

Corpus der Entscheidungen
des
Bundesverfassungsgerichts
(CE-BVerfG-Source)

COMPILATION REPORT

Version 2023-02-26

License MIT-0

DOI: 10.5281/zenodo.7659110

Titel	Source Code des »Corpus der Entscheidungen des Bundesverfassungsgerichts«
Abkürzung	CE-BVerfG-Source
Autor	Seán Fobbe
Version	2023-02-26
Download	https://doi.org/10.5281/zenodo.7659110
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2023). Source Code des »Corpus der Entscheidungen des Bundesverfassungsgerichts« (CE-BVerfG-Source). Version 2023-02-26. Zenodo. DOI: 10.5281/zenodo.7659110.

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 2023-02-26. 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.

Lizenz: MIT No Attribution (MIT-0)

Copyright — 2023 — Seán Fobbe

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the »Software«), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED »AS IS«, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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)	6
1.1	Überblick	6
1.2	Funktionsweise	6
1.3	Systemanforderungen	6
1.4	Anleitung	6
1.4.1	Schritt 1: Ordner vorbereiten	6
1.4.2	Schritt 2: Docker Image erstellen	7
1.4.3	Schritt 3: Datensatz kompilieren	7
1.4.4	Ergebnis	7
1.5	Pipeline visualisieren	7
1.6	Troubleshooting	7
1.7	Projektstruktur	8
1.8	Weitere Open Access Veröffentlichungen (Fobbe)	8
1.9	Kontakt	8
2	Packages laden	9
3	Vorbereitung	10
3.1	Definitionen	10
3.2	Aufräumen	11
3.3	Ordner erstellen	11
3.4	Vollzitate statistischer Software schreiben	11
4	Globale Variablen	12
4.1	Packages definieren	12
4.2	Konfiguration	12
4.3	Funktionen definieren	13
4.4	Metadaten für TXT-Dateien definieren	13
4.5	ZIP-Datei für Source definieren	13
5	Pipeline: Konstruktion	14
5.1	File Tracking Targets	14
5.1.1	Source Code	14
5.1.2	Changelog	14
5.1.3	Variablen für die BVerfGE	14
5.1.4	Liste aller Variablen	14
5.1.5	Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD)	15
5.1.6	Presidents and Vice-Presidents of the Federal Courts of Germany (PVP-FCG)	15
5.2	Download Targets	16
5.2.1	Vorläufige Download-Tabelle erstellen	16
5.2.2	Vorläufige Dateinamen erstellen	16
5.2.3	Finale Dateinamen erstellen	16
5.2.4	Finale Download-Tabelle erstellen	16
5.2.5	Download durchführen (PDF)	16
5.2.6	Download durchführen (HTML)	17
5.3	Convert Targets	17

