Corpus of Resolutions

UN Security Council

(CR-UNSC-Source)

Compilation Report

Version 2024-05-19

DOI: 10.5281/zenodo.11212057

Title	Source Code for the 'Corpus of Resolutions: UN Security Council'
Abbreviation	CR-UNSC-Source
Author	Seán Fobbe, Lorenzo Gasbarri and Niccoló Ridi
Version	2024-05-19
Download	https://doi.org/10.5281/zenodo.11212057
License	GNU General Public License Version 3 (GPLv3)

Citation

Seán Fobbe, Lorenzo Gasbarri and Niccoló Ridi (2024). Source Code for the 'Corpus of Resolutions: UN Security Council' (CR-UNSC-Source). Version 2024-05-19. Zenodo. DOI: 10.5281/zenodo.11212057.

Digital Object Identifiers: Concept DOI and Version DOI

This data set is uniquely identified via the Digital Object Identifier (DOI) system. DOIs are persistent identifiers that are globally unique and can be resolved as a link by entering a DOI into the web service at www.doi.org. The DOI given in this document is a *Version DOI*, which uniquely identifies version 2024-05-19. Analysts who wish to enable replication analyses are strongly advised to cite the *Version DOI* and the exact version of the data used. A *Concept DOI* is available from the page of the Zenodo record under the heading 'Cite all versions?' and will always resolve to the latest version.

Code License: GNU General Public License Version 3 (GPLv3)

Disclaimer

This data set is a personal academic initiative and is not associated with or endorsed by the the United Nations or any of its constituent organs and organizations.

Contents

1 Corpus of Resolutions: UN Security Council

1.1 Overview

This code in the R programming language downloads and processes the full set of resolutions, drafts and meeting records rendered by the United Nations Security Council (UNSC), as published by the UN Digital Library, into a rich and structured human- and machine-readable dataset. It is the basis for the **Corpus of Resolutions: UN Security Council (CR-UNSC)**.

All data sets created with this script will always be hosted permanently open access and freely available at Zenodo, the scientific repository of CERN. Each version is uniquely identified with a persistent Digitial Object Identifier (DOI), the *Version DOI*. The newest version of the data set will always available via the link of the *Concept DOI*: https://doi.org/10.5281/zenodo.7319780

1.2 Features

- 82 Variables
- Resolution texts in all six official UN languages (English, French, Spanish, Arabic, Chinese, Russian)
- Draft texts of resolutions in English
- Meeting record texts in English
- URLs to draft texts in all other languages (French, Spanish, Arabic, Chinese, Russian)
- URLs to meeting record texts in all other languages (French, Spanish, Arabic, Chinese, Russian)
- Citation data as GraphML (UNSC-to-UNSC resolutions and UNSC-to-UNGA resolutions)
- Bibliographic database in BibTeX/OSCOLA format for e.g. Zotero, Endnote and Jabref
- Extensive Codebook to explain the uses of the dataset
- Compilation Report and Quality Assurance Report explain construction and validation of the data set
- Publication quality diagrams for teaching, research and all other purposes (PDF for printing, PNG for web)
- Open and platform independent file formats (CSV, PDF, TXT, GraphML)
- Software version controlled with Docker
- Publication of full data set (Open Data)
- Publication of full source code (Open Source)
- Free Software published under the GNU General Public License Version 3 (GNU GPL v3)
- Data published under Public Domain waiver (CC Zero 1.0)
- Secure cryptographic signatures for all files in version of record (SHA2-256 and SHA3-512)

1.3 Functionality

The pipeline will produce the following results and store them in the `output/` folder:

- Codebook as PDF

- Compilation Report as PDF
- Quality Assurance Report as PDF
- ZIP archive containing the main data set as a CSV file
- ZIP archive containing only the metadata of the main data set as a CSV file
- ZIP archive containing citation data and metadata as a GraphML file
- ZIP archive containing bibliographic data as a BIBTEX file
- ZIP archive containing all resolution texts as TXT files (OCR and extracted)
- ZIP archive containing all resolution texts as PDF files (original and English OCR)
- ZIP archive containing all draft texts as PDF files (original)
- ZIP archive containing all meeting record texts as PDF files (original)
- ZIP archive containing the full Source Code
- ZIP archive containing all intermediate pipeline results ("targets")

The integrity and veracity of each ZIP archive is documented with cryptographically secure hash signatures (SHA2-256 and SHA3-512). Hashes are stored in a separate CSV file created during the data set compilation process.

1.4 System Requirements

- The reference data sets were compiled on a Debian host system. Running the Docker config on an SELinux system like Fedora will require modifications of the Docker Compose config file.
- 40 GB space on hard drive
- Multi-core CPU recommended. We used 8 cores/16 threads to compile the reference data sets. Standard config will use all cores on a system. This can be fine-tuned in the config file.
- Given these requirements the runtime of the pipeline is approximately 40 hours.

1.5 Instructions

1.5.1 Step 1: Prepare Project Folder

Copy the Github repository into an empty (!) folder, for example by:

```
$ git clone https://github.com/seanfobbe/cr-unsc
```

Please always use an *empty* folder for creating the data set. The code will delete and re-create certain subfolders without requesting additional permission.

1.5.2 Step 2: Create Docker Image

The Dockerfile contains automated instructions to create a full operating system with all necessary dependencies. To create the image from the Dockerfile, please execute:

```
$ bash docker-build-image.sh
```

1.5.3 Step 3: Compile Dataset

If you have previously compiled the data set, whether successfuly or not, you can delete all output and temporary files by executing:

```
$ Rscript delete_all_data.R
```

You can compile the full data set by executing:

```
$ bash docker-run-project.sh
```

1.5.4 Results

Once the pipeline has concluded successfuly, the data set and all results are now stored in the folder `output/`.

1.6 Visualize Pipeline

After you have run `run_project.R` at least once you can use the commands below to visually inspect the pipeline.

```
> targets::tar_glimpse()     # Only data objects
> targets::tar_visnetwork()  # All objects, including functions
```

1.7 Troubleshooting

The below commands are useful to troubleshoot the pipeline.

```
> tar_progress()  # Show progress and errors
> tar_meta()      # Show all metadata
> tar_meta(fields = "warnings", complete_only = TRUE)  # Warnings
> tar_meta(fields = "error", complete_only = TRUE)  # Errors
> tar_meta(fields = "seconds")  # Runtime for each target
```

1.8 Project Structure

This structural analysis of the project describes its most important and version-controlled components. During compilation the pipeline will create further folders in which intermediate results are stored (`files`, `temp/` `analysis` and `output/`). Final results are stored in the folder `output/`.

```
.
├ buttons                    # Buttons (for tex title pages)
├ CHANGELOG.md               # Narrative summary of changes
├ config.toml                # Primary configuration file
```

```
├ data                          # Data sets that are imported by the pipeline
├ delete_all_data.R             # Clear all results for fresh run
├ docker-build-image.sh         # Build Docker image
├ docker-compose.yaml           # Docker container runtime configuration
├ Dockerfile                    # Instructions on how to create Docker image
├ docker-run-project.sh         # Build Docker image and run full project
├ etc                           # Additional configuration files
├ functions                     # Key pipeline components
├ gpg                           # Personal Public GPG-Key for Seán Fobbe
├ instructions                  # Instructions on how to manually handle data
├ LICENSE                       # License for the software
├ pipeline.Rmd                  # Master file for data pipeline
├ README.md                     # Usage instructions
├ reports                       # Report templates
├ run_project.R                 # Run entire pipeline
└ tex                          # LaTeX templates
```

1.9 Open Access Publications (Fobbe)

Website — https://www.seanfobbe.com

Open Data — https://zenodo.org/communities/sean-fobbe-data

Code Repository — https://zenodo.org/communities/sean-fobbe-code

Regular Publications — https://zenodo.org/communities/sean-fobbe-publications

1.10 Contact

Did you discover any errors? Do you have suggestions on how to improve the data set? You can either post these to the Issue Tracker on GitHub or write me an e-mail at fobbe-data@posteo.de

2 Load Packages

```
library(targets)
library(tarchetypes)
library(RcppTOML)
library(future)
library(data.table)
library(quanteda)
#> Package version: 3.2.4
#> Unicode version: 14.0
#> ICU version: 70.1
#> Parallel computing: 16 of 16 threads used.
#> See https://quanteda.io for tutorials and examples.
library(knitr)
library(kableExtra)
library(igraph)
#>
#> Attaching package: 'igraph'
#> The following objects are masked from 'package:future':
#>
#>     %->%, %<-%
#> The following objects are masked from 'package:stats':
#>
#>     decompose, spectrum
#> The following object is masked from 'package:base':
#>
#>     union
library(ggraph)
#> Loading required package: ggplot2

tar_unscript()
```

