Corpus der Entscheidungen des Bundesverfassungsgerichts

(CE-BVerfG-Source)

Compilation Report

Version 2024-07-24

DOI: 10.5281/zenodo.12705675

Titel	Source Code des »Corpus der Entscheidungen des Bundesverfassungsgerichts«
Abkürzung	CE-BVerfG-Source
Autor	Seán Fobbe
Version	2024-07-24
Download	https://doi.org/10.5281/zenodo.12705675
Lizenz	GNU General Public License Version 3 (GPLv3)

Zitiervorschlag

Seán Fobbe (2024). Source Code des »Corpus der Entscheidungen des Bundesverfassungsgerichts« (CE-BVerfG-Source). Version 2024-07-24. Zenodo. DOI: 10.5281/zenodo.12705675.

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 2024-07-24. 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. Sie lautet 10.5281/zenodo.4308216. Die »Concept DOI« verlinkt immer die aktuellste Version.

GNU General Public License Version 3 (GPLv3)

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 README: Corpus der Entscheidungen des Bundesverfassungsgerichts (CE-BVerfGE)

1.1 Überblick

Das **Corpus der Entscheidungen des Bundesverfassungsgerichts (CE-BVerfG)** ist eine möglichst vollständige Sammlung der vom Bundesverfassungsgericht veröffentlichten Entscheidungen. Der Datensatz nutzt als seine Datenquelle die amtliche Entscheidungsdatenbank des Bundesverfassungsgerichts 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.3902658

1.2 Funktionsweise

Primäre Endprodukte des Skripts sind folgende ZIP-Archive:

- Der volle Datensatz im CSV-Format
- Die reinen Metadaten im CSV-Format (wie unter 1, nur ohne Entscheidungstexte)
- Zitationsnetzwerk des BVerfG im GraphML-Format
- (Optional) Tokenisierte Form aller Texte mit linguistischen Annotationen im CSV-Format
- Alle Entscheidungen im HTML-Format
- Alle Entscheidungen im TXT-Format (reduzierter Umfang an Metadaten)
- Alle Entscheidungen im PDF-Format (reduzierter Umfang an Metadaten)
- Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)
- Der Source Code und alle weiteren Quelldaten

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
- 5 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-bverfg
```

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 (`pdf/`, `txt/`, `temp/` `analysis` und `output/`). Die Endergebnisse werden alle in `output/` abgelegt.

```
.
├ buttons                    # Buttons (nur optische Bedeutung)
├ CHANGELOG.md               # Alle Änderungen
├ 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
├ docker-compose.yaml        # Konfiguration für Docker
├ docker-delete-all-data.sh  # Löscht Datensatz und Zwischenergebnisse via Docker
├ Dockerfile                 # Definition des Docker Images
├ docker-run-project.sh      # Docker Image und Datensatz kompilieren
├ etc                        # Weitere Konfigurationsdateien
├ functions                  # Wichtige Schritte der Pipeline
├ gpg                        # Persönlicher Public GPG-Key für Seán Fobbe
├ LICENSE                    # Volltext der Lizenz für den Source Code
├ pipeline.Rmd               # Zentrale Definition der Pipeline
├ README.md                  # Bedienungsanleitung
├ reports                    # Markdown-Dateien
├ 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 kontaktieren Sie mich via https://www.seanfobbe.de/contact

2 Packages laden

```
library(targets)
library(tarchetypes)
library(RcppTOML)
library(future)
library(data.table)
library(quanteda)
library(knitr)
library(kableExtra)
library(igraph)
library(ggraph)

tar_unscript()
```

3 Vorbereitung

3.1 Definitionen

```
## Datum
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2024-07-24"

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

## Konfiguration
config <- RcppTOML::parseTOML("config.toml")
print(config)
#> List of 10
#>  $ annotate:List of 1
#>   ..$ toggle: logi FALSE
#>  $ cores   :List of 2
#>   ..$ max   : logi TRUE
#>   ..$ number: int 8
#>  $ debug   :List of 2
#>   ..$ pages : int 20
#>   ..$ toggle: logi FALSE
#>  $ doi     :List of 4
#>   ..$ aktenzeichen : chr "10.5281/zenodo.4569564"
#>   ..$ data         :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.3902658"
#>   .. ..$ version: chr "10.5281/zenodo.12705674"
#>   ..$ personendaten: chr "10.5281/zenodo.4568682"
#>   ..$ software     :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.4308216"
#>   .. ..$ version: chr "10.5281/zenodo.12705675"
#>  $ download:List of 1
#>   ..$ timeout: int 60
#>  $ fig     :List of 3
#>   ..$ align : chr "center"
#>   ..$ dpi   : int 300
#>   ..$ format: chr [1:2] "pdf" "png"
#>  $ license :List of 2
#>   ..$ code: chr "GNU General Public License Version 3 (GPLv3)"
#>   ..$ data: chr "Creative Commons Zero 1.0 Universal (CC Zero 1.0)"
#>  $ parallel:List of 4
#>   ..$ extractPDF   : logi TRUE
#>   ..$ lingsummarize: logi TRUE
#>   ..$ multihashes  : logi TRUE
#>   ..$ spacyparse   : logi FALSE
#>  $ project :List of 3
#>   ..$ author   : chr "Seán Fobbe"
#>   ..$ fullname : chr "Corpus der Entscheidungen des Bundesverfassungsgerichts"
#>   ..$ shortname: chr "CE-BVerfG"
#>  $ 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 vom heutigen Tag sind.

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

3.3 Ordner erstellen

```
#unlink("output", recursive = TRUE)
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!
```

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
                            "igraph",       # Netzwerkanalyse
                            "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",
                 "datum",
                 "spruchkoerper_typ",
                 "spruchkoerper_az",
                 "registerzeichen",
                 "eingangsnummer",
                 "eingangsjahr_az",
                 "kollision",
                 "name",
                 "band",
                 "seite")

#> 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 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.
```

