Corpus der Entscheidungen des Bundesgerichtshofs

(CE-BGH-Source)

Compilation Report

Version 2023-03-10

DOI: 10.5281/zenodo.7699033

Titel	Source Code des »Corpus der Entscheidungen des Bundesgerichtshofs«
Abkürzung	CE-BGH-Source
Autor	Seán Fobbe
Version	2023-03-10
Download	https://doi.org/10.5281/zenodo.7699033
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2023). Source Code des »Corpus der Entscheidungen des Bundesgerichtshofs« (CE-BGH-Source). Version 2023-03-10. Zenodo. DOI: 10.5281/zenodo.7699033.

Digital Object Identifier (DOI): Concept DOI und Version DOI

Soweit nicht anders angegeben ist die DOI immer eine »Version DOI« und bezieht sich nur auf eine bestimmte Version der Software. Sie verlinkt daher nur Version 2023-03-10. Für das Gesamtkonzept der Software steht eine »Concept DOI« zur Verfügung, die auf der Zenodo-Seite jeder Version unter »Cite all versions?« zu finden ist. Die »Concept DOI« verlinkt immer die aktuellste Version.

Lizenz: MIT No Attribution (MIT-0)

Disclaimer

Dieser Datensatz ist eine private wissenschaftliche Initiative und steht in keiner Verbindung zu Behörden, Gerichten oder anderen amtlichen Stellen der Bundesrepublik Deutschland.

Inhaltsverzeichnis

1 Corpus der Entscheidungen des Bundesgerichtshofs (CE-BGH)

1.1 Überblick

Das **Corpus der Entscheidungen des Bundesgerichtshofs (CE-BGH)** ist eine möglichst vollständige Sammlung der vom Bundesgerichtshof veröffentlichten Entscheidungen. Der Datensatz nutzt als seine Datenquelle die amtliche Entscheidungsdatenbank des Bundesgerichtshofs und wertet diese vollständig aus.

Alle mit diesem Skript erstellten Datensätze werden dauerhaft kostenlos und urheberrechtsfrei auf Zenodo, dem wissenschaftlichen Archiv des CERN, veröffentlicht. Alle Versionen sind mit einem separaten und langzeit-stabilen (persistenten) Digital Object Identifier (DOI) versehen.

Aktuellster, funktionaler und zitierfähiger Release des Datensatzes: https://doi.org/10.5281/zenodo.3942742

1.2 Funktionsweise

Primäre Endprodukte des Skripts sind folgende ZIP-Archive:

- Der volle Datensatz im CSV-Format (mit zusätzlichen Metadaten)
- Die reinen Metadaten im CSV-Format (wie unter 1, nur ohne Entscheidungsinhalte)
- Alle Entscheidungen im TXT-Format
- Alle Entscheidungen im PDF-Format
- Nur Leitsatz-Entscheidungen im PDF-Format
- Nur benannte Entscheidungen im PDF-Format
- Platzhalter-Dokumente im PDF-Format
- Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)

Alle Ergebnisse werden im Ordner `output` abgelegt. Zusätzlich werden für alle ZIP-Archive kryptographische Signaturen (SHA2-256 und SHA3-512) berechnet und in einer CSV-Datei hinterlegt.

1.3 Systemanforderungen

- Docker
- Docker Compose
- 8 GB Speicherplatz auf Festplatte
- Multi-core CPU empfohlen (8 cores/16 threads für die Referenzdatensätze).

In der Standard-Einstellung wird das Skript vollautomatisch die maximale Anzahl an Rechenkernen/Threads auf dem System zu nutzen. Die Anzahl der verwendeten Kerne kann in der Konfigurationsatei angepasst werden. Wenn die Anzahl Threads auf 1 gesetzt wird, ist die Parallelisierung deaktiviert.

1.4 Anleitung

1.4.1 Schritt 1: Ordner vorbereiten

Kopieren Sie bitte den gesamten Source Code in einen leeren Ordner (!), beispielsweise mit:

```
$ git clone https://github.com/seanfobbe/ce-bgh
```

Verwenden Sie immer einen separaten und *leeren* Ordner für die Kompilierung. Die Skripte löschen innerhalb von bestimmten Unterordnern (`files/`, `temp/`, `analysis` und `output/`) alle Dateien die den Datensatz verunreinigen könnten — aber auch nur dort.

1.4.2 Schritt 2: Docker Image erstellen

Ein Docker Image stellt ein komplettes Betriebssystem mit der gesamten verwendeten Software automatisch zusammen. Nutzen Sie zur Erstellung des Images einfach:

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

1.4.3 Schritt 3: Datensatz kompilieren

Falls Sie zuvor den Datensatz schon einmal kompiliert haben (ob erfolgreich oder erfolglos), können Sie mit folgendem Befehl alle Arbeitsdaten im Ordner löschen:

```
$ Rscript delete_all_data.R
```

Den vollständigen Datensatz kompilieren Sie mit folgendem Skript:

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

1.4.4 Ergebnis

Der Datensatz und alle weiteren Ergebnisse sind nun im Ordner `output/` abgelegt.

1.5 Pipeline visualisieren

Sie können die Pipeline visualisieren, aber nur nachdem sie die zentrale .Rmd-Datei mindestens einmal gerendert haben:

```
> targets::tar_glimpse()     # Nur Datenobjekte
> targets::tar_visnetwork()  # Alle Objekte
```

1.6 Troubleshooting

Hilfreiche Befehle um Fehler zu lokalisieren und zu beheben.

```
> tar_progress()  # Zeigt Fortschritt und Fehler an
> tar_meta()      # Alle Metadaten
> tar_meta(fields = "warnings", complete_only = TRUE)  # Warnungen
> tar_meta(fields = "error", complete_only = TRUE)  # Fehlermeldungen
> tar_meta(fields = "seconds")  # Laufzeit der Targets
```

1.7 Projektstruktur

Die folgende Struktur erläutert die wichtigsten Bestandteile des Projekts. Während der Kompilierung werden weitere Ordner erstellt (`files/`, `temp/` `analysis` und `output/`). Die Endergebnisse werden alle in `output/` abgelegt.

```
.
├ buttons                    # Buttons (nur optische Bedeutung)
├ CHANGELOG.md               # Alle Änderungen
├ compose.yaml               # Konfiguration für Docker
├ config.toml                # Zentrale Konfigurations-Datei
├ data                       # Datensätze, auf denen die Pipeline aufbaut
├ delete_all_data.R          # Löscht den Datensatz und Zwischenschritte
├ docker-build-image.sh      # Docker Image erstellen
├ Dockerfile                 # Definition des Docker Images
├ docker-run-project.sh      # Docker Image und Datensatz kompilieren
├ functions                  # Wichtige Schritte der Pipeline
├ gpg                        # Persönlicher Public GPG-Key für Seán Fobbe
├ old                        # Alter Code aus früheren Versionen
├ pipeline.Rmd               # Zentrale Definition der Pipeline
├ README.md                  # Bedienungsanleitung
├ reports                    # Markdown-Dateien
├ requirements-python.txt    # Benötigte Python packages
├ requirements-R.R           # Benötigte R packages
├ requirements-system.txt    # Benötigte system dependencies
├ run_project.R              # Kompiliert den gesamten Datensatz
└ tex                       # LaTeX-Templates
```