3 Preamble

3.1 Definitions

```
## Datestamp
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2024-05-19"

## Date and Time (Begin)
begin.script <- Sys.time()

## Read Configuration
config <- RcppTOML::parseTOML("config.toml")
print(config)
#> List of 9
#>  $ cores   :List of 3
#>   ..$ max      : logi TRUE
#>   ..$ number   : int 8
#>   ..$ tesseract: int 4
#>  $ debug   :List of 3
#>   ..$ cleanrun: logi FALSE
#>   ..$ sample  : int 100
#>   ..$ toggle  : logi FALSE
#>  $ doi     :List of 2
#>   ..$ data    :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.7319780"
#>   .. ..$ version: chr "10.5281/zenodo.11212056"
#>   ..$ software:List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.7319783"
#>   .. ..$ version: chr "10.5281/zenodo.11212057"
#>  $ download:List of 2
#>   ..$ resmax : int 2722
#>   ..$ timeout: int 600
#>  $ fig     :List of 3
#>   ..$ align : chr "center"
#>   ..$ dpi   : int 300
#>   ..$ format: chr [1:2] "pdf" "png"
#>  $ license :List of 2
#>   ..$ code: chr "GNU GPL v3"
#>   ..$ data: chr "Creative Commons Zero 1.0 Universal"
#>  $ parallel:List of 3
#>   ..$ extractPDF   : logi TRUE
#>   ..$ lingsummarize: logi TRUE
#>   ..$ multihashes  : logi TRUE
#>  $ project :List of 3
#>   ..$ author   : chr "Seán Fobbe, Lorenzo Gasbarri and Niccoló Ridi"
#>   ..$ fullname : chr "Corpus of Resolutions: UN Security Council"
#>   ..$ shortname: chr "CR-UNSC"
#>  $ quanteda:List of 1
#>   ..$ tokens_locale: chr "en_US"


# Define Analysis Folder
```

```
dir.analysis <- paste0(getwd(),
                       "/analysis")
```

3.2 Clean Up

Deletes temporary files and files in the output folder that do not match the datestamp.

```

unlink(list.files(pattern = "\\.tiff"))

unlink(grep(datestamp,
            list.files("output",
                       full.names = TRUE),
            invert = TRUE,
            value = TRUE))

unlink("temp")
unlink("temp_tesseract")
```

3.3 Create Folders

```
dirs <- c("output",
          "temp")

lapply(dirs, dir.create, showWarnings = FALSE)
#> [[1]]
#> [1] FALSE
#>
#> [[2]]
#> [1] FALSE

dir.create(dir.analysis, showWarnings = FALSE)
```

3.4 Write Package Citations

```
knitr::write_bib(renv::dependencies()$Package,
                 "temp/packages.bib")
#> Finding R package dependencies ... Done!
```

4 Global Variables

4.1 Define Packages

```
tar_option_set(packages = c("tarchetypes",  # Additional targets functions
                            "RcppTOML",     # Read and write TOML files
                            "zip",          # Enhanced ZIP handling
                            "testthat",     # Unit tests
                            "rvest",        # Web Scraping
                            "knitr",        # Scientific reporting
                            "kableExtra",   # Enhanced kable tables
                            "ggplot2",      # Advanced data vizualization
                            "ggraph",       # Graph visualization
                            "igraph",       # Graph analaysis
                            "scales",       # Scaling of figures
                            "magick",       # Image processing
                            "viridis",      # Viridis palette
                            "RColorBrewer", # ColorBrewer palette
                            "pdftools",     # Extract PDF files
                            "readtext",     # Read TXT files
                            "stringi",      # String manipulation
                            "quanteda",     # Advanced computer linguistics
                            "quanteda.textstats", # Text statistics
                            "quanteda.textplots", # Text visualizations
                            "textcat",      # Language Classification of Text
                            "future",       # Parellel computing
                            "future.apply", # Higher-order functions for future
                            "data.table"    # Advanced tabular data processing
                            )
               )


tar_option_set(workspace_on_error = TRUE) # Save Workspace on Error
tar_option_set(format = "qs")

#> Establish _targets.R and _targets_r/globals/global-packages.R.
```

4.2 Configuration

```
datestamp <- Sys.Date()

config <- RcppTOML::parseTOML("config.toml")


dir.analysis <- paste0(getwd(),
                       "/analysis")

## Caption for diagrams
caption <- paste("Fobbe/Gasbarri/Ridi | DOI:",
                 config$doi$data$version)
```

```
## Prefix for figure titles
prefix.figuretitle <- paste(config$project$shortname,
                            "| Version",
                            datestamp)

## File prefix
prefix.files <- paste0(config$project$shortname,
                       "_",
                       datestamp)


if (config$cores$max == TRUE){
    fullCores <- parallel::detectCores()
    tesseractJobs <- round(fullCores / 4) + 1
}


if (config$cores$max == FALSE){
    fullCores <- as.integer(config$cores$number)
    tesseractJobs <- as.integer(config$cores$tesseract)
}

#> Establish _targets.R and _targets_r/globals/global-config.R.
```

4.3 Define Functions

```
lapply(list.files("functions", pattern = "\\.R$", full.names = TRUE), source)

#> Establish _targets.R and _targets_r/globals/global-functions.R.
```

4.4 Define Metadata for TXT Files

```
docvarnames <- c("body",
                 "doctype",
                 "res_no",
                 "year",
                 "language")

#> Establish _targets.R and _targets_r/globals/global-txtvars.R.
```

4.5 Define Source Files

```
files.source.raw <-  c(system2("git", "ls-files", stdout = TRUE), ".git")

#> Establish _targets.R and _targets_r/globals/global-sourcefiles.R.
```

5 Pipeline: Construction

5.1 File Read Targets

Defines targets that track and read input data files. Source files are not read into R, but tracked to serve as a trigger for rebuilding the relevant ZIP archive.

5.1.1 Source Code

```
tar_target(files.source,
           files.source.raw,
           format = "file")

#> Establish _targets.R and _targets_r/targets/tar.file.source.R.
```

5.1.2 Changelog

```
tar_target(changelog,
           "CHANGELOG.md",
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.file.changelog.R.
```

5.1.3 Table of Main Record Pages (Stable)

This file contains a stable and checked list of resolutions numbers and the corresponding UN Digital Library records pages for the resolution texts and meta data.

```
list(
    tar_target(csv.record.table,
               "data/UNSC_Record-Table_2722.csv",
               format = "file"),
    tar_target(record.table.stable,
               fread(csv.record.table))
)
#> Establish _targets.R and _targets_r/targets/tar.file.recordpages.main.R.
```

5.1.4 Table of Voting Record Pages (Stable)

This file contains a stable and checked list of resolutions numbers and the corresponding UN Digital Library records pages for voting data.

```
list(
    tar_target(csv.record.voting,
               "data/UNSC_Record_Voting_2722.csv",
               format = "file"),
    tar_target(record.voting.stable,
               fread(csv.record.voting))
```

```
)
#> Establish _targets.R and _targets_r/targets/tar.file.recordpages.voting.R.
```

5.1.5 Table of Variables

All variables in the data set, including their description.

```
list(
    tar_target(file.var_codebook,
               "data/CR-UNSC_Variables.csv",
               format = "file"),
    tar_target(dt.var_codebook,
               fread(file.var_codebook))
)
#> Establish _targets.R and _targets_r/targets/tar.file.vars.R.
```

5.1.6 Gold Standard Resolutions

Gold standard resolution texts are OCR texts with all extraneous information removed and errors corrected by hand.

```
list(
    tar_target(txt_res_en_gold,
               list.files("data/res_en_gold", pattern = ".txt", full.names = TRUE
    ),
               format = "file"),
    tar_target(dt_res_en_gold,
               f.readtext(x = txt_res_en_gold,
                          docvarnames = docvarnames))
)
#> Establish _targets.R and _targets_r/targets/tar.file.goldres.R.
```

5.2 Download

5.2.1 Define Scope of Resolution Numbers

The full set of resolution numbers that makes up the data set is defined. The scope() function supports selecting resolution numbers from 1 through a defined limit (full scope) and a random subset (debugging mode). You can replace the output of the function with any arbitrary set of resolution numbers and it should work as well.

```
tar_target(res.no.full,
           f.scope(limit = config$download$resmax,
                   debug.toggle = config$debug$toggle,
                   debug.sample = config$debug$sample))

#> Establish _targets.R and _targets_r/targets/tar.download.resno.R.
```

5.2.2 Build Table of Main Record Pages (Final)

This step builds the final table of UN Digital Library main records containing the links to full texts and resolution metadata. It draws on a pre-built list to speed up the download and reduce load on UN servers.

```
tar_target(dt.record.final,
           f.record_main(record.table.stable = record.table.stable,
                         res.no.full = res.no.full))
#> Establish _targets.R and _targets_r/targets/tar.download.recordtable.main.R.
```

5.2.3 Build Table of Voting Record Pages (Final)

This step builds the final table of UN Digital Library records containing voting data. It draws on a pre-built list to speed up the download and reduce load on UN servers.

```
tar_target(dt.record.voting.final,
           f.record_voting(recordtable.stable = record.voting.stable,
                           res.no.full = res.no.full))
#> Establish _targets.R and _targets_r/targets/tar.download.recordtable.voting.R.
```

5.2.4 Download HTML Main Records for Resolutions

```
tar_target(html.record.res,
                f.download(url = dt.record.final$url_record,
                           filename = paste0(stringi::stri_pad_left(dt.record.
    final$res_no, width = 4, pad = "0"), ".html"),
                           dir = "files/record_resolution",
                           sleep.min = 0.5,
                           sleep.max = 1,
                           retries = 3,
                           retry.sleep.min = 2,
                           retry.sleep.max = 5,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE),
                format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.records.res.main.R.
```

5.2.5 Download HTML Voting Records for Resolutions

```
tar_target(html.record.voting,
                f.download(url = dt.record.voting.final$url_record,
                           filename = paste0(stringi::stri_pad_left(dt.record.
    voting.final$res_no, width = 4, pad = "0"), ".html"),
                           dir = "files/record_voting",
                           sleep.min = 0.5,
                           sleep.max = 1,
```

```
                            retries = 3,
                            retry.sleep.min = 2,
                            retry.sleep.max = 5,
                            timeout = config$download$timeout,
                            debug.toggle = FALSE),
                format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.records.res.voting.R.
```

5.2.6 Parse Resolution HTML Records

```
tar_target(dt.download,
           f.parse_records_full(html.record.res))

#> Establish _targets.R and _targets_r/targets/tar.download.parse.res.R.
```

