Corpus der Entscheidungen des Bundesfinanzhofs

(CE-BFH-Source)

Compilation Report

Version 2023-10-15

DOI: 10.5281/zenodo.7691843

Titel	Source Code des »Corpus der Entscheidungen des Bundesfinanzhofs«
Abkürzung	CE-BFH-Source
Autor	Seán Fobbe
Version	2023-10-15
Download	https://doi.org/10.5281/zenodo.7691843
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2023). Source Code des »Corpus der Entscheidungen des Bundesfinanzhofs« (CE-BFH-Source). Version 2023-10-15. Zenodo. DOI: 10.5281/zenodo.7691843.

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 2023-10-15. Für das Gesamtkonzept der Software steht eine »Concept DOI« zur Verfügung, die auf der Zenodo-Seite jeder Version unter »Cite all versions?« zu finden ist. Die »Concept DOI« verlinkt immer die aktuellste Version.

Lizenz: MIT No Attribution (MIT-0)

Disclaimer

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

Inhaltsverzeichnis

1 Corpus der Entscheidungen des Bundesfinanzhofs (CE-BFH)

1.1 Überblick

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

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

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

1.2 Funktionsweise

Primäre Endprodukte des Skripts sind folgende ZIP-Archive:

- Der volle Datensatz im CSV-Format (mit zusätzlichen Metadaten)
- Die reinen Metadaten im CSV-Format (wie unter 1, nur ohne Entscheidungsinhalte)
- Alle Entscheidungen im HTML-Format
- Alle Entscheidungen im TXT-Format
- Alle Entscheidungen im PDF-Format
- Nur V-Entscheidungen (BFHE, amtliche Sammlung) im PDF-Format
- Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)

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

1.3 Systemanforderungen

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

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

1.4 Anleitung

1.4.1 Schritt 1: Ordner vorbereiten

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

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

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
├ config.toml                # Zentrale Konfigurations-Datei
├ data                       # Datensätze, auf denen die Pipeline aufbaut
├ delete_all_data.R          # Löscht den Datensatz und Zwischenergebnisse
├ docker-build-image.sh      # Docker Image erstellen
├ docker-compose.yaml        # Konfiguration für Docker
├ Dockerfile                 # Definition des Docker Images
├ docker-run-project.sh      # Docker Image und Datensatz kompilieren
├ etc                        # Konfigurations-Dateien
├ functions                  # Wichtige Schritte der Pipeline
├ gpg                        # Persönlicher Public GPG-Key für Seán Fobbe
├ pipeline.Rmd               # Zentrale Definition der Pipeline
├ README.md                  # Bedienungsanleitung
├ reports                    # Markdown-Dateien
├ run_project.R              # Kompiliert den gesamten Datensatz
└ tex                       # LaTeX-Templates
```

1.8 Weitere Open Access Veröffentlichungen (Fobbe)

Website — https://www.seanfobbe.de

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

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

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

1.9 Kontakt

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

2 Packages laden

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

tar_unscript()
```

3 Vorbereitung

3.1 Definitionen

```
## Datum
datestamp <- Sys.Date()
print(datestamp)
#> [1] "2023-10-15"

## 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 2
#>   ..$ pages : int 20
#>   ..$ toggle: logi FALSE
#>  $ doi     :List of 4
#>   ..$ aktenzeichen : chr "10.5281/zenodo.4569564"
#>   ..$ data         :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.7691840"
#>   .. ..$ version: chr "10.5281/zenodo.7691841"
#>   ..$ personendaten: chr "10.5281/zenodo.4568682"
#>   ..$ software     :List of 2
#>   .. ..$ concept: chr "10.5281/zenodo.7691842"
#>   .. ..$ version: chr "10.5281/zenodo.7691843"
#>  $ download:List of 1
#>   ..$ timeout: int 60
#>  $ fig     :List of 3
#>   ..$ align : chr "center"
#>   ..$ dpi   : int 300
#>   ..$ format: chr [1:2] "pdf" "png"
#>  $ license :List of 2
#>   ..$ code: chr "MIT-0"
#>   ..$ data: chr "Creative Commons Zero 1.0 Universal"
#>  $ parallel:List of 3
#>   ..$ extractPDF   : logi TRUE
#>   ..$ lingsummarize: logi TRUE
#>   ..$ multihashes  : logi TRUE
#>  $ project :List of 3
#>   ..$ author   : chr "Seán Fobbe"
#>   ..$ fullname : chr "Corpus der Entscheidungen des Bundesfinanzhofs"
#>   ..$ shortname: chr "CE-BFH"
#>  $ quanteda:List of 1
#>   ..$ tokens_locale: chr "de_DE"


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