1.8 Weitere Open Access Veröffentlichungen (Fobbe)

Website — https://www.seanfobbe.de

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

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

Volltexte regulärer Publikationen — https://zenodo.org/communities/sean-fobbe-publications/

1.9 Kontakt

Fehler gefunden? Anregungen? Kommentieren Sie gerne im Issue Tracker auf GitHub oder schreiben Sie mir eine E-Mail an fobbe-data@posteo.de

2 Packages laden

```
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 Vorbereitung

3.1 Definitionen

```
## Datum
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2023-03-10"

## Datum und Uhrzeit (Beginn)
begin.script <- Sys.time()

## Konfiguration
config <- RcppTOML::parseTOML("config.toml")
print(config)
#> List of 9
#>  $ cores   :List of 2
#>   ..$ max   : logi TRUE
#>   ..$ number: int 8
#>  $ debug   :List of 4
#>   ..$ cleanrun: logi FALSE
#>   ..$ pages   : int 50
#>   ..$ sample  : int 500
#>   ..$ toggle  : logi FALSE
#>  $ doi     :List of 4
#>   ..$ aktenzeichen : chr "10.5281/zenodo.4569564"
#>   ..$ data         :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.3942742"
#>   .. ..$ version: chr "10.5281/zenodo.7699032"
#>   ..$ personendaten: chr "10.5281/zenodo.4568682"
#>   ..$ software     :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.4705864"
#>   .. ..$ version: chr "10.5281/zenodo.7699033"
#>  $ download:List of 1
#>   ..$ timeout: int 300
#>  $ fig     :List of 3
#>   ..$ align : chr "center"
#>   ..$ dpi   : int 300
#>   ..$ format: chr [1:2] "pdf" "png"
#>  $ license :List of 2
#>   ..$ code: chr "MIT-0"
#>   ..$ data: chr "Creative Commons Zero 1.0 Universal"
#>  $ parallel:List of 4
#>   ..$ citations    : logi TRUE
#>   ..$ extractPDF   : logi TRUE
#>   ..$ lingsummarize: logi TRUE
#>   ..$ multihashes  : logi TRUE
#>  $ project :List of 3
#>   ..$ author   : chr "Seán Fobbe"
#>   ..$ fullname : chr "Corpus der Entscheidungen des Bundesgerichtshofs"
#>   ..$ shortname: chr "CE-BGH"
#>  $ quanteda:List of 1
#>   ..$ tokens_locale: chr "de_DE"
```

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

3.2 Aufräumen

Löscht Dateien im Output-Ordner, die nicht aktuell genug sind.

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


## files.html <- list.files("files/html", full.names = TRUE)


## if(length(files.html) > 0){

##     delete <- sort(files.html, decreasing = TRUE)[1:10]
##     unlink(delete)

## }
```

3.3 Ordner erstellen

```
#unlink("output", recursive = TRUE)
dir.create("files", showWarnings = FALSE)
dir.create("output", showWarnings = FALSE)
dir.create("temp", showWarnings = FALSE)

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

3.4 Vollzitate statistischer Software schreiben

```
knitr::write_bib(renv::dependencies()$Package,
                 "temp/packages.bib")
#> Finding R package dependencies ... Done!
#> Warning in knitr::write_bib(renv::dependencies()$Package, "temp/packages.bib")
    :
#> package(s) magick not found
```

4 Globale Variablen

4.1 Packages definieren

```
tar_option_set(packages = c("tarchetypes",
                            "RcppTOML",     # TOML-Dateien lesen und schreiben
                            "testthat",     # Unit Tests
                            "fs",           # Verbessertes File Handling
                            "zip",          # Verbessertes ZIP Handling
                            "mgsub",        # Vektorisiertes Gsub
                            "httr",         # HTTP-Werkzeuge
                            "rvest",        # HTML/XML-Extraktion
                            "knitr",        # Professionelles Reporting
                            "kableExtra",   # Verbesserte Kable Tabellen
                            "pdftools",     # Verarbeitung von PDF-Dateien
                            "ggplot2",      # Datenvisualisierung
                            "ggraph",       # Visualisierung von Graphen
                            "scales",       # Skalierung von Diagrammen
                            "data.table",   # Fortgeschrittene Datenverarbeitung
                            "readtext",     # TXT-Dateien einlesen
                            "quanteda",     # Computerlinguistik
                            "future",       # Parallelisierung
                            "future.apply"))# Funktionen für Future

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 Konfiguration

```
datestamp <- Sys.Date()

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

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

## Caption for diagrams
caption <- paste("Fobbe | 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 <- future::availableCores() - 1
}


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

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

4.3 Funktionen definieren

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

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

4.4 Metadaten für TXT-Dateien definieren

```
docvarnames <- c("gericht",
                 "spruchkoerper_db",
                 "leitsatz",
                 "datum",
                 "spruchkoerper_az",
                 "registerzeichen",
                 "eingangsnummer",
                 "eingangsjahr_az",
                 "zusatz_az",
                 "name",
                 "kollision")

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

4.5 ZIP-Datei für Source definieren

```
files.source.raw <-  c(list.files(pattern = "\\.R$|\\.toml$|\\.R?md$|\\.yaml|\\.
    sh|\\.txt"),
                       "Dockerfile",
                       "reports",
                       "data",
                       "functions",
                       "tex",
                       "gpg",
                       "buttons")

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

5 Pipeline: Konstruktion

5.1 File Tracking Targets

Mit diesem Abschnitt der Pipeline werden Input-Dateien getrackt und eingelesen. Mit der Option »format = "file"« werden für Input-Dateien Prüfsummen berechnet. Falls sich diese verändern werden alle von ihnen abhängigen Pipeline-Schritte als veraltet markiert und neu berechnet.

5.1.1 Variablen

Die Variablen des Datensatzes, inklusive ihrer Erläuterung.

```
list(
    tar_target(file.variables.codebook,
               "data/CE-BGH_Variables.csv",
               format = "file"),
    tar_target(variables.codebook,
               fread(file.variables.codebook))
)
#> Establish _targets.R and _targets_r/targets/tar.file.var.R.
```

5.1.2 Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD)

Die Tabelle der Registerzeichen und der ihnen zugeordneten Verfahrensarten stammt aus dem folgenden Datensatz: »Seán Fobbe (2021). Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD). Version 1.0.1. Zenodo. DOI: 10.5281/zenodo.4569564.«

‘

```
list(
    tar_target(file.az.brd,
           "data/AZ-BRD_1-0-1_DE_Registerzeichen_Datensatz.csv",
           format = "file"),
     tar_target(az.brd,
                fread(file.az.brd))
     )
#> Establish _targets.R and _targets_r/targets/tar.file.az.R.
```

5.1.3 Presidents and Vice-Presidents of the Federal Courts of Germany (PVP-FCG)

Die Personendaten stammen aus folgendem Datensatz: »Seán Fobbe and Tilko Swalve (2021). Presidents and Vice-Presidents of the Federal Courts of Germany (PVP-FCG). Version 2021-04-08. Zenodo. DOI: 10.5281/zenodo.4568682«.