5.2.7 Parse Voting HTML Records

```
tar_target(dt.voting,
           f.parse_records_voting(html.record.voting))

#> Establish _targets.R and _targets_r/targets/tar.download.parse.voting.R.
```

5.2.8 Download HTML Records for Drafts

```
tar_target(html.record.draft,
                f.download(url = dt.download[!is.na(url_record_draft)]$url_record
    _draft,
                           filename = paste0(stringi::stri_pad_left(dt.download[!
    is.na(url_record_draft)]$res_no, width = 4, pad = "0"), ".html"),
                           dir = "files/record_draft",
                           sleep.min = 0.5,
                           sleep.max = 1,
                           retries = 3,
                           retry.sleep.min = 5,
                           retry.sleep.max = 10,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE),
                format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.records.draft.R.
```

5.2.9 Download HTML Records for Meeting Records

```
tar_target(html.record.meeting,
                f.download(url = dt.download[!is.na(url_record_meeting)]$url_
    record_meeting,
```

```
                            filename = paste0(stringi::stri_pad_left(dt.download[!
    is.na(url_record_meeting)]$res_no, width = 4, pad = "0"), ".html"),
                            dir = "files/record_meeting",
                            sleep.min = 1,
                            sleep.max = 1.5,
                            retries = 3,
                            retry.sleep.min = 5,
                            retry.sleep.max = 10,
                            timeout = config$download$timeout,
                            debug.toggle = FALSE),
                format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.records.meeting.R.
```

5.2.10 Parse Draft HTML Records

```
tar_target(url.draft,
           f.parse_records_url(html.record.draft,
                               prefix = "url_draft_"))

#> Establish _targets.R and _targets_r/targets/tar.download.parse.draft.R.
```

5.2.11 Parse Meeting HTML Records

```
tar_target(url.meeting,
           f.parse_records_url(html.record.meeting,
                               prefix = "url_meeting_"))

#> Establish _targets.R and _targets_r/targets/tar.download.parse.meeting.R.
```

5.2.12 Finalize Download Table

```
tar_target(dt.download.final,
           f.download_manifest(dt.download = dt.download,
                               dt.record = dt.record.final,
                               url.meeting = url.meeting,
                               url.draft = url.draft))

#> Establish _targets.R and _targets_r/targets/tar.download.table.final.R.
```

5.2.13 Download, OCR, Extract and Read PDF Texts

```
values <- tibble::tibble(
                      name = c("res_ar",
                               "res_zh",
                               "res_en",
```

```
                                "res_fr",
                                "res_ru",
                                "res_es",
                                "draft_en",
                                "meeting_en"),
                      url = c(quote(dt.download.final[!is.na(url_res_ar)]$url_res
    _ar),
                              quote(dt.download.final[!is.na(url_res_zh)]$url_res
    _zh),
                              quote(dt.download.final[!is.na(url_res_en)]$url_res
    _en),
                              quote(dt.download.final[!is.na(url_res_fr)]$url_res
    _fr),
                              quote(dt.download.final[!is.na(url_res_ru)]$url_res
    _ru),
                              quote(dt.download.final[!is.na(url_res_es)]$url_res
    _es),
                              quote(dt.download.final[!is.na(url_draft_en)]$url_
    draft_en),
                              quote(dt.download.final[!is.na(url_meeting_en)]$url
    _meeting_en)),
                      filename = c(quote(paste0(dt.download.final[!is.na(url_res_
    ar)]$doc_id, "_AR.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_res_
    zh)]$doc_id, "_ZH.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_res_
    en)]$doc_id, "_EN.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_res_
    fr)]$doc_id, "_FR.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_res_
    ru)]$doc_id, "_RU.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_res_
    es)]$doc_id, "_ES.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_
    draft_en)]$doc_id, "_Draft_EN.pdf")),
                                   quote(paste0(dt.download.final[!is.na(url_
    meeting_en)]$doc_id, "_MeetingRec_EN.pdf"))),
                      dir = c("files/pdf_res_ar",
                              "files/pdf_res_zh",
                              "files/pdf_res_en",
                              "files/pdf_res_fr",
                              "files/pdf_res_ru",
                              "files/pdf_res_es",
                              "files/pdf_draft_en",
                              "files/pdf_meeting_en"),
                      tesslang = c("ara",
                                   "chi_sim",
                                   "eng",
                                   "fra",
                                   "rus",
                                   "spa",
                                   "eng",
                                   "eng"))


zip.list <- tarchetypes::tar_map(unlist = FALSE,
```

```
                                values = values,
                                names = name,
                                tar_target(pdf,
                                           f.download(url = url,
                                                      filename = filename,
                                                      dir = dir,
                                                      sleep.min = 0.3,
                                                      sleep.max = 1,
                                                      retries = 3,
                                                      retry.sleep.min = 2,
                                                      retry.sleep.max = 5,
                                                      timeout = config$download$
    timeout,
                                                      debug.toggle = FALSE),
                                           format = "file"),
                                tar_target(ocr,
                                           f.tar_pdf_ocr(grep("S_RES_
    0[0-8][0-9][0-9]",
                                                              pdf,
                                                              value = TRUE),
                                                         dpi = 300,
                                                         lang = tesslang,
                                                         crop.firstpage = 0,
                                                         crop.lastpage = 0,
                                                         output = "pdf txt",
                                                         skip = TRUE,
                                                         dir.out.pdf = paste0("
    files/pdf_tesseract_", name),
                                                         dir.out.txt = paste0("
    files/txt_tesseract_", name),
                                                         tempfile = TRUE,
                                                         quiet = TRUE,
                                                         jobs = tesseractJobs),
                                           format = "file"),
                                tar_target(txt,
                                           f.tar_pdf_extract(x = pdf,
                                                             outputdir = paste0(
    "files/txt_", name),
                                                             multicore = config$
    parallel$extractPDF,
                                                             cores = fullCores),
                                           format = "file"),
                                tar_target(dt,
                                           f.readtext(x = txt,
                                                      docvarnames = docvarnames)
    ),
                                tar_target(dt_ocr,
                                           f.readtext(x = grep("\\.txt", ocr,
    value = TRUE),
                                                      docvarnames = docvarnames)
    )
                                )

#> Establish _targets.R and _targets_r/targets/tar.download.R.
```

5.2.14 Combine PDF Targets

```
tar_target(pdf_res_all,
           c(pdf_res_ar,
             pdf_res_en,
             pdf_res_es,
             pdf_res_fr,
             pdf_res_ru,
             pdf_res_zh),
           format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.pdf.R.
```

5.2.15 Combine TXT Targets

```
tar_target(txt_all,
           c(txt_res_en_gold,
             txt_res_ar,
             txt_res_en,
             txt_res_es,
             txt_res_fr,
             txt_res_ru,
             txt_res_zh,
             txt_draft_en,
             txt_meeting_en,
             grep("\\.txt",c(ocr_res_ar,
                             ocr_res_en,
                             ocr_res_es,
                             ocr_res_fr,
                             ocr_res_ru,
                             ocr_res_zh,
                             ocr_draft_en,
                             ocr_meeting_en), value = TRUE)),
           format = "file")

#> Establish _targets.R and _targets_r/targets/tar.download.txt.R.
```

5.2.16 Combine Extracted Resolution Text DT Targets

```
tar_target(dt_extracted_res_all,
           rbind(dt_res_ar,
                 dt_res_en,
                 dt_res_es,
                 dt_res_fr,
                 dt_res_ru,
                 dt_res_zh))

#> Establish _targets.R and _targets_r/targets/tar.download.dt.res.extracted.R.
```

5.2.17 Combine OCR Resolution Text DT Targets

```
tar_target(dt_ocr_res_all,
           rbind(dt_ocr_res_ar,
                 dt_ocr_res_en,
                 dt_ocr_res_es,
                 dt_ocr_res_fr,
                 dt_ocr_res_ru,
                 dt_ocr_res_zh))

#> Establish _targets.R and _targets_r/targets/tar.download.dt.res.ocr.R.
```

5.2.18 Combine Draft Text Table Targets

```
tar_target(dt_draft_all,
           rbind(dt_draft_en,
                 dt_ocr_draft_en,
                 fill = TRUE))

#> Establish _targets.R and _targets_r/targets/tar.download.dt.draft.R.
```

5.2.19 Combine Meeting Text Table Targets

```
tar_target(dt_meeting_all,
           rbind(dt_meeting_en,
                 dt_ocr_meeting_en,
                 fill = TRUE))

#> Establish _targets.R and _targets_r/targets/tar.download.dt.meeting.R.
```

5.3 Enhance Targets

5.3.1 Merge Data

```
tar_target(dt.intermediate,
           f.merge_data(dt.extracted.res.all = dt_extracted_res_all,
                        dt.ocr.res.all = dt_ocr_res_all,
                        dt.res.en.gold = dt_res_en_gold,
                        dt.draft.all = dt_draft_all,
                        dt.meeting.all = dt_meeting_all,
                        dt.download.final = dt.download.final,
                        dt.record.final = dt.record.final,
                        dt.voting = dt.voting,
                        ocr.limit = 899))

#> Establish _targets.R and _targets_r/targets/tar.enhance.merge.R.
```

5.3.2 Create Variable 'date'

```
tar_target(var_date,
           f.date(dt.intermediate$title))
#> Establish _targets.R and _targets_r/targets/tar.enhance.date.R.
```

5.3.3 Create Variables with REGEX Extraction from Resolution Text

```
tar_target(var_regex,
           f.regex_variables(dt.intermediate$text))
#> Establish _targets.R and _targets_r/targets/tar.enhance.regex.R.
```