5.3.1	TXT-Dateien erstellen und einlesen	17
5.3.2	HTML-Dateien parsen	17
5.3.3	HTML Parse-Ergebnis splitten	17
5.4	Enhance Targets	18
5.4.1	Daten standardisieren	18
5.4.2	Variable erstellen: »verfahrensart«	18
5.4.3	Variable erstellen: »aktenzeichen«	18
5.4.4	Variable erstellen: »ecli«	19
5.4.5	Variable erstellen: »praesi«	19
5.4.6	Variable erstellen: »vpraesi«	19
5.4.7	Variablen erstellen: »zeichen, token, typen, saetze«	19
5.4.8	Konstanten erstellen	19
5.4.9	Zusätzliche Variablen zusammenführen	20
5.4.10	Datensatz und zusätzliche Variablen verbinden	20
5.4.11	Hauptdatensatz in segmentierte Variante mergen	20
5.4.12	Finalen Datensatz erstellen	20
5.4.13	Variante erstellen: Nur Metadaten	21
5.4.14	Variante erstellen: Annotiert	21
5.5	Write Targets	21
5.5.1	CSV schreiben: Voller Datensatz	21
5.5.2	CSV schreiben: Metadaten	21
5.5.3	CSV schreiben: Segmentierte Variante	22
5.6	Report Targets	22
5.6.1	LaTeX-Definitionen schreiben	22
5.6.2	Zusammenfassungen linguistischer Kennwerte berechnen	22
5.6.3	Report erstellen: Robustness Checks	22
5.6.4	Report erstellen: Codebook	23
5.7	ZIP Targets	23
5.7.1	ZIP erstellen: Source Code	23
5.7.2	ZIP erstellen: Analyse-Dateien	23
5.7.3	ZIP erstellen: CSV-Datei (voller Datensatz)	24
5.7.4	ZIP erstellen: CSV-Datei (nur Metadaten)	24
5.7.5	ZIP erstellen: CSV-Datei (Segmentiert)	24
5.7.6	ZIP erstellen: PDF-Dateien (alle Entscheidungen)	24
5.7.7	ZIP erstellen: TXT-Dateien	25
5.7.8	ZIP erstellen: HTML-Dateien	25
5.8	Kryptographische Hashes	25
5.8.1	Zu hashende ZIP-Archive definieren	25
5.8.2	Kryptographische Hashes berechnen	25
5.8.3	CSV schreiben: Kryptographische Hashes	26
6	Pipeline: Kompilierung	27
6.1	Durchführen der Kompilierung	27
6.2	Visualisierung	27
7	Pipeline: Analyse	29
7.1	Gesamte Liste	29
7.2	Timing	32
7.2.1	Gesamte Laufzeit	32
7.2.2	Laufzeit einzelner Targets	32

7.3	Warnungen	35
7.3.1	files.pdf	35
7.3.2	report.codebook	36
7.3.3	report.robustness	36
7.4	Fehlermeldungen	37
8	Dateigrößen	38
8.1	ZIP und CSV-Dateien	38
8.2	ZIP-Dateien	38
8.3	CSV-Dateien	39
8.4	PDF-Dateien (MB)	39
8.5	TXT-Dateien (MB)	39
9	Kryptographische Signaturen	40
9.1	Signaturen laden	40
9.2	Leerzeichen hinzufügen um bei SHA3-512 Zeilenumbruch zu ermöglichen . .	40
9.3	In Bericht anzeigen	40
10	Changelog	43
10.1	Version 2023-02-26	43
10.2	Version 2022-08-24	43
10.3	Version 2022-02-01	43
10.4	Version 2021-09-19	44
10.5	Version 2021-05-20	44
10.6	Version 2021-01-08	44
10.7	Version 2020-08-03	44
10.8	Version 2020-06-20	45
11	Abschluss	46
12	Parameter für strenge Replikationen	47
	Literaturverzeichnis	49

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)
- (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
- 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 Konfigurationsdatei 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
├ 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] "2023-02-26"

## 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.7659109"
#> ..$ personendaten: chr "10.5281/zenodo.4568682"
#> ..$ software :List of 2
#> .. ..$ concept: chr "10.5281/zenodo.4308216"
#> .. ..$ version: chr "10.5281/zenodo.7659110"
#> $ 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 "MIT-0"
#> ..$ data: chr "Creative Commons Zero 1.0 Universal"
#> $ 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()$Package,
                "temp/packages.bib")
#> Finding R package dependencies ... Done!
#> Warning in knitr::write_bib(renv::dependencies()$Package, "temp/packages.bib")
:
#> package(s) magick, 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",
                 "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.file1.R.
```

5.1.2 Changelog

```
tar_target(changelog,  
           "CHANGELOG.md",  
           format = "file")  
  
#> Establish _targets.R and _targets_r/targets/tar.file2.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.file3.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.file4.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.file5.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.file6.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.file7.R.

```

5.2 Download Targets

5.2.1 Vorläufige Download-Tabelle erstellen

```
tar_target(dt.download,  
           f.download_table_make(debug.toggle = config$debug$toggle,  
                                 debug.pages = config$debug$pages))  
#> Establish _targets.R and _targets_r/targets/tar.download1.R.
```

5.2.2 Vorläufige Dateinamen erstellen

