

Corpus des Deutschen Bundesrechts (C-DBR)

COMPILATION REPORT

Version 2023-01-05

License MIT-0

DOI: [10.5281/zenodo.7494477](https://doi.org/10.5281/zenodo.7494477)

| | |
|------------------|---|
| Titel | Source Code des »Corpus des Deutschen Bundesrechts« |
| Abkürzung | C-DBR-Source |
| Autor | Seán Fobbe |
| Version | 2023-01-05 |
| Download | https://doi.org/10.5281/zenodo.7494477 |
| Lizenz | MIT No Attribution (MIT-0) |

Zitiervorschlag

Seán Fobbe (2023). Source Code des »Corpus des Deutschen Bundesrechts« (C-DBR-Source). Version 2023-01-05. Zenodo. DOI: 10.5281/zenodo.7494477.

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 2023-01-05. 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.4072934. 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 öffentlichen Stellen der Bundesrepublik Deutschland.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | README: Corpus des Deutschen Bundesrechts (C-DBR) | 6 |
| 1.1 | Überblick | 6 |
| 1.2 | Funktionsweise | 6 |
| 1.3 | Systemanforderungen | 7 |
| 1.4 | Anleitung | 7 |
| 1.4.1 | Schritt 1: Ordner vorbereiten | 7 |
| 1.4.2 | Schritt 2: Installation der Programmiersprache ‘R’ | 7 |
| 1.4.3 | Schritt 3: Installation von ‘renv’ | 7 |
| 1.4.4 | Schritt 4: Installation von R Packages | 7 |
| 1.4.5 | Schritt 5: Installation von LaTeX | 8 |
| 1.4.6 | Schritt 6: Datensatz kompilieren | 8 |
| 1.4.7 | Ergebnis | 8 |
| 1.5 | Pipeline visualisieren | 8 |
| 1.6 | Troubleshooting | 8 |
| 1.7 | Projektstruktur | 9 |
| 1.8 | Weitere Open Access Veröffentlichungen (Fobbe) | 9 |
| 1.9 | Kontakt | 9 |
| 2 | Packages laden | 10 |
| 3 | Vorbereitung | 11 |
| 3.1 | Definitionen | 11 |
| 3.2 | Aufräumen | 12 |
| 3.3 | Ordner erstellen | 12 |
| 3.4 | Vollzitate statistischer Software schreiben | 12 |
| 4 | Globale Variablen | 13 |
| 4.1 | Packages definieren | 13 |
| 4.2 | Konfiguration | 13 |
| 4.3 | Funktionen definieren | 14 |
| 4.4 | ZIP-Datei für Source definieren | 14 |
| 5 | Pipeline: Konstruktion | 15 |
| 5.1 | File Tracking Targets | 15 |
| 5.1.1 | Source Code | 15 |
| 5.1.2 | Changelog | 15 |
| 5.1.3 | Liste aller Variablen im Codebook | 15 |
| 5.2 | Download Targets | 15 |
| 5.2.1 | URLs für XML-Archive | 16 |
| 5.2.2 | Tabelle der Dateinamen erstellen | 16 |
| 5.2.3 | Download Tabelle erstellen | 16 |
| 5.2.4 | Konkordanzabelle erstellen | 16 |
| 5.2.5 | Document Type Definition (DTD) herunterladen | 16 |
| 5.2.6 | XML (ZIP)-Archive herunterladen | 17 |
| 5.2.7 | PDF-Dateien herunterladen | 17 |
| 5.2.8 | EPUB-Dateien herunterladen | 17 |
| 5.3 | Convert Targets | 18 |
| 5.3.1 | Entpacken | 18 |

| | | |
|-----------|---|-----------|
| 5.3.2 | XML-Dateien bestimmen | 18 |
| 5.3.3 | PDF zu TXT konvertieren | 18 |
| 5.4 | Parse Targets | 19 |
| 5.4.1 | Datensatz erstellen: Einzelnormen | 19 |
| 5.4.2 | Datensatz erstellen: Rechtsakte (mit Text) | 19 |
| 5.4.3 | Datensatz erstellen: XML-Metadaten | 19 |
| 5.4.4 | Netzwerk-Analyse | 19 |
| 5.5 | Enhance Targets | 20 |
| 5.5.1 | Variablen erstellen: »zeichen, token, typen, saetze« | 20 |
| 5.5.2 | Finale Datensätze erstellen | 20 |
| 5.5.3 | Varianten erstellen: Nur Metadaten | 20 |
| 5.6 | Write Targets | 21 |
| 5.7 | Report Targets | 22 |
| 5.7.1 | LaTeX-Definitionen schreiben | 22 |
| 5.7.2 | Zusammenfassungen linguistischer Kennwerte berechnen | 22 |
| 5.7.3 | Report erstellen: Robustness Checks | 22 |
| 5.7.4 | Report erstellen: Codebook | 23 |
| 5.8 | ZIP Targets | 23 |
| 5.8.1 | ZIP erstellen: Static Branching | 23 |
| 5.8.2 | ZIP erstellen: Analyse-Dateien | 24 |
| 5.9 | Kryptographische Hashes | 24 |
| 5.9.1 | Zu hashende ZIP-Archive definieren | 25 |
| 5.9.2 | Kryptographische Hashes berechnen | 25 |
| 5.9.3 | CSV schreiben: Kryptographische Hashes | 25 |
| 6 | Pipeline: Kompilierung | 26 |
| 6.1 | Durchführen der Kompilierung | 26 |
| 6.2 | Visualisierung | 26 |
| 7 | Pipeline: Analyse | 28 |
| 7.1 | Gesamte Liste | 28 |
| 7.2 | Timing | 31 |
| 7.2.1 | Gesamte Laufzeit | 31 |
| 7.2.2 | Laufzeit einzelner Targets | 31 |
| 8 | Warnungen | 34 |
| 9 | Fehlermeldungen | 35 |
| 10 | Dateigrößen der Endergebnisse | 36 |
| 10.1 | ZIP-Dateien | 36 |
| 10.2 | CSV-Dateien | 37 |
| 11 | Kryptographische Signaturen | 38 |
| 11.1 | Signaturen laden | 38 |
| 11.2 | Leerzeichen hinzufügen um bei SHA3-512 Zeilenumbruch zu ermöglichen | 38 |
| 11.3 | In Bericht anzeigen | 38 |
| 12 | Changelog | 41 |
| 12.1 | Version 2023-01-05 | 41 |
| 12.2 | Version 2022-08-05 | 41 |

| | | |
|-----------|--|-----------|
| 12.3 | Version 2022-05-22 | 41 |
| 12.4 | Version 2022-01-12 | 41 |
| 12.5 | Version 2021-09-16 | 42 |
| 12.6 | Version 2021-07-30 | 42 |
| 12.7 | Version 2021-01-05 | 42 |
| 12.8 | Version 2020-10-09 | 42 |
| 12.9 | Version 2020-07-08 | 42 |
| 12.10 | Version 2020-05-18 | 43 |
| 13 | Abschluss | 44 |
| 14 | Parameter für strenge Replikationen | 45 |
| | Literaturverzeichnis | 47 |

1 README: Corpus des Deutschen Bundesrechts (C-DBR)

1.1 Überblick

Das **Corpus des deutschen Bundesrechts (C-DBR)** ist eine möglichst vollständige Sammlung der konsolidierten Fassungen aller Gesetze und Verordnungen auf Bundesebene. Der Datensatz nutzt als seine Datenquelle das amtliche Internetangebot www.gesetze-im-internet.de des Bundesministeriums der Justiz und wertet dieses 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.3832111>

Aktuellster, funktionaler und zitierfähiger Release des Source Codes: <https://doi.org/10.5281/zenodo.4072934>

1.2 Funktionsweise

Primäre Endprodukte des Skripts sind folgende ZIP-Archive:

1. Der volle Datensatz im CSV-Format, unterteilt in Einzelnormen (nur Rechtsakte mit veröffentlichtem Normtext)
2. Die Metadaten aller Einzelnormen im CSV-Format (wie 1, aber ohne Text-Variable)
3. Der volle Datensatz im CSV-Format, unterteilt in Rechtsakte (nur Rechtsakte mit veröffentlichtem Normtext)
4. Die Metadaten aller Rechtsakte im CSV-Format (wie 3, aber ohne Text-Variable)
5. Die Metadaten aller veröffentlichten Rechtsakte, im CSV-Format (unabhängig davon ob Normtext veröffentlicht wurde)
6. Der volle Datensatz im XML-Format, unterteilt in Rechtsakte (Originaldaten von GII)
7. Alle Anlagen zu den XML-Dateien im jeweiligen Original-Format (Originaldaten von GII)
8. Alle Rechtsakte im TXT-Format, unterteilt in Rechtsakte (deutlich reduzierter Umfang an Metadaten)
9. Alle Rechtstexte im PDF-Format, unterteilt in Rechtsakte (deutlich reduzierter Umfang an Metadaten)
10. Alle Rechtstexte im EPUB-Format, unterteilt in Gesetze (deutlich reduzierter Umfang an Metadaten)
11. Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)
12. Netzwerk-Strukturen (Adjazenzmatrizen, Edgelisten, GraphML, und Netzwerk-Diagramme) für alle Rechtsakte (experimentell!)

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

- Nur mit Fedora Linux getestet. Vermutlich auch funktionsfähig unter anderen Linux-Distributionen.
- 6 GB Speicherplatz auf Festplatte
- Multi-core CPU empfohlen (die Referenzdatensätze wurden mit 16 threads auf 8 physischen Cores erstellt).

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/c-dbr
```

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: Installation der Programmiersprache ‘R’

Sie müssen die Programmiersprache R und OpenSSL installiert haben. Normalerweise sind diese in Fedora Linux bereits enthalten, andernfalls führen Sie aus:

```
$ sudo dnf install R openssl
```

1.4.3 Schritt 3: Installation von ‘renv’

Starten sie eine R Session in diesem Ordner, sie sollten automatisch zur Installation von `renv` aufgefordert werden. `renv` ist ein Tool zur strengen Versionskontrolle von R packages und sichert die Reproduzierbarkeit.

1.4.4 Schritt 4: Installation von R Packages

Um durch `renv` alle R packages in der benötigten Version zu installieren, führen Sie in der R session aus:

```
> renv::restore() # In einer R-Konsole ausführen
```

Achtung: es reicht nicht, die Packages auf herkömmliche Art installiert zu haben. Sie müssen dies nochmal über `renv` tun, selbst wenn die Packages in der normalen Library schon vorhanden sind.

1.4.5 Schritt 5: Installation von LaTeX

Um die PDF Reports zu kompilieren benötigen Sie eine LaTeX-Installation. Sie können eine vollständige LaTeX-Distribution auf Fedora wie folgt installieren:

```
$ sudo dnf install texlive-scheme-full
```

Alternativ können sie das R package tinytex installieren, welches nur die benötigten LaTeX packages installiert.

```
> install.packages("tinytex") # In einer R-Konsole ausführen
```

Die für die Referenzdatensätze verwendete LaTeX-Installation ist texlive-scheme-full.

1.4.6 Schritt 6: 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:

```
> source("delete_all_data.R") # In einer R-Konsole ausführen
```

Den vollständigen Datensatz kompilieren Sie mit folgendem Befehl:

```
> source("run_project.R") # In einer R-Konsole ausführen
```

1.4.7 Ergebnis

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

1.5 Pipeline visualisieren

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

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

1.6 Troubleshooting

Hilfreiche Befehle um Fehler zu lokalisieren und zu beheben.

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


1.7 Projektstruktur

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

```
.
├ buttons                # Buttons (nur optische Bedeutung)
├ CHANGELOG.md           # Alle Änderungen
├ config.toml            # Zentrale Konfigurations-Datei
├ data                   # Datensätze, auf denen die Pipeline aufbaut
├ delete_all_data.R      # Löscht den Datensatz und Zwischenschritte
├ 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
├ renv                   # Versionskontrolle: Executables
├ renv.lock              # Versionskontrolle: Versionsinformationen
├ reports                # Markdown-Dateien
├ run_project.R          # Kompiliert den gesamten Datensatz
├ _targets_packages.R    # Versionskontrolle: Packages in targets
└ 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)
library(knitr)
library(kableExtra)
library(igraph)
library(ggraph)

tar_unscript()
```

3 Vorbereitung

3.1 Definitionen

```
## Datum
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2023-01-05"

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

## Konfiguration
config <- RcppTOML::parseTOML("config.toml")
print(config)
#> List of 9
#> $ cores :List of 2
#> ..$ max : logi TRUE
#> ..$ number: int 8
#> $ debug :List of 4
#> ..$ cleanrun: logi FALSE
#> ..$ qaSample: int 50
#> ..$ sample : int 500
#> ..$ toggle : logi FALSE
#> $ doi :List of 2
#> ..$ data :List of 2
#> .. ..$ concept: chr "10.5281/zenodo.3832111"
#> .. ..$ version: chr "10.5281/zenodo.7494474"
#> ..$ software:List of 2
#> .. ..$ concept: chr "10.5281/zenodo.4072934"
#> .. ..$ version: chr "10.5281/zenodo.7494477"
#> $ download:List of 1
#> ..$ timeout: int 10
#> $ 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 10
#> ..$ downloadEPUB : logi TRUE
#> ..$ downloadPDF : logi TRUE
#> ..$ downloadXML : logi TRUE
#> ..$ extractPDF : logi TRUE
#> ..$ htmlLandingPages : logi TRUE
#> ..$ lingsummarize : logi TRUE
#> ..$ multihashes : logi TRUE
#> ..$ parseEinzelnormen: logi FALSE
#> ..$ parseMeta : logi FALSE
#> ..$ parseNetworks : logi FALSE
#> $ project :List of 3
#> ..$ author : chr "Seán Fobbe"
#> ..$ fullname : chr "Corpus des Deutschen Bundesrechts"
```

```
#> ..$ shortname: chr "C-DBR"
#> $ 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("files", showWarnings = FALSE)
dir.create("output", showWarnings = FALSE)
dir.create("temp", showWarnings = FALSE)

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

3.4 Vollzitate statistischer Software schreiben

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

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
                             "xml2",          # XML-Extraktion
                             "rvest",         # HTML-Extraktion
                             "knitr",         # Professionelles Reporting
                             "kableExtra",    # Verbesserte Kable Tabellen
                             "pdftools",      # Verarbeitung von PDF-Dateien
                             "ggplot2",       # Datenvisualisierung
                             "ggraph",        # Visualisierung von Graphen
                             "igraph",        # Analyse 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 ZIP-Datei für Source definieren

```

files.source.raw <- c(list.files(pattern = "\\..R$|\\.toml$|\\.md$|\\.Rmd$",
                                "reports",
                                "data",
                                "functions",
                                "tex",
                                "gpg",
                                "buttons",
                                list.files(pattern = "renv\\.lock|\\.Rprofile",
                                           all.files = TRUE),
                                list.files("renv",
                                           pattern = "activate\\.R",
                                           full.names = TRUE)))

#> 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 Liste aller Variablen im Codebook

Die Variablen des Datensatzes, inklusive ihrer Erläuterung.

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

5.2 Download Targets

Es werden von www.gesetze-im-internet.de alle Rechtsakte im XML-, EPUB- und PDF-Format heruntergeladen und auf der Festplatte gespeichert. Die Document Type Definition (DTD) für die XML-Dateien wird ebenfalls archiviert.

5.2.1 URLs für XML-Archive