5.3.4 Create Variables 'ncharacters, ntokens, ntypes, nsentences'

Calculates a number of classical linguistic statistics.

```
tar_target(var_lingstats,
                f.lingstats(dt.intermediate,
                            multicore = config$parallel$lingsummarize,
                            cores = fullCores,
                            germanvars = FALSE))
#> Establish _targets.R and _targets_r/targets/tar.enhance.lingstats.R.
```

5.3.5 Create Constants

Creates a number of constants that contain important provenience information to be added to the data set. These are the version number, the version DOI, the concept DOI and the license. May become particularly relevant when merging individual rows from the data set with a larger data set.

```
tar_target(var_constants,
           data.frame(version = as.character(datestamp),
                      doi_concept = config$doi$data$concept,
                      doi_version = config$doi$data$version,
                      license = as.character(config$license$data))[rep(1, nrow(dt
    .intermediate)),])
#> Establish _targets.R and _targets_r/targets/tar.enhance.constants.R.
```

5.3.6 Combine Additional Variables

```
tar_target(vars_additional,
           data.table(date = var_date,
                      var_regex,
                      var_lingstats,
                      var_constants))

#> Establish _targets.R and _targets_r/targets/tar.enhance.combine.R.
```

5.3.7 Finalize Main Data Set

```
tar_target(dt.final,
           f.finalize(dt.intermediate = dt.intermediate,
                      vars.additional = vars_additional,
                      varnames = dt.var_codebook$varname,
                      debug.toggle = config$debug$toggle))

#> Establish _targets.R and _targets_r/targets/tar.enhance.finalize.R.
```

5.3.8 Create Variant: Metadata Only

Remove text variable to create a much smaller version of the data set with metadata only.

```
tar_target(dt.final.meta,
           dt.final[, !c("text", "text_draft", "text_meeting")])

#> Establish _targets.R and _targets_r/targets/tar.variant.meta.R.
```

5.3.9 Create Variant: Best TXT Files Only

```
tar_target(txt_en_best,
           f.tar_write_txt(text = dt.final$text,
                           doc_id = dt.final$doc_id,
                           dir = "files/txt_res_en_best",
                           cleandir = TRUE))

#> Establish _targets.R and _targets_r/targets/tar.variant.txt.best.R.
```

5.4 Citation Extraction

```
tar_target(igraph_citations,
           f.citation_extraction(dt.final = dt.final))

#> Establish _targets.R and _targets_r/targets/tar.citations.R.
```

5.5 Write Targets

Write finalized data sets and hash checksums to disk.

```
values <- tibble::tibble(
                      name = c("final",
                               "meta"),
                      input = c(quote(dt.final),
                                quote(dt.final.meta)),
```

```
                      filename = paste0(prefix.files,
                                        c("_ALL_CSV_FULL",
                                          "_ALL_CSV_META"),
                                        ".csv")
                  )


csv.all <- tarchetypes::tar_map(unlist = FALSE,
                                values = values,
                                names = name,
                                tar_target(csv,
                                           f.tar_fwrite(x = input,
                                                        filename = file.path("
    output",

    filename)),
                                           format = "file")
                                )

#> Establish _targets.R and _targets_r/targets/tar.write.csv.R.
```

```
tar_target(graphml_citations,
           f.tar_write_graph(graph = igraph_citations,
                             file = file.path("output",
                                              paste0(prefix.files,
                                                     "_GraphML_Citations.graphml"
    )),
                             format = "graphml"),
           format = "file")

#> Establish _targets.R and _targets_r/targets/tar.write.graphs.R.
```

5.5.1 Write Bibtex

```
tar_target(bibtex.oscola,
           f.bibtex(dt.final = dt.final,
                    filename = file.path("output",
                                         paste0(prefix.files,
                                                "_BIBTEX_OSCOLA.bib"))
                    )
           )

#> Establish _targets.R and _targets_r/targets/tar.write.bibtex.oscola.R.
```

5.6 Report Targets

5.6.1 Write LaTeXDefinitions

```
tar_target(latexdefs,
```

```
            f.latexdefs(config,
                        dir = "temp",
                        version = datestamp),
            format = "file")

#> Establish _targets.R and _targets_r/targets/tar.report.defs.R.
```

5.6.2 Summary Statistics for Linguistic Variables

```
tar_target(lingstats.summary,
                f.lingstats_summary(dt.final,
                                    germanvars = FALSE))

#> Establish _targets.R and _targets_r/targets/tar.report.lingsumm.R.
```

5.6.3 Test: Language Purity

```
tar_target(langtest,
           f.langtest(dt.final = dt.final))

#> Establish _targets.R and _targets_r/targets/tar.test.lang.R.
```

5.6.4 Test: OCR and Gold Quality

```
tar_target(ocrtest,
           f.ocr_quality_test(dt.res.en.gold = dt_res_en_gold[res_no < 900],
                              dt.ocr.res.all = dt_ocr_res_all[res_no < 900],
                              dt.extracted.res.all = dt_extracted_res_all[res_no
    < 900]))

#> Establish _targets.R and _targets_r/targets/tar.test.ocr.R.
```

5.6.5 Write Report: QA Test Report

```
tarchetypes::tar_render(report.test,
                        file.path("reports",
                                  "quality.Rmd"),
                        output_file = file.path("../output",
                                                paste0(config$project$shortname,
                                                       "_",
                                                       datestamp,
                                                       "_QualityAssuranceReport.
    pdf")))

#> Establish _targets.R and _targets_r/targets/tar.report.test.R.
```

5.6.6 Write Report: Codebook

```
tarchetypes::tar_render(report.codebook,
                        file.path("reports",
                                  "codebook.Rmd"),
                        output_file = file.path("../output",
                                                paste0(config$project$shortname,
                                                       "_",
                                                       datestamp,
                                                       "_Codebook.pdf")))

#> Establish _targets.R and _targets_r/targets/tar.report.codebook.R.
```

5.7 ZIP Targets

```
values <- tibble::tibble(
                        name = c("source",
                                 "txt",
                                 "pdf_res_original",
                                 "pdf_draft_original",
                                 "pdf_meeting_original",
                                 "csv_final",
                                 "csv_meta",
                                 "txt_en_best",
                                 "graphml",
                                 "bibtex_oscola"),
                        input = c(quote(files.source),
                                  quote(txt_all),
                                  quote(c(pdf_res_all, grep("\\.pdf", ocr_res_en,
    value = TRUE))),
                                  quote(pdf_draft_en),
                                  quote(pdf_meeting_en),
                                  quote(csv_final),
                                  quote(csv_meta),
                                  quote(txt_en_best),
                                  quote(graphml_citations),
                                  quote(bibtex.oscola)),
                        filename = paste0(prefix.files,
                                          "_",
                                          c("Source_Code",
                                            "ALL_TXT",
                                            "ALL_PDF_Resolutions",
                                            "EN_PDF_Drafts",
                                            "EN_PDF_Meeting-Records",
                                            "ALL_CSV_FULL",
                                            "ALL_CSV_META",
                                            "EN_TXT_BEST",
                                            "CITATIONS_GRAPHML",
                                            "BIBTEX_OSCOLA"),
                                          ".zip"),
                        mode = c(rep("mirror", 3),
                                 rep("cherry-pick", 7))
```

```
                  )



zip.list <- tarchetypes::tar_map(unlist = FALSE,
                                 values = values,
                                 names = name,
                                 tar_target(zip,
                                            f.tar_zip(x = input,
                                                      filename = filename,
                                                      dir = "output",
                                                      mode = mode),
                                            format = "file")
                                 )

#> Establish _targets.R and _targets_r/targets/tar.zip.R.
```

```
tar_target(zip_analysis,
           f.tar_zip("analysis/",
                     filename = paste(prefix.files,
                                      "ANALYSIS.zip",
                                      sep = "_"),
                     dir = "output",
                     mode = "cherry-pick",
                     report.codebook,    # manually enforced dependency
    relationship
                     report.test), # manually enforced dependency relationship
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.analysis.R.
```

5.8 Cryptographic Hashes

5.8.1 Define ZIP-Archives to be Hashed

```
tar_target(zip.all,
           c(zip_pdf_res_original,
             zip_pdf_draft_original,
             zip_pdf_meeting_original,
             zip_txt,
             zip_csv_final,
             zip_csv_meta,
             zip_txt_en_best,
             zip_bibtex_oscola,
             zip_graphml,
             zip_source,
             zip_analysis))
#> Establish _targets.R and _targets_r/targets/tar.hashes1.R.
```

5.8.2 Calculate Cryptographic Hashes

```
tar_target(hashes,
           f.tar_multihashes(c(zip.all,
                               graphml_citations,
                               report.codebook[1],
                               report.test[1]),
                             multicore = config$parallel$multihashes,
                             cores = fullCores))
#> Establish _targets.R and _targets_r/targets/tar.hashes2.R.
```

5.8.3 Write CSV: Cryptographic Hashes

```
tar_target(csv.hashes,
           f.tar_fwrite(x = hashes,
                        filename = file.path("output",
                                             paste0(prefix.files,
                                                    "_CryptographicHashes.csv"))
                        )
           )
#> Establish _targets.R and _targets_r/targets/tar.hashes3.R.
```

