous(limits = c(0, 0.90),
                     breaks=c(0, 0.25, 0.50, 0.75)) +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  theme(plot.caption = element_text(size=11))
```

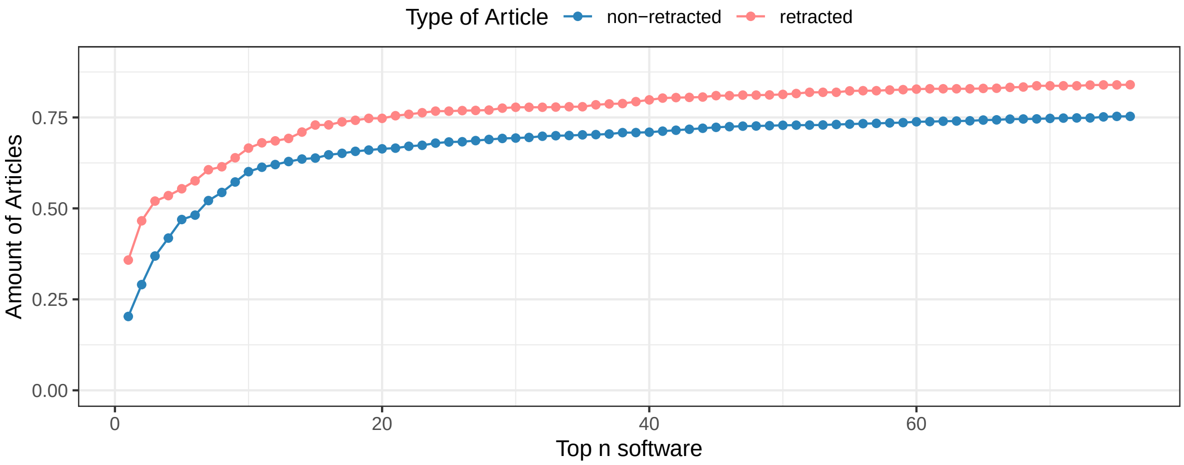


Fig S9: Relative amount of articles mentioning at least one of the top n software out of all articles that mention software.

```
# Getting breakpoints
out_df %>%
  filter(rel > 0.25) %>%
  group_by(Set_ID) %>%
  slice_min(order_by=rel, n=1) %>%
  mutate(value = .25)
```

```
# A tibble: 2 x 6
# Groups:   Set_ID [2]
  Set_ID            o     n   rel num_id value
  <chr>         <int> <int> <dbl>  <int> <dbl>
1 non-retracted 19008  5522 0.291      2  0.25
2 retracted      2067   740 0.358      1  0.25
```

```
out_df %>%
  filter(rel > 0.5) %>%
  group_by(Set_ID) %>%
  slice_min(order_by=rel, n=1) %>%
  mutate(value = .5)
```

```
# A tibble: 2 x 6
# Groups:   Set_ID [2]
  Set_ID            o     n   rel num_id value
  <chr>         <int> <int> <dbl>  <int> <dbl>
1 non-retracted 19008  9911 0.521      7   0.5
2 retracted      2067  1075 0.520      3   0.5
```

```r
out_df %>%
  filter(rel > 0.75) %>%
  group_by(Set_ID) %>%
  slice_min(order_by=rel, n=1) %>%
  mutate(value = .75)
```

```
# A tibble: 2 x 6
# Groups:   Set_ID [2]
  Set_ID            o     n   rel num_id value
  <chr>         <int> <int> <dbl>  <int> <dbl>
1 non-retracted 19008 14284 0.751     74  0.75
2 retracted      2067  1560 0.755     21  0.75
```

Software Names and Spelling Variations

Different spelling variations and names are used to refer to the same software. We analyze if there is a trend towards using the most common software name.

```r
df %>%
  filter(Set_ID == 'retracted') %>%
  select(Set_ID, Software_ID, Software_Name, Software_String) %>%
  drop_na(Software_ID) %>%
  group_by(Set_ID, Software_ID, Software_Name) %>%
  summarize(n = n_distinct(Software_String), .groups = 'drop') %>%
  filter(n > 8) -> software_to_compare

df %>%
  drop_na(Software_ID) %>%
  filter(Software_ID %in% software_to_compare$Software_ID) %>%
  select(Set_ID, Software_ID, Software_Name, Software_String) %>%
  group_by(Set_ID, Software_ID, Software_Name, Software_String) %>%
  summarize(n = n(), .groups = 'drop_last') %>%
  arrange(desc(n)) %>%
  mutate(rel = n/sum(n)) %>%
  mutate(n = sum(n)) %>%
  mutate(SE = sqrt((rel*(1-rel))/n)) %>%
  mutate(CIl = rel - (1.96*SE), CIu = rel + (1.96*SE)) %>%
  mutate(rel = rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  slice_head(n=1) %>%
  ungroup() %>%
  ggplot(aes(x=Software_Name, y=rel, group=Set_ID, color=Set_ID)) +
  geom_errorbar(aes(ymin=CIl, ymax=CIu),
                position=position_dodge(width = .6), width=.5) +
  geom_point(position=position_dodge(width=.6)) +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  labs(x='Disambiguated Software',
       y='Amount of Mentions',
       caption = 'Fig S10: Relative amount of articles mentioning a specific software
    and referring to it by its most commonly used name.') +
  theme(plot.caption = element_text(size=11))
```

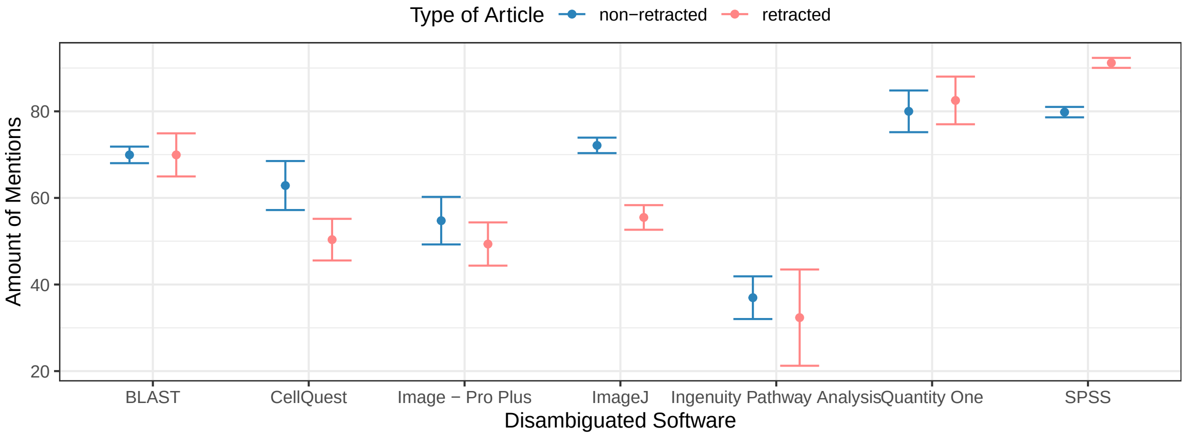


Fig S10: Relative amount of articles mentioning a specific software
  and referring to it by its most commonly used name.

Taking a closer look at the names used for SPSS and ImageJ, which we found to be differently mentioned between retracted and control set.

```
df %>%
  filter(Software_Name %in% c('SPSS', 'ImageJ')) %>%
  select(Set_ID, Paper_ID, Software_Name, Software_String) %>%
  distinct() %>%
  group_by(Set_ID, Software_Name, Software_String) %>%
  summarize(n = n(), .groups = 'drop_last') %>%
  mutate(rel = n / sum(n)) %>%
  mutate(n = sum(n)) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE)  %>%
  mutate(rel = rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  select(Set_ID, Software_Name, Software_String, rel, CIl, CIu) %>%
  slice_max(order_by = rel, n = 5) %>%
  arrange(desc(rel), .by_group=TRUE) %>% print(., n=21)
```

```
# A tibble: 20 x 6
# Groups:   Set_ID, Software_Name [4]
   Set_ID        Software_Name Software_String                  rel     CIl   CIu
   <chr>         <chr>         <chr>                          <dbl>   <dbl> <dbl>
 1 non-retracted ImageJ        ImageJ                        68.8   66.6    71.0
 2 non-retracted ImageJ        Image J                       22.9   20.9    24.9
 3 non-retracted ImageJ        Image                          3.18   2.33    4.02
 4 non-retracted ImageJ        Image - J                      1.26   0.723   1.79
 5 non-retracted ImageJ        IMAGEJ                         1.14   0.629   1.65
 6 non-retracted SPSS          SPSS                          79.4   78.2    80.7
 7 non-retracted SPSS          SPSS Statistics                7.78   6.96    8.61
 8 non-retracted SPSS          Statistical Package for the~   3.36   2.80    3.91
 9 non-retracted SPSS          Statistical Package for Soc~   2.32   1.86    2.79
10 non-retracted SPSS          PASW Statistics                1.16   0.831   1.49
11 retracted     ImageJ        ImageJ                        56.6   50.7    62.5
12 retracted     ImageJ        Image J                       27.2   21.9    32.5
13 retracted     ImageJ        Image                          9.19   5.76   12.6
14 retracted     ImageJ        Image - J                      1.84   0.242   3.43
15 retracted     ImageJ        IMAGE                          1.47   0.0400  2.90
16 retracted     SPSS          SPSS                          90.1   87.9    92.2
17 retracted     SPSS          SPSS Statistics                2.38   1.30    3.47
18 retracted     SPSS          Statistical Package for the~   2.12   1.09    3.15
19 retracted     SPSS          Statistical Package for Soc~   1.59   0.697   2.48
20 retracted     SPSS          PASW Statistics                0.662  0.0837  1.24
```

Free and Open Source Software

We analyze the use of free vs commercial software and the use of open- vs closed-source software between retracted and control set.

Overall

First, we perform an overall compare between sets.

Free

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, free) %>%
  summarize(n=n(), .groups = 'drop_last') %>%
  mutate(rel = n / sum(n)) %>%
  mutate(n = sum(n)) %>%
  ungroup() %>%
  filter(free==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  select(Set_ID, free, n, rel, CIl, CIu)
```

```
# A tibble: 2 x 6
  Set_ID         free     n   rel   CIl   CIu
  <chr>         <dbl> <int> <dbl> <dbl> <dbl>
1 non-retracted     1 31377 0.444 0.439 0.450
2 retracted         1  3646 0.369 0.354 0.385
```

Open Source

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, source) %>%
  summarize(n=n(), .groups = 'drop_last') %>%
  mutate(rel = n / sum(n)) %>%
  ungroup() %>%
  filter(source==1) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  select(Set_ID, source, n, rel, CIl, CIu)
```