```
list(
    tar_target(file.presidents,
           "data/PVP-FCG_2021-04-08_GermanFederalCourts_Presidents.csv",
```

```
            format = "file"),
     tar_target(presidents,
                fread(file.presidents))
     )
#> Establish _targets.R and _targets_r/targets/tar.file.presi.R.
```

```
list(
    tar_target(file.vpresidents,
           "data/PVP-FCG_2021-04-08_GermanFederalCourts_VicePresidents.csv",
           format = "file"),
     tar_target(vpresidents,
                fread(file.vpresidents))
     )
#> Establish _targets.R and _targets_r/targets/tar.file.vpresi.R.
```

5.1.4 Source Code

Dies sind alle Dateien, die den Source Code für den Datensatz bereitstellen.

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

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

5.1.5 Changelog

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

5.2 Download Targets

5.2.1 Maximalen Umfang der Datenbankabfrage bestimmen

```
tarchetypes::tar_age(dt.scope,
                     f.scope(debug.toggle = config$debug$toggle,
                             debug.pages = config$debug$pages),
                     age = as.difftime(1, units = "days"))

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

5.2.2 Datenbankseiten als HTML abrufen

```
tar_target(files.html,
                f.download(url = dt.scope$url,
                           filename = dt.scope$filename,
                           dir = "files/html",
                           sleep.min = 0.2,
                           sleep.max = 0.5,
                           retries = 3,
                           retry.sleep.min = 2,
                           retry.sleep.max = 5,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE,
                           debug.files = 500),
                format = "file")

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

5.2.3 Vorläufige Download-Tabelle erstellen

Abfrage der Datenbank des BGH und erstellen der vorläufigen Download-Tabelle.

```
tar_target(dt.download,
           f.download_table_make(files.html = files.html))

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

5.2.4 Download-Tabelle bereinigen

Die Aktenzeichen und Spruchkörper werden bereinigt und gespeichert.

```
list(
    tar_target(az_clean,
               f.clean_az_bgh(dt.download$az)),
    tar_target(spruch_clean,
              f.clean_spruch_bgh(dt.download$spruch))
    )


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

5.2.5 Finale Download-Tabelle erstellen

Die bereinigten Variablen werden hier mit der vorläufigen Download-Tabelle zusammengeführt, weitere Bereinigungen durchgeführt, weitere Variablen aus den Bemerkungen extrahiert und schließlich auf ihre Konformität mit der Beschreibung im Codebook getestet.

```
tar_target(dt.download.final,
           f.download_table_finalize(x = dt.download,
                                     az = az_clean,
                                     spruch = spruch_clean))
#> Establish _targets.R and _targets_r/targets/tar.download.final.R.
```

5.2.6 Download durchführen

```
tar_target(files.pdf,
                f.download(url = dt.download.final$url,
                           filename = dt.download.final$doc_id,
                           dir = "files/pdf",
                           sleep.min = 0,
                           sleep.max = 0.1,
                           retries = 3,
                           retry.sleep.min = 2,
                           retry.sleep.max = 5,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE,
                           debug.files = 500),
                format = "file")

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

5.3 Convert Targets

Durch diesen Abschnitt der Pipeline werden die PDF-Dateien in TXT konvertiert und mitsamt den Variablen in ihren Dateinamen eingelesen. Beim Einlesen werden die in PDF-Dateien üblichen über Zeilen gebrochene Wörter wieder zusammengefügt.

```
list(tar_target(files.txt,
                f.tar_pdf_extract(x = files.pdf,
                                  outputdir = "files/txt",
                                  multicore = config$parallel$extractPDF,
                                  cores = fullCores),
                format = "file"),
     tar_target(dt.bgh,
                f.readtext(x = files.txt,
                           docvarnames = docvarnames))
     )

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

5.4 Enhance Targets

Dieser Abschnitt der Pipeline berechnet diverse Verbesserungen für den Datensatz und führt diese am Ende zusammen.

5.4.1 Daten standardisieren

Das Datum wird im ISO-Format standardisiert und die Variablen »entscheidungsjahr« und »eingangsjahr_iso« hinzugefügt.

```
tar_target(dt.bgh.datecleaned,
                f.clean_dates(dt.bgh))
#> Establish _targets.R and _targets_r/targets/tar.enhance.dates.R.
```

5.4.2 Variable erstellen: »verfahrensart«

Die Variable »verfahrensart« wird aus den Registerzeichen berechnet.

```
tar_target(var_verfahrensart,
                f.var_verfahrensart(dt.bgh.datecleaned$registerzeichen,
                                    az.brd = az.brd,
                                    gericht = "BGH"))
#> Establish _targets.R and _targets_r/targets/tar.enhance.verfahrensart.R.
```

5.4.3 Variable erstellen: »aktenzeichen«

Das Aktenzeichen wird aus seinen Komponenten berechnet.

```
tar_target(var_aktenzeichen,
                f.var_aktenzeichen(x = dt.bgh.datecleaned,
                                   az.brd = az.brd,
                                   gericht = "BGH"))
#> Establish _targets.R and _targets_r/targets/tar.enhance.az.R.
```

5.4.4 Variable erstellen: »ecli«

Die ECLI wird aus ihren Komponenten berechnet.

```
tar_target(var_ecli,
           f.var_ecli_bgh(x = dt.bgh.datecleaned,
                          az.brd = az.brd))
#> Establish _targets.R and _targets_r/targets/tar.enhance.ecli.R.
```

5.4.5 Variable erstellen: »praesi«

```
tar_target(var_praesi,
           f.presidents(datum = dt.bgh.datecleaned$datum,
                        gericht = "BGH",
                        pvp.fcg = presidents))
#> Establish _targets.R and _targets_r/targets/tar.enhance.praesi.R.
```

5.4.6 Variable erstellen: »vpraesi«

```
tar_target(var_vpraesi,
           f.presidents(datum = dt.bgh.datecleaned$datum,
                        gericht = "BGH",
                        pvp.fcg = vpresidents))
#> Establish _targets.R and _targets_r/targets/tar.enhance.vpraesi.R.
```

5.4.7 Variable erstellen: »entscheidung_typ«

Der Typ der Entscheidung wird aus dem Text extrahiert.

```
tar_target(var_entscheidung_typ,
                f.var_entscheidung_typ(dt.bgh.datecleaned$text))
#> Establish _targets.R and _targets_r/targets/tar.enhance.entschtyp.R.
```

5.4.8 Variablen erstellen: »bghst, bghr, nachschlagewerk«

```
tar_target(var_sammlungen,
                f.var_sammlungen(dt.bgh.datecleaned$text))

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

5.4.9 Variablen erstellen: »zeichen, token, typen, saetze«

Berechnung klassischer linguistischer Kennzahlen.

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

5.4.10 Konstanten erstellen

Konstanten die dem Datensatz wichtige Herkunftsinformationen hinzufügen. Darunter sind die Versionsnummer, die Version DOI, die Concept DOI und die Lizenz.

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

5.4.11 Zusätzliche Variablen zusammenführen

```
tar_target(vars_additional,
           data.table(verfahrensart = var_verfahrensart,
                      aktenzeichen = var_aktenzeichen,
                      ecli = var_ecli,
                      praesi = var_praesi,
                      v_praesi = var_vpraesi,
                      entscheidung_typ = var_entscheidung_typ,
                      var_lingstats,
                      var_sammlungen,
                      var_constants))

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

5.4.12 Finalen Datensatz erstellen

Die Verbesserungen der vorherigen Schritte werden in dieser Funktion zusammengefügt um den finalen Datenatz herzustellen.