```
tar_target(url.xml,  
  f.links_xml(url = "https://www.gesetze-im-internet.de/gii-toc.xml"))  
  
#> Establish _targets.R and _targets_r/targets/tar.download.xmlurl.R.
```

5.2.2 Tabelle der Dateinamen erstellen

```
tar_target(dt.fileNames,  
  f.html_landing_pages(url.xml,  
    multicore = config$parallel$htmlLandingPages,  
    cores = fullCores))  
  
#> Establish _targets.R and _targets_r/targets/tar.download.fileNames.R.
```

5.2.3 Download Tabelle erstellen

```
tar_target(dt.download,  
  f.download_table_make(dt.fileNames = dt.fileNames,  
    url.xml = url.xml,  
    xml.toc = "https://www.gesetze-im-internet.de/  
    gii-toc.xml"))  
  
#> Establish _targets.R and _targets_r/targets/tar.download.table.R.
```

5.2.4 Konkordanzabelle erstellen

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

5.2.5 Document Type Definition (DTD) herunterladen

```
tar_target(file.dtd,  
  f.download("https://www.gesetze-im-internet.de/dtd/1.01/gii-norm.dtd",  
    paste0(prefix.files,  
      "_DE_XML_Document-Definition_v1-01.dtd"),  
    dir = "output",  
    clean = FALSE),  
  format = "file")  
  
#> Establish _targets.R and _targets_r/targets/tar.download.dtd.R.
```


5.2.6 XML (ZIP)-Archive herunterladen

```
tar_target(files.xmlzip,
  f.download(url = dt.download$url.xml,
    filename = dt.download$title.xml,
    dir = "files/xml_zip",
    clean = TRUE,
    multicore = config$parallel$downloadXML,
    cores = fullCores,
    sleep.min = 0,
    sleep.max = 0,
    retries = 3,
    retry.sleep.min = 1,
    retry.sleep.max = 2,
    timeout = config$download$timeout,
    debug.toggle = FALSE,
    debug.files = 500),
  format = "file")

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

5.2.7 PDF-Dateien herunterladen

```
tar_target(files.pdf,
  f.download(url = dt.download$url.pdf,
    filename = dt.download$title.pdf,
    dir = "files/pdf",
    clean = TRUE,
    multicore = config$parallel$downloadPDF,
    cores = fullCores,
    sleep.min = 0,
    sleep.max = 0,
    retries = 3,
    retry.sleep.min = 1,
    retry.sleep.max = 2,
    timeout = config$download$timeout,
    debug.toggle = FALSE,
    debug.files = 500),
  format = "file")

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

5.2.8 EPUB-Dateien herunterladen

```
tar_target(files.epub,
  f.download(url = dt.download$url.epub,
    filename = dt.download$title.epub,
    dir = "files/epub",
    clean = TRUE,
    multicore = config$parallel$downloadEPUB,
```

```

cores = fullCores,
sleep.min = 0,
sleep.max = 0,
retries = 3,
retry.sleep.min = 1,
retry.sleep.max = 2,
timeout = config$download$timeout,
debug.toggle = FALSE,
debug.files = 500),
format = "file")

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

```

5.3 Convert Targets

Dieser Abschnitt entpackt die ZIP-Dateien, erstellt ein Target mit allen XML-Dateien und konvertiert die PDF-Dateien in das TXT-Format.

5.3.1 Entpacken

```

tar_target(files.xml.all,
  f.tar_unzip(zipfiles = files.xml.zip,
    exdir = "files/xml"),
  format = "file")

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

```

5.3.2 XML-Dateien bestimmen

```

tar_target(files.xml,
  files.xml.all[grepl("\\.xml$", files.xml.all)],
  format = "file")

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

```

5.3.3 PDF zu TXT konvertieren

```

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

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

```

5.4 Parse Targets

Der Abschnitt zum Parsing extrahiert aus den XML-Dateien alle relevanten Normtexte und Metadaten. Es wird auch eine Netzwerk-Analyse der Struktur der Rechtsakte durchgeführt.

5.4.1 Datensatz erstellen: Einzelnormen

```
tar_target(dt.normen,
  f.dt.einzelnormen(file.xml = files.xml,
    multicore = config$parallel$parseEinzelnormen,
    cores = fullCores))

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

5.4.2 Datensatz erstellen: Rechtsakte (mit Text)

```
tar_target(dt.rechtsakte,
  f.dt.rechtsakte(dt.normen))

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

5.4.3 Datensatz erstellen: XML-Metadaten

Diese Datei unterscheidet sich von der Variante der “Rechtsakte (Metadaten),” weil sie auch Rechtsakte enthält, die ohne Text veröffentlicht wurden. Die Differenz betrifft etwa 1000 Rechtsakte, ist also erheblich.

```
tar_target(dt.meta,
  f.dt.meta(file.xml = files.xml,
    multicore = config$parallel$parseMeta,
    cores = fullCores))

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

5.4.4 Netzwerk-Analyse

```
tar_target(files.network,
  f.network.analysis(files.xml = files.xml,
    prefix.figuretitle = prefix.figuretitle,
    caption = caption,
    dir.out = "netzwerke",
    multicore = config$parallel$parseNetworks,
    cores = fullCores),
  format = "file")

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

5.5 Enhance Targets

Hier werden vereinzelte Verbesserungen vorgenommen und weitere Variablen hinzugefügt. Schließlich werden die geprüften finalen Varianten erstellt.

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

Berechnung klassischer linguistischer Kennzahlen.

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

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

5.5.2 Finale Datensätze erstellen

```
tar_target(dt.normen.final,  
           f.finalize_einzelnormen(dt.normen = dt.normen,  
                                   lingstats = var_lingstats.normen))  
#> Establish _targets.R and _targets_r/targets/tar.enhance.finalize.normen.R.
```

```
tar_target(dt.rechtsakte.final,  
           f.finalize_rechtsakte(dt.rechtsakte = dt.rechtsakte,  
                                   lingstats = var_lingstats.rechtsakte))  
#> Establish _targets.R and _targets_r/targets/tar.enhance.finalize.rechtsakte.R.
```

5.5.3 Varianten erstellen: Nur Metadaten

```
tar_target(dt.normen.meta,  
           dt.normen.final[, !"text"])  
#> Establish _targets.R and _targets_r/targets/tar.enhance.finalize.normen.meta.R.  
.
```

```
tar_target(dt.rechtsakte.meta,
           dt.rechtsakte.final[, !"text"])
#> Establish _targets.R and _targets_r/targets/tar.enhance.finalize.rechtsakte.
      meta.R.
```

5.6 Write Targets

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

```
values <- tibble::tibble(
  name = c("download",
           "conctable",
           "normen",
           "normen_meta",
           "rechtsakte",
           "rechtsakte_meta",
           "xml_meta"),
  input = c(quote(dt.download),
            quote(dt.conctable),
            quote(dt.normen.final),
            quote(dt.normen.meta),
            quote(dt.rechtsakte.final),
            quote(dt.rechtsakte.meta),
            quote(dt.meta)
            ),
  filename = paste0(prefix.files,
                    c("_02_Download-Tabelle",
                      "_DE_Alle-Rechtsakte-Verzeichnis",
                      "_DE_CSV_Einzelnormen_Datensatz",
                      "_DE_CSV_Einzelnormen_Metadaten",
                      "_DE_CSV_Rechtsakte_Datensatz",
                      "_DE_CSV_Rechtsakte_Metadaten",
                      "_DE_CSV_Metadaten-XML_Datensatz"),
                    ".csv"),
  dir = c(dir.analysis,
          rep("output", 6))
)