```
# A tibble: 2 x 6
  Set_ID        source     n   rel   CIl   CIu
  <chr>          <dbl> <int> <dbl> <dbl> <dbl>
1 non-retracted      1 11767 0.375 0.366 0.384
2 retracted          1  1047 0.287 0.260 0.315
```

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Software_Name) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  mutate(Set_ID=ifelse(Set_ID=='retracted',
                       "Retracted",
                       "Control")) %>%
  mutate(Set_ID=factor(Set_ID, levels= c("Control","Retracted"))) %>%
  mutate(free=ifelse(free, 'Free', 'Commercial')) %>%
  mutate(source=ifelse(source, 'Open Source', 'Closed Source')) %>%
  pivot_longer(c('free', 'source'), names_to="open_source") %>%
  mutate(open_source= ifelse('free'==open_source,
                             "Software availability",
                             "Source availablity")) %>%
  group_by(Set_ID, open_source, value) %>%
  mutate(value=factor(value, levels=c('Free',
                                      'Commercial',
                                      'Open Source',
                                      'Closed Source')))%>%
  summarize(n=n(), .groups = 'drop_last') %>%
  mutate(rel = n / sum(n)) %>%
  ungroup() %>%
  group_by(Set_ID, open_source) %>%
  mutate(n = sum(n)) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  mutate(rel = rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  ggplot(aes(x=value, rel)) +
  geom_point(aes(color=Set_ID),
             position = position_dodge(width = 0.7)) +
  geom_errorbar(aes(ymin=CIl, ymax=CIu, color=Set_ID),
                position = position_dodge(width = 0.7), width=0.6) +
  scale_y_continuous(limits=c(0,75), breaks=c(0,.2,.4,.6)*100,
                         labels=paste0(c(0,.2,.4,.6)*100, "%")) +
  facet_grid(~ open_source, scales='free_x') +
  scale_color_manual("Type of Article",
                         values = c("#2b83ba", "#ff8585")) +
  labs(y="Proportion of Software") +
  theme(axis.title.x = element_blank()) ->
  p_source

ggsave("Open_source_software.jpg", p_source, width=4.3, height = 3)
p_source +
  labs(caption = 'Fig S11 (Article Fig. 3.): Proportion of free or open source
    software across retracted and control articles. Error bars
    indicate 95% CIs.') +
  theme(plot.caption = element_text(size=8))
```

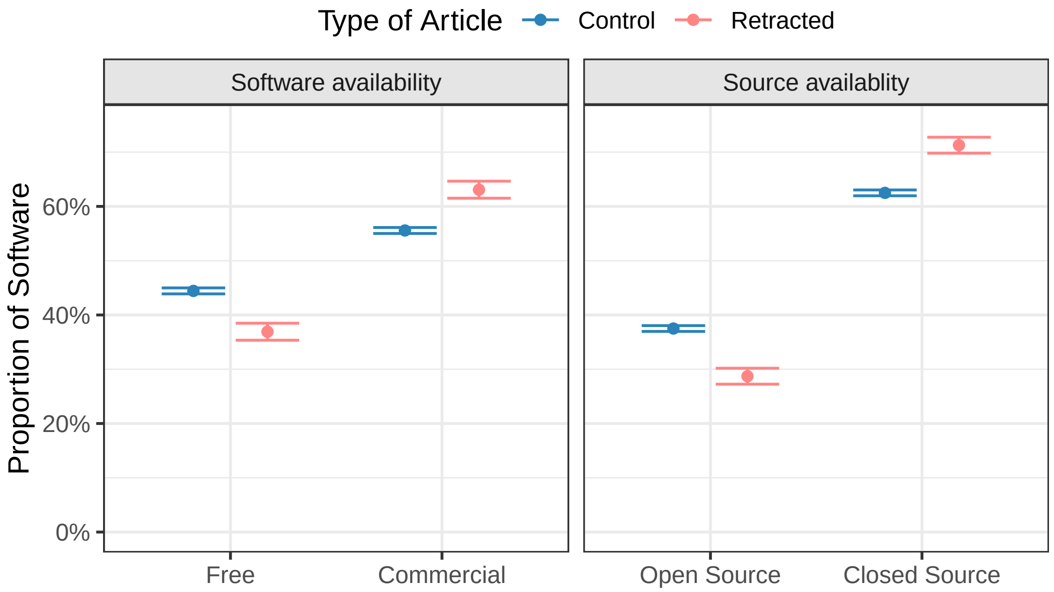


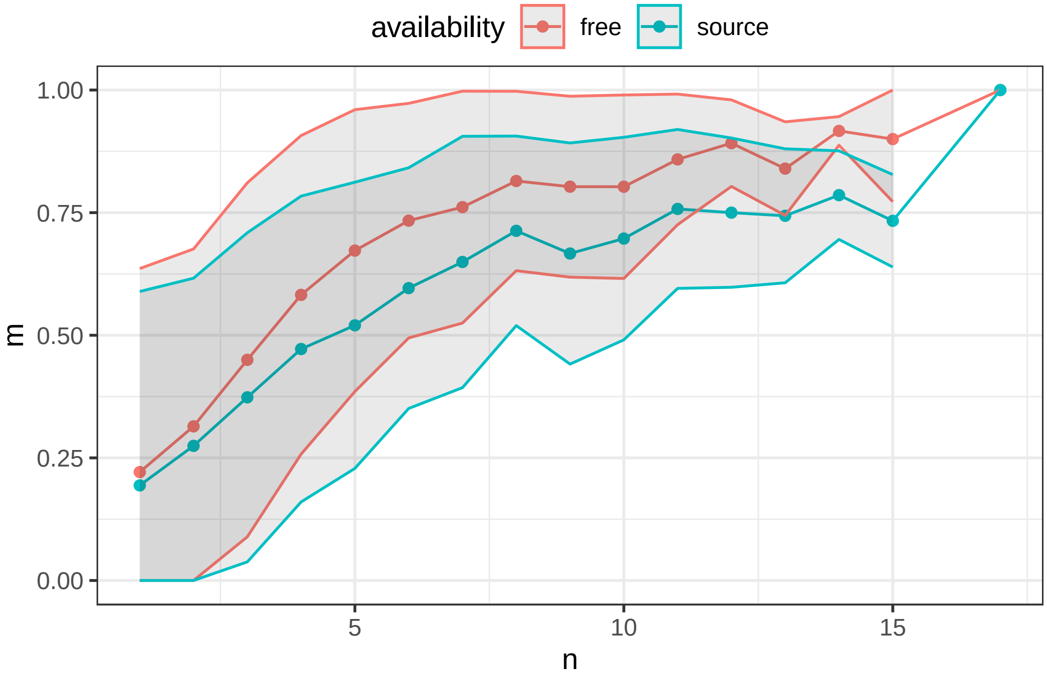
Fig S11 (Article Fig. 3.): Proportion of free or open source software across retracted and control articles. Error bars indicate 95% CIs.

Statistical Test

We further perform a statistical test to investigate if there is a difference in free and open source software usage between retracted and control articles. In this context, we observed that there is a relation between the amount of free (and open source) software and the number of software used within an article, where the ratio of free (and open source) software increases with the number of software per article. Therefore, we include this number of software in an article as a covariate in tests on free (and open source) software.

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID,
                Software_Name, Control_Sample_Origin) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Paper_ID, Control_Sample_Origin) %>%
  summarize(
    n_free = sum(free) / n(),
    n_source = sum(source) / n(),
    n=n(),
    .groups = 'drop_last') -> df_t

df_t %>%
  group_by(n) %>%
  summarize(
    m_free = mean(n_free),
    sd_free=sd(n_free),
    m_source = mean(n_source),
    sd_source=sd(n_source)
    ) %>%
  pivot_longer(-n,
               names_to=c('value_type', 'availability'),
               names_sep='_') %>%
  pivot_wider(names_from = value_type, values_from = value) %>%
  ggplot(aes(n, m, color=availability)) +
  geom_point() +
  geom_line() +
  geom_ribbon(aes(ymin=pmax(0, m-sd), ymax=pmin(1, m+sd)), alpha=.1)
```



We use a GLM and include the retraction state and the number of software in the article as covariates to predict the availability of a software, and further include their interactions for completeness. Effect sizes are then estimated through odds ratios.

```r
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID,
                Software_Name, Control_Sample_Origin) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, Paper_ID, Control_Sample_Origin) %>%
  mutate(n = n()) %>%
  ungroup() ->  df_tt

model <- glm(
  formula=free~Set_ID+n+Set_ID*n,
  data=mutate(df_tt, Set_ID=factor(Set_ID, c('retracted', 'non-retracted'))),
  family = binomial(link="logit"))
summary(model)
```

```
Call:
glm(formula = free ~ Set_ID + n + Set_ID * n, family = binomial(link = "logit"),
    data = mutate(df_tt, Set_ID = factor(Set_ID, c("retracted",
        "non-retracted"))))

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)            -1.76690    0.07589 -23.283  < 2e-16 ***
Set_IDnon-retracted     0.29666    0.07935   3.739 0.000185 ***
n                       0.41371    0.02242  18.452  < 2e-16 ***
Set_IDnon-retracted:n -0.01810    0.02337  -0.774 0.438670
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 47986  on 35022  degrees of freedom
Residual deviance: 42491  on 35019  degrees of freedom
AIC: 42499

Number of Fisher Scoring iterations: 4
```

```r
exp(summary(model)$coefficients["n",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["n",2])
```

```
[1] 1.447397 1.512421 1.580366
```

```r
exp(summary(model)$coefficients["Set_IDnon-retracted",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["Set_IDnon-retracted",2])
```

```
[1] 1.151575 1.345353 1.571739
```

```r
model <- glm(
  formula=source~Set_ID+n+Set_ID*n,
  data=mutate(df_tt, Set_ID=factor(Set_ID, c('retracted', 'non-retracted'))),
  family = binomial(link="logit"))
summary(model)
```

```
Call:
glm(formula = source ~ Set_ID + n + Set_ID * n, family = binomial(link = "logit"), 
    data = mutate(df_tt, Set_ID = factor(Set_ID, c("retracted", 
        "non-retracted"))))

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)           -1.90844    0.07644 -24.967  < 2e-16 ***
Set_IDnon-retracted    0.43722    0.07959   5.494 3.94e-08 ***
n                      0.32513    0.02095  15.520  < 2e-16 ***
Set_IDnon-retracted:n -0.03446    0.02170  -1.588    0.112    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 46001  on 35022  degrees of freedom
Residual deviance: 42383  on 35019  degrees of freedom
AIC: 42391

Number of Fisher Scoring iterations: 4
```

```r
exp(summary(model)$coefficients["n",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["n",2])
```

```
[1] 1.328522 1.384206 1.442223
```

```r
exp(summary(model)$coefficients["Set_IDnon-retracted",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["Set_IDnon-retracted",2])
```

```
[1] 1.324761 1.548394 1.809778
```

Per Reason

Then, we extend the analyses concerning individual retraction reasons.