```
tar_target(dt.bgh.final,
           f.finalize(x = dt.bgh.datecleaned,
                      download.table = dt.download.final,
                      vars.additional = vars_additional,
                      varnames = variables.codebook$varname))

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

5.4.13 Variante erstellen: Nur Metadaten

Hier wird die Text-Variable entfernt, um eine deutlich platzsparendere Variante des Datensatzes zu erstellen. Enthalten sind nur noch die Metadaten.

```
tar_target(dt.bgh.meta,
                dt.bgh.final[, !"text"])

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

5.5 Zitations-Analyse

```
tar_target(igraph.citations.raw,
           f.citation_network(dt.bgh.final = dt.bgh.final,
                              az.brd = az.brd,
                              multicore = config$parallel$citations,
                              cores = fullCores))

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

```
tar_target(igraph.citations.cleaned,
           f.clean_graph(g = igraph.citations.raw,
                         az.brd = az.brd))

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

5.6 Write Targets

Dieser Abschnitt der Pipeline schreibt den Datensatz und alle Hash-Prüfsummen auf die Festplatte.

5.6.1 CSV schreiben: Voller Datensatz

```
tar_target(csv.final,
           f.tar_fwrite(x = dt.bgh.final,
                        filename = file.path("output",
                                             paste0(prefix.files,
                                                    "_DE_CSV_Datensatz.csv"))
                        )
           )
#> Establish _targets.R and _targets_r/targets/tar.write.final.R.
```

5.6.2 CSV schreiben: Metadaten

```
tar_target(csv.meta,
           f.tar_fwrite(x = dt.bgh.meta,
                        filename = file.path("output",
                                             paste0(prefix.files,
                                                    "_DE_CSV_Metadaten.csv"))
                        )
           )
#> Establish _targets.R and _targets_r/targets/tar.write.meta.R.
```

5.6.3 GraphML schreiben: Aktenzeichen-Netzwerke

```
tar_target(graphml.citations.az,
           f.tar_write_graph(graph = igraph.citations.cleaned,
                             file = file.path("output",
                                              paste0(prefix.files,
                                                     "_DE_Zitationsnetzwerk-AZ.
    graphml")),
                             format = "graphml"),
           format = "file")

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

5.7 Report Targets

Dieser Abschnitt der Pipeline erstellt die finalen Berichte (Codebook und Robustness Checks).

5.7.1 LaTeX-Definitionen schreiben

Um Variablen aus der Pipeline in die LaTeX-Kompilierung einzuführen, müssen diese als .tex-Datei auf die Festplatte geschrieben werden.

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

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

5.7.2 Zusammenfassungen linguistischer Kennwerte berechnen

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

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

5.7.3 Report erstellen: Robustness Checks

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

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

5.7.4 Report erstellen: 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.
```

5.8 ZIP Targets

Diese Abschnitt der Pipeline erstellt ZIP-Archive für alle zentralen Rechenergebnisse und speichert diese im Ordner »output«.

5.8.1 ZIP erstellen: Analyse-Dateien

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

5.8.2 ZIP erstellen: Source Code

```
tar_target(zip.source,
               f.tar_zip(files.source,
                         filename = paste0(prefix.files,
                                           "_Source_Code.zip"),
                         dir = "output",
                         mode = "mirror"),
               format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.source.R.
```

5.8.3 ZIP erstellen: PDF-Dateien (alle Entscheidungen)

Hinweis: die verpackten PDF-Dateien werden auf die im finalen Datensatz enthaltenen Dokumente beschränkt (Bereinigung um Platzhalter-Dokumente).

```
tar_target(zip.pdf.all,
           f.tar_zip(x = intersect(files.pdf,
                                   file.path("files", "pdf",
                                             gsub("\\.txt",
                                                  "\\.pdf",
                                                  dt.bgh.final$doc_id))),
```

```
                      filename = paste(prefix.files,
                                       "DE_PDF_Datensatz.zip",
                                       sep = "_"),
                      dir = "output",
                      mode = "cherry-pick"),
           format = "file")

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

5.8.4 ZIP erstellen: PDF-Dateien (nur Leitsatzentscheidungen)

```
tar_target(zip.pdf.leitsatz,
           f.tar_zip(x = intersect(files.pdf,
                                   file.path("files", "pdf",
                                             gsub("\\.txt",
                                                  "\\.pdf",
                                                  dt.bgh.final[leitsatz == "LE"]$
    doc_id))),
                      filename = paste(prefix.files,
                                       "DE_PDF_Leitsatz-Entscheidungen.zip",
                                       sep = "_"),
                      dir = "output",
                      mode = "cherry-pick"),
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.pdf.leit.R.
```

5.8.5 ZIP erstellen: PDF-Dateien (nur benannte Entscheidungen)

```
tar_target(zip.pdf.benannt,
                f.tar_zip(x = intersect(files.pdf,
                                        file.path("files", "pdf",
                                                  gsub("\\.txt",
                                                       "\\.pdf",
                                                       dt.bgh.final[is.na(name) ==
    FALSE]$doc_id))),
                                  filename = paste(prefix.files,
                                                   "DE_PDF_Entscheidungen-mit-Namen.
    zip",
                                                   sep = "_"),
                                  dir = "output",
                                  mode = "cherry-pick"),
                format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.pdf.named.R.
```

5.8.6 ZIP erstellen: PDF-Dateien (Platzhalter)

```
tar_target(zip.pdf.platzhalter,
           f.tar_zip(x = setdiff(files.pdf,
                                 file.path("files", "pdf",
```

```
                                          gsub("\\.txt",
                                               "\\.pdf",
                                               dt.bgh.final$doc_id))),
                      filename = paste(prefix.files,
                                       "DE_PDF_Platzhalter.zip",
                                       sep = "_"),
                      dir = "output",
                      mode = "cherry-pick"),
            format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.pdf.placeholder.R.
```

5.8.7 ZIP erstellen: TXT-Dateien

```
tar_target(zip.txt,
           f.tar_zip(x = intersect(files.txt, file.path("files", "txt", dt.bgh.
    final$doc_id)),
                      filename = paste(prefix.files,
                                       "DE_TXT_Datensatz.zip",
                                       sep = "_"),
                      dir = "output",
                      mode = "cherry-pick"),
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.txt.R.
```

5.8.8 ZIP erstellen: CSV-Datei (voller Datensatz)

```
tar_target(zip.csv.final,
                f.tar_zip(csv.final,
                                 filename = gsub("\\.csv", "\\.zip", basename(csv.
    final)),
                                 dir = "output",
                                 mode = "cherry-pick"),
                f.tar_zip(csv.final),
#> Establish _targets.R and _targets_r/targets/tar.zip.csv.full.R.
```

5.8.9 ZIP erstellen: CSV-Datei (nur Metadaten)

```
tar_target(zip.csv.meta,
                f.tar_zip(csv.meta,
                                 filename = gsub("\\.csv", "\\.zip", basename(csv.
    meta)),
                                 dir = "output",
                                 mode = "cherry-pick"),
                f.tar_zip(csv.meta),
#> Establish _targets.R and _targets_r/targets/tar.zip.csv.meta.R.
```