5.1.2 Changelog

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

5.1.3 Variablen für die BVerfGE

Diese Tabelle enthält Name, Band und Seite der Entscheidung in der BVerfGE.

```
list(
    tar_target(file.var_bverfge,
               "data/BVerfGE_Variablen_NameBandSeite.csv",
               format = "file"),
    tar_target(dt.var_bverfge,
               fread(file.var_bverfge))
)
#> Establish _targets.R and _targets_r/targets/tar.file.bverfge.R.
```

5.1.4 Liste aller Variablen

Die Variablen des Datensatzes, inklusive ihrer Erläuterung.

```
list(
    tar_target(file.var_codebook,
               "data/CE-BVerfG_Variables.csv",
               format = "file"),
```

```
    tar_target(dt.var_codebook,
               fread(file.var_codebook))
)
#> Establish _targets.R and _targets_r/targets/tar.file.vars.R.
```

5.1.5 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.azbrd.R.
```

5.1.6 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.vpresi.R.
```

5.2 Download Targets

5.2.1 Vorläufige Download-Tabelle erstellen

```
tarchetypes::tar_age(dt.download,
                     f.download_table_make(debug.toggle = config$debug$toggle,
                                           debug.pages = config$debug$pages),
                     age = as.difftime(3, units = "days"))

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

5.2.2 Vorläufige Dateinamen erstellen

```
tar_target(filenames.raw,
           f.filenames_raw(url.pdf = dt.download$url_pdf))
#> Establish _targets.R and _targets_r/targets/tar.download.filenames.prelim.R.
```

5.2.3 Finale Dateinamen erstellen

```
tar_target(filenames.final,
           f.filenames_final(filenames.raw = filenames.raw,
                             var.bverfge = dt.var_bverfge))
#> Establish _targets.R and _targets_r/targets/tar.download.filenames.final.R.
```

5.2.4 Finale Download-Tabelle erstellen

```
tar_target(dt.download.final,
           f.download_finalize(dt.download = dt.download,
                               filenames.final = filenames.final))
#> Establish _targets.R and _targets_r/targets/tar.download.final.R.
```

5.2.5 Download durchführen (PDF)

```
tar_target(files.pdf,
                f.download(url = dt.download.final$url_pdf,
                           filename = dt.download.final$doc_id,
                           dir = "files/pdf",
                           sleep.min = 0.3,
                           sleep.max = 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.
```

5.2.6 Download durchführen (HTML)

```
tar_target(files.html,
                f.download(url = dt.download.final$url_html,
                           filename = basename(dt.download.final$url_html),
                           dir = "files/html",
                           sleep.min = 0,
                           sleep.max = 0.2,
                           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.
```

5.3 Convert Targets

5.3.1 TXT-Dateien erstellen und einlesen

Es 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.bverfg,
                f.readtext(x = files.txt,
                           docvarnames = docvarnames))
     )

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

5.3.2 HTML-Dateien parsen

```
tar_target(html.parsed,
           f.parse_html_bverfg(html = files.html))

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

5.3.3 HTML Parse-Ergebnis splitten

```
list(
    tar_target(dt.html.meta,
               f.clean_meta(html.parsed$dt.meta.html)),
    tar_target(dt.segmented,
               f.clean_segmented(html.parsed$dt.segmented.full))
)


#> Establish _targets.R and _targets_r/targets/tar.convert.html.split.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.bverfg.datecleaned,
           f.clean_dates(dt.bverfg))
#> Establish _targets.R and _targets_r/targets/tar.enhance.dateclean.R.
```

5.4.2 Variable erstellen: »verfahrensart«

Die Variable »verfahrensart« wird aus den Registerzeichen berechnet.

```
tar_target(var_verfahrensart,
               f.var_verfahrensart(dt.bverfg.datecleaned$registerzeichen,
                                   az.brd = az.brd,
                                   gericht = "BVerfG"))
#> 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(dt.bverfg.datecleaned,
                                  az.brd = az.brd,
                                  gericht = "BVerfG",
                                  remove.na = TRUE))
#> Establish _targets.R and _targets_r/targets/tar.enhance.aktenzeichen.R.
```

5.4.4 Variable erstellen: »ecli«

Die ECLI wird aus ihren Komponenten berechnet.

```
tar_target(var_ecli,
           f.var_ecli_bverfg(x = dt.bverfg.datecleaned))
#> 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.bverfg.datecleaned$datum,
                        gericht = "BVerfG",
                        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.bverfg.datecleaned$datum,
                        gericht = "BVerfG",
                        pvp.fcg = vpresidents))
#> Establish _targets.R and _targets_r/targets/tar.enhance.vpraesi.R.
```

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

Berechnung klassischer linguistischer Kennzahlen.

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

5.4.8 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.
    bverfg.datecleaned)),])
#> Establish _targets.R and _targets_r/targets/tar.enhance.constants.R.
```

5.4.9 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,
                      var_lingstats,
                      var_constants))

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

5.4.10 Datensatz und zusätzliche Variablen verbinden

```
tar_target(dt.bverfg.intermediate,
           cbind(dt.bverfg.datecleaned,
                 vars_additional))

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

5.4.11 Hauptdatensatz in segmentierte Variante mergen

```
tar_target(dt.segmented.final,
           f.finalize_segmented(dt.segmented = dt.segmented,
                                dt.bverfg.intermediate = dt.bverfg.intermediate,
                                dt.download.final = dt.download.final,
                                varnames = dt.var_codebook$varname))

#> Establish _targets.R and _targets_r/targets/tar.enhance.segmented.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.bverfg.final,
           f.finalize_main(dt.bverfg.intermediate = dt.bverfg.intermediate,
                           dt.download.final = dt.download.final,
                           dt.html.meta = dt.html.meta,
                           varnames = dt.var_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.bverfg.meta,
               dt.bverfg.final[, !"text"])

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

5.4.14 Variante erstellen: Annotiert