```r
df_reason_sampled %>%
  select(set, Paper_ID, Software_ID, Software_Name, OriginalReason) %>%
  distinct() %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  mutate(Set_ID=ifelse(set=='retracted',
                       "Retracted",
                       "Control")) %>%
  mutate(Set_ID=factor(set, levels= c("Control","Retracted"))) %>%
  mutate(free=ifelse(free, 'Free', 'Commercial')) %>%
  mutate(source=ifelse(source,
                       'Open Source',
                       'Closed Source')) %>%
  pivot_longer(c('free', 'source'), names_to="open_source") %>%
  mutate(open_source= ifelse('free'==open_source,
                             "Software availability",
                             "Source availablity")) %>%
  group_by(set, OriginalReason, open_source, value) %>%
  mutate(value=factor(value, levels=c('Free',
                                      'Commercial',
                                      'Open Source',
                                      'Closed Source')))%>%
  summarize(n=n(), .groups = 'drop_last') %>%
  mutate(rel = n / sum(n)) %>%
  mutate(n = sum(n)) %>%
  ungroup() %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  mutate(rel = rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  select(set, OriginalReason, open_source, value, n, rel, CIl, CIu) %>%
  print(., n=64) %>%
  ggplot(aes(x=value, rel)) +
  geom_point(aes(color=set),
             position = position_dodge(width = 0.7)) +
  geom_errorbar(aes(ymin=CIl, ymax=CIu, color=set),
                position = position_dodge(width = 0.7), width=0.6) +
  facet_grid(OriginalReason ~ open_source , scales='free_x') +
  scale_color_manual("Type of Article",
                     values=c("#2b83ba", "#ff8585")) +
  labs(y="Proportion of Software",
       caption='Fig S12: Proportion of free or open source software across
    retracted and control articles per retraction reason. A sep-
```

```
   arate control set is constructed for each retraction reasons
   by selecting the ten corresponding articles for each retrac-
   ted paper. Error bars indicate 95% CIs.') +
  theme(axis.title.x = element_blank(),
     plot.caption = element_text(size=14))
```

```
# A tibble: 56 x 8
   set           OriginalReason open_source        value     n   rel   CIl   CIu
   <chr>         <chr>          <chr>              <fct> <int> <dbl> <dbl> <dbl>
 1 non-retracted Error          Software availabi~ Free  18331  47.2  46.5  47.9
 2 non-retracted Error          Software availabi~ Comm~ 18331  52.8  52.1  53.5
 3 non-retracted Error          Source availablity Open~ 18331  40.3  39.6  41.0
 4 non-retracted Error          Source availablity Clos~ 18331  59.7  59.0  60.4
 5 non-retracted Investigation  Software availabi~ Free  10489  46.9  45.9  47.9
 6 non-retracted Investigation  Software availabi~ Comm~ 10489  53.1  52.1  54.1
 7 non-retracted Investigation  Source availablity Open~ 10489  38.5  37.6  39.4
 8 non-retracted Investigation  Source availablity Clos~ 10489  61.5  60.6  62.4
 9 non-retracted Misconduct     Software availabi~ Free   7332  49.2  48.1  50.4
10 non-retracted Misconduct     Software availabi~ Comm~  7332  50.8  49.6  51.9
11 non-retracted Misconduct     Source availablity Open~  7332  39.4  38.2  40.5
12 non-retracted Misconduct     Source availablity Clos~  7332  60.6  59.5  61.8
13 non-retracted PaperMill      Software availabi~ Free   2795  46.9  45.1  48.8
14 non-retracted PaperMill      Software availabi~ Comm~  2795  53.1  51.2  54.9
15 non-retracted PaperMill      Source availablity Open~  2795  41.0  39.2  42.9
16 non-retracted PaperMill      Source availablity Clos~  2795  59.0  57.1  60.8
17 non-retracted Plagiarism     Software availabi~ Free   2691  34.9  33.1  36.7
18 non-retracted Plagiarism     Software availabi~ Comm~  2691  65.1  63.3  66.9
19 non-retracted Plagiarism     Source availablity Open~  2691  30.2  28.5  32.0
20 non-retracted Plagiarism     Source availablity Clos~  2691  69.8  68.0  71.5
21 non-retracted SelfPlagiarism Software availabi~ Free  12096  46.5  45.6  47.4
22 non-retracted SelfPlagiarism Software availabi~ Comm~ 12096  53.5  52.6  54.4
23 non-retracted SelfPlagiarism Source availablity Open~ 12096  40.1  39.2  41.0
24 non-retracted SelfPlagiarism Source availablity Clos~ 12096  59.9  59.0  60.8
25 non-retracted other          Software availabi~ Free  12181  41.6  40.7  42.4
26 non-retracted other          Software availabi~ Comm~ 12181  58.4  57.6  59.3
27 non-retracted other          Source availablity Open~ 12181  35.9  35.1  36.8
28 non-retracted other          Source availablity Clos~ 12181  64.1  63.2  64.9
29 retracted     Error          Software availabi~ Free   2157  38.1  36.0  40.1
30 retracted     Error          Software availabi~ Comm~  2157  61.9  59.9  64.0
31 retracted     Error          Source availablity Open~  2157  30.5  28.6  32.4
32 retracted     Error          Source availablity Clos~  2157  69.5  67.6  71.4
33 retracted     Investigation  Software availabi~ Free   1312  37.7  35.0  40.3
```

```
34 retracted     Investigation  Software availabi~ Comm~  1312  62.3  59.7  65.0
35 retracted     Investigation  Source availablity Open~  1312  27.7  25.2  30.1
36 retracted     Investigation  Source availablity Clos~  1312  72.3  69.9  74.8
37 retracted     Misconduct     Software availabi~ Free    731  36.1  32.6  39.6
38 retracted     Misconduct     Software availabi~ Comm~   731  63.9  60.4  67.4
39 retracted     Misconduct     Source availablity Open~   731  25.7  22.5  28.9
40 retracted     Misconduct     Source availablity Clos~   731  74.3  71.1  77.5
41 retracted     PaperMill      Software availabi~ Free    540  40.2  36.0  44.3
42 retracted     PaperMill      Software availabi~ Comm~   540  59.8  55.7  64.0
43 retracted     PaperMill      Source availablity Open~   540  33.1  29.2  37.1
44 retracted     PaperMill      Source availablity Clos~   540  66.9  62.9  70.8
45 retracted     Plagiarism     Software availabi~ Free    230  24.8  19.2  30.4
46 retracted     Plagiarism     Software availabi~ Comm~   230  75.2  69.6  80.8
47 retracted     Plagiarism     Source availablity Open~   230  19.6  14.4  24.7
48 retracted     Plagiarism     Source availablity Clos~   230  80.4  75.3  85.6
49 retracted     SelfPlagiarism Software availabi~ Free   1534  34.0  31.6  36.3
50 retracted     SelfPlagiarism Software availabi~ Comm~  1534  66.0  63.7  68.4
51 retracted     SelfPlagiarism Source availablity Open~  1534  26.1  23.9  28.3
52 retracted     SelfPlagiarism Source availablity Clos~  1534  73.9  71.7  76.1
53 retracted     other          Software availabi~ Free   1472  37.8  35.3  40.2
54 retracted     other          Software availabi~ Comm~  1472  62.2  59.8  64.7
55 retracted     other          Source availablity Open~  1472  30.8  28.4  33.1
56 retracted     other          Source availablity Clos~  1472  69.2  66.9  71.6
```

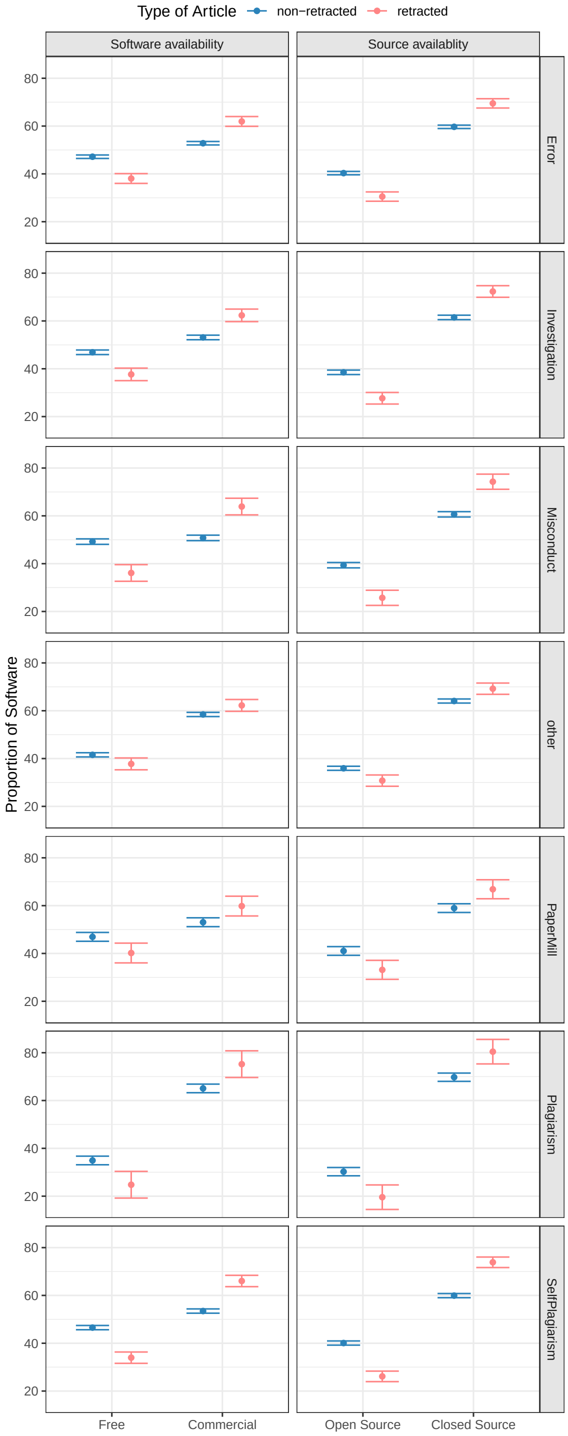


Fig S12: Proportion of free or open source software across retracted and control articles per retraction reason. A sep−arate control set is constructed for each retraction reasons by selecting the ten corresponding articles for each retrac−ted paper. Error bars indicate 95% CIs.

Software Type

We analyze the difference in software type usage between retracted and control articles based on the types: Application, PlugIn, Programming Environment, and Operating System.

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Software_Type) %>%
  drop_na() %>%
  distinct() %>%
  group_by(Set_ID, Software_Type) %>%
  summarize(n=n(), .groups = 'drop_last') %>%
  group_by(Set_ID) %>%
  mutate(rel=n/sum(n)) %>%
  group_by(Set_ID) %>%
  mutate(n=sum(n)) %>%
  ungroup() %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIu = rel + MoE, CIl = rel-MoE) %>%
  mutate(rel=rel*100, CIl=CIl*100, CIu=CIu*100) %>%
  mutate(Set_ID=ifelse(Set_ID=='non-retracted',
                       "Control",
                       "Retracted")) %>%
  select(Set_ID, Software_Type, n, rel, CIl, CIu) %>%
  print(., n=8) %>%
  ggplot(aes(rel, Software_Type)) +
  geom_col(aes(fill=Set_ID), position='dodge') +
  geom_errorbar(aes(y=Software_Type, xmin=CIl, xmax=CIu, group=Set_ID),
                position='dodge') +
  geom_text(aes(x = ifelse(abs(CIl)>10, CIl - 4, CIu + 3.5),
                y=Software_Type,
                group=Set_ID,
                label=paste0(format(abs(rel),digits=1,nsmall=1), "%")),
            position = position_dodge(width = .9), size=3) +
  scale_fill_manual('Type of Article',
                    values = c("#2b83ba", "#ff8585")) +
  theme(legend.position = 'top',
        plot.caption = element_text(size=10)) +
  labs(x='Proportion of Mentions',
       y='Software Type',
       caption = 'Fig S13: Proportion of software types on overall software mentions
    between retracted and control articles.')
```

```
# A tibble: 8 x 6
  Set_ID    Software_Type              n   rel   CIl   CIu
  <chr>     <chr>                  <int> <dbl> <dbl> <dbl>
1 Control   Application            63725 86.4  86.1  86.6
2 Control   OperatingSystem        63725  2.33  2.21  2.45
3 Control   PlugIn                 63725  5.98  5.79  6.16
4 Control   ProgrammingEnvironment 63725  5.32  5.15  5.49
5 Retracted Application             6072 89.8  89.1  90.6
6 Retracted OperatingSystem          6072  3.11  2.68  3.55
7 Retracted PlugIn                   6072  3.69  3.21  4.16
8 Retracted ProgrammingEnvironment   6072  3.36  2.91  3.81
```

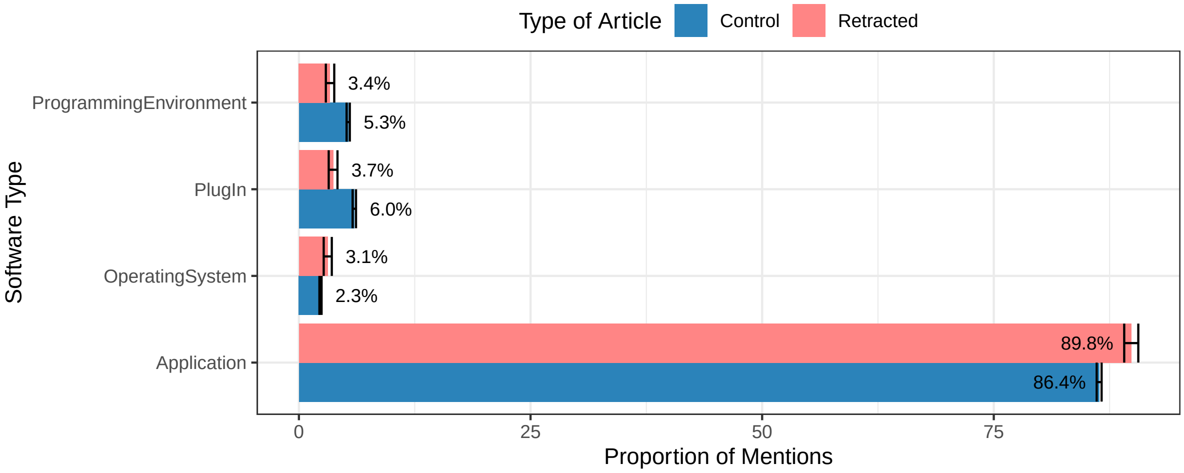


Fig S13: Proportion of software types on overall software mentions between retracted and control articles.

