Corona-Rechtsprechung des Bundesverfassungsgerichts

(BVerfG-Corona-Source)

Compilation Report

Version 2024-07-24

DOI: 10.5281/zenodo.13765530

Titel	Source Code der »Corona-Rechtsprechung des Bundesverfassungsgerichts«
Abkürzung	BVerfG-Corona-Source
Autor	Seán Fobbe
Version	2024-07-24
Download	https://doi.org/10.5281/zenodo.13765530
Lizenz	GNU General Public License Version 3 (GPLv3)

Zitiervorschlag

Seán Fobbe (2024). Source Code der »Corona-Rechtsprechung des Bundesverfassungsgerichts« (BVerfG-Corona-Source). Version 2024-07-24. Zenodo. DOI: 10.5281/zenodo.13765530.

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

Soweit nicht anders angegeben ist die DOI immer eine »Version DOI« und bezieht sich nur auf eine bestimmte Version der Software. Sie verlinkt daher nur Version 2024-07-24. Für das Gesamtkonzept der Software steht eine »Concept DOI« zur Verfügung, die auf der Zenodo-Seite jeder Version unter »Cite all versions?« zu finden ist. Sie lautet 10.5281/zenodo.4459415. Die »Concept DOI« verlinkt immer die aktuellste Version.

GNU General Public License Version 3 (GPLv3)

Disclaimer

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

Inhaltsverzeichnis

1 README

1.1 Überblick

Dieser R Code lädt den Corpus der Entscheidungen des Bundesverfassungsgerichts (CE-BVerfG) herunter, untersucht ihn auf mit SARS-CoV-2 assoziiertem Vokabular und speichert relevante Entscheidungen. Es ist die Grundlage für den Datensatz **Corona-Rechtsprechung des Bundesverfassungsgerichts (BVerfG-Corona)**.

Alle mit diesem Skript erstellten Datensätze werden dauerhaft kostenlos und urheberrechtsfrei auf Zenodo, dem wissenschaftlichen Archiv des CERN, veröffentlicht. Alle Versionen sind mit einem persistenten Digital Object Identifier (DOI) versehen. Die neueste Version des Datensatzes ist immer über den Link der Concept DOI erreichbar: https://doi.org/10.5281/zenodo.4459405

1.2 Funktionsweise

Primäre Endprodukte des Skripts (im Ordner 'output') sind folgende ZIP-Archive:

- Alle Corona-relevanten Entscheidungen im PDF-Format
- Alle Corona-relevanten Entscheidungen im TXT-Format
- Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)
- Der Source Code und alle weiteren Quelldaten

Zusätzlich werden für alle ZIP-Archive kryptographische Signaturen (SHA2-256 und SHA3-512) berechnet und in einer CSV-Datei hinterlegt. Es kann optional ein PDF-Bericht erstellt werden (siehe unter "Kompilierung").

1.3 Systemanforderungen

- Docker
- Docker Compose
- 1 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/bverfg-corona.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)
├ BVerfG-Corona_CorpusCreation.R # Zentrale Definition der Pipeline
├ CHANGELOG.md                 # Alle Änderungen
├ config.toml                  # Zentrale Konfigurations-Datei
├ delete_all_data.R            # Löscht den Datensatz und Zwischenschritte
├ 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                          # Weitere Konfigurationsdateien
├ functions                    # Wichtige Schritte der Pipeline
├ gpg                          # Persönlicher Public GPG-Key für Seán Fobbe
├ LICENSE                      # Die Lizenz für den Source Code
├ README.md                    # Bedienungsanleitung
├ run_project.R                # Kompiliert den gesamten Datensatz
└ tex                          # LaTeX-Templates
```

1.8 Weitere Open Access Veröffentlichungen (Fobbe)

Website — https://www.seanfobbe.de

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

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

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

1.9 Kontakt

Fehler gefunden? Anregungen? Kommentieren Sie gerne im Issue Tracker auf GitHub oder kontaktieren Sie mich via https://www.seanfobbe.de

2 Vorbereitung

2.1 Datumsstempel

Dieser Datumsstempel wird in alle Dateinamen eingefügt. Er wird am Anfang des Skripts gesetzt, für den den Fall, dass die Laufzeit die Datumsbarriere durchbricht.

2.2 Datum und Uhrzeit (Beginn)