```
                        "/analysis")
```

3.2 Aufräumen

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

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


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


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

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

## }
```

3.3 Ordner erstellen

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

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

3.4 Vollzitate statistischer Software schreiben

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

4 Globale Variablen

4.1 Packages definieren

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

tar_option_set(workspace_on_error = TRUE) # Save Workspace on Error
tar_option_set(format = "qs")

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

4.2 Konfiguration

```
datestamp <- Sys.Date()

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

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

## Caption for diagrams
caption <- paste("Fobbe | DOI:",
                 config$doi$data$version)


## Prefix for figure titles
prefix.figuretitle <- paste(config$project$shortname,
                            "| Version",
                            datestamp)

## File prefix
prefix.files <- paste0(config$project$shortname,
                       "_",
```

```
                        datestamp)


if (config$cores$max == TRUE){
    fullCores <- future::availableCores() - 1
}


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

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

4.3 Funktionen definieren

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

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

4.4 Metadaten für TXT-Dateien definieren

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

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

4.5 ZIP-Datei für Source definieren

```
files.source.raw <-  c(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 des Datensatzes umfassen.

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

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

5.1.2 Changelog

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

5.1.3 Variablen

Die Variablen des Datensatzes, inklusive ihrer Erläuterung.

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

5.1.4 Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD)

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

```
list(
    tar_target(file.az.brd,
```

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

5.2 Download Targets

5.2.1 Download-Manifest erstellen

Das Download-Manifest wird aus den Metadaten der Datenbank des BFH erstellt und enthält die Links zu den Unterseiten der einzelnen Entscheidungen, sowie die zugehörigen Metadaten aus der Datenbank in Tabellenform.

```
tar_target(dt.download.manifest.raw,
           f.download_manifest_make(sleep.min = 1,
                                    sleep.max = 2,
                                    verbose = FALSE,
                                    debug.toggle = config$debug$toggle,
                                    debug.pages = config$debug$pages))
#> Establish _targets.R and _targets_r/targets/tar.download.manifest.raw.R.
```

5.2.2 Download-Manifest finalisieren

Bei der Finalisierung werden einige Korrekturen und Tests vorgenommen.

```
tar_target(dt.download.manifest.final,
           f.download_manifest_finalize(dt.download.manifest.raw))
#> Establish _targets.R and _targets_r/targets/tar.download.manifest.final.R.
```

5.2.3 Entscheidungen als HTML abrufen

```
tar_target(files.html,
                f.download(url = dt.download.manifest.final$url_html,
                           filename = paste0(dt.download.manifest.final$bfh_id, "
    .html"),
                           dir = "files/html",
                           sleep.min = 0.2,
                           sleep.max = 0.5,
                           retries = 3,
                           retry.sleep.min = 2,
                           retry.sleep.max = 5,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE,
                           debug.files = 500),
                format = "file")

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

5.2.4 Entscheidungen als PDF abrufen

```
tar_target(files.pdf,
                f.download(url = dt.final$url_pdf,
                           filename = paste0(dt.final$doc_id, ".pdf"),
                           dir = "files/pdf",
                           sleep.min = 0.2,
                           sleep.max = 0.5,
                           retries = 3,
                           retry.sleep.min = 2,
                           retry.sleep.max = 5,
                           timeout = config$download$timeout,
                           debug.toggle = FALSE,
                           debug.files = 500),
                format = "file")

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

5.3 Convert Targets

```
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.R.
```

5.3.1 HTML-Seiten parsen

Die HTML-Seiten werden lokal geparsed, um eine höhere Geschwindigkeit und Robustheit zu erreichen.

```
tar_target(dt.decisionpage,
           f.extract_decisionpage(files.html))
#> Establish _targets.R and _targets_r/targets/tar.download.html.parse.R.
```

5.4 Enhance Targets

5.4.1 Download Manifest und Decision Data zusammenfügen

```
tar_target(dt.intermediate,
           f.merge(dt.download.manifest.final = dt.download.manifest.final,
                   dt.decisionpage = dt.decisionpage))
#> Establish _targets.R and _targets_r/targets/tar.enhance.merge.R.
```

```
tar_target(var_az_parts,
           f.var_az_parts(dt.intermediate))
#> Establish _targets.R and _targets_r/targets/tar.enhance.az.parts.R.
```

5.4.2 Variable erstellen: »verfahrensart«

Die Variable »verfahrensart« wird aus den Registerzeichen berechnet.