Citation Quality

After exploring the software landscape, we analyze the citation quality for software.

Overall

First, we directly compare the sets.

```
names <- c("No Info", "Incomplete Info",
           "Informal Citation", "Formal Citation")

df %>%
  select(Set_ID, Paper_ID, Software_ID, Version, Developer, Citation) %>%
  filter(!is.na(Software_ID)) %>%
  group_by(Set_ID, Paper_ID, Software_ID) %>%
  summarize(Version=any(Version),
            Developer=any(Developer),
            Citation=any(Citation),
            .groups = 'drop_last') %>%
  mutate(version_and_developer = Version & Developer & ! Citation,
         no_citation_info = ! Version & ! Developer & ! Citation,
         version_or_developer = (Version & ! Developer & ! Citation) |
           (Developer & ! Version & ! Citation),
         citation_p = Citation) %>%
  dplyr::select(-Version, -Developer, -Citation) %>%
  rename(Citation=citation_p) %>%
  distinct() %>%
  group_by(Set_ID) %>%
  summarize(`No Info` = sum(no_citation_info),
            `Incomplete Info` = sum(version_or_developer),
            `Formal Citation` = sum(Citation),
            `Informal Citation` = sum(version_and_developer),
            n = n(),
            .groups = 'drop_last') %>%
  pivot_longer(c(`No Info`, `Incomplete Info`,
                 `Formal Citation`, `Informal Citation`)) %>%
  mutate(rel = value/n) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIl = rel-MoE, CIu = rel + MoE) %>%
  group_by(name) %>%
  mutate(order=sum(rel))  %>%
  ungroup() %>%
  mutate(name=factor(name, levels=names)) %>%
  mutate(rel=rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(Set_ID = ifelse('non-retracted'==Set_ID,
                         "Control",
                         "Retracted")) %>%
  select(Set_ID, name, value, rel, CIl, CIu) %>%
  print(.) %>%
  ggplot(aes(name, rel)) +
  geom_point(aes(x=name, color=Set_ID),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=name, ymin=CIl, ymax=CIu, color=Set_ID),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions',
       caption = 'Fig S14: Proportion of software across different levels of citation
    completeness, separated by retracted and control articles. No Info:
    Neither the version, nor the developer of a software are provided;
    Incomplete Info: Either version or developer is provided; Informal
    Citation: Version and developer are provided; Formal citation: soft-
    ware mention is accompanied by bibliographic citation. Error bars
    indicate 95% CIs.') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  scale_y_continuous(limits=c(0,45),
                     breaks=c(0, 20, 40),
                     labels=c("0%", "20%", "40%")) +
  theme(legend.position = 'top',
        axis.title.x = element_blank(),
        plot.margin = unit(c(0,1,0,1), 'mm'),
        plot.caption = element_text(size=8))
```

```
# A tibble: 8 x 6
  Set_ID    name              value   rel   CIl   CIu
  <chr>     <fct>             <int> <dbl> <dbl> <dbl>
1 Control   No Info           21994  34.8 34.5   35.2
2 Control   Incomplete Info   18019  28.5 28.2   28.9
3 Control   Formal Citation   12970  20.5 20.2   20.9
4 Control   Informal Citation 10174  16.1 15.8   16.4
5 Retracted No Info            2200  36.5 35.2   37.7
6 Retracted Incomplete Info    1901  31.5 30.3   32.7
7 Retracted Formal Citation     645  10.7  9.91  11.5
8 Retracted Informal Citation  1289  21.4 20.3   22.4
```

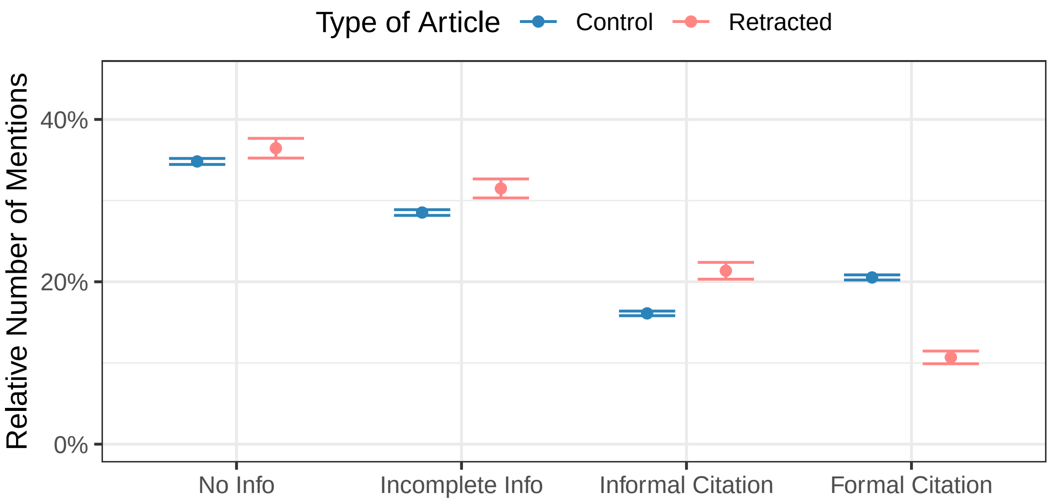


Fig S14: Proportion of software across different levels of citation completeness, separated by retracted and control articles. No Info: Neither the version, nor the developer of a software are provided; Incomplete Info: Either version or developer is provided; Informal Citation: Version and developer are provided; Formal citation: software mention is accompanied by bibliographic citation. Error bars indicate 95% CIs.

Free and Commercial Software

Then, we look at the citation quality divided by free and commercial software because prior work has shown that there are differences in their citation (Du et al. 2022).

```
df %>%
  select(Set_ID, Paper_ID, Software_ID, Software_Name,
         Version, Developer, Citation) %>%
  filter(!is.na(Software_ID)) %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, free, Paper_ID, Software_ID) %>%
  summarize(Version=any(Version),
            Developer=any(Developer),
            Citation=any(Citation),
            .groups = 'drop_last') %>%
  mutate(version_and_developer = Version & Developer & ! Citation,
         no_citation_info = ! Version & ! Developer & ! Citation,
         version_or_developer = (Version & ! Developer & ! Citation) |
           (Developer & ! Version & ! Citation),
```

```r
         citation_p = Citation) %>%
  select(-Version, -Developer, -Citation) %>%
  rename(Citation=citation_p) %>%
  distinct() %>%
  group_by(Set_ID, free) %>%
  summarize(`No Info` = sum(no_citation_info),
            `Incomplete Info` = sum(version_or_developer),
            `Formal Citation` = sum(Citation),
            `Informal Citation` = sum(version_and_developer),
            n = n(),
            .groups = 'drop_last') %>%
  pivot_longer(c(`No Info`, `Incomplete Info`,
                 `Formal Citation`, `Informal Citation`)) %>%
  mutate(rel = value/n) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIl = rel-MoE, CIu = rel + MoE) %>%
  group_by(name) %>%
  mutate(order=sum(rel))  %>%
  ungroup() %>%
  mutate(name=factor(name, levels=names)) %>%
  mutate(free=ifelse(free, "Free", "Commercial")) %>%
  mutate(rel=rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(Set_ID = ifelse('non-retracted'==Set_ID,
                         "Control",
                         "Retracted")) %>%
  select(Set_ID, free, name, value, rel, CIl, CIu) %>%
  print(.) ->
  df_ccomplete
```

```
# A tibble: 16 x 7
   Set_ID    free       name              value   rel   CIl   CIu
   <chr>     <chr>      <fct>             <int> <dbl> <dbl> <dbl>
 1 Control   Commercial No Info            3198 18.3  17.8  18.9 
 2 Control   Commercial Incomplete Info    6667 38.2  37.5  39.0 
 3 Control   Commercial Formal Citation     536  3.07  2.82  3.33
 4 Control   Commercial Informal Citation  7034 40.3  39.6  41.1 
 5 Control   Free       No Info            5144 36.9  36.1  37.7 
 6 Control   Free       Incomplete Info    3751 26.9  26.2  27.6 
 7 Control   Free       Formal Citation    4274 30.7  29.9  31.4 
 8 Control   Free       Informal Citation   773  5.54  5.16  5.92
 9 Retracted Commercial No Info             418 18.2  16.6  19.7 
```

```
10 Retracted Commercial Incomplete Info     879 38.2   36.2    40.2
11 Retracted Commercial Formal Citation      34  1.48   0.985   1.97
12 Retracted Commercial Informal Citation   969 42.1   40.1    44.1
13 Retracted Free       No Info             663 49.3   46.6    51.9
14 Retracted Free       Incomplete Info     357 26.5   24.2    28.9
15 Retracted Free       Formal Citation     217 16.1   14.2    18.1
16 Retracted Free       Informal Citation   109  8.10   6.64    9.56
```

```
df_ccomplete %>%
  ggplot(aes(free,rel)) +
  geom_point(aes(x=free, color=Set_ID),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=free, ymin=CIl, ymax=CIu, color=Set_ID),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  scale_y_continuous(breaks=c(0, 20, 40),
                     labels=c("0%", "20%", "40%")) +
  theme(legend.position = 'top',
        axis.title.x = element_blank(),
        plot.margin = unit(c(0,1,0,1), 'mm')) +
  facet_grid(~name) -> pp

ggsave("one_column_plot.jpg", pp, width = 8, height = 3)
pp +
  labs(caption = 'Fig S15 (Article Fig. 4): Proportion of software across different
    levels of citation completeness, separated by retracted and control
    articles and between free and commercial software. No Info: Neither
    the version, nor the developer of a software are provided; Incomplete
    Info: Either version or developer is provided; Informal Citation:
    Version and developer are provided; Formal citation: software mention
    is accompanied by bibliographic citation. Error bars indicate 95% CIs.') +
  theme(plot.caption = element_text(size=10))
```

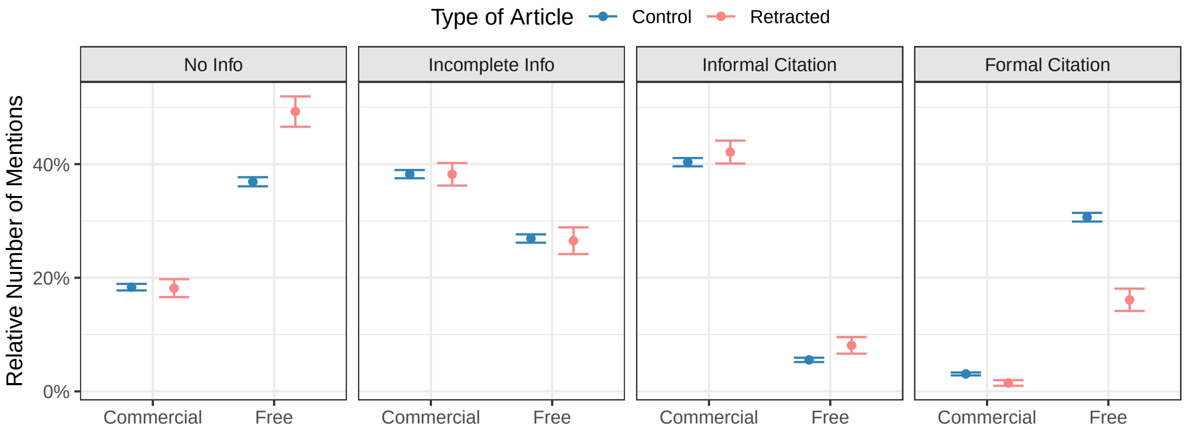


Fig S15 (Article Fig. 4): Proportion of software across different levels of citation completeness, separated by retracted and control articles and between free and commercial software. No Info: Neither the version, nor the developer of a software are provided; Incomplete Info: Either version or developer is provided; Informal Citation: Version and developer are provided; Formal citation: software mention is accompanied by bibliographic citation. Error bars indicate 95% CIs.