```
begin.script <- Sys.time()
print(begin.script)
```

```
## [1] "2024-09-15 19:32:08 CEST"
```

2.3 Packages Laden

```
library(stringr)      # String-Manipulation
library(magick)       # Cropping von PNG-Dateien
```

```
## Linking to ImageMagick 6.9.11.60
## Enabled features: fontconfig, freetype, fftw, heic, lcms, pango, webp, x11
## Disabled features: cairo, ghostscript, raw, rsvg
```

```
## Using 16 threads
```

```
library(RcppTOML)     # Verarbeitung von TOML-Format
library(ggplot2)      # Fortgeschrittene Datenvisualisierung
library(rmarkdown)    # Wissenschaftliches Reporting
library(knitr)        # Wissenschaftliches Reporting
library(kableExtra)   # Verbesserte Kable Tabellen
library(data.table)   # Fortgeschrittene Datenverarbeitung
library(quanteda)     # Fortgeschrittenes Natural Language Processing
```

```
## Package version: 4.0.2
## Unicode version: 14.0
## ICU version: 70.1
```

```
## Parallel computing: disabled
```

```
## See https://quanteda.io for tutorials and examples.
```

```
library(quanteda.textplots) # Quanteda: Diagramme
library(future)       # Parallelisierung mit Futures
```

```
##
## Attaching package: 'future'
```

```
## The following object is masked from 'package:rmarkdown':
##
##     run
```

```
library(future.apply) # Apply-Funtionen für Futures
library(zip)
```

```
##
## Attaching package: 'zip'
```

```
## The following objects are masked from 'package:utils':
##
##     unzip, zip
```

2.4 Zusätzliche Funktionen einlesen

Hinweis: Die hieraus verwendeten Funktionen werden jeweils vor der ersten Benutzung in vollem Umfang angezeigt um den Lesefluss zu verbessern.

```
source("functions/f.fast.freqtable.R")
source("functions/f.future_multihashes.R")
```

2.5 Verzeichnis für Analyse-Ergebnisse und Diagramme definieren

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

2.6 Allgemeine Konfiguration

2.6.1 Konfiguration einlesen

```
config <- parseTOML("config.toml")
```

2.6.2 Konfiguration anzeigen

```
print(config)
```

```
## List of 11
##  $ cebverfg :List of 2
##   ..$ date: chr "2024-07-24"
##   ..$ doi :List of 1
##   .. ..$ data:List of 1
##   .. .. ..$ version: chr "10.5281/zenodo.12705674"
##  $ cores    :List of 2
##   ..$ max   : logi TRUE
##   ..$ number: int 8
##  $ debug    :List of 1
##   ..$ cleanrun: logi TRUE
##  $ doi      :List of 2
##   ..$ data    :List of 2
##   .. ..$ concept: chr "10.5281/zenodo.4459405"
##   .. ..$ version: chr "10.5281/zenodo.13765529"
##   ..$ software:List of 2
##   .. ..$ concept: chr "10.5281/zenodo.4459415"
##   .. ..$ version: chr "10.5281/zenodo.13765530"
##  $ download :List of 1
##   ..$ timeout: int 600
##  $ fig      :List of 3
##   ..$ align : chr "center"
##   ..$ dpi   : int 300
##   ..$ format: chr [1:2] "pdf" "png"
##  $ freqtable:List of 1
##   ..$ ignore: chr [1:19] "text" "eingangsnummer" "datum" "doc_id" ...
##  $ license  :List of 2
##   ..$ code: chr "GNU General Public License Version 3 (GPLv3)"
##   ..$ data: chr "Creative Commons Zero 1.0 Universal"
##  $ parallel :List of 1
##   ..$ multihashes: logi TRUE
##  $ project  :List of 3
##   ..$ author   : chr "Seán Fobbe"
##   ..$ fullname : chr "Corona-Rechtsprechung des Bundesverfassungsgerichts"
##   ..$ shortname: chr "BVerfG-Corona"
##  $ quanteda :List of 1
##   ..$ tokens_locale: chr "de_DE"
```

2.6.3 Knitr Optionen setzen

