Corpus des Deutschen Bundesrechts

(C-DBR)

Compilation Report

Version 2024-04-03

DOI: 10.5281/zenodo.10908140

Titel	Source Code des »Corpus des Deutschen Bundesrechts«
Abkürzung	C-DBR-Source
Autor	Seán Fobbe
Version	2024-04-03
Download	https://doi.org/10.5281/zenodo.10908140
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2024). Source Code des »Corpus des Deutschen Bundesrechts« (C-DBR-Source). Version 2024-04-03. Zenodo. DOI: 10.5281/zenodo.10908140.

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

Soweit nicht anders angegeben ist die DOI immer eine »Version DOI« und bezieht sich nur auf eine bestimmte Version der Software. Sie verlinkt daher nur Version 2024-04-03. 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)

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)

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, Edgelists, 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

- Docker

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

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

1.4 Anleitung

1.4.1 Schritt 1: Ordner vorbereiten

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

```
$ git clone https://github.com/seanfobbe/c-dbr.git
```

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

1.4.2 Schritt 2: Docker Image erstellen

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

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

1.4.3 Schritt 3: Datensatz kompilieren

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

```
$ Rscript delete_all_data.R
```

Den vollständigen Datensatz kompilieren Sie mit folgendem Skript:

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

1.4.4 Ergebnis

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

1.5 Pipeline visualisieren

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

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

1.6 Troubleshooting

Hilfreiche Befehle um Fehler zu lokalisieren und zu beheben.

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

1.7 Projektstruktur

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

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

1.8 Weitere Open Access Veröffentlichungen (Fobbe)

Website — https://www.seanfobbe.de

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

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

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

1.9 Kontakt

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

2 Packages laden

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

tar_unscript()
```

3 Vorbereitung

3.1 Definitionen

```
## Datum
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2024-04-03"

## 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.10908139"
#>   ..$ software:List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.4072934"
#>   .. ..$ version: chr "10.5281/zenodo.10908140"
#>  $ 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(system2("git", "ls-files", stdout = TRUE),
                       ".git")

#> 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

```
tarchetypes::tar_age(url.xml,
                     f.links_xml(url = "https://www.gesetze-im-internet.de/gii-
    toc.xml"),
                     age = as.difftime(1, units = "days"))

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

5.2.2 Tabelle der Dateinamen erstellen

```
tarchetypes::tar_age(dt.filenames,
                     f.html_landing_pages(url.xml,
                                          multicore = config$parallel$
    htmlLandingPages,
                                          cores = fullCores),
                     age = as.difftime(1, units = "days"))

#> 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-Type-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.xmlzip,
                       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_txt,
             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 Pipeline archivieren