Open and Closed Source Software

We perform the same analysis for open and closed source software expecting similar results as the attributes free and open source are strongly correlated.

```
df %>%
  dplyr::select(Set_ID, Paper_ID, Software_ID, Software_Name,
                Version, Developer, Citation) %>%
  filter(!is.na(Software_ID)) %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, source, Paper_ID, Software_ID) %>%
  summarize(Version=any(Version),
            Developer=any(Developer),
            Citation=any(Citation),
            .groups = 'drop_last') %>%
  mutate(version_and_developer = Version & Developer & ! Citation,
         no_citation_info = ! Version & ! Developer & ! Citation,
         version_or_developer = (Version & ! Developer & ! Citation) |
           (Developer & ! Version & ! Citation),
         citation_p = Citation) %>%
  dplyr::select(-Version, -Developer, -Citation) %>%
  rename(Citation=citation_p) %>%
  distinct() %>%
  group_by(Set_ID, source) %>%
  summarize(`No Info` = sum(no_citation_info),
            `Incomplete Info` = sum(version_or_developer),
            `Formal Citation` = sum(Citation),
            `Informal Citation` = sum(version_and_developer),
            n = n(),
            .groups = 'drop_last') %>%
  pivot_longer(c(`No Info`, `Incomplete Info`,
                 `Formal Citation`, `Informal Citation`)) %>%
  mutate(rel = value/n) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIl = rel-MoE, CIu = rel + MoE) %>%
  group_by(name) %>%
  mutate(order=sum(rel))  %>%
  ungroup() %>%
  mutate(name=factor(name, levels=names)) %>%
  mutate(source=ifelse(source, "Open", "Closed")) %>%
  mutate(rel=rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(Set_ID = ifelse('non-retracted'==Set_ID,
                         "Control",
                         "Retracted")) %>%
  select(Set_ID, source, name, value, rel, CIl, CIu) %>%
  print(.) ->
  df_ccomplete
```

```
# A tibble: 16 x 7
   Set_ID    source name              value   rel   CIl   CIu
   <chr>     <chr>  <fct>             <int> <dbl> <dbl> <dbl>
 1 Control   Closed No Info            3840 19.6  19.0  20.1
 2 Control   Closed Incomplete Info    7255 37.0  36.3  37.7
 3 Control   Closed Formal Citation    1253  6.39  6.05  6.73
 4 Control   Closed Informal Citation  7262 37.0  36.4  37.7
 5 Control   Open   No Info            4502 38.3  37.4  39.1
 6 Control   Open   Incomplete Info    3163 26.9  26.1  27.7
 7 Control   Open   Formal Citation    3557 30.2  29.4  31.1
 8 Control   Open   Informal Citation   545  4.63  4.25  5.01
 9 Retracted Closed No Info             543 20.9  19.3  22.5
10 Retracted Closed Incomplete Info     973 37.4  35.6  39.3
11 Retracted Closed Formal Citation      76  2.92  2.28  3.57
12 Retracted Closed Informal Citation  1007 38.7  36.9  40.6
```

```
13 Retracted Open   No Info             538 51.4   48.4   54.4
14 Retracted Open   Incomplete Info     263 25.1   22.5   27.7
15 Retracted Open   Formal Citation     175 16.7   14.5   19.0
16 Retracted Open   Informal Citation    71  6.78   5.26   8.30
```

```
df_ccomplete %>%
  ggplot(aes(source,rel)) +
  geom_point(aes(x=source, color=Set_ID),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=source, ymin=CIl, ymax=CIu, color=Set_ID),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions',
       caption = 'Fig S16: Proportion of software across different levels of citation
    completeness, separated by retracted and control articles and between
    open-source and closed-source software. No Info: Neither the version,
    nor the developer of a software are provided; Incomplete Info: Either
    version or developer is provided; Informal Citation: Version and devel-
    oper are provided; Formal citation: software mention is accompanied
    by bibliographic citation. Error bars indicate 95% CIs.') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  scale_y_continuous(breaks=c(0, 20, 40),
                     labels=c("0%", "20%", "40%")) +
  theme(legend.position = 'top',
        axis.title.x = element_blank(),
        plot.margin = unit(c(0,1,0,1), 'mm'),
        plot.caption = element_text(size=11)) +
  facet_grid(~name)
```

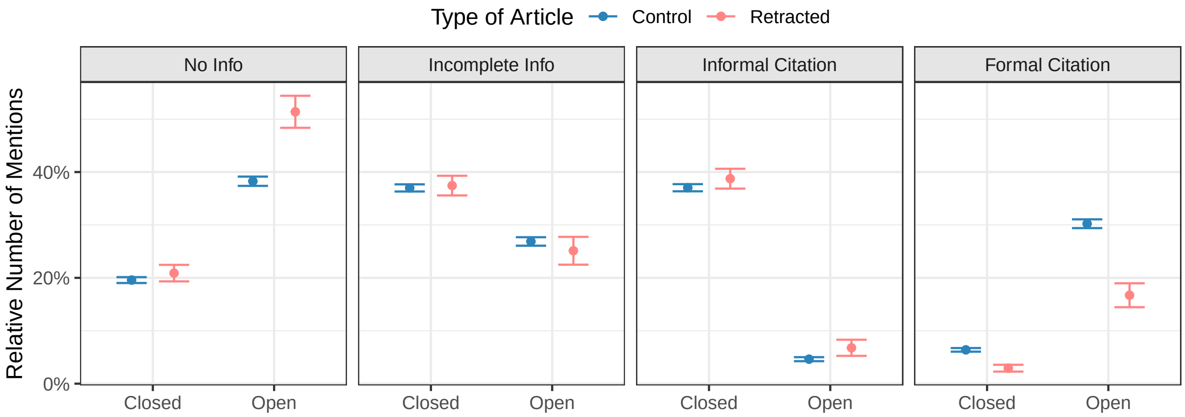


Fig S16: Proportion of software across different levels of citation completeness, separated by retracted and control articles and between open−source and closed−source software. No Info: Neither the version, nor the developer of a software are provided; Incomplete Info: Either version or developer is provided; Informal Citation: Version and devel−oper are provided; Formal citation: software mention is accompanied by bibliographic citation. Error bars indicate 95% CIs.

Statistical Test

We include a further statistical test to test whether the citation quality in terms of formal software citation differs between retracted and control articles. We use a GLM and include retraction state, number of software, and availability (free vs commercial or open source vs close source) as covariates to predict whether a software was formally cited. We further include their interaction, particularly, retraction state and availability, which we know to be related from a prior test. Finally, we estimate effect sizes through odds ratios.

```
df %>%
  select(Set_ID, Paper_ID, Software_ID, Software_Name, Version,
         Developer, Citation, Control_Sample_Origin) %>%
  filter(!is.na(Software_ID)) %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(Set_ID, Paper_ID,  Software_ID,
           Control_Sample_Origin, free, source) %>%
  summarise(citation=any(Citation), .groups = 'drop') %>%
  group_by(Set_ID, Paper_ID, Control_Sample_Origin) %>%
  mutate(n=n()) %>%
  ungroup() -> df_tt

model <- glm(
  citation~Set_ID+source+free+n+.*.,
  data=select(
    mutate(
      df_tt,
      Set_ID=factor(Set_ID, c('retracted', 'non-retracted'))),
    -c(Paper_ID, Software_ID, Control_Sample_Origin)),
  family=binomial(link='logit'))
summary(model)
```

```
Call:
glm(formula = citation ~ Set_ID + source + free + n + . * .,
    family = binomial(link = "logit"), data = select(mutate(df_tt,
        Set_ID = factor(Set_ID, c("retracted", "non-retracted"))),
        -c(Paper_ID, Software_ID, Control_Sample_Origin)))

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -4.889008   0.204945 -23.855  < 2e-16 ***
Set_IDnon-retracted         1.060401   0.202860   5.227 1.72e-07 ***
source                      2.591774   0.337023   7.690 1.47e-14 ***
free                        2.081472   0.260686   7.985 1.41e-15 ***
n                           0.247544   0.034424   7.191 6.43e-13 ***
Set_IDnon-retracted:free    0.424635   0.252053   1.685 0.092046 .
Set_IDnon-retracted:source -0.274121   0.195446  -1.403 0.160754
Set_IDnon-retracted:n      -0.116372   0.030190  -3.855 0.000116 ***
source:free                -2.349954   0.271951  -8.641  < 2e-16 ***
free:n                      0.001585   0.025209   0.063 0.949854
source:n                   -0.014990   0.017934  -0.836 0.403260
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 28933  on 35022  degrees of freedom
Residual deviance: 23002  on 35012  degrees of freedom
AIC: 23024

Number of Fisher Scoring iterations: 6
```

```r
exp(summary(model)$coefficients["n",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["n",2])
```

```
[1] 1.197307 1.280876 1.370278
```

```r
exp(summary(model)$coefficients["free",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["free",2])
```

```
[1]  4.809234  8.016264 13.361896
```

```r
exp(summary(model)$coefficients["source",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["source",2])
```

```
[1]  6.897924 13.353433 25.850413
```

```r
exp(summary(model)$coefficients["Set_IDnon-retracted",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["Set_IDnon-retracted",2])
```

```
[1] 1.940223 2.887529 4.297352
```

```r
exp(summary(model)$coefficients["source:free",1] + qnorm(c(0.025,0.5,0.975)) *
      summary(model)$coefficients["source:free",2])
```

```
[1] 0.05596839 0.09537356 0.16252238
```

Per Reason (Free and Commercial)