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

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

Berechnung klassischer linguistischer Kennzahlen.

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

5.4.4 Konstanten erstellen

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

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

5.4.5 Zusätzliche Variablen zusammenführen

```
tar_target(vars_additional,
           data.table(verfahrensart = var_verfahrensart,
                      var_az_parts,
                      var_lingstats,
                      var_constants))
```

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

5.4.6 Finalen Datensatz erstellen

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

```
tar_target(dt.final,
           f.finalize(dt.intermediate = dt.intermediate,
                      vars.additional = vars_additional,
                      varnames = variables.codebook$varname))

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

5.4.7 Variante erstellen: Nur Metadaten

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

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

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

5.5 Write Targets

Dieser Abschnitt der Pipeline schreibt den Datensatz auf die Festplatte.

5.5.1 CSV schreiben: Voller Datensatz

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

5.5.2 CSV schreiben: Metadaten

```
tar_target(csv.meta,
           f.tar_fwrite(x = dt.meta,
                        filename = file.path("output",
                                             paste0(prefix.files,
```

```
                                                    "_DE_CSV_Metadaten.csv"))
                        )
          )
#> Establish _targets.R and _targets_r/targets/tar.write.meta.R.
```

5.6 Report Targets

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

5.6.1 LaTeX-Definitionen schreiben

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

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

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

5.6.2 Zusammenfassungen linguistischer Kennwerte berechnen

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

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

5.6.3 Report erstellen: Robustness Checks

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

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

5.6.4 Report erstellen: Codebook

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

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

5.7 ZIP Targets

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

5.7.1 ZIP erstellen: Source Code

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

5.7.2 ZIP erstellen: Analyse-Dateien

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

5.7.3 ZIP erstellen: PDF-Dateien (alle Entscheidungen)

```
tar_target(zip.pdf.all,
           f.tar_zip(x = files.pdf,
                     filename = paste(prefix.files,
                                      "DE_PDF_Datensatz.zip",
                                      sep = "_"),
                     dir = "output",
                     mode = "cherry-pick"),
           format = "file")

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

5.7.4 ZIP erstellen: PDF-Dateien (nur V-Entscheidungen)

```
tar_target(zip.pdf.bfhe,
           f.tar_zip(x = grep("BFH_V", files.pdf, value = TRUE),
                     filename = paste(prefix.files,
                                      "DE_PDF_BFHE_V-Entscheidungen.zip",
                                      sep = "_"),
                     dir = "output",
                     mode = "cherry-pick"),
           format = "file")
#> Establish _targets.R and _targets_r/targets/tar.zip.pdf.leit.R.
```

5.7.5 ZIP erstellen: TXT-Dateien

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

5.7.6 ZIP erstellen: HTML-Dateien

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

5.7.7 ZIP erstellen: CSV-Datei (voller Datensatz)

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

5.7.8 ZIP erstellen: CSV-Datei (nur Metadaten)

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

5.8 Kryptographische Hashes

5.8.1 Zu hashende ZIP-Archive definieren

```
tar_target(zip.all,
           c(zip.pdf.all,
             zip.pdf.bfhe,
             zip.txt,
             zip.html,
             zip.csv.final,
             zip.csv.meta,
             zip.analysis,
             zip.source))
#> Establish _targets.R and _targets_r/targets/tar.hashes.all.R.
```