```
knitr::opts_chunk$set(fig.path = paste0(dir.analysis, "/"),
                      dev = config$fig$format,
                      dpi = config$fig$dpi,
                      fig.align = config$fig$align)
```

2.6.4 Download Timeout setzen

```
options(timeout = config$download$timeout)
```

2.6.5 Quellenangabe für Diagramme definieren

```
caption <- paste("Fobbe | DOI:",
                 config$doi$data$version)
print(caption)
```

```
## [1] "Fobbe | DOI: 10.5281/zenodo.13765529"
```

2.6.6 Präfix für Dateien definieren

```
prefix.files <- paste0(config$project$shortname,
                 "_",
                 config$cebverfg$date)
print(prefix.files)
```

```
## [1] "BVerfG-Corona_2024-07-24"
```

2.6.7 Präfix für Diagramme definieren

```
prefix.figuretitle <- paste(config$project$shortname,
                            "| Version",
                            config$cebverfg$date)
```

2.6.8 Quanteda-Optionen setzen

```
quanteda_options(tokens_locale = config$quanteda$tokens_locale)
```

2.7 Dateien aus vorherigen Runs bereinigen

```
if(config$debug$cleanrun == TRUE){

    source("delete_all_data.R")

    }
```

2.8 Weitere Verzeichnisse definieren

```
dirs <- c("output",
          "temp",
          "files",
          "txt",
          "pdf")
```

2.9 Verzeichnisse anlegen

```
dir.create(dir.analysis)

lapply(dirs, dir.create)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
```

2.10 Vollzitate statistischer Software schreiben

```
knitr::write_bib(c(.packages()),
                 "temp/packages.bib")
```

2.11 LaTeX Konfiguration

2.11.1 LaTeX Parameter definieren

```
latexdefs <- c("%==========================\n% Definitionen\n
    %==========================",
                "\n% NOTE: Diese Datei wurde während des Kompilierungs-Prozesses
    automatisch erstellt.\n",
                "\n%-----Autor-----",
                paste0("\\newcommand{\\projectauthor}{",
                       config$project$author,
                       "}"),
                "\n%-----Version-----",
                paste0("\\newcommand{\\version}{",
                       config$cebverfg$date,
                       "}"),
                "\n%-----Titles-----",
                paste0("\\newcommand{\\datatitle}{",
                       config$project$fullname,
                       "}"),
                paste0("\\newcommand{\\datashort}{",
                       config$project$shortname,
                       "}"),
                paste0("\\newcommand{\\softwaretitle}{Source Code der \\enquote{",
                       config$project$fullname,
                       "}}"),
                paste0("\\newcommand{\\softwareshort}{",
                       config$project$shortname,
                       "-Source}"),
                "\n%-----Data DOIs-----",
                paste0("\\newcommand{\\dataconceptdoi}{",
                       config$doi$data$concept,
                       "}"),
                paste0("\\newcommand{\\dataversiondoi}{",
                       config$doi$data$version,
                       "}"),
                paste0("\\newcommand{\\dataconcepturldoi}{https://doi.org/",
                       config$doi$data$concept,
                       "}"),
                paste0("\\newcommand{\\dataversionurldoi}{https://doi.org/",
                       config$doi$data$version,
                       "}"),
                "\n%-----Software DOIs-----",
                paste0("\\newcommand{\\softwareconceptdoi}{",
                       config$doi$software$concept,
                       "}"),
                paste0("\\newcommand{\\softwareversiondoi}{",
                       config$doi$software$version,
                       "}"),
                paste0("\\newcommand{\\softwareconcepturldoi}{https://doi.org/",
                       config$doi$software$concept,
                       "}"),
                paste0("\\newcommand{\\softwareversionurldoi}{https://doi.org/",
                       config$doi$software$version,
                       "}"))
```

2.11.2 LaTeX Parameter schreiben

```
writeLines(latexdefs,
           paste0("temp/",
                  config$project$shortname,
                  "_Definitions.tex"))
```

2.12 Parallelisierung aktivieren

Parallelisierung wird zur Beschleunigung der Konvertierung von PDF zu TXT und der Datenanalyse mittels **quanteda** und **data.table** verwendet. Die Anzahl threads wird automatisch auf das verfügbare Maximum des Systems gesetzt, kann aber auch nach Belieben auf das eigene System angepasst werden. Die Parallelisierung kann deaktiviert werden, indem die Variable **fullCores** auf 1 gesetzt wird.

2.12.1 Anzahl logischer Kerne festlegen