Then, we extend the analysis to cover retraction reasons.

```r
df_reason_sampled %>%
  filter(!is.na(Software_ID)) %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(set, free, OriginalReason, Paper_ID, Software_ID) %>%
  summarize(Version=any(Version),
            Developer=any(Developer),
            Citation=any(Citation),
            .groups = 'drop_last') %>%
  distinct() %>%
  mutate(version_and_developer = Version & Developer & ! Citation,
         no_citation_info = ! Version & ! Developer & ! Citation,
         version_or_developer = (Version & ! Developer & ! Citation) |
           (Developer & ! Version & ! Citation),
         citation_p = Citation) %>%
  dplyr::select(-Version, -Developer, -Citation) %>%
  rename(Citation=citation_p) %>%
  distinct() %>%
  group_by(set, free, OriginalReason) %>%
  summarize(`No Info` = sum(no_citation_info),
            `Incomplete Info` = sum(version_or_developer),
            `Formal Citation` = sum(Citation),
            `Informal Citation` = sum(version_and_developer),
            n = n(),
            .groups = 'drop_last') %>%
  pivot_longer(c(`No Info`, `Incomplete Info`,
                 `Formal Citation`, `Informal Citation`)) %>%
  mutate(rel = value/n) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIl = rel-MoE, CIu = rel + MoE) %>%
  group_by(name) %>%
  ungroup() %>%
  mutate(name=factor(name, levels=names)) %>%
  mutate(rel=rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(free=ifelse(free, "Free", "Commercial")) %>%
  mutate(set = ifelse('non-retracted'==set,
                      "Control",
                      "Retracted")) %>%
  mutate(OriginalReason = factor(OriginalReason, levels=reasons)) %>%
  select(set, free, OriginalReason, name, rel, CIl, CIu) %>%
  print(., n=128) ->
  df_creasons
```

```
# A tibble: 112 x 7
    set       free       OriginalReason name                 rel    CIl   CIu
    <chr>     <chr>      <fct>          <fct>              <dbl>  <dbl> <dbl>
```

```
 1 Control   Commercial Error          No Info           17.5   16.7   18.2
 2 Control   Commercial Error          Incomplete Info   38.5   37.5   39.4
 3 Control   Commercial Error          Formal Citation    3.05   2.70   3.39
 4 Control   Commercial Error          Informal Citation 41.0   40.0   42.0
 5 Control   Commercial Investigation  No Info           17.6   16.6   18.6
 6 Control   Commercial Investigation  Incomplete Info   38.8   37.5   40.1
 7 Control   Commercial Investigation  Formal Citation    2.64   2.22   3.06
 8 Control   Commercial Investigation  Informal Citation 41.0   39.7   42.3
 9 Control   Commercial Misconduct     No Info           18.4   17.1   19.6
10 Control   Commercial Misconduct     Incomplete Info   42.6   41.0   44.2
11 Control   Commercial Misconduct     Formal Citation    2.71   2.19   3.23
12 Control   Commercial Misconduct     Informal Citation 36.3   34.8   37.9
13 Control   Commercial PaperMill      No Info           12.9   11.2   14.6
14 Control   Commercial PaperMill      Incomplete Info   30.6   28.3   33.0
15 Control   Commercial PaperMill      Formal Citation    1.82   1.14   2.50
16 Control   Commercial PaperMill      Informal Citation 54.7   52.2   57.2
17 Control   Commercial Plagiarism     No Info           21.6   19.7   23.5
18 Control   Commercial Plagiarism     Incomplete Info   37.1   34.9   39.4
19 Control   Commercial Plagiarism     Formal Citation    3.37   2.52   4.21
20 Control   Commercial Plagiarism     Informal Citation 37.9   35.6   40.2
21 Control   Commercial SelfPlagiarism No Info           16.4   15.5   17.3
22 Control   Commercial SelfPlagiarism Incomplete Info   38.1   36.9   39.3
23 Control   Commercial SelfPlagiarism Formal Citation    3.02   2.60   3.43
24 Control   Commercial SelfPlagiarism Informal Citation 42.5   41.3   43.7
25 Control   Commercial other          No Info           18.3   17.4   19.2
26 Control   Commercial other          Incomplete Info   37.1   36.0   38.2
27 Control   Commercial other          Formal Citation    2.92   2.53   3.31
28 Control   Commercial other          Informal Citation 41.6   40.5   42.8
29 Control   Free       Error          No Info           36.3   35.3   37.3
30 Control   Free       Error          Incomplete Info   26.9   26.0   27.9
31 Control   Free       Error          Formal Citation   31.9   30.9   32.8
32 Control   Free       Error          Informal Citation  4.90   4.45   5.36
33 Control   Free       Investigation  No Info           36.6   35.2   37.9
34 Control   Free       Investigation  Incomplete Info   26.5   25.3   27.7
35 Control   Free       Investigation  Formal Citation   30.3   29.0   31.6
36 Control   Free       Investigation  Informal Citation  6.63   5.93   7.32
37 Control   Free       Misconduct     No Info           34.8   33.2   36.3
38 Control   Free       Misconduct     Incomplete Info   27.2   25.8   28.7
39 Control   Free       Misconduct     Formal Citation   31.4   29.9   32.9
40 Control   Free       Misconduct     Informal Citation  6.62   5.81   7.43
41 Control   Free       PaperMill      No Info           38.5   35.9   41.1
42 Control   Free       PaperMill      Incomplete Info   27.0   24.6   29.4
43 Control   Free       PaperMill      Formal Citation   28.5   26.1   30.9
```

```
44 Control   Free       PaperMill      Informal Citation  6.02   4.73   7.31
45 Control   Free       Plagiarism     No Info           39.3   36.1   42.4
46 Control   Free       Plagiarism     Incomplete Info   26.0   23.2   28.8
47 Control   Free       Plagiarism     Formal Citation   30.4   27.5   33.4
48 Control   Free       Plagiarism     Informal Citation  4.36   3.06   5.67
49 Control   Free       SelfPlagiarism No Info           36.7   35.4   38.0
50 Control   Free       SelfPlagiarism Incomplete Info   25.2   24.0   26.3
51 Control   Free       SelfPlagiarism Formal Citation   33.6   32.3   34.8
52 Control   Free       SelfPlagiarism Informal Citation  4.57   4.02   5.11
53 Control   Free       other          No Info           38.6   37.3   40.0
54 Control   Free       other          Incomplete Info   26.5   25.3   27.7
55 Control   Free       other          Formal Citation   29.6   28.4   30.9
56 Control   Free       other          Informal Citation  5.22   4.60   5.83
57 Retracted Commercial Error          No Info           16.3   14.3   18.3
58 Retracted Commercial Error          Incomplete Info   38.7   36.1   41.3
59 Retracted Commercial Error          Formal Citation    1.05   0.502  1.59
60 Retracted Commercial Error          Informal Citation 43.9   41.3   46.6
61 Retracted Commercial Investigation  No Info           15.5   13.0   18.0
62 Retracted Commercial Investigation  Incomplete Info   35.5   32.2   38.7
63 Retracted Commercial Investigation  Formal Citation    0.733  0.149  1.32
64 Retracted Commercial Investigation  Informal Citation 48.3   44.9   51.7
65 Retracted Commercial Misconduct     No Info           19.1   15.5   22.6
66 Retracted Commercial Misconduct     Incomplete Info   44.3   39.8   48.8
67 Retracted Commercial Misconduct     Formal Citation    1.71   0.536  2.89
68 Retracted Commercial Misconduct     Informal Citation 34.9   30.6   39.2
69 Retracted Commercial PaperMill      No Info            7.43   4.57  10.3
70 Retracted Commercial PaperMill      Incomplete Info   23.5   18.9   28.2
71 Retracted Commercial PaperMill      Formal Citation    0      0      0
72 Retracted Commercial PaperMill      Informal Citation 69.0   64.0   74.1
73 Retracted Commercial Plagiarism     No Info           28.3   21.6   35.0
74 Retracted Commercial Plagiarism     Incomplete Info   34.7   27.6   41.8
75 Retracted Commercial Plagiarism     Formal Citation    1.73  -0.211  3.68
76 Retracted Commercial Plagiarism     Informal Citation 35.3   28.1   42.4
77 Retracted Commercial SelfPlagiarism No Info           16.1   13.8   18.4
78 Retracted Commercial SelfPlagiarism Incomplete Info   37.5   34.5   40.5
79 Retracted Commercial SelfPlagiarism Formal Citation    0.987  0.378  1.60
80 Retracted Commercial SelfPlagiarism Informal Citation 45.4   42.3   48.5
81 Retracted Commercial other          No Info           17.9   15.4   20.4
82 Retracted Commercial other          Incomplete Info   34.0   30.9   37.0
83 Retracted Commercial other          Formal Citation    1.42   0.653  2.19
84 Retracted Commercial other          Informal Citation 46.7   43.5   50.0
85 Retracted Free       Error          No Info           51.6   48.2   55.1
86 Retracted Free       Error          Incomplete Info   24.7   21.8   27.7
```

```
 87 Retracted Free       Error          Formal Citation   16.7   14.1   19.2
 88 Retracted Free       Error          Informal Citation  6.94   5.20   8.68
 89 Retracted Free       Investigation  No Info           51.8   47.4   56.2
 90 Retracted Free       Investigation  Incomplete Info   27.9   24.0   31.9
 91 Retracted Free       Investigation  Formal Citation    8.70   6.22  11.2
 92 Retracted Free       Investigation  Informal Citation 11.5    8.72  14.4
 93 Retracted Free       Misconduct     No Info           41.3   35.3   47.2
 94 Retracted Free       Misconduct     Incomplete Info   34.1   28.4   39.8
 95 Retracted Free       Misconduct     Formal Citation   11.0    7.21  14.8
 96 Retracted Free       Misconduct     Informal Citation 13.6    9.50  17.8
 97 Retracted Free       PaperMill      No Info           73.7   67.9   79.6
 98 Retracted Free       PaperMill      Incomplete Info   18.4   13.3   23.6
 99 Retracted Free       PaperMill      Formal Citation    0      0      0
100 Retracted Free       PaperMill      Informal Citation  7.83   4.26  11.4
101 Retracted Free       Plagiarism     No Info           29.8   17.9   41.7
102 Retracted Free       Plagiarism     Incomplete Info   42.1   29.3   54.9
103 Retracted Free       Plagiarism     Formal Citation   24.6   13.4   35.7
104 Retracted Free       Plagiarism     Informal Citation  3.51  -1.27   8.29
105 Retracted Free       SelfPlagiarism No Info           62.6   58.4   66.7
106 Retracted Free       SelfPlagiarism Incomplete Info   24.0   20.3   27.7
107 Retracted Free       SelfPlagiarism Formal Citation    6.14   4.08   8.20
108 Retracted Free       SelfPlagiarism Informal Citation  7.29   5.06   9.53
109 Retracted Free       other          No Info           51.4   47.3   55.6
110 Retracted Free       other          Incomplete Info   21.9   18.5   25.4
111 Retracted Free       other          Formal Citation   19.6   16.3   22.9
112 Retracted Free       other          Informal Citation  7.01   4.89   9.14
```

```r
filter(df_creasons, OriginalReason %in% c("Error",
                                          "Investigation",
                                          "Plagiarism",
                                          "SelfPlagiarism")) %>%
  ggplot(aes(free,rel)) +
  geom_point(aes(color=set),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=free, ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  theme(legend.position = 'top',
        axis.title.x = element_blank(), axis.text.y = element_blank(),
        axis.ticks.y.left = element_blank(),
        axis.text.y.right = element_text(hjust=.55),
        axis.title.y.right = element_blank(),
        plot.margin = unit(c(0,1,0,0), 'mm'),
        )  +
  facet_grid(OriginalReason~name, switch='y') +
    scale_y_continuous(
      limits = c(-2, 82),
      breaks=c(0, 20, 40, 60, 80),
      labels=c("0%", "20%", "40%", "60%", "80%"),
      sec.axis = sec_axis(~., breaks=c(0, 20, 40, 60, 80),
                          labels=c("0%", "20%", "40%", "60%", "80%"))) ->
    p_creasons1

filter(df_creasons, OriginalReason %in% c("Misconduct",
                                          "PaperMill",
                                          "other")) %>%
  ggplot(aes(free,rel)) +
  geom_point(aes(color=set),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=free, ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  scale_y_continuous(limits = c(-2, 82)) +
  theme(legend.position = 'none',
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y.left = element_blank()) +
  facet_grid(OriginalReason~name) ->
  p_creasons2

pp <- p_creasons1 + (p_creasons2 / plot_spacer() +
                       plot_layout(heights = c(33.5,10)))
ggsave("two_column_plot.jpg", pp, width=12, height=7, bg = 'white')
pp +
  plot_annotation(caption = 'Fig S17 (Article Fig. 5): Proportion of software mentions acr
    different levels of citation completeness per retraction reason,
    separated by retracted and control articles. No Info: Neither the
```

```
      version, nor the developer of a software is provided; Incomplete
      Info: Either version or developer are provided; Informal Citation:
      Version and developer are provided; Formal citation: software
      mention is accompanied by bibliographic citation (independent
      from any associated information). Error bars indicate 95% CIs.') &
    theme(plot.caption = element_text(size=14))
```