```
tar_target(filenamees.raw,  
           f.filenamees_raw(url.pdf = dt.download$url_pdf))  
#> Establish _targets.R and _targets_r/targets/tar.download2.R.
```

5.2.3 Finale Dateinamen erstellen

```
tar_target(filenamees.final,  
           f.filenamees_final(filenamees.raw = filenamees.raw,  
                              var.bverfge = dt.var_bverfge))  
#> Establish _targets.R and _targets_r/targets/tar.download3.R.
```

5.2.4 Finale Download-Tabelle erstellen

```
tar_target(dt.download.final,  
           f.download_finalize(dt.download = dt.download,  
                              filenamees.final = filenamees.final))  
#> Establish _targets.R and _targets_r/targets/tar.download4.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.download5.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.download6.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.convert1.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.convert2.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.convert3.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.enhance1.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.enhance2.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.enhance3.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.enhance4.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.enhance5.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.enhance6.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.enhance8.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,
```

```

    bverfg.datecleaned)),])
#> Establish _targets.R and _targets_r/targets/tar.enhance9.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.enhance10.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.enhance11.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.enhance12.R.

```

5.4.12 Finalen Datensatz erstellen

Die Verbesserungen der vorherigen Schritte werden in dieser Funktion zusammengefügt um den finalen Datensatz 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.enhance13.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.enhance14.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 Write Targets

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

5.5.1 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.write1.R.
```

5.5.2 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.write2.R.
```

5.5.3 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.write3.R.
```

5.6 Report Targets

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

5.6.1 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.report1.R.
```

5.6.2 Zusammenfassungen linguistischer Kennwerte berechnen

```
tar_target(lingstats.summary,
  f.lingstats_summary(dt.bverfg.final,
    germanvars = TRUE))
#> Establish _targets.R and _targets_r/targets/tar.report2.R.
```

5.6.3 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.report3.R.
```

5.6.4 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.report4.R.
```

5.7 ZIP Targets

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

5.7.1 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.zip1.R.
```

5.7.2 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.zip2.R.
```

5.7.3 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.zip3.R.
```

5.7.4 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.zip4.R.
```

5.7.5 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.zip5.R.
```

5.7.6 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.zip6.R.
```


5.7.7 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.zip7.R.
```

5.7.8 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.zip8.R.
```

5.8 Kryptographische Hashes

5.8.1 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.analysis,
    zip.source))
#> Establish _targets.R and _targets_r/targets/tar.hashes1.R.
```

5.8.2 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.hashes2.R.
```

5.8.3 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.hashes3.R.
```