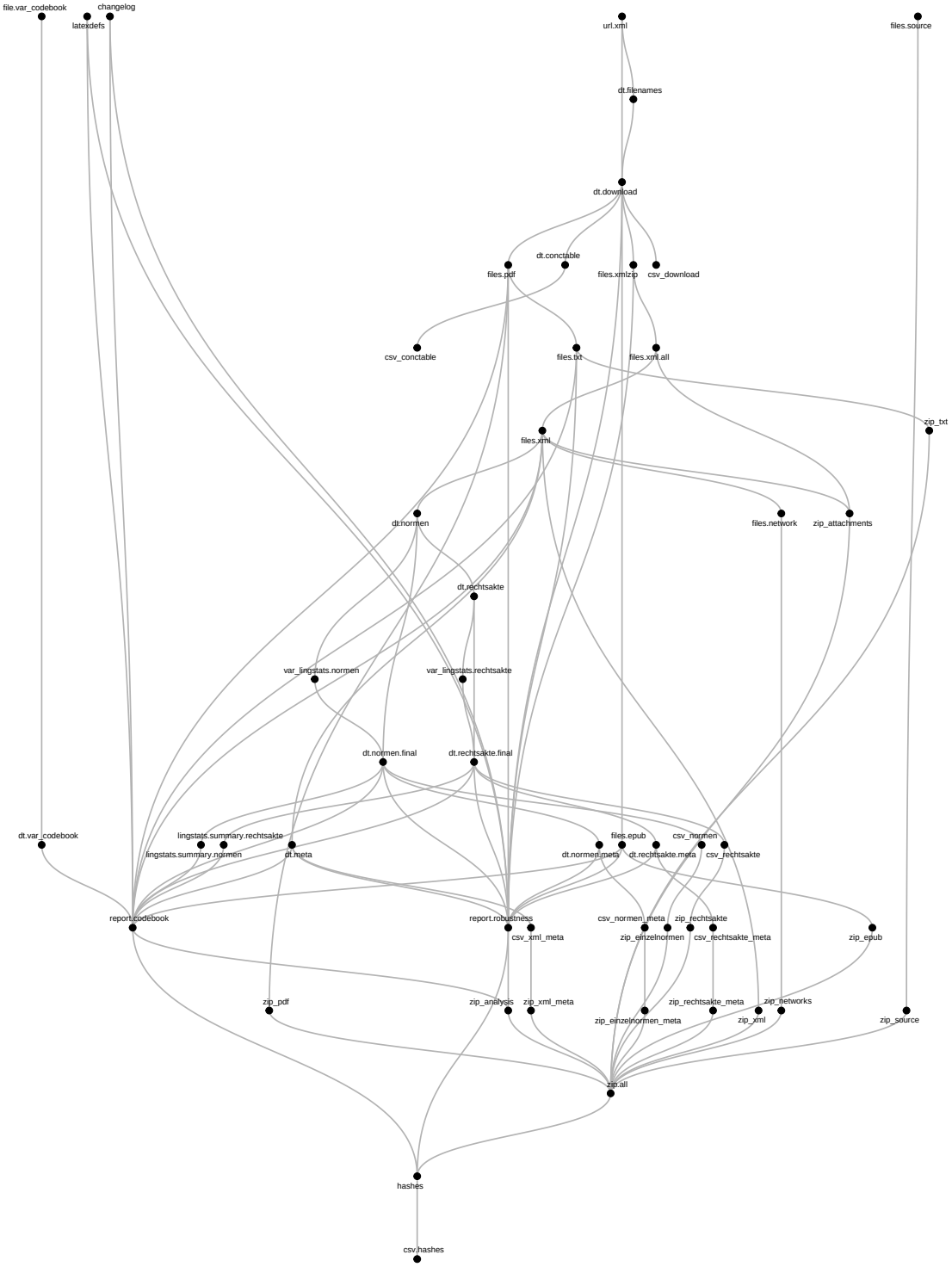
```
zip(paste0("output/",
           paste0(config$project$shortname,
                  "_",
                  datestamp),
           "_Targets_Storage.zip"),
    "_targets/")
```

6.3 Visualisierung

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

g  <- igraph::graph.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.
```


file.var_codebook
changelog
latexdefs
url.xml
files.source
dt.filenames
dt.download
dt.conctable
files.pdf
files.xmlzip
csv_download
csv_conctable
files.txt
files.xml.all
zip_txt
files.xml
dt.normen
files.network
zip_attachments
dt.rechtsakte
var_lingstats.normen
var_lingstats.rechtsakte
dt.normen.final
dt.rechtsakte.final
dt.var_codebook
lingstats.summary.rechtsakte
lingstats.summary.normen
dt.meta
files.epub
csv_normen
dt.normen.meta
dt.rechtsakte.meta
csv_rechtsakte
report.codebook
report.robustness
csv_normen_meta
zip_rechtsakte
csv_xml_meta
zip_einzelnormen
csv_rechtsakte_meta
zip_epub
zip_pdf
zip_analysis
zip_xml_meta
zip_rechtsakte_meta
zip_networks
zip_einzelnormen_meta
zip_xml
zip_source
zip.all
hashes
csv.hashes

7 Pipeline: Analyse

7.1 Gesamte Liste

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

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

# Gesamter Speicherplatzverbrauch
sum(meta$MB, na.rm = TRUE)
#> [1] 4969.2

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	5603	file	0.01
csv.hashes	stem	90	qs	0.00
csv_conctable	stem	1008977	file	1.01
csv_download	stem	6402171	file	6.40
csv_normen	stem	222234688	file	222.23
csv_normen_meta	stem	66983347	file	66.98
csv_rechtsakte	stem	157563796	file	157.56
csv_rechtsakte_meta	stem	2509449	file	2.51
csv_xml_meta	stem	3049892	file	3.05
dt.conctable	stem	236136	qs	0.24
dt.download	stem	1339616	qs	1.34
dt.filenames	stem	74213	qs	0.07
dt.meta	stem	474702	qs	0.47
dt.normen	stem	37918317	qs	37.92
dt.normen.final	stem	38336367	qs	38.34
dt.normen.meta	stem	2080434	qs	2.08

(continued)

name	type	bytes	format	MB
dt.rechtsakte	stem	36935174	qs	36.94
dt.rechtsakte.final	stem	36973661	qs	36.97
dt.rechtsakte.meta	stem	417560	qs	0.42
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	473410256	file	473.41
files.network	stem	78902450	file	78.90
files.pdf	stem	703748752	file	703.75
files.source	stem	382404	file	0.38
files.txt	stem	214737900	file	214.74
files.xml	stem	283756458	file	283.76
files.xml.all	stem	598508599	file	598.51
files.xmlzip	stem	375747924	file	375.75
hashes	stem	2124	qs	0.00
latexdefs	stem	1199	file	0.00
lingstats.summary.normen	stem	385	qs	0.00
lingstats.summary.rechtsakte	stem	390	qs	0.00
report.codebook	stem	662824	file	0.66
report.robustness	stem	524932	file	0.52
url.xml	stem	47778	qs	0.05
var_lingstats.normen	stem	453033	qs	0.45
var_lingstats.rechtsakte	stem	38406	qs	0.04
zip.all	stem	234	qs	0.00
zip_analysis	stem	4031962	file	4.03
zip_attachments	stem	284503318	file	284.50
zip_einzelnormen	stem	40894097	file	40.89
zip_einzelnormen_meta	stem	3004852	file	3.00
zip_epub	stem	458945342	file	458.95

(continued)

name	type	bytes	format	MB
zip_networks	stem	73189682	file	73.19
zip_pdf	stem	617835232	file	617.84
zip_rechtsakte	stem	37136561	file	37.14
zip_rechtsakte_meta	stem	507405	file	0.51
zip_source	stem	721385	file	0.72
zip_txt	stem	50260831	file	50.26
zip_xml	stem	52126557	file	52.13
zip_xml_meta	stem	555576	file	0.56

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] 2375.52

## Minuten
runtime.sum / 60
#> [1] 39.592

## Stunden
runtime.sum / 3600
#> [1] 0.6598667
```