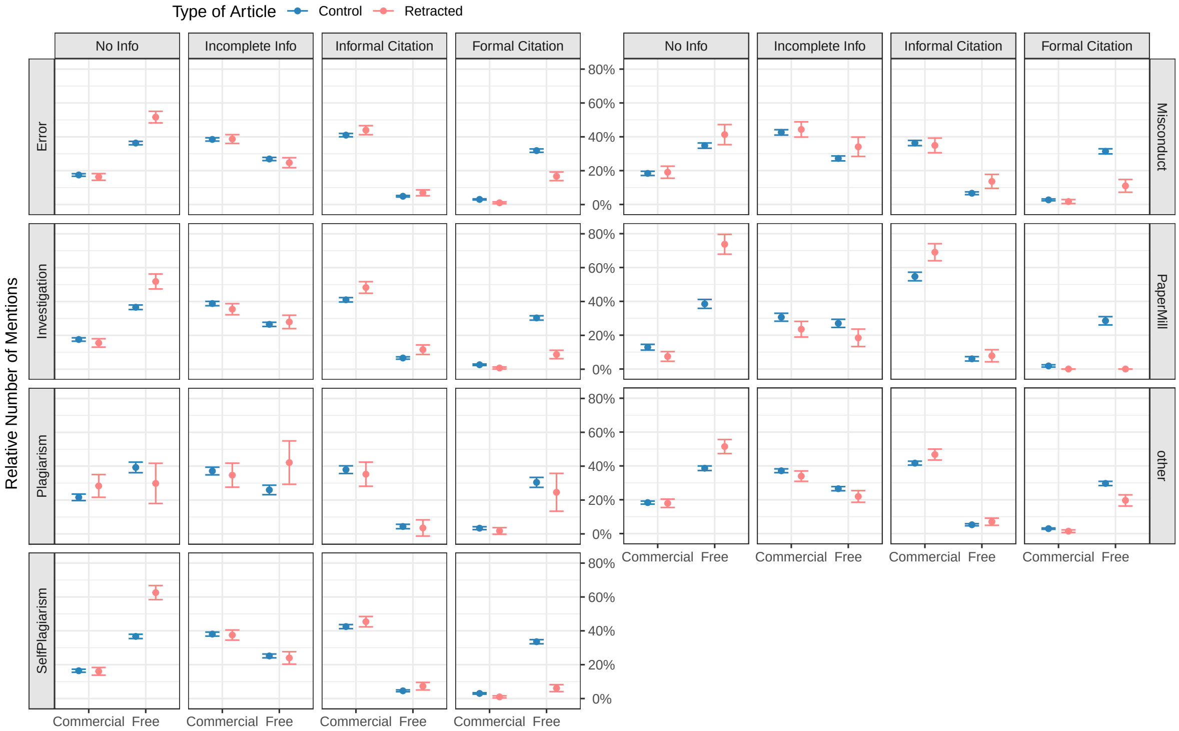


Fig S17 (Article Fig. 5): Proportion of software mentions across different levels of citation completeness per retraction reason, separated by retracted and control articles. No Info: Neither the version, nor the developer of a software is provided; Incomplete Info: Either version or developer are provided; Informal Citation: Version and developer are provided; Formal citation: software mention is accompanied by bibliographic citation (independent from any associated information). Error bars indicate 95% CIs.

Per Reason (Open and Closed Source)

We also repeat this analysis for open and closed source software.

```
df_reason_sampled %>%
  filter(!is.na(Software_ID)) %>%
  inner_join(software_enrichment, by=c(
    'Software_ID'='Software_ID',
    'Software_Name'='Software_Name')) %>%
  group_by(set, source, OriginalReason, Paper_ID, Software_ID) %>%
  summarize(Version=any(Version),
            Developer=any(Developer),
            Citation=any(Citation),
            .groups = 'drop_last') %>%
  distinct() %>%
  mutate(version_and_developer = Version & Developer & ! Citation,
         no_citation_info = ! Version & ! Developer & ! Citation,
         version_or_developer = (Version & ! Developer & ! Citation) |
           (Developer & ! Version & ! Citation),
         citation_p = Citation) %>%
  dplyr::select(-Version, -Developer, -Citation) %>%
  rename(Citation=citation_p) %>%
  distinct() %>%
  group_by(set, source, OriginalReason) %>%
  summarize(`No Info` = sum(no_citation_info),
            `Incomplete Info` = sum(version_or_developer),
            `Formal Citation` = sum(Citation),
            `Informal Citation` = sum(version_and_developer),
            n = n(),
            .groups = 'drop_last') %>%
  pivot_longer(c(`No Info`, `Incomplete Info`,
                 `Formal Citation`, `Informal Citation`)) %>%
  mutate(rel = value/n) %>%
  mutate(SEM=sqrt((rel * (1-rel))/n),
         MoE = sqrt((rel * (1-rel))/n) * 1.96) %>%
  mutate(CIl = rel-MoE, CIu = rel + MoE) %>%
  group_by(name) %>%
  ungroup() %>%
  mutate(name=factor(name, levels=names)) %>%
  mutate(rel=rel*100, CIu=CIu*100, CIl=CIl*100) %>%
  mutate(source=ifelse(source, "Open", "Closed")) %>%
  mutate(set = ifelse('non-retracted'==set,
                      "Control",
                      "Retracted")) %>%
  mutate(OriginalReason = factor(OriginalReason, levels=reasons)) %>%
  select(set, source, OriginalReason, name, rel, CIl, CIu) %>%
  print(., n=128) ->
  df_creasons
```

```
# A tibble: 112 x 7
    set       source OriginalReason name                 rel     CIl   CIu
    <chr>     <chr>  <fct>          <fct>              <dbl>   <dbl> <dbl>
  1 Control   Closed Error          No Info           19.2  18.5    19.9
  2 Control   Closed Error          Incomplete Info   36.9  36.0    37.8
  3 Control   Closed Error          Formal Citation    6.79  6.32    7.26
  4 Control   Closed Error          Informal Citation 37.1  36.2    38.0
  5 Control   Closed Investigation  No Info           18.4  17.5    19.4
  6 Control   Closed Investigation  Incomplete Info   37.8  36.7    39.0
  7 Control   Closed Investigation  Formal Citation    6.25  5.66    6.84
  8 Control   Closed Investigation  Informal Citation 37.5  36.3    38.7
  9 Control   Closed Misconduct     No Info           18.6  17.5    19.7
 10 Control   Closed Misconduct     Incomplete Info   40.8  39.3    42.2
 11 Control   Closed Misconduct     Formal Citation    7.31  6.54    8.08
 12 Control   Closed Misconduct     Informal Citation 33.3  31.9    34.7
 13 Control   Closed PaperMill      No Info           15.9  14.1    17.7
 14 Control   Closed PaperMill      Incomplete Info   29.9  27.6    32.1
 15 Control   Closed PaperMill      Formal Citation    4.85  3.82    5.89
 16 Control   Closed PaperMill      Informal Citation 49.4  47.0    51.8
 17 Control   Closed Plagiarism     No Info           22.4  20.5    24.3
 18 Control   Closed Plagiarism     Incomplete Info   36.1  33.9    38.2
 19 Control   Closed Plagiarism     Formal Citation    5.81  4.75    6.87
 20 Control   Closed Plagiarism     Informal Citation 35.7  33.6    37.9
 21 Control   Closed SelfPlagiarism No Info           18.4  17.5    19.3
 22 Control   Closed SelfPlagiarism Incomplete Info   36.2  35.1    37.3
 23 Control   Closed SelfPlagiarism Formal Citation    6.90  6.32    7.48
 24 Control   Closed SelfPlagiarism Informal Citation 38.5  37.4    39.6
 25 Control   Closed other          No Info           20.0  19.1    20.9
 26 Control   Closed other          Incomplete Info   35.7  34.7    36.8
 27 Control   Closed other          Formal Citation    5.87  5.35    6.39
 28 Control   Closed other          Informal Citation 38.4  37.3    39.5
 29 Control   Open   Error          No Info           37.0  35.9    38.1
 30 Control   Open   Error          Incomplete Info   27.3  26.2    28.3
 31 Control   Open   Error          Formal Citation   31.2  30.2    32.3
 32 Control   Open   Error          Informal Citation  4.55  4.07    5.02
 33 Control   Open   Investigation  No Info           39.3  37.8    40.8
 34 Control   Open   Investigation  Incomplete Info   25.3  24.0    26.7
 35 Control   Open   Investigation  Formal Citation   30.5  29.1    32.0
 36 Control   Open   Investigation  Informal Citation  4.78  4.12    5.43
 37 Control   Open   Misconduct     No Info           38.5  36.8    40.3
 38 Control   Open   Misconduct     Incomplete Info   26.2  24.6    27.8
 39 Control   Open   Misconduct     Formal Citation   31.5  29.8    33.2
 40 Control   Open   Misconduct     Informal Citation  3.81  3.11    4.51
```

```
41 Control   Open   PaperMill      No Info           37.8  35.0    40.6
42 Control   Open   PaperMill      Incomplete Info   27.6  25.0    30.1
43 Control   Open   PaperMill      Formal Citation   28.0  25.4    30.6
44 Control   Open   PaperMill      Informal Citation  6.63  5.19    8.07
45 Control   Open   Plagiarism     No Info           40.2  36.8    43.5
46 Control   Open   Plagiarism     Incomplete Info   26.7  23.6    29.7
47 Control   Open   Plagiarism     Formal Citation   29.0  25.9    32.1
48 Control   Open   Plagiarism     Informal Citation  4.18  2.80    5.55
49 Control   Open   SelfPlagiarism No Info           37.0  35.6    38.3
50 Control   Open   SelfPlagiarism Incomplete Info   25.9  24.7    27.1
51 Control   Open   SelfPlagiarism Formal Citation   32.7  31.4    34.0
52 Control   Open   SelfPlagiarism Informal Citation  4.43  3.86    5.01
53 Control   Open   other          No Info           38.8  37.3    40.2
54 Control   Open   other          Incomplete Info   27.4  26.0    28.7
55 Control   Open   other          Formal Citation   28.6  27.2    29.9
56 Control   Open   other          Informal Citation  5.30  4.64    5.96
57 Retracted Closed Error          No Info           19.8  17.8    21.8
58 Retracted Closed Error          Incomplete Info   37.8  35.3    40.2
59 Retracted Closed Error          Formal Citation    2.47  1.68    3.25
60 Retracted Closed Error          Informal Citation 40.0  37.5    42.4
61 Retracted Closed Investigation  No Info           18.8  16.3    21.2
62 Retracted Closed Investigation  Incomplete Info   35.4  32.4    38.4
63 Retracted Closed Investigation  Formal Citation    1.69  0.867   2.51
64 Retracted Closed Investigation  Informal Citation 44.2  41.0    47.3
65 Retracted Closed Misconduct     No Info           20.1  16.7    23.4
66 Retracted Closed Misconduct     Incomplete Info   44.4  40.2    48.6
67 Retracted Closed Misconduct     Formal Citation    2.39  1.11    3.68
68 Retracted Closed Misconduct     Informal Citation 33.1  29.2    37.1
69 Retracted Closed PaperMill      No Info           15.8  12.0    19.6
70 Retracted Closed PaperMill      Incomplete Info   22.2  17.9    26.4
71 Retracted Closed PaperMill      Formal Citation    0     0       0
72 Retracted Closed PaperMill      Informal Citation 62.0  57.0    67.1
73 Retracted Closed Plagiarism     No Info           29.2  22.6    35.7
74 Retracted Closed Plagiarism     Incomplete Info   34.6  27.7    41.4
75 Retracted Closed Plagiarism     Formal Citation    2.16  0.0663  4.26
76 Retracted Closed Plagiarism     Informal Citation 34.1  27.2    40.9
77 Retracted Closed SelfPlagiarism No Info           20.9  18.5    23.3
78 Retracted Closed SelfPlagiarism Incomplete Info   36.3  33.5    39.1
79 Retracted Closed SelfPlagiarism Formal Citation    1.41  0.725   2.10
80 Retracted Closed SelfPlagiarism Informal Citation 41.4  38.5    44.3
81 Retracted Closed other          No Info           21.3  18.8    23.8
82 Retracted Closed other          Incomplete Info   32.8  29.9    35.7
83 Retracted Closed other          Formal Citation    3.34  2.23    4.44
```

```
 84 Retracted Closed other          Informal Citation 42.6  39.6    45.6
 85 Retracted Open   Error          No Info           52.4  48.6    56.2
 86 Retracted Open   Error          Incomplete Info   23.4  20.2    26.6
 87 Retracted Open   Error          Formal Citation   17.3  14.4    20.2
 88 Retracted Open   Error          Informal Citation  6.84  4.91    8.77
 89 Retracted Open   Investigation  No Info           56.5  51.4    61.6
 90 Retracted Open   Investigation  Incomplete Info   25.3  20.9    29.8
 91 Retracted Open   Investigation  Formal Citation    9.09  6.13   12.0
 92 Retracted Open   Investigation  Informal Citation  9.09  6.13   12.0
 93 Retracted Open   Misconduct     No Info           47.3  40.2    54.5
 94 Retracted Open   Misconduct     Incomplete Info   29.8  23.2    36.3
 95 Retracted Open   Misconduct     Formal Citation   12.8   8.00   17.5
 96 Retracted Open   Misconduct     Informal Citation 10.1   5.80   14.4
 97 Retracted Open   PaperMill      No Info           70.9  64.3    77.6
 98 Retracted Open   PaperMill      Incomplete Info   20.1  14.2    26.0
 99 Retracted Open   PaperMill      Formal Citation    0     0       0
100 Retracted Open   PaperMill      Informal Citation  8.94  4.76   13.1
101 Retracted Open   Plagiarism     No Info           26.7  13.7    39.6
102 Retracted Open   Plagiarism     Incomplete Info   44.4  29.9    59.0
103 Retracted Open   Plagiarism     Formal Citation   28.9  15.6    42.1
104 Retracted Open   Plagiarism     Informal Citation  0     0       0
105 Retracted Open   SelfPlagiarism No Info           62.8  58.1    67.6
106 Retracted Open   SelfPlagiarism Incomplete Info   23.4  19.3    27.6
107 Retracted Open   SelfPlagiarism Formal Citation    6.48  4.07    8.89
108 Retracted Open   SelfPlagiarism Informal Citation  7.23  4.70    9.77
109 Retracted Open   other          No Info           51.4  46.8    56.0
110 Retracted Open   other          Incomplete Info   21.9  18.0    25.7
111 Retracted Open   other          Formal Citation   19.4  15.8    23.1
112 Retracted Open   other          Informal Citation  7.28  4.89    9.68
```

```r
filter(df_creasons, OriginalReason %in% c("Error",
                                           "Investigation",
                                           "Plagiarism",
                                           "SelfPlagiarism")) %>%
  ggplot(aes(source,rel)) +
  geom_point(aes(color=set),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=source, ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  theme(legend.position = 'top',
        axis.title.x = element_blank(), axis.text.y = element_blank(),
        axis.ticks.y.left = element_blank(),
        axis.text.y.right = element_text(hjust=.55),
        axis.title.y.right = element_blank(),
        plot.margin = unit(c(0,1,0,0), 'mm'),
        )  +
  facet_grid(OriginalReason~name, switch='y') +
    scale_y_continuous(
      limits = c(-2, 82),
      breaks=c(0, 20, 40, 60, 80),
      labels=c("0%", "20%", "40%", "60%", "80%"),
      sec.axis = sec_axis(~., breaks=c(0, 20, 40, 60, 80),
                          labels=c("0%", "20%", "40%", "60%", "80%"))) ->
    p_creasons1

filter(df_creasons, OriginalReason %in% c("Misconduct",
                                          "PaperMill",
                                          "other")) %>%
  ggplot(aes(source,rel)) +
  geom_point(aes(color=set),
             position=position_dodge(width=.7)) +
  geom_errorbar(aes(x=source, ymin=CIl, ymax=CIu, color=set),
                position=position_dodge(width=.7), width=.5) +
  labs(x='Relative Number of Mentions',
       y='Relative Number of Mentions') +
  scale_color_manual('Type of Article',
                     values = c("#2b83ba", "#ff8585")) +
  scale_y_continuous(limits = c(-2, 82)) +
  theme(legend.position = 'none',
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y.left = element_blank()) +
  facet_grid(OriginalReason~name) ->
  p_creasons2


p_creasons1 + (p_creasons2 / plot_spacer() +
                 plot_layout(heights = c(33.5,10)))  +
```

```r
  plot_annotation(
    caption = 'Fig S18: Proportion of software mentions across different levels
    of citation completeness per retraction reason, separated by
    retracted and control articles. No Info: Neither the version,
    nor the developer of a software is provided; Incomplete Info:
    Either version or developer are provided; Informal Citation:
    Version and developer are provided; Formal citation: software
    mention is accompanied by bibliographic citation (independent
    from any associated information). Error bars indicate 95% CIs.') &
  theme(plot.caption = element_text(size=14))
```