5.8.10 ZIP erstellen: GraphML

```
tar_target(zip.citations.az,
               f.tar_zip(graphml.citations.az,
                             filename = paste0(prefix.files,
                                               "_DE_GraphML_Netzwerke.zip"),
                             dir = "output",
                             mode = "cherry-pick"),
               format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.graphml.R.
```

5.9 Kryptographische Hashes

5.9.1 Zu hashende ZIP-Archive definieren

```
tar_target(zip.all,
           c(zip.pdf.all,
             zip.pdf.leitsatz,
             zip.pdf.benannt,
             zip.pdf.platzhalter,
             zip.txt,
             zip.citations.az,
             zip.csv.final,
             zip.csv.meta,
             zip.analysis,
             zip.source))
#> Establish _targets.R and _targets_r/targets/tar.hashes.all.R.
```

5.9.2 Kryptographische Hashes berechnen

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

5.9.3 CSV schreiben: Kryptographische Hashes

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

6 Pipeline: Kompilierung

6.1 Durchführen der Kompilierung

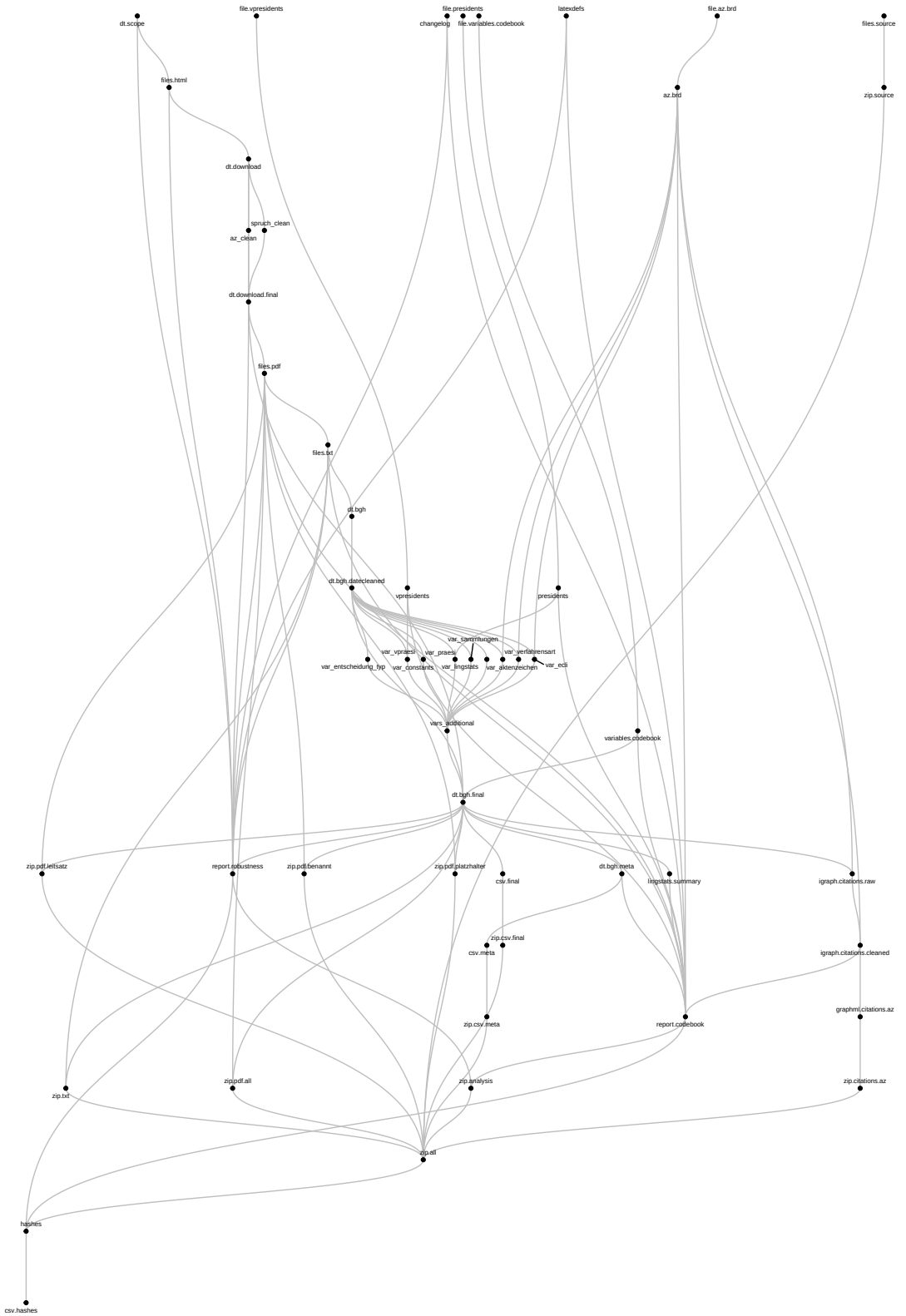
```
tar_make()
```

6.2 Visualisierung

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

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


ggraph(g,
       'sugiyama') +
    geom_edge_diagonal(colour = "grey")+
    geom_node_point()+
    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.
```


file.vpresidents
file.presidents
latexdefs
file.az.brd
dt.scope
changelog
file.variables.codebook
files.source
files.html
az.brd
zip.source
dt.download
spruch_clean
az_clean
dt.download.final
files.pdf
files.txt
dt.bgh
dt.bgh.datecleaned
vpresidents
presidents
var_sammlungen
var_vpraesi
var_praesi
var_verfahrensart
var_entscheidung_typ
var_constants
var_lingstats
var_aktenzeichen
var_ecli
vars_additional
variables.codebook
dt.bgh.final
zip.pdf.teilsatz
report.robustness
zip.pdf.benannt
zip.pdf.platzhalter
csv.final
dt.bgh.meta
lingstats.summary
igraph.citations.raw
zip.csv.final
csv.meta
igraph.citations.cleaned
graphml.citations.az
zip.csv.meta
report.codebook
zip.pdf.all
zip.analysis
zip.citations.az
zip.txt
zip.all
hashes
csv.hashes

7 Pipeline: Analyse

7.1 Gesamte Liste

Die vollständige Liste aller Targets, inklusive ihres Types und ihrer Größe. Targets die auf Dateien verweisen (z.B. alle PDF-Dateien) geben die Gesamtgröße der Dateien auf der Festplatte an.

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

# Gesamter Speicherplatzverbrauch
sum(meta$MB, na.rm = TRUE)
#> [1] 18698.54

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

name	type	bytes	format	MB
az.brd	stem	5509	qs	0.01
az_clean	stem	266477	qs	0.27
changelog	stem	4127	file	0.00
csv.final	stem	85	qs	0.00
csv.hashes	stem	91	qs	0.00
csv.meta	stem	85	qs	0.00
dt.bgh	stem	220737432	qs	220.74
dt.bgh.datecleaned	stem	227867438	qs	227.87
dt.bgh.final	stem	222678753	qs	222.68
dt.bgh.meta	stem	2654938	qs	2.65
dt.download	stem	1051554	qs	1.05
dt.download.final	stem	1761302	qs	1.76
dt.scope	stem	6496	qs	0.01
file.az.brd	stem	32883	file	0.03
file.presidents	stem	7249	file	0.01
file.variables.codebook	stem	9460	file	0.01

(continued)

name	type	bytes	format	MB
file.vpresidents	stem	9193	file	0.01
files.html	stem	116055359	file	116.06
files.pdf	stem	6992526543	file	6992.53
files.source	stem	1013459	file	1.01
files.txt	stem	871550087	file	871.55
graphml.citations.az	stem	40863555	file	40.86
hashes	stem	1793	qs	0.00
igraph.citations.cleaned	stem	1855849	qs	1.86
igraph.citations.raw	stem	2425927	qs	2.43
latexdefs	stem	1271	file	0.00
lingstats.summary	stem	390	qs	0.00
presidents	stem	1959	qs	0.00
report.codebook	stem	6662617	file	6.66
report.robustness	stem	419031	file	0.42
spruch_clean	stem	69194	qs	0.07
var_aktenzeichen	stem	251657	qs	0.25
var_constants	stem	29871	qs	0.03
var_ecli	stem	396710	qs	0.40
var_entscheidung_typ	stem	15603	qs	0.02
var_lingstats	stem	438079	qs	0.44
var_praesi	stem	187	qs	0.00
var_sammlungen	stem	28130	qs	0.03
var_verfahrensart	stem	119248	qs	0.12
var_vpraesi	stem	207	qs	0.00
variables.codebook	stem	3502	qs	0.00
vars_additional	stem	1256457	qs	1.26
vpresidents	stem	2451	qs	0.00
zip.all	stem	221	qs	0.00
zip.analysis	stem	10713273	file	10.71

(continued)

name	type	bytes	format	MB
zip.citations.az	stem	1962601	file	1.96
zip.csv.final	stem	235170585	file	235.17
zip.csv.meta	stem	3262829	file	3.26
zip.pdf.all	stem	6400976911	file	6400.98
zip.pdf.benannt	stem	614092986	file	614.09
zip.pdf.leitsatz	stem	2395428223	file	2395.43
zip.pdf.platzhalter	stem	1695854	file	1.70
zip.source	stem	264483	file	0.26
zip.txt	stem	321875152	file	321.88

7.2 Timing

7.2.1 Gesamte Laufzeit