```
if (config$cores$max == TRUE){
    fullCores <- availableCores()
}


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



print(fullCores)
```

```
## system
##     16
```

2.12.2 Quanteda

```
quanteda_options(threads = fullCores)
```

```
## Warning: Setting threads instead to maximum available 1
```

2.12.3 Data.table

```
setDTthreads(threads = fullCores)
```

3 Stamm-Datensatz einlesen (CE-BVerfG)

Der Stamm-Datensatz ist der »Corpus der Entscheidungen des Bundesverfassungsgerichts« (CE-BVerfG). Dieser enthält alle vom Bundesverfassungsgericht seit 1998 veröffentlichten Entscheidungen. Dessen **aktuellste** Version ist immer über diesen Digital Object Identifier (DOI) abrufbar: https://doi.org/10.5281/zenodo.3902658

3.1 Download der CSV-Datei

Der Datensatz im CSV-Format wird automatisch über einen verschlüsselten und langzeitstabilen Link aus dem wissenschaftlichen Archiv des CERN heruntergeladen. Dieses Vorgehen garantiert die Verwendung einer authentischen Version des Datensatzes.

```
zip.csv <- paste0("CE-BVerfG_",
                  config$cebverfg$date,
                  "_DE_CSV_Datensatz.zip")

print(zip.csv)
```

```
## [1] "CE-BVerfG_2024-07-24_DE_CSV_Datensatz.zip"
```

```
link.csv <- paste0("https://zenodo.org/record/",
                   gsub("10\\.5281/zenodo\\.([0-9]+)",
                        "\\1",
                        config$cebverfg$doi$data$version),
                   "/files/",
                   zip.csv,
                   "?download=1")

print(link.csv)
```

```
## [1] "https://zenodo.org/record/12705674/files/CE-BVerfG_2024-07-24_DE_CSV_
    Datensatz.zip?download=1"
```

```
if (file.exists(file.path("files", zip.csv)) == FALSE){

    download.file(link.csv,
                  file.path("files", zip.csv))

}
```

3.2 CSV-Datei einlesen

```
dt.bverfg <- fread(cmd = paste("unzip -cq",
                               file.path("files", zip.csv)))
```

3.3 Korpus-Objekt erstellen

```
corpus.bverfg <- corpus(dt.bverfg)
```

4 Keywords in Context (KWIC)

Bei einer KWIC-Analyse (keywords in context) wird nach einer bestimmten Zeichengefolge gesucht und sowohl diese, als auch die angrenzenden Wörter werden angezeigt. Konkret wird an dieser Stelle eine alternative Suche nach den Mustern “Corona”, “COVID” oder “SARS-CoV” durchgeführt. Groß- und Kleinschreibung wird ignoriert um eventuelle Tippfehler zu vernachlässigen. Das Sichtfenster wird auf 15 Tokens vor und nach dem Treffer gesetzt.

4.1 Tokenisierung

```
tokens <- tokens(corpus.bverfg,
                 what = "word",
                 remove_punct = FALSE,
                 remove_symbols = FALSE,
                 remove_numbers = FALSE,
                 remove_url = FALSE,
                 remove_separators = TRUE,
                 split_hyphens = FALSE,
                 include_docvars = TRUE,
                 padding = FALSE)
```

4.2 KWIC-Analyse durchführen

```
kwic <- kwic(tokens,
             pattern = "(Corona)|(COVID)|(SARS-CoV)",
             window = 15,
             valuetype = "regex",
             case_insensitive = TRUE)
```

4.3 KWIC-Tabelle speichern