```
tar_target(dt.bverfg.annotated,
           f.tar_udpipe(dt.bverfg.final,
                        language = "german-hdt",
                        model_dir = "temp",
                        cores = fullCores))
```

5.5 Zitate extrahieren

```
tar_target(igraph_citations,
           f.citation_extraction_bverfg(dt.final = dt.bverfg.final))

#> Establish _targets.R and _targets_r/targets/tar.citations.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.bverfg.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.bverfg.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 CSV schreiben: Segmentierte Variante

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

5.6.4 GraphML schreiben

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

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

5.7.2 Zusammenfassungen linguistischer Kennwerte berechnen

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

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

5.8.2 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.
```

5.8.3 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"),
                format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.csv.full.R.
```

5.8.4 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"),
                format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.csv.meta.R.
```

5.8.5 ZIP erstellen: CSV-Datei (Segmentiert)

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

5.8.6 ZIP erstellen: PDF-Dateien (alle Entscheidungen)

```
tar_target(zip.pdf.all,
           f.tar_zip(files.pdf,
                     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.
```

5.8.7 ZIP erstellen: TXT-Dateien

```
tar_target(zip.txt,
           f.tar_zip(files.txt,
                     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: HTML-Dateien

```
tar_target(zip.html,
           f.tar_zip(files.html,
                     filename = paste(prefix.files,
                                      "DE_HTML_Datensatz.zip",
                                      sep = "_"),
                     dir = "output",
                     mode = "cherry-pick"),
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.html.R.
```

5.8.9 ZIP erstellen: GraphML

```
tar_target(zip.graphml,
           f.tar_zip(graphml_citations,
                     filename = paste(prefix.files,
                                      "DE_GraphML_Zitationsnetzwerk.zip",
                                      sep = "_"),
                     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.txt,
             zip.html,
             zip.csv.final,
             zip.csv.meta,
             zip.csv.segmented,
             zip.graphml,
             zip.analysis,
             zip.source))
#> Establish _targets.R and _targets_r/targets/tar.hashes.manifest.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.write.R.
```

6 Pipeline: Kompilierung

6.1 Durchführen der Kompilierung

```
tar_make()
```

6.2 Pipeline archivieren

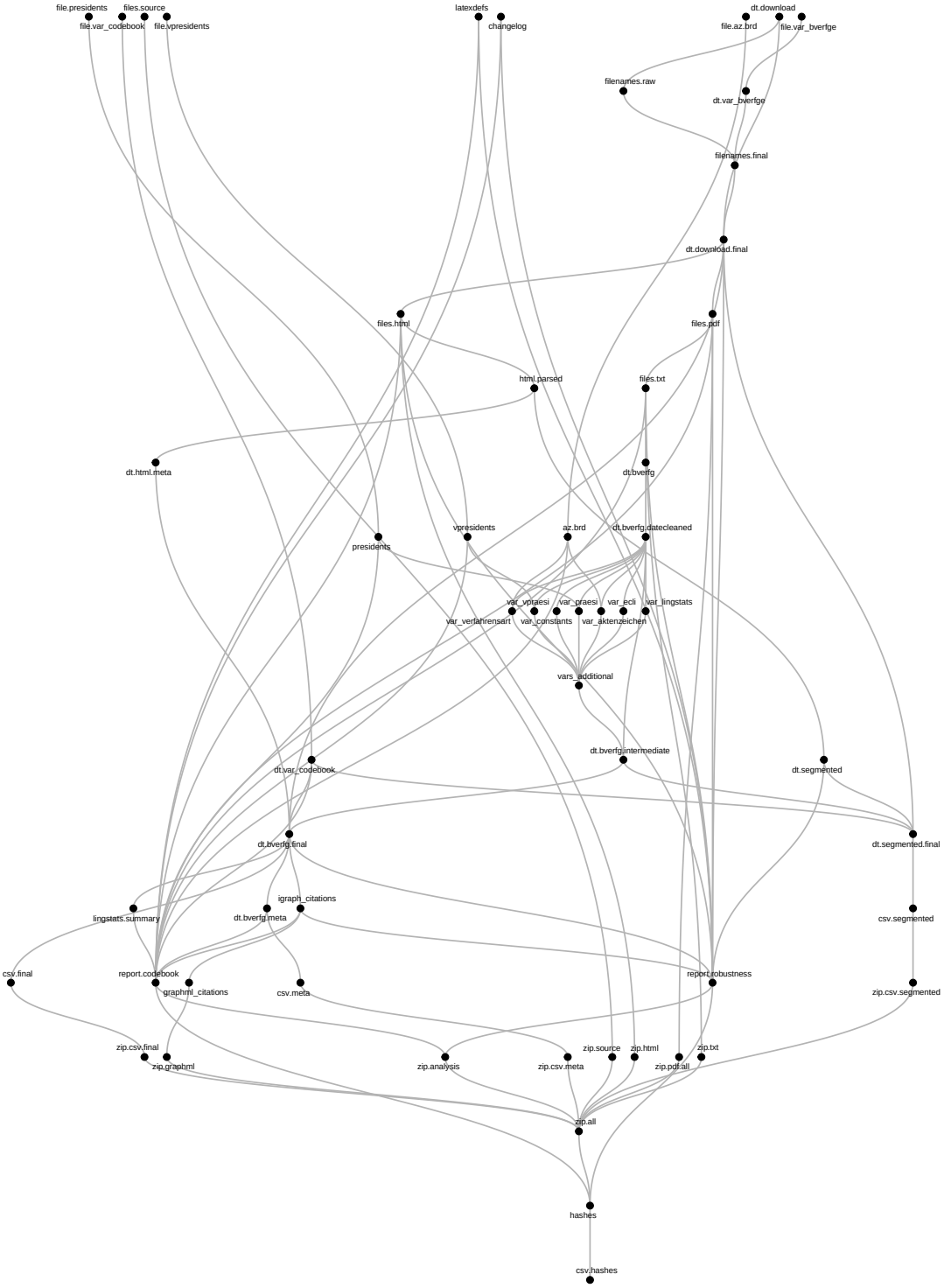
```
zip(paste0("output/",
           paste0(config$project$shortname,
                  "_",
                  datestamp),
           "_Targets_Storage.zip"),
    "_targets/")
```