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	2024-04-03 12:56:20	698.719	11.65
var_lingstats.normen	2024-04-03 13:01:53	327.514	5.46
dt.normen	2024-04-03 12:26:58	250.188	4.17
files.pdf	2024-04-03 12:17:42	183.687	3.06
files.xmlzip	2024-04-03 12:20:27	165.385	2.76
files.epub	2024-04-03 12:14:37	162.820	2.71
var_lingstats.rechtsakte	2024-04-03 13:04:01	122.075	2.03
lingstats.summary.rechtsakte	2024-04-03 13:06:32	115.257	1.92
dt.filenames	2024-04-03 12:11:55	91.052	1.52
dt.meta	2024-04-03 12:22:46	60.704	1.01

(continued)

name	time	seconds	mins
lingstats.summary.normen	2024-04-03 13:04:37	33.848	0.56
zip_pdf	2024-04-03 12:21:27	26.491	0.44
zip_epub	2024-04-03 12:20:47	17.543	0.29
report.codebook	2024-04-03 13:06:58	16.065	0.27
report.robustness	2024-04-03 13:07:14	15.700	0.26
files.txt	2024-04-03 12:21:00	13.440	0.22
zip_attachments	2024-04-03 12:27:10	12.063	0.20
zip_txt	2024-04-03 12:21:45	11.510	0.19
zip_xml	2024-04-03 12:27:20	10.371	0.17
zip_rechtsakte	2024-04-03 13:07:23	9.463	0.16
zip_einzelnormen	2024-04-03 13:06:42	9.390	0.16
files.xml.all	2024-04-02 21:40:19	5.555	0.09
zip_networks	2024-04-03 13:01:58	5.212	0.09
hashes	2024-04-03 13:07:28	3.082	0.05
dt.rechtsakte	2024-04-03 12:56:25	2.659	0.04
changelog	2024-04-03 11:43:27	1.366	0.02
zip_einzelnormen_meta	2024-04-03 13:07:24	0.988	0.02
dt.download	2024-04-03 12:11:56	0.829	0.01
url.xml	2024-04-03 12:10:23	0.479	0.01
dt.normen.final	2024-04-03 13:04:02	0.382	0.01
file.dtd	2024-04-03 12:10:24	0.352	0.01
zip_analysis	2024-04-03 13:07:25	0.331	0.01
dt.conctable	2024-04-03 12:20:29	0.238	0.00
csv_normen	2024-04-03 13:04:03	0.186	0.00
zip_xml_meta	2024-04-03 13:01:59	0.130	0.00
zip_rechtsakte_meta	2024-04-03 13:07:24	0.104	0.00
csv_rechtsakte	2024-04-03 13:06:33	0.083	0.00
zip_source	2024-04-03 12:11:55	0.057	0.00
dt.rechtsakte.final	2024-04-03 13:04:03	0.053	0.00