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

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

5.7.2 Zusammenfassungen linguistischer Kennwerte berechnen

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

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

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

#> Establish _targets.R and _targets_r/targets/tar.report.lingstatsummary.
rechtsakte.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: Static Branching

```

values <- tibble::tibble(
  name = c("source",
    "networks",
    "pdf",
    "txt",
    "epub",
    "xml",
    "attachments",
    "einzelnormen",
    "einzelnormen_meta",
    "rechtsakte",
    "rechtsakte_meta",
    "xml_meta"),
  input = c(quote(files.source),
    quote(files.network),
    quote(files.pdf),
    quote(files.txt),
    quote(files.epub),
    quote(files.xml),
    quote(setdiff(files.xml.all, files.xml)),
    quote(csv_normen),
    quote(csv_normen_meta),
    quote(csv_rechtsakte),
    quote(csv_rechtsakte_meta),
    quote(csv_xml_meta)),
  filename = paste0(prefix.files,
    c("_Source_Code",

```

```

        "_DE_Netzwerke",
        "_DE_PDF_Datensatz",
        "_DE_TXT_Datensatz",
        "_DE_EPUB_Datensatz",
        "_DE_XML_Datensatz",
        "_DE_XML_Anlagen",
        "_DE_CSV_Einzelnormen_Datensatz",
        "_DE_CSV_Einzelnormen_Metadaten",
        "_DE_CSV_Rechtsakte_Datensatz",
        "_DE_CSV_Rechtsakte_Metadaten",
        "_DE_CSV_Metadaten-XML"),
        ".zip"),
    mode = c(rep("mirror", 2),
             rep("cherry-pick", 10))
)

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

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

```

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.

```

5.9 Kryptographische Hashes

Zum Ende hin werden für alle wichtigen Ergebnisse kryptographische Prüfsummen berechnet, die abschließend (außerhalb der Pipeline) mit dem persönlichen GPG-Key von Seán Fobbe signiert werden.

5.9.1 Zu hashende ZIP-Archive definieren

```
tar_target(zip.all,  
           c(zip_source,  
             zip_analysis,  
             zip_networks,  
             zip_pdf,  
             zip_epub,  
             zip_xml,  
             zip_attachments,  
             zip_einzelnormen,  
             zip_einzelnormen_meta,  
             zip_rechtsakte,  
             zip_rechtsakte_meta,  
             zip_xml_meta))  
#> Establish _targets.R and _targets_r/targets/tar.hashes.list.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.compute.R.
```

5.9.3 CSV schreiben: Kryptographische Hashes

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

6 Pipeline: Kompilierung

6.1 Durchführen der Kompilierung

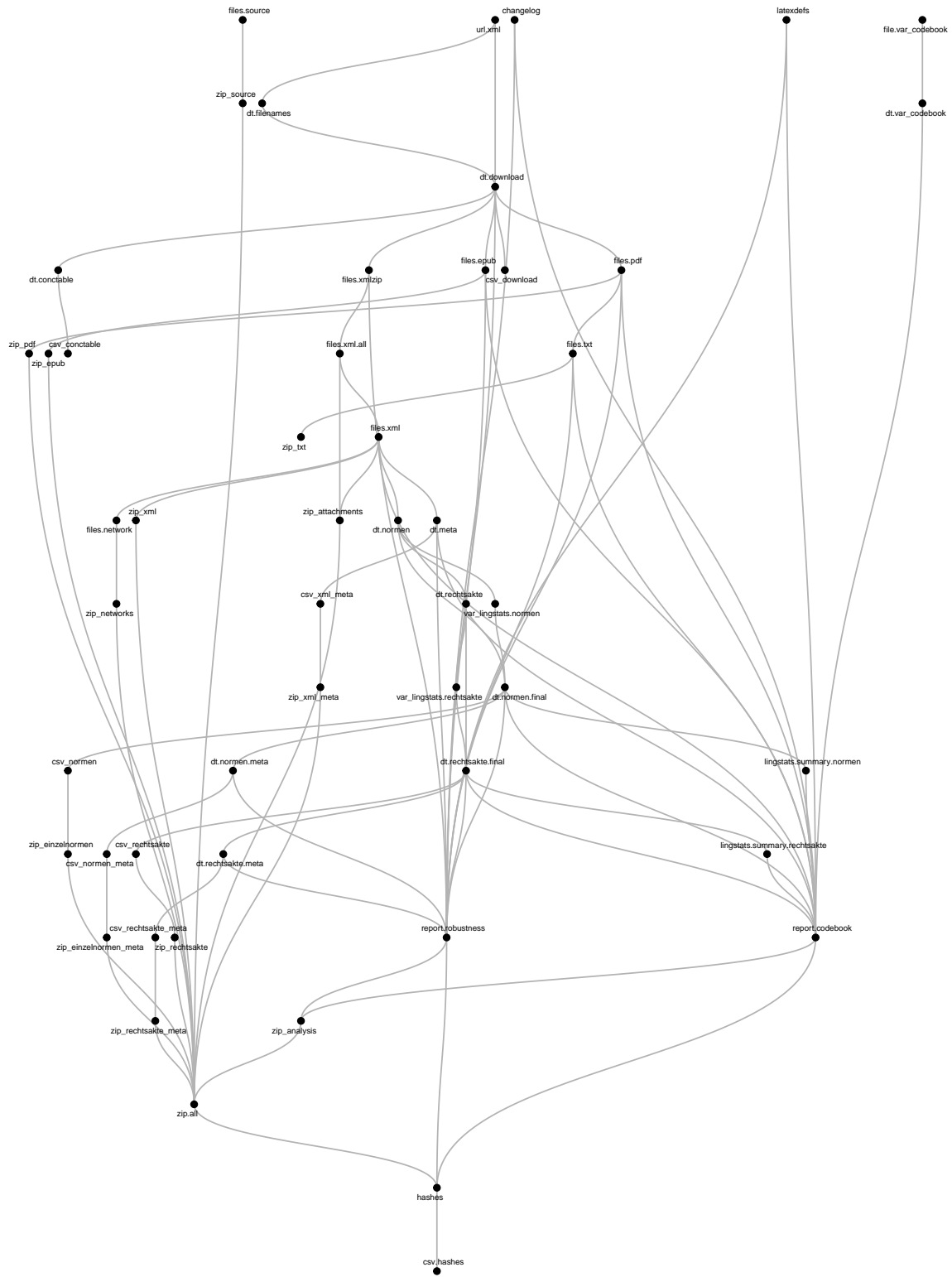
```
tar_make()
```

6.2 Visualisierung

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

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

ggraph(g,
       'sugiyama') +
  geom_edge_diagonal(colour = "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] 4789.75

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

| | name | type | bytes | format | MB |
|--|---------------------|------|-----------|--------|--------|
| | changelog | stem | 4310 | file | 0.00 |
| | csv.hashes | stem | 90 | qs | 0.00 |
| | csv_conctable | stem | 992504 | file | 0.99 |
| | csv_download | stem | 6301992 | file | 6.30 |
| | csv_normen | stem | 220995772 | file | 221.00 |
| | csv_normen_meta | stem | 71315402 | file | 71.32 |
| | csv_rechtsakte | stem | 151998048 | file | 152.00 |
| | csv_rechtsakte_meta | stem | 2510974 | file | 2.51 |
| | csv_xml_meta | stem | 3060525 | file | 3.06 |
| | dt.conctable | stem | 233160 | qs | 0.23 |
| | dt.download | stem | 1319851 | qs | 1.32 |
| | dt.filenames | stem | 73540 | qs | 0.07 |
| | dt.meta | stem | 473075 | qs | 0.47 |
| | dt.normen | stem | 36666176 | qs | 36.67 |
| | dt.normen.final | stem | 37137645 | qs | 37.14 |
| | dt.normen.meta | stem | 2048975 | qs | 2.05 |

(continued)

| | name | type | bytes | format | MB |
|--|------------------------------|------|-----------|--------|--------|
| | dt.rechtsakte | stem | 35683835 | qs | 35.68 |
| | dt.rechtsakte.final | stem | 35720847 | qs | 35.72 |
| | dt.rechtsakte.meta | stem | 413813 | qs | 0.41 |
| | dt.var_codebook | stem | 3615 | qs | 0.00 |
| | file.dtd | stem | 8915 | file | 0.01 |
| | file.var_codebook | stem | 11497 | file | 0.01 |
| | files.epub | stem | 459696538 | file | 459.70 |
| | files.network | stem | 77214163 | file | 77.21 |
| | files.pdf | stem | 653970723 | file | 653.97 |
| | files.source | stem | 119356 | file | 0.12 |
| | files.txt | stem | 206049121 | file | 206.05 |
| | files.xml | stem | 273018839 | file | 273.02 |
| | files.xml.all | stem | 575431604 | file | 575.43 |
| | files.xmlzip | stem | 367489263 | file | 367.49 |
| | hashes | stem | 2012 | qs | 0.00 |
| | latexdefs | stem | 1195 | file | 0.00 |
| | lingstats.summary.normen | stem | 385 | qs | 0.00 |
| | lingstats.summary.rechtsakte | stem | 392 | qs | 0.00 |
| | report.codebook | stem | 664757 | file | 0.66 |
| | report.robustness | stem | 520866 | file | 0.52 |
| | url.xml | stem | 47209 | qs | 0.05 |
| | var_lingstats.normen | stem | 444141 | qs | 0.44 |
| | var_lingstats.rechtsakte | stem | 37669 | qs | 0.04 |
| | zip.all | stem | 229 | qs | 0.00 |
| | zip_analysis | stem | 4226708 | file | 4.23 |
| | zip_attachments | stem | 274993963 | file | 274.99 |
| | zip_einzelnormen | stem | 39820821 | file | 39.82 |
| | zip_einzelnormen_meta | stem | 3014007 | file | 3.01 |
| | zip_epub | stem | 445470201 | file | 445.47 |

(continued)

| name | type | bytes | format | MB |
|---------------------|------|-----------|--------|--------|
| zip_networks | stem | 71652283 | file | 71.65 |
| zip_pdf | stem | 592583407 | file | 592.58 |
| zip_rechtsakte | stem | 35964757 | file | 35.96 |
| zip_rechtsakte_meta | stem | 503272 | file | 0.50 |
| zip_source | stem | 100450 | file | 0.10 |
| zip_txt | stem | 48670963 | file | 48.67 |
| zip_xml | stem | 50555442 | file | 50.56 |
| zip_xml_meta | stem | 554672 | file | 0.55 |

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] 2046.98

## Minuten
runtime.sum / 60
#> [1] 34.11633

## Stunden
runtime.sum / 3600
#> [1] 0.5686056
```