6.3 Visualisierung

```
edgelist <- tar_network(targets_only = TRUE)$edges
#> -\|/-\|/-\|/-\
setDT(edgelist)

g  <- igraph::graph_from_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()
```


file.presidents
files.source
file.var_codebook
file.vpresidents
latexdefs
changelog
dt.download
file.az.brd
file.var_bverfge
filenames.raw
dt.var_bverfge
filenames.final
dt.download.final
files.html
files.pdf
html.parsed
files.txt
dt.html.meta
dt.bverfg
vpresidents
az.brd
dt.bverfg.datecleaned
presidents
var_vpraesi
var_praesi
var_ecli
var_lingstats
var_verfahrensart
var_constants
var_aktenzeichen
vars_additional
dt.bverfg.intermediate
dt.segmented
dt.var_codebook
dt.segmented.final
dt.bverfg.final
igraph_citations
lingstats.summary
dt.bverfg.meta
csv.segmented
csv.final
report.codebook
graphml_citations
csv.meta
report.robustness
zip.csv.segmented
zip.csv.final
zip.graphml
zip.analysis
zip.source
zip.html
zip.txt
zip.csv.meta
zip.pdf.all
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] 2819.17

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
changelog	stem	6328	file	0.01
csv.final	stem	88	qs	0.00
csv.hashes	stem	94	qs	0.00
csv.meta	stem	88	qs	0.00
csv.segmented	stem	90	qs	0.00
dt.bverfg	stem	49979342	qs	49.98
dt.bverfg.datecleaned	stem	49980662	qs	49.98
dt.bverfg.final	stem	50788279	qs	50.79
dt.bverfg.intermediate	stem	50141819	qs	50.14
dt.bverfg.meta	stem	960040	qs	0.96
dt.download	stem	135385	qs	0.14
dt.download.final	stem	219647	qs	0.22
dt.html.meta	stem	573091	qs	0.57
dt.segmented	stem	47375869	qs	47.38
dt.segmented.final	stem	48646504	qs	48.65

(continued)

name	type	bytes	format	MB
dt.var_bverfge	stem	21487	qs	0.02
dt.var_codebook	stem	3868	qs	0.00
file.az.brd	stem	36533	file	0.04
file.presidents	stem	7249	file	0.01
file.var_bverfge	stem	69777	file	0.07
file.var_codebook	stem	10254	file	0.01
file.vpresidents	stem	9193	file	0.01
filenames.final	stem	83515	qs	0.08
filenames.raw	stem	59165	qs	0.06
files.html	stem	541282589	file	541.28
files.pdf	stem	725800549	file	725.80
files.source	stem	1071021	file	1.07
files.txt	stem	185284618	file	185.28
graphml_citations	stem	19674044	file	19.67
hashes	stem	1660	qs	0.00
html.parsed	stem	47933846	qs	47.93
igraph_citations	stem	764062	qs	0.76
latexdefs	stem	1295	file	0.00
lingstats.summary	stem	384	qs	0.00
presidents	stem	1959	qs	0.00
report.codebook	stem	3193339	file	3.19
report.robustness	stem	5538532	file	5.54
var_aktenzeichen	stem	34785	qs	0.03
var_constants	stem	6043	qs	0.01
var_ecli	stem	59346	qs	0.06
var_lingstats	stem	60594	qs	0.06
var_praesi	stem	165	qs	0.00
var_verfahrensart	stem	6113	qs	0.01
var_vpraesi	stem	192	qs	0.00

(continued)

name	type	bytes	format	MB
vars_additional	stem	161118	qs	0.16
vpresidents	stem	2450	qs	0.00
zip.all	stem	198	qs	0.00
zip.analysis	stem	10835096	file	10.84
zip.csv.final	stem	52132803	file	52.13
zip.csv.meta	stem	1104031	file	1.10
zip.csv.segmented	stem	58503361	file	58.50
zip.graphml	stem	723147	file	0.72
zip.html	stem	113788638	file	113.79
zip.pdf.all	stem	688285176	file	688.29
zip.source	stem	268016	file	0.27
zip.txt	stem	63550500	file	63.55