6 Pipeline: Compilation

6.1 Run Pipeline

```
tar_make()
```

6.2 Archive Pipeline

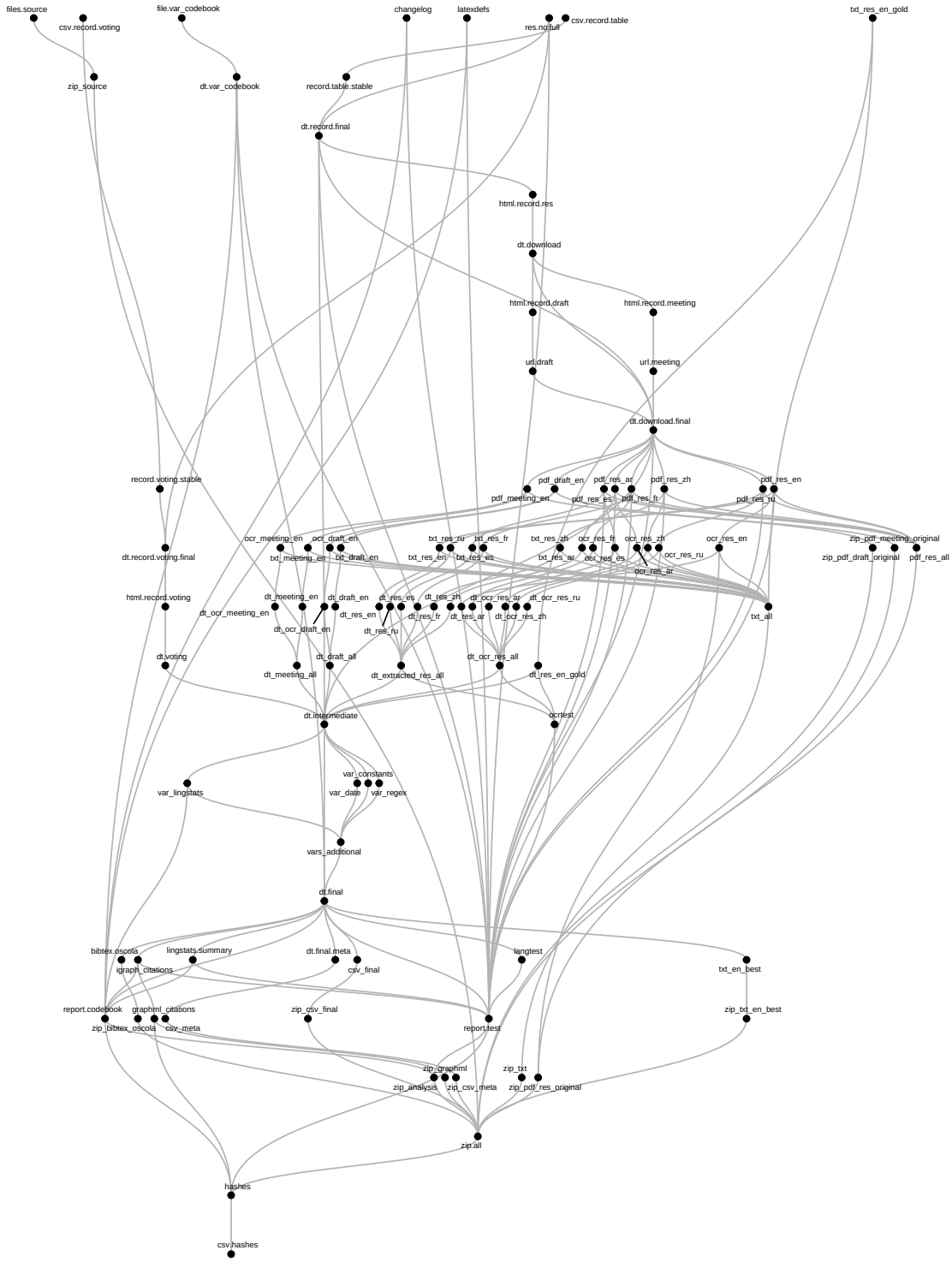
```
zip(paste0("output/",
           paste0(config$project$shortname,
                  "_",
                  datestamp),
           "_Targets_Storage.zip"),
    "_targets/")
```

6.3 Visualize Pipeline

```
edgelist <- tar_network(targets_only = TRUE)$edges
setDT(edgelist)

g  <- igraph::graph.data.frame(edgelist,
                               directed = TRUE)


ggraph(g,
       'sugiyama') +
    geom_edge_diagonal(colour = "grey70")+
    geom_node_point(size = 2)+
    geom_node_text(aes(label = name),
                   size = 2,
                   repel = TRUE)+
    theme_void()
#> Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2
    3.4.0.
#> i Please use `linewidth` in the `default_aes` field and elsewhere instead.
#> Warning: ggrepel: 3 unlabeled data points (too many overlaps). Consider
#> increasing max.overlaps
```


files.source
csv.record.voting
file.var_codebook
changelog
latexdefs
csv.record.table
res.no.full
txt_res_en_gold
zip_source
dt.var_codebook
record.table.stable
dt.record.final
html.record.res
dt.download
html.record.draft
html.record.meeting
url.draft
url.meeting
dt.download.final
record.voting.stable
pdf_draft_en
pdf_res_ar
pdf_res_zh
pdf_res_en
pdf_meeting_en
pdf_res_es
pdf_res_fr
pdf_res_ru
ocr_meeting_en
ocr_draft_en
txt_res_ru
txt_res_fr
txt_res_zh
ocr_res_fr
ocr_res_zh
ocr_res_en
zip_pdf_meeting_original
dt.record.voting.final
txt_meeting_en
txt_draft_en
txt_res_en
txt_res_es
txt_res_ar
ocr_res_es
ocr_res_ru
ocr_res_ar
zip_pdf_draft_original
pdf_res_all
html.record.voting
dt_meeting_en
dt_draft_en
dt_res_es
dt_res_zh
dt_ocr_res_ar
dt_ocr_res_ru
dt_ocr_meeting_en
dt_res_en
dt_res_fr
dt_res_ar
dt_ocr_res_zh
txt_all
dt_ocr_draft_en
dt_res_ru
dt.voting
dt_draft_all
dt_ocr_res_all
dt_meeting_all
dt_extracted_res_all
dt_res_en_gold
dt.intermediate
ocrtest
var_constants
var_lingstats
var_date
var_regex
vars_additional
dt.final
bibtex.oscola
lingstats.summary
dt.final.meta
langtest
txt_en_best
graphml_citations
csv_final
report.codebook
graphml_citations
zip_csv_final
report.test
zip_txt_en_best
zip_bibtex_oscola
csv_meta
zip_graphml
zip_txt
zip_analysis
zip_csv_meta
zip_pdf_res_original
zip.all
hashes
csv.hashes

7 Full Target List

```
meta <- tar_meta(fields = c("type", "bytes", "format"), complete_only = TRUE)
setDT(meta)
meta$MB <- round(meta$bytes / 1e6, digits = 2)

# Total Disk Usage
sum(meta$MB, na.rm = TRUE)
#> [1] 41849.56

kable(meta[order(type, name)],
      format = "latex",
      align = "r",
      booktabs = TRUE,
      longtable = TRUE) %>% kable_styling(latex_options = "repeat_header")
```

name	type	bytes	format	MB
bibtex.oscola	stem	83	qs	0.00
changelog	stem	666	file	0.00
csv.hashes	stem	89	qs	0.00
csv.record.table	stem	132239	file	0.13
csv.record.voting	stem	148997	file	0.15
csv_final	stem	364311673	file	364.31
csv_meta	stem	196518393	file	196.52
dt.download	stem	752300	qs	0.75
dt.download.final	stem	892348	qs	0.89
dt.final	stem	84062344	qs	84.06
dt.final.meta	stem	41246598	qs	41.25
dt.intermediate	stem	84201391	qs	84.20
dt.record.final	stem	10454	qs	0.01
dt.record.voting.final	stem	7985	qs	0.01
dt.var_codebook	stem	3634	qs	0.00
dt.voting	stem	23762	qs	0.02
dt_draft_all	stem	6635874	qs	6.64
dt_draft_en	stem	6036693	qs	6.04
dt_extracted_res_all	stem	48619814	qs	48.62
dt_meeting_all	stem	51318895	qs	51.32

(continued)

name	type	bytes	format	MB
dt_meeting_en	stem	33757653	qs	33.76
dt_ocr_draft_en	stem	601149	qs	0.60
dt_ocr_meeting_en	stem	17365597	qs	17.37
dt_ocr_res_all	stem	5357037	qs	5.36
dt_ocr_res_ar	stem	569748	qs	0.57
dt_ocr_res_en	stem	936500	qs	0.94
dt_ocr_res_es	stem	845727	qs	0.85
dt_ocr_res_fr	stem	777032	qs	0.78
dt_ocr_res_ru	stem	1244311	qs	1.24
dt_ocr_res_zh	stem	983011	qs	0.98
dt_res_ar	stem	11045254	qs	11.05
dt_res_en	stem	6688630	qs	6.69
dt_res_en_gold	stem	478553	qs	0.48
dt_res_es	stem	7330913	qs	7.33
dt_res_fr	stem	7449137	qs	7.45
dt_res_ru	stem	10597765	qs	10.60
dt_res_zh	stem	5500615	qs	5.50
file.var_codebook	stem	15711	file	0.02
files.source	stem	6269288	file	6.27
graphml_citations	stem	11579009	file	11.58
hashes	stem	2071	qs	0.00
html.record.draft	stem	134323594	file	134.32
html.record.meeting	stem	116345057	file	116.35
html.record.res	stem	143448003	file	143.45
html.record.voting	stem	76480729	file	76.48
igraph_citations	stem	1511155	qs	1.51
langtest	stem	690	qs	0.00
latexdefs	stem	1261	file	0.00
lingstats.summary	stem	390	qs	0.00