6 Pipeline: Kompilierung

6.1 Durchführen der Kompilierung

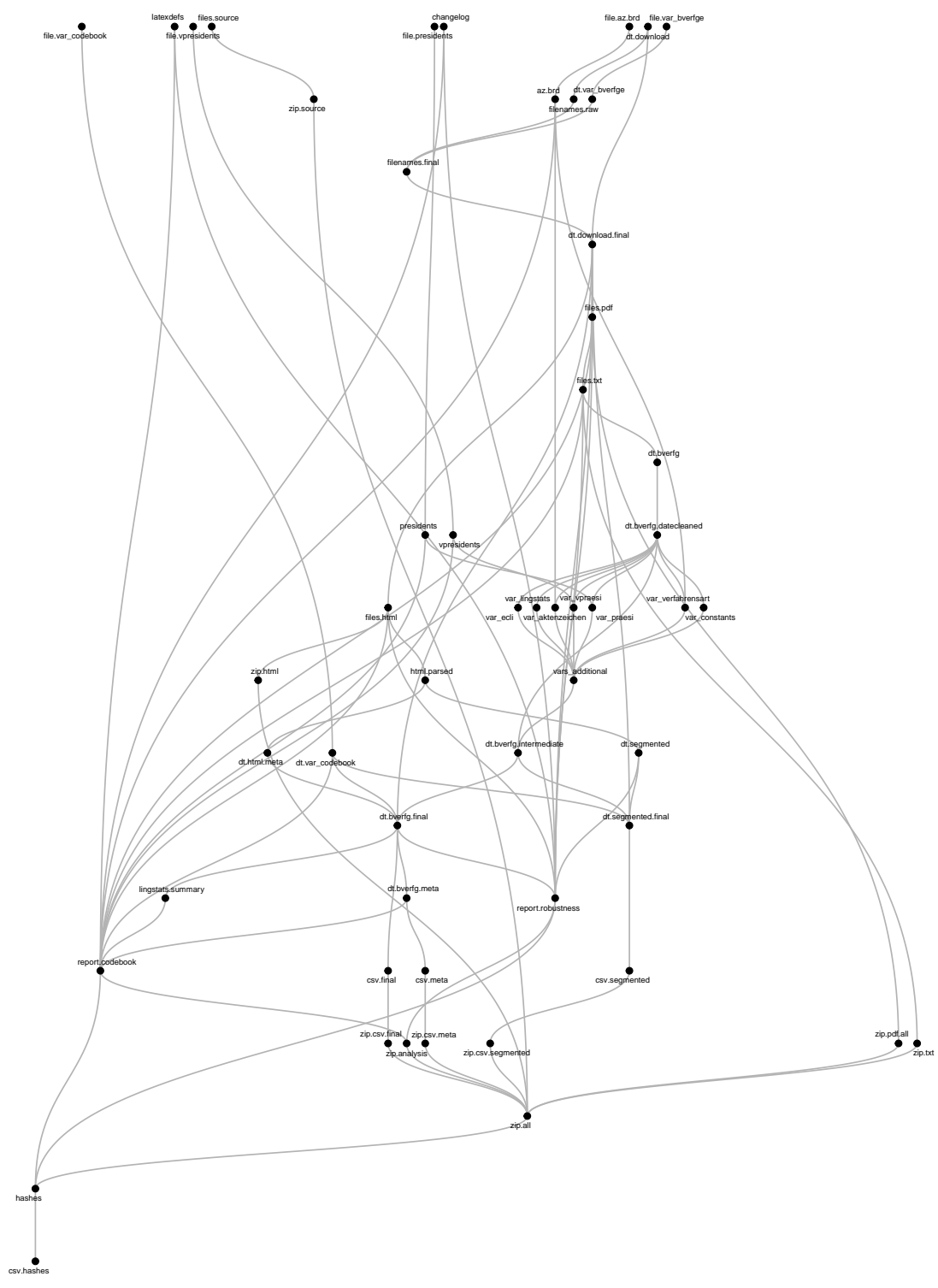
```
tar_make()
```

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 = "grey70")+
  geom_node_point(size = 2)+
  geom_node_text(aes(label = name),
                size = 2,
                repel = TRUE)+
  theme_void()
#> Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2
3.4.0.
#> i Please use `linewidth` in the `default_aes` field and elsewhere instead.
```



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] 2659.47

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	5115	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	47272446	qs	47.27
	dt.bverfg.datecleaned	stem	47274541	qs	47.27
	dt.bverfg.final	stem	48044541	qs	48.04
	dt.bverfg.intermediate	stem	47428843	qs	47.43
	dt.bverfg.meta	stem	893026	qs	0.89
	dt.download	stem	128837	qs	0.13
	dt.download.final	stem	207712	qs	0.21
	dt.html.meta	stem	549474	qs	0.55
	dt.segmented	stem	45048322	qs	45.05
	dt.segmented.final	stem	46297830	qs	46.30

(continued)

	name	type	bytes	format	MB
	dt.var_bverfge	stem	19370	qs	0.02
	dt.var_codebook	stem	3818	qs	0.00
	file.az.brd	stem	36533	file	0.04
	file.presidents	stem	7249	file	0.01
	file.var_bverfge	stem	62643	file	0.06
	file.var_codebook	stem	10151	file	0.01
	file.vpresidents	stem	9193	file	0.01
	filenames.final	stem	78530	qs	0.08
	filenames.raw	stem	56228	qs	0.06
	files.html	stem	513401456	file	513.40
	files.pdf	stem	699118546	file	699.12
	files.source	stem	680299	file	0.68
	files.txt	stem	174876356	file	174.88
	hashes	stem	1509	qs	0.00
	html.parsed	stem	45608188	qs	45.61
	latexdefs	stem	1291	file	0.00
	lingstats.summary	stem	390	qs	0.00
	presidents	stem	1959	qs	0.00
	report.codebook	stem	613986	file	0.61
	report.robustness	stem	460801	file	0.46
	var_aktENZEICHEN	stem	33248	qs	0.03
	var_constants	stem	5817	qs	0.01
	var_ecli	stem	56657	qs	0.06
	var_lingstats	stem	58018	qs	0.06
	var_praesi	stem	165	qs	0.00
	var_verfahrensart	stem	5769	qs	0.01
	var_vpPraesi	stem	183	qs	0.00
	vars_additional	stem	153785	qs	0.15
	vpresidents	stem	2451	qs	0.00

(continued)

name	type	bytes	format	MB
zip.all	stem	174	qs	0.00
zip.analysis	stem	3044964	file	3.04
zip.csv.final	stem	49331059	file	49.33