```
file.kwic.sansdate <- paste(config$project$shortname,
                            "02_KeywordsInContext.csv",
                            sep = "_")

file.kwic.date <- paste(prefix.files,
                        "ANALYSE_02_KeywordsInContext.csv",
                        sep = "_")


fwrite(data.frame(kwic),
       file.path(dir.analysis,
              file.kwic.sansdate))


fwrite(data.frame(kwic),
       file.path("output",
                 file.kwic.date))
```

5 Lexical Dispersion Plot

Lexical Dispersion Plots zeigen mit einem vertikalen Strich an, an welcher Stelle in einem Dokument sich ein Token befindet. Alle Dokumente sind auf eine Länge von 1.0 normalisiert, d.h. ein Wert von 0.5 heißt immer, dass sich das Token in der Mitte des jeweiligen Dokumentes befindet. Viele und/oder dicke Striche deuten auf eine große Häufigkeit des Tokens hin.

5.1 Aufbereitung der Labels

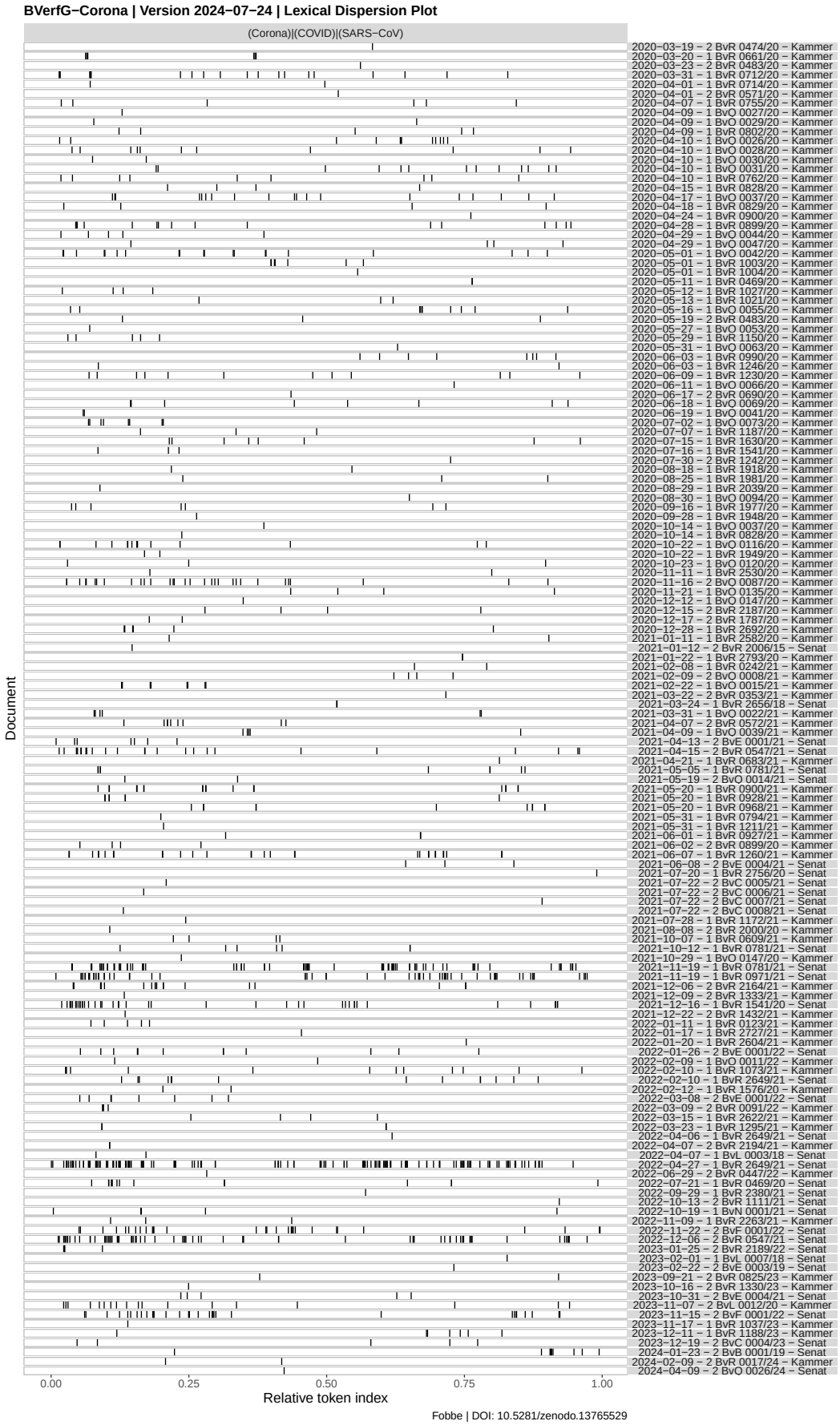
Die Labels müssen speziell aufbereitet werden, damit sie in der Grafik noch sinnvoll anzeigbar sind. Insbesondere die Dateinamen mit Entscheidungsnamen sind zu lang.