5.8.2 Kryptographische Hashes berechnen

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

5.8.3 CSV schreiben: Kryptographische Hashes

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

6 Pipeline: Kompilierung

6.1 Durchführen der Kompilierung

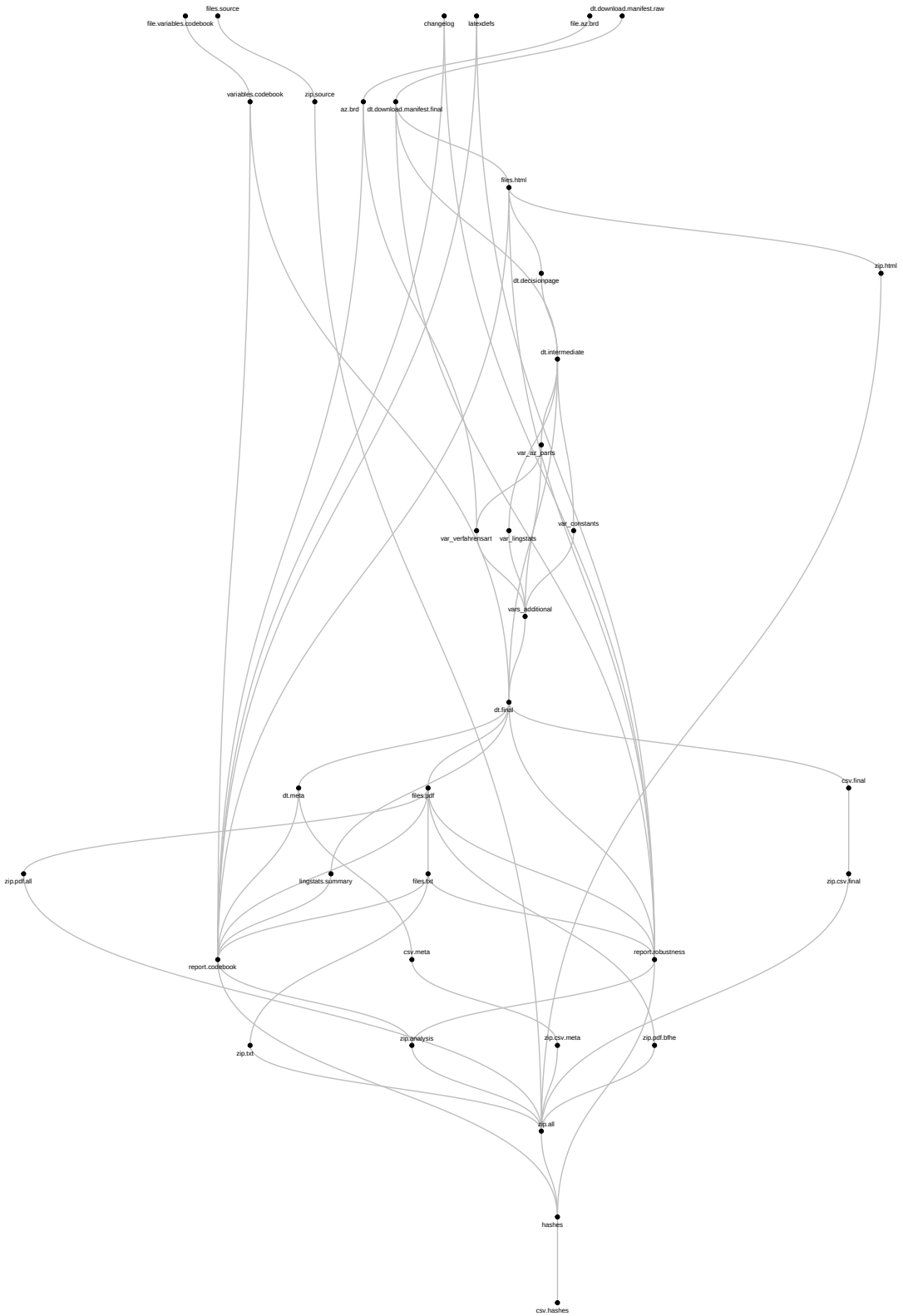
```
tar_make()
```

6.2 Visualisierung

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

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


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


files.source
file.variables.codebook
changelog
latexdefs
dt.download.manifest.raw
file.az.brd
variables.codebook
zip.source
az.brd
dt.download.manifest.final
files.html
zip.html
dt.decisionpage
dt.intermediate
var_az_parts
var_dotstants
var_verfahrensart
var_lingstats
vars_additional
dt.final
csv.final
dt.meta
files.pdf
zip.pdf.all
lingstats.summary
files.txt
zip.csv.final
csv.meta
report.robustness
report.codebook
zip.analysis
zip.csv.meta
zip.pdf.bfhe
zip.txt
zip.all
hashes
csv.hashes