zip.csv.meta	stem	1047114	file	1.05
zip.csv.segmented	stem	55413512	file	55.41
zip.html	stem	108360808	file	108.36
zip.pdf.all	stem	663265153	file	663.27
zip.source	stem	191123	file	0.19
zip.txt	stem	60286311	file	60.29

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] 16829.4

## Minuten
runtime.sum / 60
#> [1] 280.49

## Stunden
runtime.sum / 3600
#> [1] 4.674834
```

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	2023-02-26 22:36:26	8784.962	146.42
	files.html	2023-02-26 20:11:03	5442.290	90.70
	dt.download	2023-02-26 18:40:20	1999.978	33.33
	html.parsed	2023-02-26 22:42:28	298.045	4.97
	var_lingstats	2023-02-26 22:45:55	62.354	1.04
	lingstats.summary	2023-02-26 22:46:59	52.647	0.88
	dt.segmented	2023-02-26 22:44:09	47.997	0.80
	dt.bverfg	2023-02-26 22:44:48	28.594	0.48
	zip.pdf.all	2023-02-26 22:43:20	18.071	0.30
	zip.html	2023-02-26 22:42:45	16.800	0.28

(continued)

	name	time	seconds	mins
	files.txt	2023-02-26 22:43:01	15.867	0.26
	report.codebook	2023-02-26 22:47:22	12.095	0.20
	zip.csv.segmented	2023-02-26 22:47:33	10.946	0.18
	zip.csv.final	2023-02-26 22:47:10	10.113	0.17
	zip.txt	2023-02-26 22:44:19	9.437	0.16
	report.robustness	2023-02-26 22:46:07	8.876	0.15
	var_aktENZEICHEN	2023-02-26 22:44:52	3.084	0.05
	hashes	2023-02-26 22:47:36	3.045	0.05
	file.az.brd	2023-02-22 13:33:31	1.283	0.02
	dt.html.meta	2023-02-26 22:44:10	0.909	0.02
	filenames.raw	2023-02-26 18:40:21	0.412	0.01
	dt.segmented.final	2023-02-26 22:45:58	0.284	0.00
	filenames.final	2023-02-26 18:40:21	0.254	0.00
	zip.csv.meta	2023-02-26 22:47:33	0.204	0.00
	dt.bverfg.final	2023-02-26 22:45:56	0.186	0.00
	csv.segmented	2023-02-26 22:47:00	0.144	0.00
	zip.analysis	2023-02-26 22:47:33	0.120	0.00
	var_ecli	2023-02-26 22:44:52	0.104	0.00
	csv.final	2023-02-26 22:46:07	0.075	0.00
	var_praesi	2023-02-26 22:45:55	0.067	0.00
	dt.download.final	2023-02-26 18:40:21	0.045	0.00
	var_vpPraesi	2023-02-26 22:45:55	0.037	0.00
	zip.source	2023-02-26 18:40:21	0.027	0.00
	latexdefs	2023-02-26 18:40:20	0.016	0.00
	csv.meta	2023-02-26 22:47:00	0.007	0.00
	dt.bverfg.datecleaned	2023-02-26 22:44:49	0.006	0.00
	dt.bverfg.meta	2023-02-26 22:45:58	0.004	0.00
	var_constants	2023-02-26 22:44:49	0.004	0.00
	az.brd	2023-02-26 18:40:21	0.003	0.00