```
kwic.display <- kwic

split <- str_split(kwic$docname, pattern = "_")

f.recombine <- function(x){

    recombine <- paste0(x[2],
                        " - ",
                        x[4],
                        " ",
                        x[5],
                        " ",
                        x[6],
                        "/",
                        x[7],
                        " - ",
                        x[3] )

    recombine <- gsub("S$", "Senat", recombine)
    recombine <- gsub("K$", "Kammer", recombine)

    return(recombine)

}

filenames.display <- unlist(lapply(split, f.recombine))
kwic.display$docname <- filenames.display

split.attr <- str_split(attr(attr(kwic, "ntoken"), "names"), pattern = "_")
attrs.display <- unlist(lapply(split.attr, f.recombine))
attr(attr(kwic.display, "ntoken"), "names") <- attrs.display
```

5.2 Rechteckiges Format

```
textplot_xray(kwic.display,
              scale = "relative")+
    labs(
```

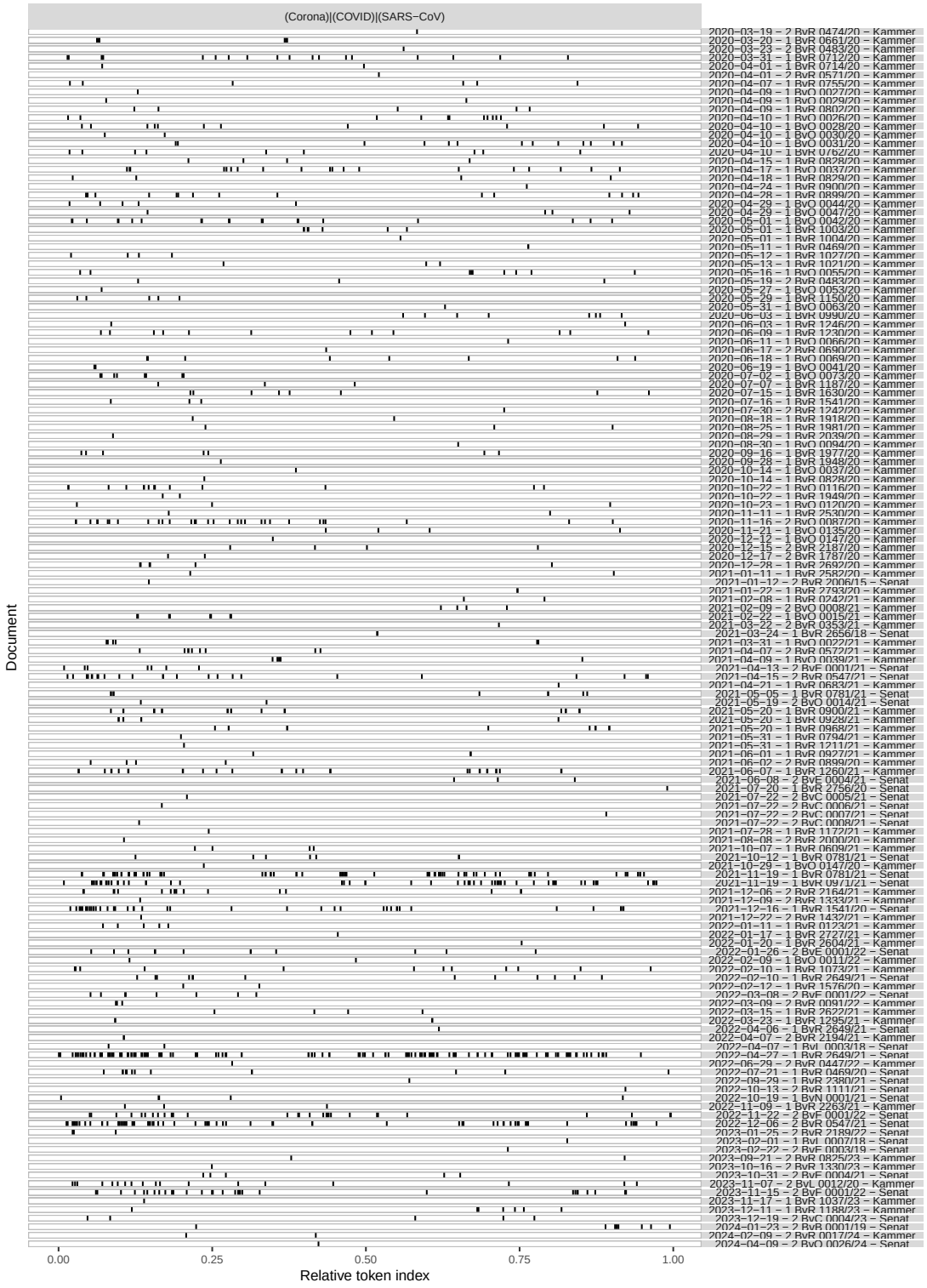
```
        title = paste(prefix.figuretitle,
                      "| Lexical Dispersion Plot"),
        caption = caption)+
    theme(
        text = element_text(size = 14),
        plot.title = element_text(size = 14,
                                  face = "bold"),
        legend.position = "none",
        plot.margin = margin(10, 20, 10, 10)
    )
```


BVerfG−Corona | Version 2024−07−24 | Lexical Dispersion Plot
(Corona)|(COVID)|(SARS−CoV)
Document
2020−03−19 − 2 BvR 0474/20 − Kammer
2020−03−20 − 1 BvR 0661/20 − Kammer