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] 16906.14

## Minuten
runtime.sum / 60
#> [1] 281.769

## Stunden
runtime.sum / 3600
#> [1] 4.69615
```

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	2024-07-24 08:24:37	9589.521	159.83
files.html	2024-07-24 05:45:46	4798.256	79.97
dt.download	2024-07-24 04:25:47	1771.765	29.53
html.parsed	2024-07-24 08:31:16	319.537	5.33
var_lingstats	2024-07-24 08:35:07	82.459	1.37
lingstats.summary	2024-07-24 08:36:08	55.711	0.93
dt.segmented	2024-07-24 08:32:52	53.527	0.89
report.codebook	2024-07-24 16:24:36	51.324	0.86
dt.bverfg	2024-07-24 08:33:40	34.867	0.58
report.robustness	2024-07-24 16:19:07	32.898	0.55

(continued)

name	time	seconds	mins
zip.pdf.all	2024-07-24 08:31:40	23.985	0.40
zip.html	2024-07-24 08:25:55	18.575	0.31
files.txt	2024-07-24 08:31:57	17.116	0.29
zip.csv.segmented	2024-07-24 08:38:04	12.302	0.21
zip.csv.final	2024-07-24 08:37:52	11.310	0.19
zip.txt	2024-07-24 08:33:04	10.761	0.18
igraph_citations	2024-07-24 08:36:15	6.102	0.10
var_aktenzeichen	2024-07-24 08:33:44	3.542	0.06
hashes	2024-07-24 16:24:40	3.149	0.05
latexdefs	2024-07-24 03:56:15	1.218	0.02
changelog	2024-07-24 16:11:58	1.215	0.02
files.source	2024-07-24 16:23:34	1.197	0.02
dt.segmented.final	2024-07-24 08:35:13	1.051	0.02
dt.html.meta	2024-07-24 08:32:53	0.968	0.02
dt.bverfg.final	2024-07-24 08:35:10	0.871	0.01
filenames.raw	2024-07-24 04:25:48	0.531	0.01
zip.analysis	2024-07-24 16:24:37	0.450	0.01
zip.graphml	2024-07-24 08:38:05	0.339	0.01
zip.csv.meta	2024-07-24 08:38:05	0.223	0.00
graphml_citations	2024-07-24 08:36:15	0.214	0.00
csv.segmented	2024-07-24 08:36:15	0.210	0.00
var_ecli	2024-07-24 08:33:45	0.201	0.00
filenames.final	2024-07-24 04:25:48	0.191	0.00
dt.download.final	2024-07-24 04:25:48	0.142	0.00
var_praesi	2024-07-24 08:35:07	0.129	0.00
csv.final	2024-07-24 08:36:15	0.122	0.00
var_vpraesi	2024-07-24 08:35:07	0.080	0.00
zip.source	2024-07-24 16:23:44	0.040	0.00
dt.bverfg.datecleaned	2024-07-24 08:33:41	0.016	0.00

(continued)

name	time	seconds	mins
csv.meta	2024-07-24 08:37:41	0.011	0.00
var_constants	2024-07-24 08:33:44	0.004	0.00
dt.bverfg.meta	2024-07-24 08:36:15	0.003	0.00
dt.var_bverfge	2024-07-24 04:25:48	0.002	0.00
var_verfahrensart	2024-07-24 08:35:07	0.002	0.00
az.brd	2024-07-24 04:25:48	0.001	0.00
dt.bverfg.intermediate	2024-07-24 08:35:08	0.001	0.00
dt.var_codebook	2024-07-24 04:25:48	0.001	0.00
vars_additional	2024-07-24 08:35:07	0.001	0.00
csv.hashes	2024-07-24 16:24:40	0.000	0.00
file.az.brd	2023-05-09 12:40:41	0.000	0.00
file.presidents	2023-05-09 12:40:41	0.000	0.00
file.var_bverfge	2024-07-17 23:33:37	0.000	0.00
file.var_codebook	2024-07-23 18:51:27	0.000	0.00
file.vpresidents	2023-05-09 12:40:41	0.000	0.00
presidents	2024-07-24 04:25:48	0.000	0.00
vpresidents	2024-07-24 04:25:48	0.000	0.00
zip.all	2024-07-24 16:24:37	0.000	0.00

7.3 Warnungen

Hinweis: für die folgenden Entscheidungen waren auch nach manueller Nachprüfung keine PDF-Downloads verfügbar:

- ECLI:DE:BVerfG:2001:bs20011001.2bvb000101
- ECLI:DE:BVerfG:2001:rk20011023.2bvr123601
- ECLI:DE:BVerfG:2004:rk20040519.1bvr071104
- ECLI:DE:BVerfG:2014:rk20140319.1bvr141710
- ECLI:DE:BVerfG:2014:rk20140610.1bvr066914

```
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

cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201406rk20140610_1bvr06

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201403rk20140319_1bvr141710.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200405rk20040519_1bvr071104.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200110rk20011023_2bvr123601.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200110bs20011001_2bvb000101.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201406rk20140610_1bvr066914.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201403rk20140319_1bvr141710.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL httpswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200405rk20040519_1bvr071104.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL htt-
pswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200110rk20011023_2bvr123601.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL htt-
pswww.bundesverfassungsgericht.deSharedDocsDownloadsDE200110bs20011001_2bvb000101.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL htt-
pswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201406rk20140610_1bvr066914.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL htt-
pswww.bundesverfassungsgericht.deSharedDocsDownloadsDE201403rk20140319_1

7.3.2 report.codebook

Package microtype Warning Unable to apply patch footnote on input line 207.

7.3.3 report.robustness

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-BVerfG_2024-07-24_DE_Analyse.zip	10.84
CE-BVerfG_2024-07-24_DE_CSV_Datensatz.zip	52.13
CE-BVerfG_2024-07-24_DE_CSV_Metadaten.zip	1.10
CE-BVerfG_2024-07-24_DE_CSV_Segmentiert.zip	58.50
CE-BVerfG_2024-07-24_DE_GraphML_Zitationsnetzwerk.zip	0.72
CE-BVerfG_2024-07-24_DE_HTML_Datensatz.zip	113.79
CE-BVerfG_2024-07-24_DE_PDF_Datensatz.zip	688.29
CE-BVerfG_2024-07-24_DE_TXT_Datensatz.zip	63.55
CE-BVerfG_2024-07-24_Source_Code.zip	0.27
CE-BVerfG_2024-07-24_Targets_Storage.zip	348.19

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-BVerfG_2024-07-24_DE_CSV_Datensatz.csv	191.09
CE-BVerfG_2024-07-24_DE_CSV_Metadaten.csv	8.57
CE-BVerfG_2024-07-24_DE_CSV_Segmentiert.csv	454.99
CE-BVerfG_2024-07-24_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] 725.8005
```

8.5 TXT-Dateien (MB)