(continued)

	name	time	seconds	mins
	var_verfahrensart	2023-02-26 22:44:49	0.003	0.00
	dt.var_bverfge	2023-02-26 18:40:21	0.002	0.00
	csv.hashes	2023-02-26 22:47:36	0.001	0.00
	dt.bverfg.intermediate	2023-02-26 22:45:56	0.001	0.00
	file.presidents	2023-02-22 13:33:31	0.001	0.00
	presidents	2023-02-26 18:40:21	0.001	0.00
	vars_additional	2023-02-26 22:45:55	0.001	0.00
	vpresidents	2023-02-26 18:40:21	0.001	0.00
	changelog	2023-02-26 17:37:37	0.000	0.00
	dt.var_codebook	2023-02-26 18:40:21	0.000	0.00
	file.var_bverfge	2023-02-24 16:54:41	0.000	0.00
	file.var_codebook	2023-02-22 13:33:31	0.000	0.00
	file.vpresidents	2023-02-22 13:33:31	0.000	0.00
	files.source	2023-02-26 17:49:31	0.000	0.00
	zip.all	2023-02-26 22:47:33	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

URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/10/rk20141009_2bvr287410.pdf

__blobpublicationFilev1 status was Failure when receiving data from the peer. cannot open

URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/06/rk20140610_1bvr066914.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/03/rk20140319_1bvr141710.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2004/05/rk20040519_1bvr071104.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2001/10/rk20011023_2bvr123601.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2001/10/bs20011001_2bvb000101.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/06/rk20140610_1bvr066914.pdf

__blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/03/rk20140319_1bvr141710.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2004/05/rk20040519_1bvr0711104.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2001/10/rk20011023_2bvr123601.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2001/10/bs20011001_2bvb000101.pdf

___blobpublicationFilev1 HTTP status was 404 Not Found. cannot open URL <https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/2014/06/rk201406r>

7.3.2 report.codebook

Package microtype Warning Unable to apply patch footnote on input line 193.

7.3.3 report.robustness

The file CEBVerfG_2023/02/26_RobustnessChecks.log is not encoded in UTF8. These lines contain invalid UTF8 characters 1342, 1343, 1348, 1354, 1364, 1369, 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_2023-02-26_DE_Analyse.zip	3.04
CE-BVerfG_2023-02-26_DE_CSV_Datensatz.zip	49.33
CE-BVerfG_2023-02-26_DE_CSV_Metadaten.zip	1.05
CE-BVerfG_2023-02-26_DE_CSV_Segmentiert.zip	55.41
CE-BVerfG_2023-02-26_DE_HTML_Datensatz.zip	108.36
CE-BVerfG_2023-02-26_DE_PDF_Datensatz.zip	663.27
CE-BVerfG_2023-02-26_DE_TXT_Datensatz.zip	60.29
CE-BVerfG_2023-02-26_Source_Code.zip	0.19

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_2023-02-26_DE_CSV_Datensatz.csv	180.29
CE-BVerfG_2023-02-26_DE_CSV_Metadaten.csv	8.01
CE-BVerfG_2023-02-26_DE_CSV_Segmentiert.csv	425.50
CE-BVerfG_2023-02-26_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] 699.1185
```

8.5 TXT-Dateien (MB)