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.network | 2023-01-05 11:57:13 | 696.048 | 11.60 |
| | var_lingstats.normen | 2023-01-05 12:02:39 | 318.803 | 5.31 |
| | dt.normen | 2023-01-05 11:45:23 | 242.413 | 4.04 |
| | files.epub | 2023-01-05 11:34:43 | 105.947 | 1.77 |
| | var_lingstats.rechtsakte | 2023-01-05 12:04:27 | 102.692 | 1.71 |
| | files.pdf | 2023-01-05 11:36:22 | 98.646 | 1.64 |
| | lingstats.summary.rechtsakte | 2023-01-05 12:06:37 | 97.259 | 1.62 |
| | files.xmlzip | 2023-01-05 11:37:54 | 91.409 | 1.52 |
| | dt.meta | 2023-01-05 11:41:20 | 56.189 | 0.94 |
| | dt.filenames | 2023-01-05 11:32:58 | 55.449 | 0.92 |

(continued)

| | name | time | seconds | mins |
|--------------------------|---------------------|--------|---------|------|
| lingstats.summary.normen | 2023-01-05 12:05:00 | 31.121 | 0.52 | |
| report.codebook | 2023-01-05 12:07:08 | 22.228 | 0.37 | |
| zip_pdf | 2023-01-05 11:38:44 | 19.476 | 0.32 | |
| report.robustness | 2023-01-05 12:07:24 | 16.725 | 0.28 | |
| zip_epub | 2023-01-05 11:38:10 | 14.200 | 0.24 | |
| files.txt | 2023-01-05 11:38:23 | 13.363 | 0.22 | |
| zip_txt | 2023-01-05 11:39:01 | 10.247 | 0.17 | |
| zip_attachments | 2023-01-05 11:45:32 | 9.428 | 0.16 | |
| zip_xml | 2023-01-05 11:45:41 | 9.190 | 0.15 | |
| zip_rechtsakte | 2023-01-05 12:07:33 | 8.482 | 0.14 | |
| zip_einzelnormen | 2023-01-05 12:06:45 | 8.221 | 0.14 | |
| files.xml.all | 2023-01-04 21:40:20 | 6.084 | 0.10 | |
| zip_networks | 2023-01-05 12:02:44 | 4.743 | 0.08 | |
| hashes | 2023-01-05 12:07:37 | 2.945 | 0.05 | |
| dt.rechtsakte | 2023-01-05 11:57:20 | 1.440 | 0.02 | |
| changelog | 2023-01-04 13:32:38 | 1.018 | 0.02 | |
| zip_einzelnormen_meta | 2023-01-05 12:07:34 | 0.864 | 0.01 | |
| dt.download | 2023-01-05 11:32:58 | 0.624 | 0.01 | |
| dt.normen.final | 2023-01-05 12:04:28 | 0.341 | 0.01 | |
| url.xml | 2023-01-05 11:32:02 | 0.298 | 0.00 | |
| zip_analysis | 2023-01-05 12:07:34 | 0.281 | 0.00 | |
| file.dtd | 2023-01-05 11:32:02 | 0.154 | 0.00 | |
| csv_normen | 2023-01-05 12:04:28 | 0.147 | 0.00 | |
| zip_xml_meta | 2023-01-05 12:02:44 | 0.119 | 0.00 | |
| dt.rechtsakte.final | 2023-01-05 12:04:28 | 0.098 | 0.00 | |
| zip_rechtsakte_meta | 2023-01-05 12:07:34 | 0.095 | 0.00 | |
| csv_rechtsakte | 2023-01-05 12:06:37 | 0.077 | 0.00 | |
| csv_normen_meta | 2023-01-05 12:06:46 | 0.032 | 0.00 | |
| dt.normen.meta | 2023-01-05 12:05:00 | 0.019 | 0.00 | |

(continued)

| | name | time | seconds | mins |
|--|---------------------|---------------------|---------|------|
| | latexdefs | 2023-01-05 11:32:02 | 0.016 | 0.00 |
| | zip_source | 2023-01-05 11:32:58 | 0.014 | 0.00 |
| | csv_download | 2023-01-05 11:37:56 | 0.006 | 0.00 |
| | dt.conctable | 2023-01-05 11:37:56 | 0.006 | 0.00 |
| | csv_xml_meta | 2023-01-05 11:57:18 | 0.005 | 0.00 |
| | csv_conctable | 2023-01-05 11:38:51 | 0.004 | 0.00 |
| | csv_rechtsakte_meta | 2023-01-05 12:07:08 | 0.004 | 0.00 |
| | files.xml | 2023-01-04 21:40:20 | 0.003 | 0.00 |
| | csv.hashes | 2023-01-05 12:07:37 | 0.002 | 0.00 |
| | dt.rechtsakte.meta | 2023-01-05 12:06:37 | 0.002 | 0.00 |
| | dt.var_codebook | 2023-01-05 11:32:58 | 0.002 | 0.00 |
| | files.source | 2023-01-05 11:09:30 | 0.001 | 0.00 |
| | file.var_codebook | 2022-12-31 05:16:01 | 0.000 | 0.00 |
| | zip.all | 2023-01-05 12:07:34 | 0.000 | 0.00 |