```
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)

## Sekunden
print(runtime.sum)
#> [1] 32776.91

## Minuten
runtime.sum / 60
#> [1] 546.2818

## Stunden
runtime.sum / 3600
#> [1] 9.104697
```

7.2.2 Laufzeit einzelner Targets

Der Zeitpunkt an dem die Targets berechnet wurden und ihre jeweilige Laufzeit in Sekunden.

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

name	time	seconds	mins
files.pdf	2023-03-10 09:57:27	29034.058	483.90
files.html	2023-03-10 01:51:30	1630.645	27.18
igraph.citations.raw	2023-03-10 10:20:23	340.428	5.67
var_lingstats	2023-03-10 10:14:04	324.504	5.41
var_ecli	2023-03-10 10:08:39	232.697	3.88
files.txt	2023-03-10 09:59:11	203.701	3.40
lingstats.summary	2023-03-10 10:23:37	193.249	3.22
zip.pdf.all	2023-03-10 10:27:54	187.063	3.12
dt.bgh	2023-03-10 10:03:34	143.776	2.40
dt.download	2023-03-10 01:53:02	91.884	1.53

(continued)

name	time	seconds	mins
zip.pdf.leitsatz	2023-03-10 10:24:47	69.840	1.16
var_aktenzeichen	2023-03-10 10:04:47	68.966	1.15
zip.txt	2023-03-10 10:28:52	57.067	0.95
zip.csv.final	2023-03-10 10:29:46	51.327	0.86
report.codebook	2023-03-10 10:30:20	33.034	0.55
spruch_clean	2023-03-10 01:53:39	31.641	0.53
hashes	2023-03-10 10:30:49	28.135	0.47
zip.pdf.benannt	2023-03-10 10:14:42	18.538	0.31
report.robustness	2023-03-10 10:14:23	11.904	0.20
dt.scope	2023-03-10 01:24:19	6.732	0.11
az_clean	2023-03-10 01:53:07	4.653	0.08
var_sammlungen	2023-03-10 10:14:07	3.032	0.05
igraph.citations.cleaned	2023-03-10 10:28:55	2.025	0.03
files.source	2023-03-10 15:48:27	1.342	0.02
file.az.brd	2023-03-09 18:00:48	1.274	0.02
dt.download.final	2023-03-10 01:53:40	1.117	0.02
zip.citations.az	2023-03-10 10:30:21	0.931	0.02
zip.csv.meta	2023-03-10 10:29:47	0.575	0.01
dt.bgh.final	2023-03-10 10:14:11	0.531	0.01
zip.pdf.platzhalter	2023-03-10 10:20:23	0.452	0.01
graphml.citations.az	2023-03-10 10:30:20	0.355	0.01
zip.analysis	2023-03-10 10:30:20	0.338	0.01
csv.final	2023-03-10 10:20:23	0.320	0.01
var_entscheidung_typ	2023-03-10 10:03:38	0.232	0.00
var_vpraesi	2023-03-10 10:14:07	0.224	0.00
var_praesi	2023-03-10 10:14:04	0.148	0.00
dt.bgh.datecleaned	2023-03-10 10:03:37	0.038	0.00
var_constants	2023-03-10 10:03:38	0.032	0.00
zip.source	2023-03-10 15:48:51	0.031	0.00

(continued)

name	time	seconds	mins
csv.meta	2023-03-10 10:28:52	0.028	0.00
latexdefs	2023-03-10 01:24:13	0.016	0.00
dt.bgh.meta	2023-03-10 10:14:12	0.012	0.00
var_verfahrensart	2023-03-10 10:03:38	0.005	0.00
az.brd	2023-03-10 01:24:19	0.002	0.00
csv.hashes	2023-03-10 10:30:49	0.002	0.00
vars_additional	2023-03-10 10:14:07	0.002	0.00
presidents	2023-03-10 01:51:30	0.001	0.00
variables.codebook	2023-03-10 01:51:30	0.001	0.00
vpresidents	2023-03-10 01:24:19	0.001	0.00
changelog	2023-03-09 17:48:12	0.000	0.00
file.presidents	2022-07-16 22:40:26	0.000	0.00
file.variables.codebook	2023-03-09 17:48:12	0.000	0.00
file.vpresidents	2022-07-16 22:40:26	0.000	0.00
zip.all	2023-03-10 15:48:52	0.000	0.00

7.3 Warnungen

```
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.")