(continued)

name	time	seconds	mins
csv_xml_meta	2024-04-03 12:56:22	0.048	0.00
csv_normen_meta	2024-04-03 13:06:42	0.036	0.00
dt.normen.meta	2024-04-03 13:04:37	0.017	0.00
latexdefs	2024-04-03 12:10:23	0.017	0.00
csv_download	2024-04-03 12:20:29	0.009	0.00
csv_rechtsakte_meta	2024-04-03 13:06:58	0.006	0.00
files.xml	2024-04-02 21:40:19	0.005	0.00
csv_conctable	2024-04-03 12:21:34	0.004	0.00
csv.hashes	2024-04-03 13:07:28	0.002	0.00
dt.rechtsakte.meta	2024-04-03 13:06:33	0.002	0.00
dt.var_codebook	2024-04-03 12:11:55	0.002	0.00
files.source	2024-04-03 12:09:58	0.001	0.00
file.var_codebook	2023-05-09 12:40:36	0.000	0.00
zip.all	2024-04-03 13:07:25	0.000	0.00

8 Warnungen

```
meta <- tar_meta(fields = "warnings", complete_only = TRUE)
setDT(meta)
meta <- meta[name != "files.network"]
meta$warnings <- gsub("(\\.pdf|\\.html?|\\.txt)|\\.xml", "\\1 \n\n", meta$
    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.")

}
```

8.1 report.codebook

Package microtype Warning Unable to apply patch footnote on input line 210.

8.2 report.robustness

Package microtype Warning Unable to apply patch footnote on input line 220.