7 Liste aller Targets

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] 2566.59

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

name	type	bytes	format	MB
az.brd	stem	5837	qs	0.01
changelog	stem	58	file	0.00
csv.final	stem	85	qs	0.00
csv.hashes	stem	91	qs	0.00
csv.meta	stem	85	qs	0.00
dt.decisionpage	stem	46284435	qs	46.28
dt.download.manifest.final	stem	439148	qs	0.44
dt.download.manifest.raw	stem	413511	qs	0.41
dt.final	stem	45950902	qs	45.95
dt.intermediate	stem	45828352	qs	45.83
dt.meta	stem	2919132	qs	2.92
file.az.brd	stem	33787	file	0.03
file.variables.codebook	stem	8397	file	0.01
files.html	stem	662413544	file	662.41
files.pdf	stem	586803938	file	586.80
files.source	stem	220188	file	0.22
files.txt	stem	174403475	file	174.40

(continued)

name	type	bytes	format	MB
hashes	stem	1504	qs	0.00
latexdefs	stem	1267	file	0.00
lingstats.summary	stem	374	qs	0.00
report.codebook	stem	585813	file	0.59
report.robustness	stem	351944	file	0.35
var_az_parts	stem	27685	qs	0.03
var_constants	stem	7050	qs	0.01
var_lingstats	stem	66073	qs	0.07
var_verfahrensart	stem	6705	qs	0.01
variables.codebook	stem	2921	qs	0.00
vars_additional	stem	104241	qs	0.10
zip.all	stem	181	qs	0.00
zip.analysis	stem	2936429	file	2.94
zip.csv.final	stem	46454537	file	46.45
zip.csv.meta	stem	3449909	file	3.45
zip.html	stem	122514563	file	122.51
zip.pdf.all	stem	529500815	file	529.50
zip.pdf.bfhe	stem	231479474	file	231.48
zip.source	stem	1241298	file	1.24
zip.txt	stem	62150185	file	62.15

8 Laufzeit aller Targets

```
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] 22716.66

## Minuten
runtime.sum / 60
#> [1] 378.611

## Stunden
runtime.sum / 3600
#> [1] 6.310183
```

9 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.html	2023-10-15 19:49:06	12255.275	204.25
files.pdf	2023-10-15 22:22:45	8105.830	135.10
dt.download.manifest.raw	2023-10-15 16:24:51	2009.158	33.49
dt.decisionpage	2023-10-15 19:51:42	154.487	2.57
var_lingstats	2023-10-15 20:07:39	43.040	0.72
lingstats.summary	2023-10-15 22:23:21	32.748	0.55
files.txt	2023-10-15 22:23:46	24.682	0.41
zip.html	2023-10-15 19:52:03	20.507	0.34
zip.pdf.all	2023-10-15 22:24:06	19.250	0.32
report.codebook	2023-10-15 22:24:41	10.619	0.18
zip.csv.final	2023-10-15 22:24:24	9.767	0.16
zip.txt	2023-10-15 22:24:50	9.151	0.15
zip.pdf.bfhe	2023-10-15 22:24:14	8.191	0.14
report.robustness	2023-10-15 22:24:31	6.754	0.11
hashes	2023-10-15 22:24:54	2.440	0.04
dt.final	2023-10-15 20:07:42	1.767	0.03
file.az.brd	2023-10-08 20:36:26	1.283	0.02
zip.csv.meta	2023-10-15 22:24:51	0.571	0.01
dt.download.manifest.final	2023-10-15 16:24:51	0.381	0.01
zip.analysis	2023-10-15 22:24:51	0.196	0.00
var_az_parts	2023-10-15 19:52:04	0.183	0.00
dt.intermediate	2023-10-15 19:52:04	0.132	0.00
csv.final	2023-10-15 22:22:48	0.113	0.00

(continued)

name	time	seconds	mins
zip.source	2023-10-15 16:24:51	0.072	0.00
latexdefs	2023-10-15 15:51:21	0.016	0.00
csv.meta	2023-10-15 22:24:24	0.014	0.00
vars_additional	2023-10-15 20:07:39	0.010	0.00
var_verfahrensart	2023-10-15 20:07:39	0.007	0.00
var_constants	2023-10-15 19:52:04	0.006	0.00
dt.meta	2023-10-15 22:22:48	0.003	0.00
az.brd	2023-10-15 16:24:51	0.002	0.00
csv.hashes	2023-10-15 22:24:54	0.001	0.00
variables.codebook	2023-10-15 16:24:51	0.001	0.00
zip.all	2023-10-15 22:24:51	0.001	0.00
changelog	2023-10-04 12:40:58	0.000	0.00
file.variables.codebook	2023-10-15 15:22:24	0.000	0.00
files.source	2023-10-15 15:49:26	0.000	0.00