}
```

7.3.1 files.pdf

URL httpsjuris.bundesgerichtshof.decgibinrechtsprechungdocument.pyGerichtbghArtenDatum2009Sort1Seite1

status was Stream error in the HTTP2 framing layer. URL httpsjuris.bundesgerichtshof.decgibinrechtsprechung

status was Stream error in the HTTP2 framing layer. URL httpsjuris.bundesgerichtshof.decgibinrechtsprechung

status was Stream error in the HTTP2 framing layer. URL httpsjuris.bundesgerichtshof.decgibinrechtsprechung

status was Stream error in the HTTP2 framing layer. URL httpsjuris.bundesgerichtshof.decgibinrechtsprechung

status was Stream error in the HTTP2 framing layer. URL httpsjuris.bundesgerichtshof.decgibinrechtsprechung

status was Stream error in the HTTP2 framing layer

7.3.2 igraph.citations.cleaned

Warnung 3 Nodes entfernt, weil Registerzeichen fehlerhaft.

7.3.3 report.codebook

The file CEBGH_20230310_Codebook.log is not encoded in UTF8. These lines contain invalid UTF8 characters 1135. LaTeX Warning Label variablen multiply defined.. Package microtype Warning Unable to apply patch footnote on input line 193.. LaTeX Warning There were multiplydefined labels.

7.3.4 report.robustness

The file CEBGH_20230310_RobustnessChecks.log is not encoded in UTF8. These lines contain invalid UTF8 characters 1187, 1227, 1237, 1267, 1282, 1307. Package microtype Warning Unable to apply patch footnote on input line 202.

7.4 Fehlermeldungen

```
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.
```

8 Dateigrößen

8.1 ZIP und CSV-Dateien

8.2 ZIP-Dateien

```
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("Datei",
                    "Größe in MB"))
```

Datei	Größe in MB
CE-BGH_2023-03-10_DE_Analyse.zip	10.71
CE-BGH_2023-03-10_DE_CSV_Datensatz.zip	235.17
CE-BGH_2023-03-10_DE_CSV_Metadaten.zip	3.26
CE-BGH_2023-03-10_DE_GraphML_Netzwerke.zip	1.96
CE-BGH_2023-03-10_DE_PDF_Datensatz.zip	6,400.98
CE-BGH_2023-03-10_DE_PDF_Entscheidungen-mit-Namen.zip	614.09
CE-BGH_2023-03-10_DE_PDF_Leitsatz-Entscheidungen.zip	2,395.43
CE-BGH_2023-03-10_DE_PDF_Platzhalter.zip	1.70
CE-BGH_2023-03-10_DE_TXT_Datensatz.zip	321.88
CE-BGH_2023-03-10_Source_Code.zip	0.26

8.3 CSV-Dateien

```
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("Datei",
                    "Größe in MB"))
```

Datei	Größe in MB
CE-BGH_2023-03-10_DE_CSV_Datensatz.csv	898.98
CE-BGH_2023-03-10_DE_CSV_Metadaten.csv	46.59
CE-BGH_2023-03-10_KryptographischeHashes.csv	0.00

8.4 PDF-Dateien (MB)

```
tar_load(files.pdf)
pdf.MB <- file.size(files.pdf) / 10^6
sum(pdf.MB)
#> [1] 6992.527
```

8.5 TXT-Dateien (MB)

```
tar_load(files.txt)
txt.MB <- file.size(files.txt) / 10^6
sum(txt.MB)
#> [1] 871.5501
```

9 Kryptographische Signaturen

9.1 Signaturen laden