8 Warnungen

```
meta <- tar_meta(fields = "warnings", complete_only = TRUE)
setDT(meta)
meta <- meta[name != "files.network"]

kable(meta,
  format = "latex",
  align = c("P{4cm}", "p{10cm}"),
  booktabs = TRUE,
  longtable = TRUE)
```

| name | warnings |
|-----------------|--|
| report.codebook | Removed 1 rows containing missing values position_stack. |

9 Fehlermeldungen

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

kable(meta,
      format = "latex",
      align = c("P{4cm}", "p{10cm}"),
      booktabs = TRUE,
      longtable = TRUE)
```

| name | error |
|------|-------|
|------|-------|

10 Dateigrößen der Endergebnisse

10.1 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 |
|--|-------------|
| C-DBR_2023-01-05_DE_Analyse.zip | 4.23 |
| C-DBR_2023-01-05_DE_CSV_Einzelnormen_Datensatz.zip | 39.82 |
| C-DBR_2023-01-05_DE_CSV_Einzelnormen_Metadaten.zip | 3.01 |
| C-DBR_2023-01-05_DE_CSV_Metadaten-XML.zip | 0.55 |
| C-DBR_2023-01-05_DE_CSV_Rechtsakte_Datensatz.zip | 35.96 |
| C-DBR_2023-01-05_DE_CSV_Rechtsakte_Metadaten.zip | 0.50 |
| C-DBR_2023-01-05_DE_EPUB_Datensatz.zip | 445.47 |
| C-DBR_2023-01-05_DE_Netzwerke.zip | 71.65 |
| C-DBR_2023-01-05_DE_PDF_Datensatz.zip | 592.58 |
| C-DBR_2023-01-05_DE_TXT_Datensatz.zip | 48.67 |
| C-DBR_2023-01-05_DE_XML_Anlagen.zip | 274.99 |
| C-DBR_2023-01-05_DE_XML_Datensatz.zip | 50.56 |
| C-DBR_2023-01-05_Source_Code.zip | 0.10 |

10.2 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 |
|---|-------------|
| C-DBR_2023-01-05_DE_Alle-Rechtsakte-Verzeichnis.csv | 0.99 |
| C-DBR_2023-01-05_DE_CSV_Einzelnormen_Datensatz.csv | 221.00 |
| C-DBR_2023-01-05_DE_CSV_Einzelnormen_Metadaten.csv | 71.32 |
| C-DBR_2023-01-05_DE_CSV_Metadaten-XML_Datensatz.csv | 3.06 |
| C-DBR_2023-01-05_DE_CSV_Rechtsakte_Datensatz.csv | 152.00 |
| C-DBR_2023-01-05_DE_CSV_Rechtsakte_Metadaten.csv | 2.51 |
| C-DBR_2023-01-05_KryptographischeHashes.csv | 0.00 |

11 Kryptographische Signaturen

11.1 Signaturen laden

```
tar_load(hashes)
```

11.2 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))
```

11.3 In Bericht anzeigen

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

| index | filename |
|-------|---|
| 1 | output/C-DBR_2023-01-05_Source_Code.zip |
| 2 | output/C-DBR_2023-01-05_DE_Analyse.zip |
| 3 | output/C-DBR_2023-01-05_DE_Netzwerke.zip |
| 4 | output/C-DBR_2023-01-05_DE_PDF_Datensatz.zip |
| 5 | output/C-DBR_2023-01-05_DE_EPUB_Datensatz.zip |
| 6 | output/C-DBR_2023-01-05_DE_XML_Datensatz.zip |
| 7 | output/C-DBR_2023-01-05_DE_XML_Anlagen.zip |
| 8 | output/C-DBR_2023-01-05_DE_CSV_Einzelnormen_Datensatz.zip |
| 9 | output/C-DBR_2023-01-05_DE_CSV_Einzelnormen_Metadaten.zip |
| 10 | output/C-DBR_2023-01-05_DE_CSV_Rechtsakte_Datensatz.zip |

- 11 output/C-DBR_2023-01-05_DE_CSV_Rechtsakte_Metadaten.zip
 - 12 output/C-DBR_2023-01-05_DE_CSV_Metadaten-XML.zip
 - 13 output/C-DBR_2023-01-05_Codebook.pdf
 - 14 output/C-DBR_2023-01-05_RobustnessChecks.pdf
-

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

| index | sha2.256 |
|-------|--|
| <hr/> | |
| 1 | 8549626d7106f044a8b5d7e0d6a0d27677cddcb8e685bfbd3d66c494d6ade198 |
| 2 | 35d9c3295ea1866513a77bf93c98687e69c93616d422f4df09a9e905b9b6ec67 |
| 3 | 2f93b9f5871466c48225fd7562c78dbd203e22721d6a9460442e143b002747e9 |
| 4 | cc295470882f7c4dbaa8ac60d7fc8057886940a52723aa42602677daed89f4c7 |
| 5 | 9d8be58ba214a9fabcea0ccd02b8ce85802d8e25d0f198e94563614d70fd16fa |
| 6 | 8137f3fd7e7b4e048d609d0fa96b8cf2262418443572686d1660b6f1faa9b60c |
| 7 | 4309feeabce3e85f3c56e77fb83dfd743c466b786399f5f6c56c69a9cfe6a8b8 |
| 8 | 3925f368a5c4856783503cd43e104ca2c91d1002f7db48217eda3aa2090d828b |
| 9 | 66819982e018b233ddba9b85c545ef12bdd59bf882dfba988acb7aa8e8e3f06e |
| 10 | e9eb3b5a22c14f30b56e83f0c03225cef6255799753895a73047b0746b28eb59 |
| 11 | 53d93ee4b99c740ab5da53156714dc8b42012f8f86a8a08261101bf9be50c74a |
| 12 | 3ab7be05f6344257a8abdc6f97fce4b359976153a9ba7df51974afcfab30fa3f |
| 13 | 5212af4ea2c33ea993022cf7b9066021aecf4fe3833697f7afb4f0a5786e9e6c |
| 14 | 98d842d3cfdd5916f8750e31cddd3dba1c069f253d4e22376e301830e7364cd3 |