(continued)

name	type	bytes	format	MB
ocr_draft_en	stem	1036160294	file	1036.16
ocr_meeting_en	stem	18855640567	file	18855.64
ocr_res_ar	stem	483343558	file	483.34
ocr_res_en	stem	1553889516	file	1553.89
ocr_res_es	stem	1516991620	file	1516.99
ocr_res_fr	stem	1196220964	file	1196.22
ocr_res_ru	stem	1571985334	file	1571.99
ocr_res_zh	stem	1282946897	file	1282.95
ocrtest	stem	355	qs	0.00
pdf_draft_en	stem	266830579	file	266.83
pdf_meeting_en	stem	2045867702	file	2045.87
pdf_res_all	stem	2002912035	file	2002.91
pdf_res_ar	stem	310447555	file	310.45
pdf_res_en	stem	271564149	file	271.56
pdf_res_es	stem	302273031	file	302.27
pdf_res_fr	stem	263895559	file	263.90
pdf_res_ru	stem	412168952	file	412.17
pdf_res_zh	stem	442562789	file	442.56
record.table.stable	stem	10454	qs	0.01
record.voting.stable	stem	7985	qs	0.01
report.codebook	stem	2704137	file	2.70
report.test	stem	5868425	file	5.87
res.no.full	stem	353	qs	0.00
txt_all	stem	480983219	file	480.98
txt_draft_en	stem	25947710	file	25.95
txt_en_best	stem	3528	qs	0.00
txt_meeting_en	stem	133782657	file	133.78
txt_res_ar	stem	50830252	file	50.83
txt_res_en	stem	31575824	file	31.58

(continued)

name	type	bytes	format	MB
txt_res_en_gold	stem	2085565	file	2.09
txt_res_es	stem	35183522	file	35.18
txt_res_fr	stem	34297659	file	34.30
txt_res_ru	stem	55046119	file	55.05
txt_res_zh	stem	22205181	file	22.21
url.draft	stem	58876	qs	0.06
url.meeting	stem	50454	qs	0.05
var_constants	stem	2126	qs	0.00
var_date	stem	3990	qs	0.00
var_lingstats	stem	16030	qs	0.02
var_regex	stem	43757	qs	0.04
vars_additional	stem	64627	qs	0.06
zip.all	stem	237	qs	0.00
zip_analysis	stem	11075378	file	11.08
zip_bibtex_oscola	stem	1104762	file	1.10
zip_csv_final	stem	100415914	file	100.42
zip_csv_meta	stem	50712951	file	50.71
zip_graphml	stem	1190530	file	1.19
zip_pdf_draft_original	stem	236068667	file	236.07
zip_pdf_meeting_original	stem	1859126603	file	1859.13
zip_pdf_res_original	stem	3250289644	file	3250.29
zip_source	stem	4352096	file	4.35
zip_txt	stem	144125073	file	144.13
zip_txt_en_best	stem	8166779	file	8.17

8 Runtime Total

```
meta <- tar_meta(fields = c("time", "seconds"), complete_only = TRUE)
setDT(meta)
meta$mins <- round(meta$seconds / 60, digits = 2)

runtime.sum <- sum(meta$seconds)

## Seconds
print(runtime.sum)
#> [1] 775.075

## Minutes
runtime.sum / 60
#> [1] 12.91792

## Hours
runtime.sum / 3600
#> [1] 0.2152986
```

9 Runtime of Individual Targets

```
kable(meta[order(-seconds)],
      format = "latex",
      align = "r",
      booktabs = TRUE,
      longtable = TRUE) %>% kable_styling(latex_options = "repeat_header")
```

name	time	seconds	mins
zip_pdf_res_original	2024-05-19 00:22:07	127.734	2.13
zip_pdf_meeting_original	2024-05-19 00:19:59	79.172	1.32
langtest	2024-05-19 00:24:11	76.583	1.28
dt.download	2024-03-26 03:08:36	66.721	1.11
var_regex	2024-05-18 23:43:15	34.028	0.57
zip_txt	2024-05-19 00:22:41	33.299	0.55
report.codebook	2024-05-19 00:24:46	32.837	0.55
dt.voting	2024-03-13 04:14:36	32.399	0.54
zip_csv_final	2024-05-19 00:25:44	29.151	0.49
report.test	2024-05-19 00:25:15	26.829	0.45
dt_meeting_en	2024-03-26 03:12:10	22.123	0.37
zip_csv_meta	2024-05-19 00:26:02	17.866	0.30
hashes	2024-05-19 00:26:17	14.530	0.24
url.draft	2024-03-26 02:11:22	14.172	0.24
txt_meeting_en	2024-05-17 20:21:15	13.136	0.22
var_lingstats	2024-05-17 20:21:28	10.878	0.18
dt_ocr_meeting_en	2024-03-14 06:08:11	10.781	0.18
zip_pdf_draft_original	2024-05-19 00:18:39	9.327	0.16
url.meeting	2024-03-26 02:11:31	9.086	0.15
dt_res_es	2024-03-26 03:11:42	6.504	0.11
ocrtest	2024-05-03 18:20:02	6.322	0.11
dt_res_ru	2024-03-26 03:14:44	6.269	0.10
dt_res_en	2024-03-26 03:52:28	6.060	0.10
dt_res_fr	2024-03-26 03:12:24	5.937	0.10

(continued)

name	time	seconds	mins
dt_res_ar	2024-03-26 02:29:30	4.990	0.08
dt_draft_en	2024-03-26 03:11:47	4.941	0.08
txt_res_ru	2024-05-17 20:20:55	4.576	0.08
txt_res_ar	2024-05-17 20:20:47	4.464	0.07
lingstats.summary	2024-05-19 00:22:53	4.259	0.07
txt_res_es	2024-05-17 20:20:42	4.122	0.07
txt_res_en	2024-05-17 20:20:38	3.797	0.06
txt_res_fr	2024-05-17 20:20:51	3.687	0.06
txt_draft_en	2024-05-17 20:21:02	3.429	0.06
dt.final	2024-05-19 00:22:46	2.984	0.05
dt_res_zh	2024-03-26 03:12:13	2.678	0.04
txt_res_zh	2024-05-17 20:20:58	2.496	0.04
ocr_res_es	2024-03-13 21:48:20	2.466	0.04
bibtex.oscola	2024-05-19 00:24:13	2.363	0.04
ocr_res_ru	2024-03-14 06:07:42	2.279	0.04
ocr_res_en	2024-03-14 05:21:46	2.164	0.04
ocr_res_ar	2024-03-14 04:54:01	1.916	0.03
igraph_citations	2024-05-19 00:22:49	1.809	0.03
zip_txt_en_best	2024-05-19 00:24:48	1.754	0.03
dt.intermediate	2024-04-30 23:37:41	1.577	0.03
changelog	2024-05-18 23:37:36	1.536	0.03
txt_res_en_gold	2024-04-30 23:34:09	1.478	0.02
latexdefs	2024-05-19 00:18:29	1.465	0.02
csv.record.voting	2024-03-26 03:06:02	1.454	0.02
ocr_draft_en	2024-03-13 22:08:48	1.258	0.02
ocr_res_fr	2024-03-14 05:38:30	1.250	0.02
dt_ocr_res_es	2024-03-14 06:07:55	1.114	0.02
dt_ocr_res_en	2024-03-14 06:08:44	1.106	0.02
dt.download.final	2024-03-26 03:52:01	1.096	0.02

(continued)

name	time	seconds	mins
dt_ocr_res_ru	2024-03-14 06:11:06	0.995	0.02
dt_ocr_res_fr	2024-03-14 06:08:56	0.872	0.01
txt_en_best	2024-05-19 00:22:54	0.660	0.01
dt_ocr_res_zh	2024-03-14 06:08:34	0.643	0.01
dt_ocr_draft_en	2024-03-14 06:07:55	0.627	0.01
dt_res_en_gold	2024-04-30 23:37:01	0.611	0.01
zip_source	2024-05-19 00:18:30	0.547	0.01
dt_extracted_res_all	2024-03-26 03:55:13	0.537	0.01
zip_analysis	2024-05-19 00:26:03	0.449	0.01
dt_ocr_res_ar	2024-03-14 06:08:37	0.365	0.01
ocr_res_zh	2024-03-14 04:43:18	0.339	0.01
ocr_meeting_en	2024-03-14 04:20:28	0.332	0.01
zip_graphml	2024-05-19 00:26:02	0.266	0.00
zip_bibtex_oscola	2024-05-19 00:25:15	0.221	0.00
csv_final	2024-05-19 00:24:13	0.183	0.00
graphml_citations	2024-05-19 00:24:13	0.145	0.00
html.record.res	2024-03-26 02:06:11	0.101	0.00
csv_meta	2024-05-19 00:24:13	0.090	0.00
pdf_res_en	2024-03-26 03:52:13	0.084	0.00
dt_meeting_all	2024-03-26 03:14:46	0.077	0.00
html.record.draft	2024-03-26 02:09:00	0.074	0.00
pdf_meeting_en	2024-03-26 03:08:58	0.062	0.00
pdf_res_es	2024-03-26 03:08:39	0.061	0.00
pdf_draft_en	2024-03-26 03:08:53	0.060	0.00
pdf_res_ru	2024-03-26 03:09:09	0.060	0.00
pdf_res_zh	2024-03-26 03:09:01	0.059	0.00
pdf_res_fr	2024-03-26 03:09:06	0.057	0.00
html.record.voting	2024-03-13 02:40:09	0.051	0.00
html.record.meeting	2024-03-26 02:11:06	0.048	0.00