10 Warnungen

```
meta <- tar_meta(fields = "warnings", complete_only = TRUE)
setDT(meta)
meta$warnings <- gsub("(\\.pdf|\\.html?|\\.txt)", "\\1 \n\n", meta$warnings)

if (meta[,.N > 0]){

    for(i in 1:meta[,.N]){

        cat(paste("###", meta[i]$name), "\n\n")
        cat(paste(meta[i]$warnings, "\n\n"))

    }

}else{

    cat("No warnings to report.")

}
```

10.0.1 files.pdf

URL httpswww.bundesfinanzhof.dedeentscheidungentscheidungenonlinedetailpdfSTRE201750242type1646225 status was Failure when receiving data from the peer

10.0.2 lingstats.summary

NA is replaced by empty string

10.0.3 report.codebook

LaTeX Warning Label variablen multiply defined.. Package microtype Warning Unable to apply patch footnote on input line 193.. LaTeX Warning There were multiplydefined labels.

10.0.4 report.robustness

Package microtype Warning Unable to apply patch footnote on input line 202.

10.0.5 var_lingstats

NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string. NA is replaced by empty string

11 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.
```

12 Dateigrößen

12.1 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
CE-BFH_2023-10-15_DE_Analyse.zip	2.94
CE-BFH_2023-10-15_DE_CSV_Datensatz.zip	46.45
CE-BFH_2023-10-15_DE_CSV_Metadaten.zip	3.45
CE-BFH_2023-10-15_DE_HTML_Datensatz.zip	122.51
CE-BFH_2023-10-15_DE_PDF_BFHE_V-Entscheidungen.zip	231.48
CE-BFH_2023-10-15_DE_PDF_Datensatz.zip	529.50
CE-BFH_2023-10-15_DE_TXT_Datensatz.zip	62.15
CE-BFH_2023-10-15_Source_Code.zip	1.24

12.2 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
CE-BFH_2023-10-15_DE_CSV_Datensatz.csv	173.96
CE-BFH_2023-10-15_DE_CSV_Metadaten.csv	14.56
CE-BFH_2023-10-15_KryptographischeHashes.csv	0.00

12.3 PDF-Dateien (MB)

```
tar_load(files.pdf)
pdf.MB <- file.size(files.pdf) / 10^6
sum(pdf.MB)
#> [1] 586.8039
```

12.4 TXT-Dateien (MB)

```
tar_load(files.txt)
txt.MB <- file.size(files.txt) / 10^6
sum(txt.MB)
#> [1] 174.4035
```

12.5 HTML-Dateien (MB)

```
tar_load(files.html)
html.MB <- file.size(files.html) / 10^6
sum(html.MB)
#> [1] 662.4135
```

13 Kryptographische Signaturen

13.1 Signaturen laden

```
tar_load(hashes)
```

13.2 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))
```

13.3 In Bericht anzeigen

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

index	filename
1	output/CE-BFH_2023-10-15_DE_PDF_Datensatz.zip
2	output/CE-BFH_2023-10-15_DE_PDF_BFHE_V-Entscheidungen.zip
3	output/CE-BFH_2023-10-15_DE_TXT_Datensatz.zip
4	output/CE-BFH_2023-10-15_DE_HTML_Datensatz.zip
5	output/CE-BFH_2023-10-15_DE_CSV_Datensatz.zip
6	output/CE-BFH_2023-10-15_DE_CSV_Metadaten.zip
7	output/CE-BFH_2023-10-15_DE_Analyse.zip
8	output/CE-BFH_2023-10-15_Source_Code.zip
9	output/CE-BFH_2023-10-15_Codebook.pdf

10 output/CE-BFH_2023-10-15_RobustnessChecks.pdf

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

index	sha2.256
1	d3b30dab67a86936733c099d9a803adc058b81069427b7184cc8628bafac3116
2	eadf06fc4f309e886cf534a440cf2e55aecb53a34543783f705a5704541920f6
3	4e28a713180cb29b72871750ae2768fa005b4540cae62856fc343984220e986f
4	9ab6fde3c75aab6080be8c1b57f4a903712410408d150740bf2da7149766764d
5	366bea150b0c941b7fbabec8be5e14063686d60e919264e8144642e0e9d69c1d
6	b85223baa2d0631de1774372662f2625330a4ba443fdb925f974eda3d76839fa
7	0d5a967a8458241cf92f7a3e0448dbbb783ebeaf2249b934acb8654a9b5b932a
8	a61ce4029addf314f76e3e3bb89613b8bb628e5d97a301bcab94a73223743bb2
9	634f5ed4f747d7a6ea4ea7455bcf96b77a303c03c8d591e28daf8be6559f680a
10	d41e6ea7d9feb147bca8fa15cde44aef9420094a4e1f903ce9a0446c746fa42b