```
tar_load(files.txt)
txt.MB <- file.size(files.txt) / 10^6
sum(txt.MB)
#> [1] 185.2846
```

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-BVerfG_2024-07-24_DE_PDF_Datensatz.zip
2	output/CE-BVerfG_2024-07-24_DE_TXT_Datensatz.zip
3	output/CE-BVerfG_2024-07-24_DE_HTML_Datensatz.zip
4	output/CE-BVerfG_2024-07-24_DE_CSV_Datensatz.zip
5	output/CE-BVerfG_2024-07-24_DE_CSV_Metadaten.zip
6	output/CE-BVerfG_2024-07-24_DE_CSV_Segmentiert.zip
7	output/CE-BVerfG_2024-07-24_DE_GraphML_Zitationsnetzwerk.zip
8	output/CE-BVerfG_2024-07-24_DE_Analyse.zip
9	output/CE-BVerfG_2024-07-24_Source_Code.zip
10	output/CE-BVerfG_2024-07-24_Codebook.pdf
11	output/CE-BVerfG_2024-07-24_RobustnessChecks.pdf

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

index	sha2.256
1	2220ee883bbb73c56b55f64cda187d641f7573b4c1abe4bae34c968f394e49e1
2	1e47a28e6db65763a7ca291c84ab49ce1706925caaf60ad65f0e019a6efdab3a
3	72c447d93022c882957264926bf27ceea6ab1826a4217b13f439d893c6075b2e
4	d57f508825c9acb5562dfdf784c33f5fc66943fa6dedc3dd3a68d7691ee82feb
5	49236b37d12007abdcad0ed59a7b748ac2bfa46792adab83bf9b7d29e718733d
6	9fa50bc5efe2110967d87883e25837e04510fa77ac3b8ca0ed19e8904e2a6038
7	234e73bdcf380b7b249b83f0a590d9b2e6e1580ae69d401f8a5ac6197602aea7
8	655b2af58d9927cabbe1eeb32ca17ae4ddbb1172dd660eaa3a219a4563c72569
9	4c30a52a3b84d13386ac54881208e4d5a90918824a04651b369fd9dfdb956316
10	ae3b05e4460358b45717fb12bc43ab9f93c654795334840c2e29a55944b258c5
11	8cf6a11028db681f2de02a383721f9ff45da00acda9e68d5f0440750edee07ef

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

index	sha3.512
1	1d323cdf5397dfbec6514a39087575bf1c8fe50c0ebd7ca6f37b964f86b019b3 e50881fc8c1047a72a338266674c50612775331a3b3ea25c37effd7145409ad3
2	b07bf69fa7b9bc01fd80a3a665354b71fac729a664413ce2329e72284fb75aac 82c5911b17ef41d85f773e184ca24ae839cb0d5490da7f8adcbef300ed888c6a
3	ff5157a3029416b20035898d9d7ee55eb6657a0cb730481da98e6949dc49a53a f8a168e93858c762e4f7a1ff117a0f1f677a340679c6c46a1f885ed188753be4
4	bbd41f14aeb92bfdfe053f29c6338b7efb37d1afaaefd5c0d02e2240ba46e731 e8e7301ab374676b38647bf47e19661b014f94f29d0da614e88d28b5156ee3bf
5	216e7fd65dafd5f9352aa6ac966270cca1bc283ae3ac3d29b71f20fc89834a4a 894801cd223ce38f95336b12d1598a992a668134c035e3afa901d32932b21a51

6	52376c72eaa1d2934bb960b8fb8c642724e341149528f317c2a9b844bb4d7c7e 7ff1f658b2d8d4667d0da98b149b6e4c1ddb8f51874e827d30a7070dcee54a88
7	8d01b19a0f08f5dc6b112b76e62210d1f8593703dcc4341b0726040a39cce883 21cbd845c3dcc880d15a59af3c2a0609dad2e7f3dd44f20a435ec82a89c028aa
8	55f347f43de360be6ea0b9583454584203c8c8a944aee1a7716aec9f1fe9b93a 59c3a78ceb8cc344fc916468135e7c964127c6a4c20f192a040684643fe3cd4c
9	88480e23d3a17254d78e8ac1cce5e58e1f20d919c10b987153f8077561cdedb2 377abecc892e9d69ed91f3607a7ee3adfed6f37d70103ee385f4d461cabb4d9f
10	1d921a2e5158c5ffdca2dbc5a7d9507685bbea53fcd942f255fa551f4f8736f7 d95147c834cecc87be31f937a0ed63f970701a4fe7d1cec5dcd51f2efeb69696
11	620bd255f85bac68802177040d7c8d9a73040a8801fe6ede7aed207718b789cc 136b86ee0556abe81d6b13b09fb8dd9be31d49c1ff853e000a4c3746707a5d27

10 Changelog

10.1 Version 2024-07-24

- LIZENZÄNDERUNG: Source Code jetzt unter GNU General Public License Version 3 (GPLv3) oder später lizenziert
- NEU: Zitationsnetzwerke BVerfGE-zu-BVerfGE, BVerfGE-zu-Aktenzeichen und Aktenzeichen-zu-Aktenzeichen als GraphML
- Neue Variable "bverfge": TRUE/FALSE, ob eine Entscheidung in der BVerfGE enthalten ist
- Vollständige Aktualisierung der Daten
- Amtliche Sammlung bis inklusive Band 164 mit Name, Band und Seite versehen
- Die Pipeline mit allen Zwischenergebnissen wird nun automatisch in "output/" archiviert
- R-Version auf 4.4.0 aktualisiert (wegen CVE-2024-27322)
- Python Toolchain aktualisiert
- Vereinfachung der Repository-Struktur mit Ordner etc/ für Config Files
- Anpassung Docker Compose an Debian 11
- Docker Zeitzone auf Berlin eingestellt
- Aktualisierung von Public GPG Key im Repository
- Zusätzliches Lösch-Skript mit Docker-Integration
- Neuer Test auf Vollständigkeit der Dokumentation von Variablen im Codebook
- Viele neue Tests zur Sicherung der Datenqualität des finalen tabellarischen Datensatzes
- Vollständigkeit der Datenbankabfrage wird nun alle 3 Tage automatisch überprüft
- Diagramme nicht mehr nummeriert, sondern nach Typ sortiert

10.2 Version 2023-02-26