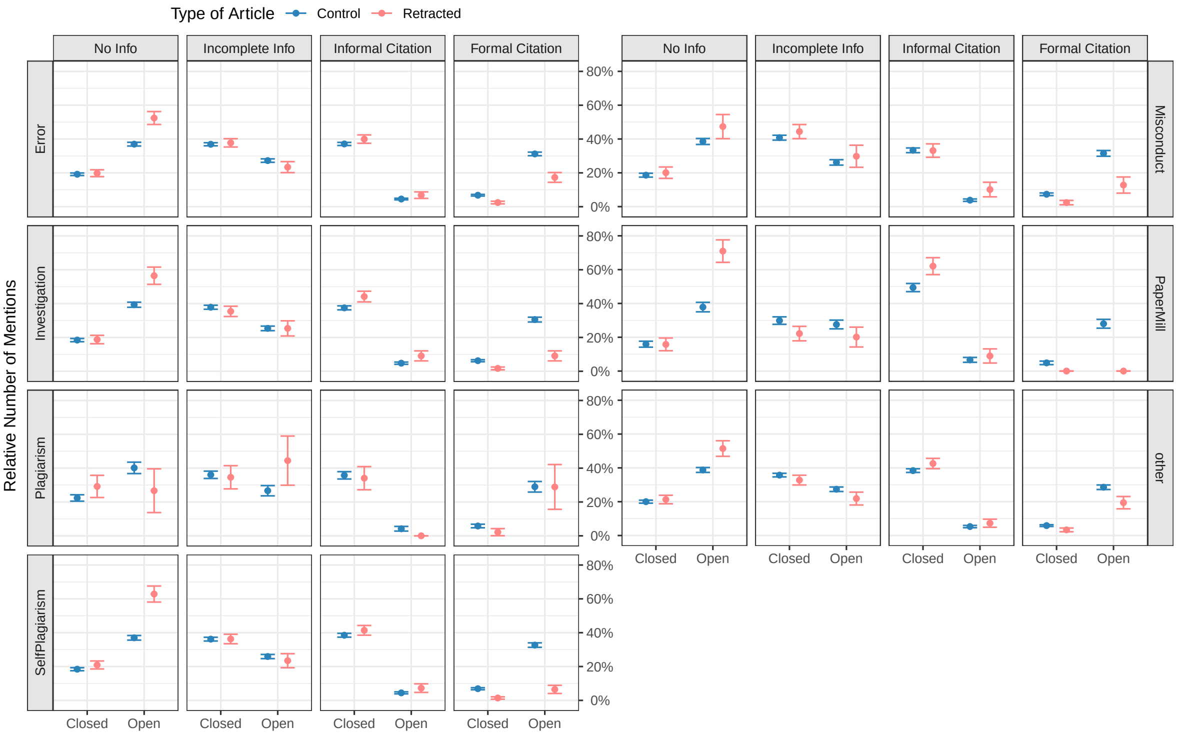


Fig S18: Proportion of software mentions across different levels of citation completeness per retraction reason, separated by retracted and control articles. No Info: Neither the version, nor the developer of a software is provided; Incomplete Info: Either version or developer are provided; Informal Citation: Version and developer are provided; Formal citation: software mention is accompanied by bibliographic citation (independent from any associated information). Error bars indicate 95% CIs.

```r
sessionInfo()
```

```
R version 4.3.1 (2023-06-16)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.6 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C

time zone: Europe/Berlin
tzcode source: system (glibc)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] effectsize_0.8.3 patchwork_1.1.2  magrittr_2.0.3   lubridate_1.9.2
 [5] forcats_1.0.0    stringr_1.5.0    dplyr_1.1.2      purrr_1.0.1
 [9] readr_2.1.4      tidyr_1.3.0      tibble_3.2.1     ggplot2_3.4.2
[13] tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] utf8_1.2.3        generics_0.1.3    stringi_1.7.12    hms_1.1.3
 [5] digest_0.6.31     evaluate_0.20     grid_4.3.1        timechange_0.2.0
 [9] fastmap_1.1.1     jsonlite_1.8.4    fansi_1.0.4       scales_1.2.1
[13] textshaping_0.3.6 cli_3.6.1         rlang_1.1.1       crayon_1.5.2
[17] bit64_4.0.5       munsell_0.5.0     withr_2.5.0       yaml_2.3.7
[21] parallel_4.3.1    tools_4.3.1       datawizard_0.8.0  tzdb_0.3.0
[25] colorspace_2.1-0  bayestestR_0.13.1 vctrs_0.6.2       R6_2.5.1
[29] lifecycle_1.0.3   bit_4.0.5         vroom_1.6.1       ragg_1.2.5
[33] insight_0.19.3    pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.3
[37] glue_1.6.2        systemfonts_1.0.4 xfun_0.39         tidyselect_1.2.0
[41] rstudioapi_0.14   parameters_0.21.1 knitr_1.42        farver_2.1.1
[45] htmltools_0.5.5   labeling_0.4.2    rmarkdown_2.21    compiler_4.3.1
```