```
tar_load(files.txt)
txt.MB <- file.size(files.txt) / 10^6
sum(txt.MB)
#> [1] 174.8764
```

9 Kryptographische Signaturen

9.1 Signaturen laden

```
tar_load(hashses)
```

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.

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

9.3 In Bericht anzeigen

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

index	filename
1	output/CE-BVerfG_2023-02-26_DE_PDF_Datensatz.zip
2	output/CE-BVerfG_2023-02-26_DE_TXT_Datensatz.zip
3	output/CE-BVerfG_2023-02-26_DE_HTML_Datensatz.zip
4	output/CE-BVerfG_2023-02-26_DE_CSV_Datensatz.zip
5	output/CE-BVerfG_2023-02-26_DE_CSV_Metadaten.zip
6	output/CE-BVerfG_2023-02-26_DE_CSV_Segmentiert.zip
7	output/CE-BVerfG_2023-02-26_DE_Analyse.zip
8	output/CE-BVerfG_2023-02-26_Source_Code.zip
9	output/CE-BVerfG_2023-02-26_Codebook.pdf
10	output/CE-BVerfG_2023-02-26_RobustnessChecks.pdf

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

index	sha2.256
1	09fa1fe61e5f3d045e11d2a57030e017f982e7809ba7b5c474e4df605d4c5bc3
2	7a94b7c7d2367e9349e6f40cf99946ee86e0c1f2b9b40158dcd75ed1be598bd4
3	79a98dbaa09ba29a62c8701f5efa006d2ee70c2332e93272b6881e3e5c29aece
4	1c086c5bf211f93383479a5f9d764cded2fab9767f637f0516b2db40376feefe
5	ab9917b3d513a4e7bf969900c1b6c57fcf8c854b9763869c236cbfd310aef678
6	a2164dca98492a92b7e06b794dec4b655cdad36383383e967b662478788392ef
7	7fdaa449010acba5a42d0f0eb308393f913f80c71aff359eed2df298acf821a
8	424e1e430bf7397a29fefe2ca740e7d087804c0404ff35be850a56f6194b8114
9	0ae553d3d0d0115dfd3f3e05a8598341538d627a83ffd5baeb5b2551bf83b3d5
10	9b0a263b1409bb44a95f85009eb4d32a1709683bb08b153df4b2dd0aa050a6f4

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

index	sha3.512
1	6989441cc747a665c8931a3781cd34dfadfa73c5a38cd5ff18f8e6694d250dd4 6db46c937bd61c1e4f763284e93e621e0e6c2eb743805d35be1a52f59257425b
2	8490e61233ced0aa26a4a68e1d2ead46beaed42e448fa4445623e33d297cb6c1 0dfb68f6a0c0ae0571e6fb449e83759078fb9945abfd59015b84870f25971bb9
3	128bb5651282e5ff6504c121ca42b5d228df44429a5612ba1ff52d22cb8ce686 0ff66313c0992ac705201a760b71d9efbc06593f3d3da4953d034720d0fa4920
4	b71cc7d9be30b6094738ab37cc97bc41ee9ff8265c50c4426f5cab56301011e3 99384766f141c83a5e60d9297ec5a5ef218a8dd82e075948ba1019b280348443
5	c883d500f2f85f6e68e83405b03ae719bed8f7e5a8acd103fd5d27f3d86bfdb9 121a9bdc8ed3c805427002c6c5224964e14ba18097569764509ec810cc5e4548
6	46e91b45217ea3cf1ad9ab6d6cfc0a73689db10679daed1cbd1a6deb023011b5 a49dde193f0730867ec91e254fb78bce1f2db99569e2e88d92666fd378392cea

- 7 385fd6538aeafbbdd6029bad50d98b72158662c0911ea44977e662bc9713b7c79edef6121bbb3c3184601ac868c9167896ce17aeef3661100fad5e13f3cdc503
 - 8 1eebe3c1d9db35c07724aa69d338ee2d00dbdba11ad8c68957f22e934e599c45fee08048c42a081e4f9c0119dcf2dba3710598d61dad9fb8b60b4074f93fdb4d
 - 9 63cb545864dae8d9be7711196c7e144aadfca15e7abe460f3167e084d2cc801225b0a9f95a384929e4a947785da7daf9866b18e10ccc2d40b132aca477e07f49
 - 10 c126dac32e928036597833e4eb76dc0a66e70a4df63b097cd1bbbf72308a0584d222d9d48a6c7ad3620bc824c639b0381c5fb8a0ae3a65737dca715eb03d1339
-

10 Changelog

10.1 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

10.2 Version 2022-08-24

- Vollständige Aktualisierung der Daten
- Neuentwurf des 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

10.3 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

10.4 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 Genauigkeit zu gewährleisten

10.5 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

10.6 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.

10.7 Version 2020-08-03

- Vollständige Aktualisierung der Daten

- Angleichung der Variablen-Namen an andere Datensätze der CE-Serie¹
- 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.

10.8 Version 2020-06-20

- Erstveröffentlichung

¹ Siehe: <https://zenodo.org/communities/sean-fobbe-data/>

11 Abschluss