- Vollständige Aktualisierung der Daten
- Gesamte Laufzeitumgebung mit Docker versionskontrolliert
- Amtliche Sammlung bis inklusive Band 160 mit Name, Band und Seite versehen
- 50 neue historische Entscheidungen aus dem Zeitraum 1951 bis 1998 (u.a. Elfes, Schleyer-Entführung, Kurzarbeitergeld, Nachtarbeiterinnen)
- Aktenzeichen aus dem Eingangszeitraum 2000 bis 2009 nun korrekt mit führender Null formatiert (z.B. 1 BvR 44/02 statt 1 BvR 44/2)
- Überarbeitung der Namen der Entscheidungen, u.a. Einfügung von Bindestrichen um Lesbarkeit zu verbessern und weitere Standardisierung
- Verbesserte Formatierung von Warnungen und Fehlermeldungen im Compilation Report
- Duplikat-Prüfung für Dateinamen eingeführt
- Update für allgemeine Download-Funktion
- Verbesserung des Robustness Check Reports
- Download-Timeout auf 60 Sekunden reduziert
- Überflüssige Warnung in f.future_lingsummarize-Funktion entfernt

10.3 Version 2022-08-24

- Vollständige Aktualisierung der Daten
- Neuentwurf das gesamten Source Codes im {targets} framework

- Entfernung von englischen Zusammenfassungen aus dem Korpus
- Vielzahl zusätzlicher Unit Tests
- Zusätzliche Variablen mit URLs zu originalen HTML- und PDF-Dateien
- Variante mit linguistischen Annotationen temporär nicht mehr verfügbar
- Robustness Checks sind nun in einem separaten Bericht dokumentiert
- Frequenztabellen-Test berücksichtigt nun alle Variablen
- Neues Diagramm: Visualisierung von Kompilierungs-Prozess
- Diagramme sind in neuer Reihenfolge nummeriert, um die Reihenfolge im Codebook abzubilden

10.4 Version 2022-02-01

- Vollständige Aktualisierung der Daten
- Strenge Versionskontrolle von R packages mit {renv}
- Kompilierung jetzt detailliert konfigurierbar, insbesondere die Parallelisierung
- Parallelisierung nun vollständig mit {future} statt mit {foreach} und {doParallel}
- Codebook-Erstellung stark beschleunigt durch Verwendung vorberechneter Diagramme
- Fehlerhafte Kompilierungen werden 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
- Source Code des Changelogs zu Markdown konvertiert
- REGEX-Tests im Detail kommentiert

10.5 Version 2021-09-19

- Vollständige Aktualisierung der Daten
- Neue Variablen: Pressemitteilung, Zitiervorschlag, Aktenzeichen (alle), Kurzbeschreibung und Richter
- Neue Variante: Segmentiert
- Neue Variante: HTML
- Erweiterung der Codebook-Dokumentation
- Strenge Kontrolle und semantische Sortierung der Variablen-Namen
- Abgleich der selbst berechneten ECLI mit der in der HTML-Fassung dokumentierten ECLI
- Variable für Entscheidungstyp wird nun aus dem Zitiervorschlag berechnet um eine höhere Genaugikeit zu gewährleisten

10.6 Version 2021-05-20

- Vollständige Aktualisierung der Daten
- Einführung eines Debugging-Modus
- Einführung von Variablen für Verfahrensart, Lizenz, Typ der Entscheidung und Zeichenzahl
- Zusätzliche Diagramme für Typ der Entscheidung, Verteilung der Zeichen und Verteilung der Dateigrößen (TXT)
- Neue Datenquellen für Präsident:in, Vize-Präsident:in und für Registerzeichen/Verfahrensarten

- Zusammenfügen von über Zeilengrenzen getrennten Wörtern in der Variable »text« (nur CSV-Formate)
- Einige Verbesserungen im Codebook

10.7 Version 2021-01-08

- 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 Versionsnummer, Concept DOI, Version DOI, ECLI, Entscheidungsnamen, BVerfGE-Band, BVerfGE-Seite, Typ des Spruchkörpers, Präsident:in, Vize-Präsident:in und linguistische Kennzahlen (Tokens, Typen, Sätze)
- 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 »Suffix« in »kollision« umbenannt.
- Variable »Ordinalzahl« in »eingangsnummer« umbenannt.

10.8 Version 2020-08-03

- Vollständige Aktualisierung der Daten
- Angleichung der Variablen-Namen an andere Datensätze der CE-Serie[1]
- Einführung der Variable »Suffix« um weitere Entscheidungen korrekt erfassen zu können; aufgrund der fehlenden Berücksichtigung des Suffix sind die Metadaten von 36 Entscheidungen der Version 2020-06-20 fehlerhaft. Bitte verwenden Sie daher nur die neue Version. Alternativ können Sie die fehlerhaften Dateien (erkennbar an einem dreistelligen Eingangsjahr) aus der Analyse ausschließen oder per Hand korrigieren.

10.9 Version 2020-06-20

- Erstveröffentlichung

[1] Siehe: https://zenodo.org/communities/sean-fobbe-data/

11 Abschluss