9 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.
```

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_2024-04-03_DE_Analyse.zip	4.03
C-DBR_2024-04-03_DE_CSV_Einzelnormen_Datensatz.zip	40.89
C-DBR_2024-04-03_DE_CSV_Einzelnormen_Metadaten.zip	3.00
C-DBR_2024-04-03_DE_CSV_Metadaten-XML.zip	0.56
C-DBR_2024-04-03_DE_CSV_Rechtsakte_Datensatz.zip	37.14
C-DBR_2024-04-03_DE_CSV_Rechtsakte_Metadaten.zip	0.51
C-DBR_2024-04-03_DE_EPUB_Datensatz.zip	458.95
C-DBR_2024-04-03_DE_Netzwerke.zip	73.19
C-DBR_2024-04-03_DE_PDF_Datensatz.zip	617.84
C-DBR_2024-04-03_DE_TXT_Datensatz.zip	50.26
C-DBR_2024-04-03_DE_XML_Anlagen.zip	284.50
C-DBR_2024-04-03_DE_XML_Datensatz.zip	52.13
C-DBR_2024-04-03_Source_Code.zip	0.72
C-DBR_2024-04-03_Targets_Storage.zip	156.25

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_2024-04-03_DE_Alle-Rechtsakte-Verzeichnis.csv	1.01
C-DBR_2024-04-03_DE_CSV_Einzelnormen_Datensatz.csv	222.23
C-DBR_2024-04-03_DE_CSV_Einzelnormen_Metadaten.csv	66.98
C-DBR_2024-04-03_DE_CSV_Metadaten-XML_Datensatz.csv	3.05
C-DBR_2024-04-03_DE_CSV_Rechtsakte_Datensatz.csv	157.56
C-DBR_2024-04-03_DE_CSV_Rechtsakte_Metadaten.csv	2.51
C-DBR_2024-04-03_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_2024-04-03_Source_Code.zip
2	output/C-DBR_2024-04-03_DE_Analyse.zip
3	output/C-DBR_2024-04-03_DE_Netzwerke.zip
4	output/C-DBR_2024-04-03_DE_PDF_Datensatz.zip
5	output/C-DBR_2024-04-03_DE_TXT_Datensatz.zip
6	output/C-DBR_2024-04-03_DE_EPUB_Datensatz.zip
7	output/C-DBR_2024-04-03_DE_XML_Datensatz.zip
8	output/C-DBR_2024-04-03_DE_XML_Anlagen.zip
9	output/C-DBR_2024-04-03_DE_CSV_Einzelnormen_Datensatz.zip
10	output/C-DBR_2024-04-03_DE_CSV_Einzelnormen_Metadaten.zip

11	output/C-DBR_2024-04-03_DE_CSV_Rechtsakte_Datensatz.zip
12	output/C-DBR_2024-04-03_DE_CSV_Rechtsakte_Metadaten.zip
13	output/C-DBR_2024-04-03_DE_CSV_Metadaten-XML.zip
14	output/C-DBR_2024-04-03_Codebook.pdf
15	output/C-DBR_2024-04-03_RobustnessChecks.pdf

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

index	sha2.256
1	9530684674eac9dcd7d158515ea2aad04a6de905d56e04ba5a9527b5f85f6447
2	dfc464b9c6eb208039b52e04918e73d6be6d02a6ad73cc04633685622a231dc9
3	d5f117f54a97449fa9d860e107a3901b0ba6fb2ea54405cfd46f6423e35e8769
4	6caf049ae9cea91304e83ba67fdd9868a6c44e340406569f7b52f59a61812e55
5	f09bc5573af2d99a26eb48c0ca34c584303d5c43e4beda07cd20582510fa0675
6	3ec5a4a9d74191c3171dc13c9e4af0c4feda9121c20a70c9e5464797fe60236f
7	90c8c0adefc048cf3393998f16265ffa9659847c9e7578c7a108c2fe452daf49
8	a98b4a23c15ec67818599b4d68bddf9ee875d99e5d5d252e7ee353653bba42db
9	d89a342be26374ed07abb828823a2efd3d334ffaf9fabe5b29485e57fc442491
10	e8c54f982a273a7eb3be09a61acab4193c1dec79f00da719fc3c6a436e5d2601
11	92d9c4d3fadf8b827d9e3affdaa74ccde29c8e41294af199f4e9fc13cf4b4bf6
12	26b5d17976d7fbb3d5ae6cab8b5932a81d0eb05dc6d0538176cf5753a2a14fad
13	99a22b87f6cc7646dd7889161bd455b18bc5fb2c31b468aee28eea8d67566832
14	4c928b3067fae09a1cbf6d648b4bd518524ae26a17606bd7ac4e516e491f42e2
15	9692789e84dec140066d44dfc15701d06a87337c3f981fd2d3954e8fa606413c

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

index	sha3.512
1	98325acd4a36d3adb866ad60526e2458fa99a01fa2cb20a647f8e492e753e983 31a01031d24904f021fe8c8bd1c20f86752d539fe8dade149262d1bf2c44918c
2	ebc5d5bbe59d94371fd7432a0ca0a9bfaded5f7a85067f6cc8693942ecb32fd8 f31abd1742c6b0ad8fde6bacc953fe34caf66ccd8140ab4f2eb0550fa9fc8669
3	ef08f2958d08c10a4522fcec89f1d8f945fc5b2d108f0480ca38b83cd13d4de5 292eba6a8158e080b60ddf12cf213ed6cdff2d7d971a5f051d5316961e0abb7e
4	c8275c76f31cebde34dd1a285048ea8a738668af69c4b43eb4d0e3f1f2aa5148 4e1166286a8453d926132757d353e5df0c98a5ac9e1f2303643409b50a5b2344
5	661b288f54cfa92667b1f5bf75d29d3cf58fb2c7b8f5ff01023a9732a715c846 845dcfbba7430853499401510b6d8566e41c67226b2c3cd219547d63c4eb84a6
6	c957ebaea39b46316dbaec076e3b985fb014b3df7d13bc0f6061d1983e1c754d fd84ddffdc750287452343e52f0e5fad0860394d00b35c257e31f17177a5103d
7	be630e4db2ddfbfdfb73d117732ed018919ea6eba0475a470a2d230888e1b814 743177bfe8137d14ba18bdb6cb61764e67562f136e9825e29cba00d183d62318
8	9bd06147be2ed6939cc2fd3e2d8f02d36cd03f90605e2e2c1b095965b02239fb a10cdc6183d91687ba59201f6748f002afab1c12519ba6f252ee4a9a4f990731
9	ee78abc5c4e7fe50b11a7f7d2a2d0e63f8c94cf1912d08722408e53d4baadcb2 2f89af20652bc8dbe35d64ac0be28a6abaa5771200d85af5f957383c1d490140
10	264992304b757b38cf40b98a0a540acba757c0ba77b09f7e74b842a55fec28e1 ddf107ba091478cb7b9bbe5b8d090a16c2ec2d1eee8c0c933b845344a799d371
11	17c72a75c6819a337782fd8f0e8d549f280872b810ae33d9bbbf7f3663643ab4 704369e8b947358ea151734ad05229b26dab8677691c60bc20a4485024ee5c18
12	8f8b3e17e2cdce281f588cb5defcb976357769c59d4261bd3f803a0b2e591d08 d1fdb5839b0bc13bfa932260f3fb4383d2df0e0adc69d2aa3b63e802eef90e20
13	3641e309331e21677520007be309bb1d9907e3ae805aea01a789bb4eed0cb539 0151a7311d14837f789db37516768d4c0d1dd74b5d452bb79c4285425648f668
14	050f654386389195b60abc042fa7d8b3ee17df7cdb45e079f189d370fc9937c3 9acd663de92c8c692eb416bccd2573360c39124ad6f2334adf7ea2b57237256c
15	fb50ec823efe31c50a1fc7a5167f90777135984b4dc8191b5ed57141066bb8d4 11a7db538439b2fda3ccddfeed548406fd55cc2439b2c229b9bf3a58c268712e

12 Changelog

12.1 Version 2024-04-03

- Vollständige Aktualisierung der Daten
- Aktualisierung der Python Toolchain
- Aktualisierung des Public GPG Keys im Repository

12.2 Version 2024-01-07

- Vollständige Aktualisierung der Daten

12.3 Version 2023-10-03

- Vollständige Aktualisierung der Daten

12.4 Version 2023-07-09

- Vollständige Aktualisierung der Daten
- Konfigurations-Dateien in etc/ verschoben

12.5 Version 2023-04-07

- Vollständige Aktualisierung der Daten
- Gesamte Laufzeitumgebung nun mit Docker versionskontrolliert
- Download-Manifest wird nun spätestens nach 24h Stunden invalidiert, damit keine alten Daten aus früheren Kompilierungen den Prozess zum Absturz bringen
- ZIP-Archiv der TXT-Dateien wird nun auch gehasht
- Verbesserte Formatierung von Warnungen und Fehlermeldungen im Compilation Report
- Veränderung der Download-Reihenfolge
- Falls im ersten Download-Durchlauf Dateien fehlen werden die Folgeversuche nun korrekt durchgeführt
- Die Pipeline mit allen Zwischenergebnissen wird nun automatisch in “output/” archiviert
- Source Code ZIP-Archiv wird nun anhand des git-Manifestes generiert
- README im Hinblick auf Docker aktualisiert
- Struktur des Compilation Reports angepasst, um Warnungen und Fehler prominenter anzuzeigen
- Zusätzliche Unit Tests

12.6 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

12.7 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

12.8 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

12.9 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 deaktivert, 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

12.10 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

12.11 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

12.12 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

12.13 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.

12.14 Version 2020-07-08

- Vollständige Aktualisierung der Daten

12.15 Version 2020-05-18

- Erstveröffentlichung

13 Abschluss