```
tar_load(hashes)
```

9.2 Leerzeichen hinzufügen um bei SHA3-512 Zeilenumbruch zu ermöglichen

Hierbei handelt es sich lediglich um eine optische Notwendigkeit. Die normale 128 Zeichen lange Zeichenfolge von SHA3-512-Signaturen wird ansonsten nicht umgebrochen und verschwindet über die Seitengrenze. Das Leerzeichen erlaubt den automatischen Zeilenumbruch und damit einen für Menschen sinnvoll lesbaren Abdruck im Codebook. Diese Variante wird nur zur Anzeige verwendet und danach verworfen.

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

9.3 In Bericht anzeigen

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

index	filename
1	output/CE-BGH_2023-03-10_DE_PDF_Datensatz.zip
2	output/CE-BGH_2023-03-10_DE_PDF_Leitsatz-Entscheidungen.zip
3	output/CE-BGH_2023-03-10_DE_PDF_Entscheidungen-mit-Namen.zip
4	output/CE-BGH_2023-03-10_DE_PDF_Platzhalter.zip
5	output/CE-BGH_2023-03-10_DE_TXT_Datensatz.zip
6	output/CE-BGH_2023-03-10_DE_GraphML_Netzwerke.zip
7	output/CE-BGH_2023-03-10_DE_CSV_Datensatz.zip
8	output/CE-BGH_2023-03-10_DE_CSV_Metadaten.zip
9	output/CE-BGH_2023-03-10_DE_Analyse.zip
10	output/CE-BGH_2023-03-10_Source_Code.zip

11	output/CE-BGH_2023-03-10_Codebook.pdf
12	output/CE-BGH_2023-03-10_RobustnessChecks.pdf

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

index	sha2.256
1	c42bb013ad16dd87e17a9d43a0ae810c91b8c1370ea0e674403d9efaee123ff3
2	1e1def9d3aee984d21a77e97f08aecbe5c3a738e2f930b197b1c52ebdc9905b2
3	74c5f41c861cf73200c1ac4ed1e014a644d82d7c43fd74e56b73b848d6494b43
4	458abfbb5f65b0b31fb193d519dd166ef53dc91b20ea4c29e2228371baeeefbe
5	a8fcfe9f22d3934780c564d9fc78f1cdc7272232e26d42c212ce17c0b8838e28
6	7fbbbfc43fb988e09f5fd25edf4effb80313614266ab53d1b724ae7f91474450
7	c9a77f9bc41a36798b018fd7bd1a8381d506e2fa3d00a4f3a4c1be6e4381b768
8	24fe23b2dc132bc7a58208b964319c72f3cd9d5db5ab843c0e7283d6d19fd67c
9	1cbf0e35eebe35ddaf56cb78971215a22c8223689a2f239706b48c974f32ec56
10	514c7963f81486dd54af702998c1f2c9d3744d87a425d1c6d6bf6ef64a23f19a
11	354f15bf61a7291af6532809248d7198782adbdf98877812f621134a9d153cf7
12	000bb9534602f4f08a48781d8963769cd0e9c66cdf5a52a3a856438b7b58c9d6

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

index	sha3.512
1	4bb8b21f978f4a25dd3c2ffc94d5f7b0c4ad14b85e2b7b3f214bfea25f68e450 abcb908eb6183b116677cbed86dc28f4143fdcdb22ff156fca623d38d85af37e
2	3173cfd82c8e27ec2585fe2b8c4a356235019b07090a16c97594b909f50327cc 8ba1431c2d0f916614a275557e9dee4e1de53d2ce0e2024833407713d34a5050
3	1fbdcc227fcf4ca8652df32b34e3baa704e93108937b6a95626da05ab4739301 1fa5d867ed5527b22380512dde01071baea1a1673179a7265d65fe24a9e30bf6
4	c815b91a81557db2be892cdd07d873e725e7ee6019d4325f20f5333a20d85e4e 235739ab6d2cd19740cb1d133dacd57c9794c014788ba7ee51d2dd7947e99857
5	26bee62b7e1c44df120b74c7eed45ab5364b0e5fda4f259d0e92c57d945c1e6b aca92e12257d71c7d35c22a5681b0062c9d2a18e6d6ebc5eb2de2df052fe3010
6	9fdce7ddbfd86cc356c217ac3fd7cadd0b57a2b7655fc5fc6b72f396bd6827f5 1044c32cc1cfec7027054d4156f4103716d9b24b750b003ae2b5cfac84ac8aa2
7	4460d349c9ed56ac1fffc56da3a6e363a2efce746e357f9a05be9d845f0efc58 c1051e87d14c6984468d3fc6e9e22de6163ca1d5006dcfd6b51efe0d53ac125a
8	56212c5ebb99a1de12d9275f8a5136520213464bad83ca48c6e17e4fffc33d76 8359abd5b6906fd538c0d56fd5208c801484c146af08fbcd33a6e93b887a8469
9	44e0a3f7f1d4f9738e36570c935e3f6140c5d5cb35d0c2033f5af5d8c9123bdb d81a9b5d1b980c956bc898418061ecef123e4e2f27dc945de8f0728858f35bfb
10	d8d2ca144e9acc8b880a7723f6484a3cfec8381f07e75596afc4b631ba241916 67323a3e2d616c49d06aa027c65e8786466482c7c9c62386eb4442de470bc42f
11	44355aa01d12bc3569fc1b143e91d136d4a1f5a0e5c7abf1b4656bf970af56b9 19521f75b47d4f9ef7d91bf1164b16f5b29cb74835afc40e049d07b75e7096c8
12	10d3320d31918ed8711c4ac4c25e7e1db9b1a3eb3005b058e5ffacf7d438c252 6dc- cd12dd901a5562c9fe8c03bf378abe15b78b04854e46917b97cf2ea3c997d

10 Changelog

10.1 Version 2023-03-10

- Vollständige Aktualisierung der Daten
- NEU Zitations-Netzwerk zwischen allen Aktenzeichen des Bundesgerichtshofs als GraphML zur freien Verwendung (EXPERIMENTELL!)
- Gesamte Laufzeitumgebung mit Docker versionskontrolliert
- Aktenzeichen aus dem Eingangszeitraum 2000 bis 2009 nun korrekt mit führender Null formatiert (z.B. 1 BvR 44/02 statt 1 BvR 44/2)
- Vereinfachung der Konfigurations-Datei
- Verbesserte Formatierung von Warnungen und Fehlermeldungen im Compilation Report
- Verbesserung des Robustness Check Reports
- Neue Funktion für automatischen clean run (Löschung aller Zwischenergebnisse)
- Update der Download-Funktion
- Überflüssige Warnung in f.future_lingsummarize-Funktion entfernt
- Alle Roh-Dateien werden nun im Unterordner "files" gespeichert
- Korrektur für RiSt-Aktenzeichen eingefügt

10.2 Version 2022-08-16

- Vollständige Aktualisierung der Daten
- Neuer Entwurf des gesamten Source Codes im {targets} framework
- Die Zivilsenate in der Variable "spruchkoerper_db" sind jetzt arabisch nummeriert, damit sie automatisch sortiert werden können. Die originale römische Nummerierung ist weiterhin in der Variable "spruchkoerper_az" zu finden.
- Variable "comment" wurde in "bemerkung" umbenannt
- Variable "berichtigung" ist jetzt entweder mit TRUE oder FALSE codiert
- Diagramme sind in neuer Reihenfolge nummeriert, um die Reihenfolge im Codebook abzubilden
- Umfang der Datenbankabfrage ist nun vollautomatisiert

10.3 Version 2022-02-12

- Vollständige Aktualisierung der Daten
- Strenge Kontrolle und semantische Sortierung aller Variablen (entsprechend der Reihenfolge im Codebook)
- Datenstruktur wird nicht mehr im Codebook angezeigt um Fehler mit der UTF8-Kodierung und *listings* für LaTeX zu vermeiden
- Strenge Versionskontrolle aller R packages mit *renv*
- Der Prozess der Kompilierung ist jetzt detailliert konfigurierbar, insbesondere die Parallelisierung
- Parallelisierung nun vollständig mit *future* statt mit *foreach* und *doParallel*
- Fehlerhafte Kompilierungen werden beim vor der nächsten Kompilierung vollautomatisch aufgeräumt
- Alle Ergebnisse werden automatisch fertig verpackt in den Ordner 'output' sortiert
- README und CHANGELOG sind jetzt externe Markdown-Dateien, die bei der Kompilierung automatisiert eingebunden werden

- Issue #1 fixed: Senate normalisiert; die Variable “spruchkoerper_db” enthält nun die Präfixe “Strafsenat” und “Zivilsenat” vor der jeweiligen Senatsnummer um in den Dateinamen eine einfachere Orientierung zu ermöglichen
- Issue #2 fixed: Variablen nicht mehr doppelt definiert
- Issue #3 fixed: Alle Dateinamen-Präfixe nun korrekt
- Source Code des Changelogs zu Markdown konvertiert
- In der Vergangenheit fälschlich als “Platzhalter” aussortierte drei Dokumente sind nun im Datensatz enthalten
- Das Diagramm “Entscheidungen je Registerzeichen” ist nun zu einer Log-Skala konvertiert um die Darstellung informativer zu gestalten

10.4 Version 2021-04-27

- Vollständige Aktualisierung der Daten
- Veröffentlichung des vollständigen Source Codes
- Deutliche Erweiterung des inhaltlichen Umfangs des Codebooks
- Einführung der vollautomatischen Erstellung von Datensatz und Codebook
- Einführung von Compilation Reports um den Erstellungsprozess exakt zu dokumentieren
- Einführung von Variablen für Lizenz, Versionsnummer, Concept DOI, Version DOI, ECLI, Typ der Entscheidung, Präsident:in, Vize-Präsident:in, Verfahrensart, Name, Leitsatz, Bemerkungen, Berichtigungen, und linguistische Kennzahlen (Zeichen, Tokens, Typen, Sätze)
- Einführung von PDF-Varianten für Leitsatzentscheidungen und namentlich gekennzeichneten Entscheidungen.
- Zusammenfügung von über Zeilengrenzen getrennten Wörtern in der CSV-Variante
- Automatisierung und Erweiterung der Qualitätskontrolle
- Einführung von Diagrammen zur Visualisierung von Prüfergebnissen
- Einführung kryptographischer Signaturen
- Alle Variablen sind nun in Kleinschreibung und Snake Case gehalten
- Variable ‘Ordinalzahl’ in ‘eingangsnummer’ umbenannt

10.5 Version 2020-07-09

- Erstveröffentlichung

11 Abschluss

```r
## Datumsstempel
print(datestamp)
#> [1] "2023-03-10"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2023-03-10 15:48:47 UTC"


## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2023-03-10 15:48:57 UTC"


## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 10.30461 secs
```

12 Parameter für strenge Replikationen

```
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_1.31.0
#>  [9] RcppTOML_0.2.2    tarchetypes_0.7.4 targets_0.14.2
#>
#> 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_3.1.8
#> [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
```

Literaturverzeichnis

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.*

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, and Akitaka Matsuo. 2020. *Spacyr: Wrapper to the spaCy 'Nlp' Library.* https://spacyr.quanteda.io.

Benoit, Kenneth, and Adam Obeng. 2021. *Readtext: Import and Handling for Plain and Formatted Text Files.* https://github.com/quanteda/readtext.

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.

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.

Ewing, Mark. 2021. *Mgsub: Safe, Multiple, Simultaneous String Substitution.*

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.*

Hester, Jim, Hadley Wickham, and Gábor Csárdi. 2023. *Fs: Cross-Platform File System Operations Based on Libuv.*

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*.

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. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

———. 2022. *Rvest: Easily Harvest (Scrape) Web Pages*.

———. 2023. *Httr: Tools for Working with Urls and Http*.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2023. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*.

Wickham, Hadley, and Dana Seidel. 2022. *Scales: Scale Functions for Visualization*.

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*.