```
## Datumsstempel
print(datestamp)
#> [1] "2023-02-26"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2023-02-26 18:06:57 UTC"

## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2023-02-26 22:47:40 UTC"

## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 4.678588 hours
```

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 22.04.1 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/libopenblas-p-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_1.4.0 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_1.4.4 tidyr_1.3.0
#> [4] tidygraph_1.2.3 viridisLite_0.4.1 RcppParallel_5.1.6
#> [7] highr_0.10 future.callr_0.8.1 base64url_1.4
#> [10] renv_0.16.0 yaml_2.3.7 ggrepel_0.9.3
#> [13] globals_0.16.2 pillar_1.8.1 backports_1.4.1
#> [16] lattice_0.20-45 glue_1.6.2 digest_0.6.31
#> [19] polyclip_1.10-4 rvest_1.0.3 stringfish_0.15.7
#> [22] colorspace_2.1-0 htmltools_0.5.4 Matrix_1.5-1
#> [25] pkgconfig_2.0.3 listenv_0.9.0 purrr_1.0.1
#> [28] scales_1.2.1 webshot_0.5.4 processx_3.8.0
#> [31] svglite_2.1.1 tweenr_2.0.2 RApiSerialize_0.1.2
#> [34] ggforce_0.4.1 tibble_3.1.8 generics_0.1.3
#> [37] farver_2.1.1 withr_2.5.0 furrr_0.3.1
#> [40] cli_3.6.0 magrittr_2.0.3 evaluate_0.20
#> [43] ps_1.7.2 stopwords_2.3 fs_1.6.1
#> [46] fansi_1.0.4 parallelly_1.34.0 MASS_7.3-58.1
#> [49] xml2_1.3.3 tools_4.2.2 lifecycle_1.0.3
#> [52] stringr_1.5.0 munsell_0.5.0 callr_3.7.3
#> [55] compiler_4.2.2 qs_0.25.4 systemfonts_1.0.4
#> [58] rlang_1.0.6 grid_4.2.2 rstudioapi_0.14
#> [61] labeling_0.4.2 rmarkdown_2.20 gtable_0.3.1
#> [64] codetools_0.2-18 graphlayouts_0.8.4 R6_2.5.1
#> [67] gridExtra_2.3 dplyr_1.1.0 fastmap_1.1.0
#> [70] utf8_1.2.3 fastmatch_1.1-3 stringi_1.7.12
```

```
#> [73] parallel_4.2.2      Rcpp_1.0.10        vctrs_0.5.2
#> [76] tidyselect_1.2.0   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*.
- 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. 2022. *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>.
- . 2022a. *Httr: Tools for Working with Urls and Http*.
- . 2022b. *Rvest: Easily Harvest (Scrape) Web Pages*.
- 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*.
- Wijffels, Jan. 2023. *Udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the Udpipe 'Nlp' Toolkit*.
- 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 Golemund. 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*.