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

| index | sha3.512 |
|-------|--|
| 1 | 284d319ab190f8b70bf538f3df2b7b8cbb125386cd83ee1834ff6774208ac844be9a91b5a2e83438490d4642aa49c834efb9b3371fe68b31bb0fd1f2318d6e22 |
| 2 | f981cff4b2d0f8815bae67c708860e8ccda6f036fe6c25eec68039a83534e4d82cb670abc2c40f15a541678ecb85d6ff4183fbd4017d6cc880a3327ab5007015 |
| 3 | de311bcd4f430648b37fdc4945876f6e86fa1ab93936cef7fbd2aa68f2d37b9a4bb743bc9abb9b0ad7f8a899e1f8f21d84999d1510fa3441323a6fba37aed37 |
| 4 | 8ffc4e12572ac751b819b6eb38a3be126fab2aeb6ba1ac251a20f28d096bc19f347e9ec862a50f0a7a10acc016a4a08ec055cef411dab5e6cad970c138a67686 |
| 5 | d65589a067f2ad6262d7423f93c53f2a5e71521cdef449ec52189e3a48fce15d54a3d6f6a6a7a2feec29db07f42231349436b2c1308fece5bbab759f95994def |
| 6 | 5c832f1105bda901ac30b4aa211fff8fdf599cd7b33072d9faadc46d5f55a0de0a917dfd442de12e5d3304e156acf2c36a3babf124e5fc2b1719bac0210cf478 |
| 7 | bda2af8e08ba842829ea2df1342771bded29ae7dc3a1feb6bc40356aa3c05e53046edd900fa9d215f0a6346180fcd033d30b862270778e13a14e171195ee4b0d |
| 8 | 3c35e7318c9d6ee351d48d360d75cf11643dd8424c596bb7ff0222c6aaa4470b2216dae27714e1d9142e1780fc7cf11761b36ea0142c7459d1b93a0b00e3612d |
| 9 | 5a392ae14059fe621bda0d3ae930243adba0c86967b770c893840243aef5d2314319869ddfe4864069b4d6d3da07cc2cc2ed80bd527a745f4b6a6a2635bd4660 |
| 10 | a74139051445fff80aa367ef9ba8c458a11d137d91ab86fa92b88049e64a12b9294ceb4f61b2481ff85607fe0be9d1e19955121899e8c86a625fecabbbb700de |
| 11 | 65eee4235cc4a000ff58bd33618ed55125a2e6d19ecdb942f6abef052060a63d44f112333a0a6a940acdd5886ad5020ca5ee84185a4b0b98362081a632643666 |
| 12 | 7fc16829fc343b9077b1c3a733bcf3870191948baf0ccb51ecb49e0023f23d301fc8bdb942de2cb2fb1dc62a76919a7d792bd0f5d384c8b6b046bebe8e04838f |
| 13 | 23b3def4d24283115adc51dcbef5f778d33f10e0252257d4983c60e42fc6168af21b3a3bef26f2e04356e9d37af73f3fa3cbb1bdde3652d222484640f835cebf |
| 14 | 8da797d773291afe42837bb9f9ac0ee7208c27e762f16d4a6a2055e14a8278d347c40520a2d6700b3203d00f3dad2ad3b0e1e8d4b2272eddd954aa00f0075420 |

12 Changelog

12.1 Version 2023-01-05

- Vollständige Aktualisierung der Daten
- Neuer Entwurf des gesamten Source Codes im {targets} Framework
- Zusätzliche Netzwerk-Diagramme für alle Rechtsakte: Sunburst und Circlepacking
- Reguläre Netzwerk-Diagramme nun in blau auf schwarzem Hintergrund
- Manche finale Dateinamen nun mit Trennstrichen statt Pascal Case
- TXT-Konvertierung bricht bei Fehler nicht ab, dokumentiert aber fehlende TXT-Dateien
- Einführung eines separaten Berichts für Robustness Checks

12.2 Version 2022-08-05

- Vollständige Aktualisierung der Daten
- Wenn der Download einer Datei scheitert wird der Kompilierungs-Prozess nicht mehr abgebrochen; Kontrolle über Datenabgleich im Compilation Report
- Diagramme für Norm/Rechtsakt/Metadaten je Periodikum sind nun logarithmisch skaliert
- Technischer Bugfix bei der Berechnung von Netzwerkdiagrammen
- Neuer Unit Test um identische Länge von HTML-Links und extrahierten PDF- und EPUB-Dateinamen
- Fehlende PDF- oder EPUB-Dateien führen nun nicht mehr zu Fehlern in der Pipeline
- Unterscheidung zwischen VBVG 2005 und VBVG 2023

12.3 Version 2022-05-22

- Vollständige Aktualisierung der Daten
- README und CHANGELOG sind nun externe Dateien die bei der Kompilierung automatisch eingebunden werden
- Das für *renv* notwendige Skript *activate.R* ist im ZIP-Archiv in den Ordner “*renv*” sortiert

12.4 Version 2022-01-12

- Vollständige Aktualisierung der Daten
- Strenge Versionskontrolle aller R packages
- Der Prozess der Kompilierung ist jetzt detailliert konfigurierbar, insbesondere die Parallelisierung
- Parallelisierung der XML-Parser deaktiviert, weil instabil
- Parallelisierung nun vollständig mit *future* statt mit *foreach* und *doParallel*
- Fehlerhafte Kompilierungen werden beim vor der nächsten Kompilierung vollautomatisch aufgeräumt
- Alle Ergebnisse werden automatisch fertig verpackt in den Ordner »output« sortiert
- Source Code des Changelogs zu Markdown konvertiert
- Einführung eines Debugging-Modus um die Entwicklung zu beschleunigen

12.5 Version 2021-09-16

- Vollständige Aktualisierung der Daten
- Einfügung von Kurzbezeichnungen der Rechtsakte in die Dateinamen der Netzwerkanalysen
- Einfügung der ID der Rechtsakte in die CSV-Tabelle aller Kurz- und Langtitel

12.6 Version 2021-07-30

- Vollständige Aktualisierung der Daten
- Einführung von neuen Variablen für letzte Änderung (Datum), Neufassung (Datum), Aufhebung (Datum jeweils für Verkündung und Wirkung), Lizenz und hierarchische Ketten von Gliederungsbezeichnungen und -titeln
- Parallelisierung der Downloads um Kompilierung des Korpus zu beschleunigen
- Korrektur bei den Dateinamen der Allgemeinen Eisenbahngesetze: GII weist zwei gleichnamige Rechtsakte (»Allgemeines Eisenbahngesetz«) nach. Beide werden nun mit dem Jahr ihrer Ausfertigung 1951 und 1993 im Langtitel differenziert. In der Vorversion wurde das neuere AEG noch mit dem Jahr 1994 (Inkrafttreten) beschriftet und das andere AEG ohne Jahreszahl.
- Einführung von Netzwerkanalysen (experimentell!)
- Variablen in CSV-Dateien sind nun semantisch sortiert
- Neues Diagramm für Verteilung von Zeichen
- Falls die XML-Datei mehrere Bemerkungen für Hinweise, Änderung, Neufassung, den Stand oder sonstige Angaben aufweist werden diese nun durch einen vertikalen Strich getrennt (vorher nur mehrere Leerzeichen).
- Kleinere Korrekturen und Ergänzungen im Codebook

12.7 Version 2021-01-05

- Vollständige Aktualisierung der Daten
- Komplette Überarbeitung des Source Codes
- Erstveröffentlichung eines Codebooks
- Einführung der vollautomatischen Erstellung von Datensatz und Codebook
- Einführung von Compilation Reports um den Erstellungsprozess exakt zu dokumentieren
- CSV-Dateien werden durch Parsing der XML-Dateien erstellt
- Automatisierung und deutliche Erweiterung der Qualitätskontrolle
- Einführung von Diagrammen zur Visualisierung von Prüfergebnissen
- Einführung kryptographischer Signaturen

12.8 Version 2020-10-09

- Vollständige Aktualisierung der Daten
- Erstveröffentlichung des Source Codes
- XML-Daten nun fehlerfrei. In Version 2020-07-08 waren XML-Dateien mit Anhängen fehlerhaft.