(continued)

name	time	seconds	mins
pdf_res_ar	2024-03-26 02:20:12	0.048	0.00
var_date	2024-04-30 23:38:32	0.025	0.00
dt.record.voting.final	2024-03-26 03:07:29	0.023	0.00
dt.record.final	2024-03-26 02:04:29	0.022	0.00
txt_all	2024-05-17 20:21:15	0.009	0.00
dt_draft_all	2024-03-26 03:14:44	0.005	0.00
dt.final.meta	2024-05-19 00:22:47	0.003	0.00
vars_additional	2024-05-19 00:22:41	0.003	0.00
record.table.stable	2024-03-26 02:04:29	0.002	0.00
record.voting.stable	2024-03-26 03:07:29	0.002	0.00
var_constants	2024-05-19 00:22:41	0.002	0.00
csv.hashes	2024-05-19 00:26:17	0.001	0.00
dt_ocr_res_all	2024-03-14 06:11:15	0.001	0.00
dt.var_codebook	2024-05-03 20:19:02	0.001	0.00
csv.record.table	2024-03-26 02:04:02	0.000	0.00
file.var_codebook	2024-05-03 20:12:14	0.000	0.00
files.source	2024-05-19 00:18:13	0.000	0.00
pdf_res_all	2024-03-26 03:52:13	0.000	0.00
res.no.full	2024-05-17 20:20:27	0.000	0.00
zip.all	2024-05-19 00:26:03	0.000	0.00

10 Warnings

```
meta <- tar_meta(fields = "warnings", complete_only = TRUE)
setDT(meta)
meta$warnings <- gsub("(\\.pdf|\\.html?|\\.txt)", "\\1 \n\n", meta$warnings)

if (meta[,.N > 0]){

    for(i in 1:meta[,.N]){

        cat(paste("###", meta[i]$name), "\n\n")
        cat(paste(meta[i]$warnings, "\n\n"))

    }

}else{

    cat("No warnings to report.")

}
```

10.0.1 dt_draft_en

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.2 dt_meeting_en

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.3 dt_ocr_draft_en

Fewer docnames supplied than existing docvars last 2 docvars given generic names.

10.0.4 dt_ocr_meeting_en

Fewer docnames supplied than existing docvars last 2 docvars given generic names.

10.0.5 dt_ocr_res_ar

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.6 dt_ocr_res_en

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.7 dt_ocr_res_es

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.8 dt_ocr_res_fr

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.9 dt_ocr_res_ru

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.10 dt_ocr_res_zh

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.11 dt_res_en_gold

Fewer docnames supplied than existing docvars last 1 docvar given generic names.

10.0.12 dt.voting

NAs introduced by coercion. NAs introduced by coercion

10.0.13 report.codebook

Package microtype Warning Unable to apply patch footnote on input line 203.

10.0.14 report.test

Package microtype Warning Unable to apply patch footnote on input line 213.

11 Errors

```
meta <- tar_meta(fields = "error", complete_only = TRUE)
setDT(meta)

if (meta[,.N > 0]){

    for(i in 1:meta[,.N]){

        cat(paste("###", meta[i]$name), "\n\n")
        cat(paste(meta[i]$error, "\n\n"))

    }

}else{

    cat("No errors to report.")

}
#> No errors to report.
```

12 File Sizes

12.1 ZIP Files

```
files <- list.files("output", pattern = "\\.zip", full.names = TRUE)

filesize <- round(file.size(files) / 10^6, digits = 2)

table.size <- data.table(basename(files),
                         filesize)


kable(table.size,
      format = "latex",
      align = c("l", "r"),
      format.args = list(big.mark = ","),
      booktabs = TRUE,
      longtable = TRUE,
      col.names = c("File",
                    "MB"))
```

File	MB
CR-UNSC_2024-05-19_ALL_CSV_FULL.zip	100.42
CR-UNSC_2024-05-19_ALL_CSV_META.zip	50.71
CR-UNSC_2024-05-19_ALL_PDF_Resolutions.zip	3,250.29
CR-UNSC_2024-05-19_ALL_TXT.zip	144.13
CR-UNSC_2024-05-19_ANALYSIS.zip	11.08
CR-UNSC_2024-05-19_BIBTEX_OSCOLA.zip	1.10
CR-UNSC_2024-05-19_CITATIONS_GRAPHML.zip	1.19
CR-UNSC_2024-05-19_EN_PDF_Drafts.zip	236.07
CR-UNSC_2024-05-19_EN_PDF_Meeting-Records.zip	1,859.13
CR-UNSC_2024-05-19_EN_TXT_BEST.zip	8.17
CR-UNSC_2024-05-19_Source_Code.zip	4.35
CR-UNSC_2024-05-19_Targets_Storage.zip	437.48

12.2 CSV Files

```
files <- list.files("output", pattern = "\\.csv", full.names = TRUE)

filesize <- round(file.size(files) / 10^6, digits = 2)

table.size <- data.table(basename(files),
                         filesize)


kable(table.size,
      format = "latex",
      align = c("l", "r"),
      format.args = list(big.mark = ","),
      booktabs = TRUE,
      longtable = TRUE,
      col.names = c("File",
                    "MB"))
```

File	MB
CR-UNSC_2024-05-19_ALL_CSV_FULL.csv	364.31
CR-UNSC_2024-05-19_ALL_CSV_META.csv	196.52
CR-UNSC_2024-05-19_CryptographicHashes.csv	0.00

13 Cryptographic Signatures

13.1 Load Signatures

```
tar_load(hashes)
```

13.2 Add Whitespace to Enable Linebreaks in SHA3-512 Signatures

This is an optical necessity only. SHA3-512 signatures are 128 characters long and are otherwise not broken across lines, vanishing beyond the page boundary instead. This variant of the signatures is intended for display purposes in this Compilation Report only.

```
hashes$sha3.512 <- paste(substr(hashes$sha3.512, 1, 64),
                         substr(hashes$sha3.512, 65, 128))
```

13.3 Show Signatures in Report

```
kable(hashes[,.(index,filename)],
      format = "latex",
      align = c("p{1cm}",
                "p{13cm}"),
      booktabs = TRUE,
      longtable = TRUE)
```

index	filename
1	output/CR-UNSC_2024-05-19_ALL_PDF_Resolutions.zip
2	output/CR-UNSC_2024-05-19_EN_PDF_Drafts.zip
3	output/CR-UNSC_2024-05-19_EN_PDF_Meeting-Records.zip
4	output/CR-UNSC_2024-05-19_ALL_TXT.zip
5	output/CR-UNSC_2024-05-19_ALL_CSV_FULL.zip
6	output/CR-UNSC_2024-05-19_ALL_CSV_META.zip
7	output/CR-UNSC_2024-05-19_EN_TXT_BEST.zip
8	output/CR-UNSC_2024-05-19_BIBTEX_OSCOLA.zip
9	output/CR-UNSC_2024-05-19_CITATIONS_GRAPHML.zip
10	output/CR-UNSC_2024-05-19_Source_Code.zip
11	output/CR-UNSC_2024-05-19_ANALYSIS.zip
12	output/CR-UNSC_2024-05-19_GraphML_Citations.graphml

13	output/CR-UNSC_2024-05-19_Codebook.pdf
14	output/CR-UNSC_2024-05-19_QualityAssuranceReport.pdf

```
kable(hashes[,.(index,sha2.256)],
      format = "latex",
      align = c("c",
                "p{13cm}"),
      booktabs = TRUE,
      longtable = TRUE)
```

index	sha2.256
1	115cdd5881e9087baabd939796aa75b14e9d3c2573c506ab80ab78c930fae831
2	b836c2e6bfaee5276f3f2ae5625280a97a03c290179b7d34b747f4414a0a73f7
3	1cf2ba4ead20cd04803e3fa0105733a01689ca2e8db084089f4c20110126c613
4	ee3d5afa4236f8d2c740d09130743b67b8c77ed1d655a65160cac660e70029fd
5	0e1ad2c6ddd927c803f7c5366a377190f5038440a46f66a062e265a6e9166e95
6	a3b8372cbfc68b722ad30ca0e80d8799a5f3167e0d4e70e773a5913881e6ed5f
7	e124f5d2dbc310992c39bc711bd56cae176d831cb3661b6182d2efd06838e367
8	2d722dbe3345b6bc4c68449a819cf9ec048cfd7214c711da686c0e61068d95a5
9	7f15409c883a17ff6619e2428fd6198cdfcf47d603547e9613cf151e13ca34c5
10	0cd7bc59cd82af8b9c49fc167f97c12a025e255f0c86bc0849faaa857a8e8616
11	086513a1ffca471af66b254f4b69ae2a4e44bb09e9a30f79a6d38ae08900bf89
12	913d503c6e5f1d12040c4d11d67c7b770902a4de221ff5aa085d491cf81fcca9
13	2a37b8916393cea04f417b9b08ade77d8e36100a727f8543857e6e56be4dfe0e
14	ec88e88d6f0639a0b9ca538310b7cbfd6c01c8794a7a8d265963a8f9316e0a73

```
kable(hashes[,.(index,sha3.512)],
      format = "latex",
      align = c("c",
                "p{13cm}"),
      booktabs = TRUE,
      longtable = TRUE)
```