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

index	sha3.512
1	c2328d9aa12d19d4fd16602d831d429dd2067ecf2442154eae13f4c5896668dc369cbe55518bb2199e4a1901dead33155a4ad24e38f67c607193467302ccfe59
2	a62c10cfe11bd9a9d96da4771fa16f9301f2bfab8b5921dc34ba61e6b8803939525ab96c496d9e1efaa8bb61b5118b51e565aa5fac8f8c5822c8e0cd9fa1daad
3	4670164ef2c45bf8b3eb1eff19b32227c977fe510b49fe4edeb4144ed4ebd6097bda9f5184a1984d6cf4b5d70a20b95a7fe908146c63909aa6de029a876c91c2
4	c3eb40b48a4d35ac4ef8438c0414860a36bd13280ee6358f7881cd6839b033f98c92da024977a84009a683df823bdc1dc2e70e80f9e5782756165284638d87ec
5	1bbd3217609989c497fa39cc834cbe09c6188b85561b70d32a5c98da82b044ca11dd3884d477e5201a362ac41fcdedc86d966fdb8e0a77ae71d69200a18b3c9a
6	90c2d7aa5732fee4e0c7d5ce0e2edf49beb797c9aed26970889bc1332430367cb124f27c069ed49cd27d71e481014db5c8b979d401db82d0b1114c09e7a31549
7	639386886352eb60fd595ccc52772f1b5a1d7cb5745882f50a688f7feb036e0a652367eaf9bd1d7f66141783881d7ac31fe7ca4a05f17623aafb813d1f0baf80
8	cd74c5cec2cda9b4b6325ef8b658f2b80d8ee23b2161adc8da4cddbc568700b2e25622528be920024b23a7a239766117ef693b7210f333dff115aee268284162
9	5fea83ac07fd6b81ae4fc60a67e0b4d9b8e9e3f724904f6d31e52c371debffa59bc7e62cf046b2547ea4d73d294fe6c129074ad1b93ba942ff66a7f2e754a3ca
10	deb1733950f544a5936f342bc0ac748508ca0db705cfbfb575ef84b185ccc4aa3b93df53df4cc6215638a897d1d69113a5c05e448e613058054cf1ce4eaa8a55

14 Changelog

14.1 Version 2023-10-15

- Erstveröffentlichung

15 Abschluss

```
## Datumsstempel
print(datestamp)
#> [1] "2023-10-15"

## Datum und Uhrzeit (Anfang)
print(begin.script)
#> [1] "2023-10-15 20:06:52 UTC"


## Datum und Uhrzeit (Ende)
end.script <- Sys.time()
print(end.script)
#> [1] "2023-10-15 22:24:57 UTC"


## Laufzeit des gesamten Skriptes
print(end.script - begin.script)
#> Time difference of 2.301142 hours
```

16 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, 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.*

Gagolewski, Marek. 2022. "stringi: Fast and Portable Character String Processing in R." *Journal of Statistical Software* 103 (2): 1–59. https://doi.org/10.18637/jss.v103.i02.

Gagolewski, Marek, Bartek Tartanus, others; Unicode, Inc., and others. 2023. *Stringi: Fast and Portable Character String Processing Facilities.*

Landau, William Michael. 2021a. *Tarchetypes: Archetypes for Targets.*

———. 2021b. "The Targets R Package: A Dynamic Make-Like Function-Oriented Pipeline Toolkit for Reproducibility and High-Performance Computing." *Journal of Open Source Software* 6 (57): 2959. https://doi.org/10.21105/joss.02959.

———. 2023a. *Tarchetypes: Archetypes for Targets.*

———. 2023b. *Targets: Dynamic Function-Oriented Make-Like Declarative Pipelines.*

Ooms, Jeroen. 2023. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents.*

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

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

Wickham, Hadley. 2022. *Rvest: Easily Harvest (Scrape) Web Pages.*

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*.