12.9 Version 2020-07-08

- Vollständige Aktualisierung der Daten

12.10 Version 2020-05-18

- Erstveröffentlichung

13 Abschluss

```
## Datumsstempel
print(datestamp)
#> [1] "2023-01-05"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2023-01-05 11:40:20 CET"

## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2023-01-05 12:07:41 CET"

## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 27.36108 mins
```

14 Parameter für strenge Replikationen

```
system2("openssl", "version", stdout = TRUE)
#> [1] "OpenSSL 3.0.5 5 Jul 2022 (Library: OpenSSL 3.0.5 5 Jul 2022)"

sessionInfo()
#> R version 4.1.3 (2022-03-10)
#> Platform: x86_64-redhat-linux-gnu (64-bit)
#> Running under: Fedora Linux 36 (Workstation Edition)
#>
#> Matrix products: default
#> BLAS/LAPACK: /usr/lib64/libflexiblas.so.3.2
#>
#> 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
#>
#> attached base packages:
#> [1] stats      graphics  grDevices  utils      datasets  methods   base
#>
#> other attached packages:
#>  [1] gggraph_2.1.0      ggplot2_3.4.0      igraph_1.3.5      kableExtra_1.3.4
#>  [5] knitr_1.41         quanteda_3.2.3     data.table_1.14.6 future_1.29.0
#>  [9] RcppTOML_0.1.7     magrittr_2.0.3     tarchetypes_0.7.2 targets_0.14.0
#>
#> loaded via a namespace (and not attached):
#>  [1] viridis_0.6.2      httr_1.4.4         tidyr_1.2.1
#>  [4] tidygraph_1.2.2    viridisLite_0.4.1  RcppParallel_5.1.5
#>  [7] highr_0.9          base64url_1.4       renv_0.16.0
#> [10] yaml_2.2.1         ggrepel_0.9.2      globals_0.16.1
#> [13] pillar_1.8.1       backports_1.4.1    lattice_0.20-45
#> [16] glue_1.6.2         digest_0.6.30      polyclip_1.10-4
#> [19] rvest_1.0.3        stringfish_0.15.7  colorspace_2.0-3
#> [22] htmltools_0.5.4    Matrix_1.4-0       pkgconfig_2.0.3
#> [25] listenv_0.8.0      purrr_0.3.5        scales_1.2.1
#> [28] webshot_0.5.4      processx_3.8.0     svglite_2.1.0
#> [31] tweenr_2.0.2       RApiSerialize_0.1.2 ggforce_0.4.1
#> [34] tibble_3.1.8       generics_0.1.3     farver_2.1.1
#> [37] withr_2.5.0        cli_3.4.1          evaluate_0.18
#> [40] ps_1.7.2           stopwords_2.3       fs_1.5.2
#> [43] fansi_1.0.3        parallelly_1.32.1  MASS_7.3-55
#> [46] xml2_1.3.3         tools_4.1.3        lifecycle_1.0.3
#> [49] stringr_1.4.1      munsell_0.5.0      callr_3.7.3
#> [52] compiler_4.1.3     qs_0.25.4          systemfonts_1.0.4
#> [55] rlang_1.0.6        grid_4.1.3         rstudioapi_0.14
#> [58] labeling_0.4.2     rmarkdown_2.18     gtable_0.3.1
#> [61] codetools_0.2-18   graphlayouts_0.8.3 R6_2.5.1
#> [64] gridExtra_2.3      dplyr_1.0.10       fastmap_1.1.0
#> [67] utf8_1.2.2         fastmatch_1.1-3    stringi_1.6.2
#> [70] parallel_4.1.3     Rcpp_1.0.9         vctrs_0.5.1
#> [73] tidyselect_1.2.0   xfun_0.35
```

Literaturverzeichnis

- Allaire, JJ, Jeffrey Horner, Yihui Xie, Vicent Marti, and Natacha Porte. 2019. *Markdown: Render Markdown with the c Library Sundown*. <https://github.com/rstudio/markdown>.
- Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2022. *Rmarkdown: Dynamic Documents for r*. <https://CRAN.R-project.org/package=rmarkdown>.
- Almende B.V. and Contributors, and Benoit Thieurmél. 2022. *visNetwork: Network Visualization Using Vis.js Library*. <https://datastorm-open.github.io/visNetwork/>.
- 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>.
- . 2022a. *Future: Unified Parallel and Distributed Processing in r for Everyone*. <https://CRAN.R-project.org/package=future>.
- . 2022b. *Future.apply: Apply Function to Elements in Parallel Using Futures*. <https://CRAN.R-project.org/package=future.apply>.
- 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>.
- Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2022. *Shiny: Web Application Framework for r*. <https://shiny.rstudio.com/>.
- Ching, Travers. 2022. *Qs: Quick Serialization of r Objects*. <https://github.com/traversc/qs>.
- Csardi, Gabor, and Tamas Nepusz. 2006. “The Igraph Software Package for Complex Network Research.” *InterJournal Complex Systems*: 1695. <https://igraph.org>.
- Csárdi, Gábor. 2022. *Pingr: Check If a Remote Computer Is up*. <https://github.com/r-lib/pingr#readme>.
- 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. 2022. *Data.table: Extension of ‘Data.frame’*. <https://CRAN.R-project.org/package=data.table>.
- Eddelbuettel, Dirk. 2020. *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*. <https://CRAN.R-project.org/package=mgsub>.
- file., See AUTHORS. 2022. *Igraph: Network Analysis and Visualization*. <https://CRAN.R-project.org/package=igraph>.
- Granjon, David. 2022. *bs4Dash: A Bootstrap 4 Version of Shinydashboard*. <https://CRAN.R-project.org/package=bs4Dash>.
- Hester, Jim, Hadley Wickham, and Gábor Csárdi. 2021. *Fs: Cross-Platform File System Operations Based on Libuv*. <https://CRAN.R-project.org/package=fs>.
- Iannone, Richard, Joe Cheng, Barret Schloerke, Ellis Hughes, and JooYoung Seo. 2022. *Gt: Easily Create Presentation-Ready Display Tables*. <https://CRAN.R-project.org/package=gt>.
- 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>.
- . 2022a. *Tarchetypes: Archetypes for Targets*. <https://CRAN.R-project.org/package=tarchetypes>.
- . 2022b. *Targets: Dynamic Function-Oriented Make-Like Declarative Workflows*. <https://CRAN.R-project.org/package=targets>.
- Meyer, Fanny, and Victor Perrier. 2022. *Shinybusy: Busy Indicators and Notifications for Shiny Applications*. <https://github.com/dreamRs/shinybusy>.
- Ooms, Jeroen. 2021. *Magick: Advanced Graphics and Image-Processing in r*. <https://CRAN.R-project.org/package=magick>.
- . 2022a. *Openssl: Toolkit for Encryption, Signatures and Certificates Based on OpenSSL*. <https://github.com/jeroen/openssl>.
- . 2022b. *Pdftools: Text Extraction, Rendering and Converting of PDF Documents*. <https://CRAN.R-project.org/package=pdftools>.
- Pedersen, Thomas Lin. 2022. *Ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. <https://CRAN.R-project.org/package=ggraph>.
- Perrier, Victor, Fanny Meyer, and David Granjon. 2022. *shinyWidgets: Custom Inputs Widgets for Shiny*. <https://CRAN.R-project.org/package=shinyWidgets>.
- Schubert, Michael. 2022. *Clustermq: Evaluate Function Calls on HPC Schedulers (LSF, SGE, SLURM, PBS/Torque)*. <https://mschubert.github.io/clustermq/>.
- Ushey, Kevin. 2022. *Renv: Project Environments*. <https://rstudio.github.io/renv/>.
- Ushey, Kevin, JJ Allaire, Hadley Wickham, and Gary Ritchie. 2022. *Rstudioapi: Safely Access the RStudio API*. <https://CRAN.R-project.org/package=rstudioapi>.
- Wickham, Hadley. 2011. “Testthat: Get Started with Testing.” *The R Journal* 3: 5–10. https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.
- . 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.
- . 2022a. *Httr: Tools for Working with URLs and HTTP*. <https://CRAN.R-project.org/package=httr>.
- . 2022b. *Rvest: Easily Harvest (Scrape) Web Pages*. <https://CRAN.R-project.org/package=rvest>.
- . 2022c. *Testthat: Unit Testing for r*. <https://CRAN.R-project.org/package=testthat>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2022. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Wickham, Hadley, Jim Hester, and Jeroen Ooms. 2021. *Xml2: Parse XML*. <https://CRAN.R-project.org/package=xml2>.
- Wickham, Hadley, and Dana Seidel. 2022. *Scales: Scale Functions for Visualization*. <https://CRAN.R-project.org/package=scales>.
- 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. <http://www.crcpress.com/product/ISBN/9781466561595>.
- . 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <https://yihui.org/knitr/>.
- . 2022. *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*. <https://CRAN.R-project.org/package=kableExtra>.