index	sha3.512
1	63ac58317a10c283fbb10cab7dfb1633f686fc646d54fc4e8fb8e5440b3382e2 cb245504e54dce6626e23557a87039565b52a1debb45db2f2ddbf67891d494c6
2	7728ff40bb53857b11a140ce326863e749335b4df938f6aa00ceb0c855dc2542 063ef0366b472c2edb5cb50bd2f53830b424c32e88ebd79010273230b001bf85
3	5c77a5a9328f73b224e8d29e7f76915f554c82a37e3ea5963d72b94d23ac5d2a d288f378b01d1ee659effffbed0b3286723e2edde93fedc5243eb51ff603a424
4	00cf8d92fc82bacca71d1ff4b4a8daf02fcd11dfd0cb492865c3635e405d661a 9beddc20e700ad1c369b4ea780d130c20c6631d8919c8fa6db1804130201672f
5	292b7247482c4fa489b50a01100c636172c9f1651fb6cdfe751b7bc4446b8020 d1aca4519096ac7daa7fd4e5ebee6e7a879ac57b095e16541f0528a03696b494
6	e9d771375dc848f9ee9bf13267a385bd85f2eb550d980e7e233d7e93f672b0e2 930e44e349fdbcfaa4d745d75a76505b4200a9b13eb42b19137272d052918263
7	ac16adc51c9647fb802a2e9edc7b07493c89d3e535ea9191f82fec9a10d16d8b da42c63b90da81144cd51f60f2b50b8a62fe4443d9541e716c5d4279b2d24c01
8	53e30888e66f11026aa448fef4655b618a63f053efa1d11b78f158211418ec11 5a1bda5c05d2a1fc32c9d34cab98ecd01b5ed107d29049a0ba5e8119acd8627b
9	c4ee607258abe1f04b6a9a96cad9a7f1fff838a02beeeaba527714a97594f80e 7c1d015f34c6570d65ec00e646a0fe00fd147409f22a69ce2f0e653858e0bece
10	61cdb884915fab5f4a808134924c33d93253ebe93eefd442209032a5984b1852 53178ec12dfa1f6d0b72cd30adf074c63c57b7cccefc982ac3c0b6ef05cece44
11	9ccdb74f212bfb03a2dc0aa331112c77cf913eeebd7ecdea3af729c44c8472d8 25b68cd71bb4f1b5b383688efb6d2a4822c33dede29276dd9ca191f7efb841c2
12	6703c70bd1ed67f180e043dd97e5fbbc5d45e8a63d63fbf11dd3135329c7c7a4 d4aeaaf4a47cc6e992cf65da5a8e68b0782dab0df1b6588c3d15795e67bebc35
13	421d13dec361710301c18e2d8b52ee09a382947b6c912ec011536d2b2914baf3 6167885a83908ad42fd8619e3e611adf640c56018a9b944cfca562a500051d00
14	3d23fd45e11fa68241364e812554909939438faf541ef0c5ef6ae1982a5d2d74 e7f536d28828788f779b54b4c0ae222ba8716f781a9f2a8453f442bac27a1ecf

14 Changelog

14.1 Version 2024-05-19

- New variant: EN_TXT_BEST containing a write-out of the English resolution texts equivalent to the CSV file text variable
- New diagrams: bar charts of top M49 regions and sub-regions of countries mentioned in resolution texts
- Fixed naming mix-up of BIBTEX and GRAPHML zip archives
- Fixed whitespace character detection in citation extraction (adds ca. 10% more citations)
- Fixed improper merging of weights in citation network
- Fixed "cannot xtfrm data frames" warning
- Improve REGEX detection for certain geographic entities
- Improve Codebook (headings, citation network docs)

14.2 Version 2024-05-03

- Initial Release

15 Runtime of Script

```
## Date Stamp
print(datestamp)
#> [1] "2024-05-19"

## Date and Time (Begin)
print(begin.script)
#> [1] "2024-05-19 00:18:24 UTC"


## Date and Time (End)
end.script <- Sys.time()
print(end.script)
#> [1] "2024-05-19 00:26:33 UTC"


## Runtime for Script
print(end.script - begin.script)
#> Time difference of 8.148426 mins
```

16 Parameters for Strict Replications

```
system2("openssl", "version", stdout = TRUE)
#> [1] "OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)"

sessionInfo()
#> R version 4.2.2 (2022-10-31)
#> Platform: x86_64-pc-linux-gnu (64-bit)
#> Running under: Ubuntu 22.04.2 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so
#>
#> locale:
#>  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
#>  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
#>  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
#>  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
#>  [9] LC_ADDRESS=C               LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] ggraph_2.1.0      ggplot2_3.4.1     igraph_1.4.1      kableExtra_1.3.4
#>  [5] knitr_1.42        quanteda_3.2.4    data.table_1.14.8 future_1.32.0
#>  [9] RcppTOML_0.2.2    tarchetypes_0.7.5 targets_0.14.3
#>
#> loaded via a namespace (and not attached):
#>  [1] viridis_0.6.2       httr_1.4.5          tidyr_1.3.0
#>  [4] bit64_4.0.5         tidygraph_1.2.3     viridisLite_0.4.1
#>  [7] RcppParallel_5.1.7  highr_0.10          future.callr_0.8.1
#> [10] base64url_1.4       renv_0.17.0         yaml_2.3.7
#> [13] ggrepel_0.9.3       globals_0.16.2      pillar_1.8.1
#> [16] backports_1.4.1     lattice_0.20-45     glue_1.6.2
#> [19] digest_0.6.31       polyclip_1.10-4     rvest_1.0.3
#> [22] stringfish_0.15.7   colorspace_2.1-0    htmltools_0.5.4
#> [25] Matrix_1.5-1        pkgconfig_2.0.3     listenv_0.9.0
#> [28] purrr_1.0.1         scales_1.2.1        webshot_0.5.4
#> [31] processx_3.8.0      svglite_2.1.1       tweenr_2.0.2
#> [34] RApiSerialize_0.1.2 ggforce_0.4.1       tibble_3.2.0
#> [37] generics_0.1.3      farver_2.1.1        withr_2.5.0
#> [40] furrr_0.3.1         cli_3.6.0           magrittr_2.0.3
#> [43] evaluate_0.20       ps_1.7.2            stopwords_2.3
#> [46] fs_1.6.1            fansi_1.0.4         parallelly_1.34.0
#> [49] MASS_7.3-58.1       xml2_1.3.3          tools_4.2.2
#> [52] lifecycle_1.0.3     stringr_1.5.0       munsell_0.5.0
#> [55] callr_3.7.3         compiler_4.2.2      qs_0.25.5
#> [58] systemfonts_1.0.4   rlang_1.0.6         grid_4.2.2
#> [61] rstudioapi_0.14     labeling_0.4.2      rmarkdown_2.20
#> [64] gtable_0.3.1        codetools_0.2-18    graphlayouts_0.8.4
#> [67] R6_2.5.1            gridExtra_2.3       dplyr_1.1.0
#> [70] bit_4.0.5           fastmap_1.1.1       utf8_1.2.3
```

```
#> [73] fastmatch_1.1-3     stringi_1.7.12      parallel_4.2.2
#> [76] Rcpp_1.0.10         vctrs_0.5.2         tidyselect_1.2.0
#> [79] xfun_0.37
```

Bibliography

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2023. *Rmarkdown: Dynamic Documents for R.*

Arel-Bundock, Vincent. 2022. *Countrycode: Convert Country Names and Country Codes.* https://vincentarelbundock.github.io/countrycode/.

Arel-Bundock, Vincent, Nils Enevoldsen, and CJ Yetman. 2018. "Countrycode: An R Package to Convert Country Names and Country Codes." *Journal of Open Source Software* 3 (28): 848. https://doi.org/10.21105/joss.00848.

Bengtsson, Henrik. 2021. "A Unifying Framework for Parallel and Distributed Processing in R Using Futures." *The R Journal* 13 (2): 208–27. https://doi.org/10.32614/RJ-2021-048.

———. 2022. *Future.apply: Apply Function to Elements in Parallel Using Futures.*

———. 2023. *Future: Unified Parallel and Distributed Processing in R for Everyone.*

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. "Quanteda: An R Package for the Quantitative Analysis of Textual Data." *Journal of Open Source Software* 3 (30): 774. https://doi.org/10.21105/joss.00774.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, Akitaka Matsuo, and William Lowe. 2022. *Quanteda: Quantitative Analysis of Textual Data.* https://quanteda.io.

Buchta, Christian, and Kurt Hornik. 2022. *ISOcodes: Selected Iso Codes.*

Csardi, Gabor, and Tamas Nepusz. 2006. "The Igraph Software Package for Complex Network Research." *InterJournal* Complex Systems: 1695. https://igraph.org.

Csárdi, Gábor, Kuba Podgórski, and Rich Geldreich. 2022. *Zip: Cross-Platform Zip Compression.* https://github.com/r-lib/zip#readme.

Dowle, Matt, and Arun Srinivasan. 2023. *Data.table: Extension of 'Data.frame'.*

Eddelbuettel, Dirk. 2023. *RcppTOML: Rcpp Bindings to Parser for "Tom's Obvious Markup Language".* http://dirk.eddelbuettel.com/code/rcpp.toml.html.

file., See AUTHORS. 2023. *Igraph: Network Analysis and Visualization.*

Gagolewski, Marek. 2022. "stringi: Fast and Portable Character String Processing in R." *Journal of Statistical Software* 103 (2): 1–59. https://doi.org/10.18637/jss.v103.i02.

Gagolewski, Marek, Bartek Tartanus, others; Unicode, Inc., and others. 2023. *Stringi: Fast and Portable Character String Processing Facilities.*

Landau, William Michael. 2021a. *Tarchetypes: Archetypes for Targets.*

———. 2021b. "The Targets R Package: A Dynamic Make-Like Function-Oriented Pipeline Toolkit for Reproducibility and High-Performance Computing." *Journal of Open Source Software* 6 (57): 2959. https://doi.org/10.21105/joss.02959.

———. 2023a. *Tarchetypes: Archetypes for Targets.*

———. 2023b. *Targets: Dynamic Function-Oriented Make-Like Declarative Pipelines*.

Ooms, Jeroen. 2023. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents*.

Ottolinger, Philipp. 2019. *Bib2df: Parse a Bibtex File to a Data Frame*. https://github.com/ropensci/bib2df.

Pedersen, Thomas Lin. 2022. *Ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*.

Ushey, Kevin. 2023. *Renv: Project Environments*. https://rstudio.github.io/renv/.

Wickham, Hadley. 2022. *Rvest: Easily Harvest (Scrape) Web Pages*.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2023. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook.

Zhu, Hao. 2021. *KableExtra: Construct Complex Table with Kable and Pipe Syntax*.