```
## Datumsstempel
print(datestamp)
#> [1] "2024-04-03"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2024-04-03 13:12:40 CEST"


## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2024-04-03 13:12:51 CEST"


## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 10.69919 secs
```

14 Parameter für strenge Replikationen

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

sessionInfo()
#> R version 4.2.2 (2022-10-31)
#> Platform: x86_64-pc-linux-gnu (64-bit)
#> Running under: Ubuntu 22.04.2 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so
#>
#> locale:
#>  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
#>  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
#>  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
#>  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
#>  [9] LC_ADDRESS=C               LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] ggraph_2.1.0      ggplot2_3.4.1     igraph_1.4.1      kableExtra_1.3.4
#>  [5] knitr_1.42        quanteda_3.2.4    data.table_1.14.8 future_1.32.0
#>  [9] RcppTOML_0.2.2    tarchetypes_0.7.5 targets_0.14.3
#>
#> loaded via a namespace (and not attached):
#>  [1] viridis_0.6.2       httr_1.4.5          tidyr_1.3.0
#>  [4] tidygraph_1.2.3     viridisLite_0.4.1   RcppParallel_5.1.7
#>  [7] highr_0.10          future.callr_0.8.1  base64url_1.4
#> [10] renv_0.17.0         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_3.2.0        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_0.25.5           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_1.1.1
#> [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 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.

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. 2023a. *Magick: Advanced Graphics and Image-Processing in R.*

———. 2023b. *Openssl: Toolkit for Encryption, Signatures and Certificates Based on Openssl.* https://github.com/jeroen/openssl.

———. 2023c. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents.*

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

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

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

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

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, Jim Hester, and Jeroen Ooms. 2021. *Xml2: Parse Xml*.

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

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

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

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

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

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

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