```r
## Datumsstempel
print(datestamp)
#> [1] "2024-07-24"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2024-07-24 16:23:40 CEST"


## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2024-07-24 16:24:53 CEST"


## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 1.212563 mins
```

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 4.4.0 (2024-04-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 22.04.4 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;
    LAPACK version 3.10.0
#>
#> locale:
#>  [1] LC_CTYPE=en_US.utf8       LC_NUMERIC=C
#>  [3] LC_TIME=en_US.utf8        LC_COLLATE=en_US.utf8
#>  [5] LC_MONETARY=en_US.utf8    LC_MESSAGES=en_US.utf8
#>  [7] LC_PAPER=en_US.utf8       LC_NAME=C
#>  [9] LC_ADDRESS=C              LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=en_US.utf8 LC_IDENTIFICATION=C
#>
#> time zone: Europe/Berlin
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] ggraph_2.2.1      ggplot2_3.5.1     igraph_2.0.3      kableExtra_1.4.0
#>  [5] knitr_1.48        quanteda_4.0.2    data.table_1.15.4 future_1.33.2
#>  [9] RcppTOML_0.2.2    magrittr_2.0.3    tarchetypes_0.9.0 targets_1.7.1
#>
#> loaded via a namespace (and not attached):
#>  [1] fastmatch_1.1-4     gtable_0.3.5        xfun_0.46
#>  [4] ggrepel_0.9.5       processx_3.8.4      RApiSerialize_0.1.3
#>  [7] lattice_0.22-6      callr_3.7.6         vctrs_0.6.5
#> [10] tools_4.4.0         ps_1.7.7            generics_0.1.3
#> [13] base64url_1.4       parallel_4.4.0      tibble_3.2.1
#> [16] fansi_1.0.6         highr_0.11          pkgconfig_2.0.3
#> [19] Matrix_1.7-0        secretbase_1.0.1    RcppParallel_5.1.8
#> [22] lifecycle_1.0.4     farver_2.1.2        compiler_4.4.0
#> [25] stringr_1.5.1       tinytex_0.52        munsell_0.5.1
#> [28] ggforce_0.4.2       qs_0.26.3           graphlayouts_1.1.1
#> [31] codetools_0.2-20    htmltools_0.5.8.1   yaml_2.3.9
#> [34] pillar_1.9.0        tidyr_1.3.1         MASS_7.3-60.2
#> [37] cachem_1.1.0        viridis_0.6.5       parallelly_1.37.1
#> [40] stopwords_2.3       tidyselect_1.2.1    digest_0.6.36
#> [43] stringi_1.8.4       dplyr_1.1.4         purrr_1.0.2
#> [46] listenv_0.9.1       labeling_0.4.3      polyclip_1.10-6
#> [49] fastmap_1.2.0       grid_4.4.0          colorspace_2.1-0
#> [52] cli_3.6.3           tidygraph_1.3.1     utf8_1.2.4
#> [55] withr_3.0.0         scales_1.3.0        backports_1.5.0
#> [58] rmarkdown_2.27      globals_0.16.3      gridExtra_2.3
```

```
#> [61] stringfish_0.16.0   memoise_2.0.1       evaluate_0.24.0
#> [64] viridisLite_0.4.2   rlang_1.1.4         Rcpp_1.0.13
#> [67] glue_1.7.0          renv_1.0.7          tweenr_2.0.3
#> [70] xml2_1.3.6          svglite_2.1.3       rstudioapi_0.16.0
#> [73] R6_2.5.1            systemfonts_1.1.0   fs_1.6.4
```

Literaturverzeichnis

Allaire, JJ, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, et al. 2024. *Rmarkdown: Dynamic Documents for R*. https://github.com/rstudio/rmarkdown.

Barrett, Tyson, Matt Dowle, Arun Srinivasan, Jan Gorecki, Michael Chirico, and Toby Hocking. 2024. *Data.table: Extension of 'Data.frame'*. https://r-datatable.com.

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.

———. 2024a. *Future.apply: Apply Function to Elements in Parallel Using Futures*. https://future.apply.futureverse.org.

———. 2024b. *Future: Unified Parallel and Distributed Processing in R for Everyone*. https://future.futureverse.org.

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. 2024. *Quanteda: Quantitative Analysis of Textual Data*. https://quanteda.io.

Csárdi, Gábor. 2024. *Zip: Cross-Platform Zip Compression*. https://github.com/r-lib/zip.

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

Csárdi, Gábor, Tamás Nepusz, Vincent Traag, Szabolcs Horvát, Fabio Zanini, Daniel Noom, and Kirill Müller. 2024. *Igraph: Network Analysis and Visualization*. https://r.igraph.org/.

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*.

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. 2024. *Stringi: Fast and Portable Character String Processing Facilities*. https://stringi.gagolewski.com/.

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.

———. 2024a. *Tarchetypes: Archetypes for Targets*. https://docs.ropensci.org/tarchetypes/.

———. 2024b. *Targets: Dynamic Function-Oriented Make-Like Declarative Pipelines*. https://docs.ropensci.org/targets/.

Ooms, Jeroen. 2023. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents*. https://docs.ropensci.org/pdftools/.

Pedersen, Thomas Lin. 2024. *Ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. https://ggraph.data-imaginist.com.

Ushey, Kevin, and Hadley Wickham. 2024. *Renv: Project Environments*. https://rstudio.github.io/renv/.

Wickham, Hadley. 2024. *Rvest: Easily Harvest (Scrape) Web Pages*. https://rvest.tidyverse.org/.

Wijffels, Jan. 2023. *Udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the Udpipe 'Nlp' Toolkit*. https://bnosac.github.io/udpipe/en/index.html.

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/.

———. 2024. *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. 2024. *KableExtra: Construct Complex Table with Kable and Pipe Syntax*. http://haozhu233.github.io/kableExtra/.