2020−03−23 − 2 BvR 0483/20 − Kammer
2020−03−31 − 1 BvR 0712/20 − Kammer
2020−04−01 − 1 BvR 0714/20 − Kammer
2020−04−01 − 2 BvR 0571/20 − Kammer
2020−04−07 − 1 BvR 0755/20 − Kammer
2020−04−09 − 1 BvQ 0027/20 − Kammer
2020−04−09 − 1 BvQ 0029/20 − Kammer
2020−04−09 − 1 BvR 0802/20 − Kammer
2020−04−10 − 1 BvQ 0026/20 − Kammer
2020−04−10 − 1 BvQ 0028/20 − Kammer
2020−04−10 − 1 BvQ 0030/20 − Kammer
2020−04−10 − 1 BvQ 0031/20 − Kammer
2020−04−10 − 1 BvR 0762/20 − Kammer
2020−04−15 − 1 BvR 0828/20 − Kammer
2020−04−17 − 1 BvQ 0037/20 − Kammer
2020−04−18 − 1 BvR 0829/20 − Kammer
2020−04−24 − 1 BvR 0900/20 − Kammer
2020−04−28 − 1 BvR 0899/20 − Kammer
2020−04−29 − 1 BvQ 0044/20 − Kammer
2020−04−29 − 1 BvQ 0047/20 − Kammer
2020−05−01 − 1 BvQ 0042/20 − Kammer
2020−05−01 − 1 BvR 1003/20 − Kammer
2020−05−01 − 1 BvR 1004/20 − Kammer
2020−05−11 − 1 BvR 0469/20 − Kammer
2020−05−12 − 1 BvR 1027/20 − Kammer
2020−05−13 − 1 BvR 1021/20 − Kammer
2020−05−16 − 1 BvQ 0055/20 − Kammer
2020−05−19 − 2 BvR 0483/20 − Kammer
2020−05−27 − 1 BvQ 0053/20 − Kammer
2020−05−29 − 1 BvR 1150/20 − Kammer
2020−05−31 − 1 BvQ 0063/20 − Kammer
2020−06−03 − 1 BvR 0990/20 − Kammer
2020−06−03 − 1 BvR 1246/20 − Kammer
2020−06−09 − 1 BvR 1230/20 − Kammer
2020−06−11 − 1 BvQ 0066/20 − Kammer
2020−06−17 − 2 BvR 0690/20 − Kammer
2020−06−18 − 1 BvQ 0069/20 − Kammer
2020−06−19 − 1 BvQ 0041/20 − Kammer
2020−07−02 − 1 BvQ 0073/20 − Kammer
2020−07−07 − 1 BvR 1187/20 − Kammer
2020−07−15 − 1 BvR 1630/20 − Kammer
2020−07−16 − 1 BvR 1541/20 − Kammer
2020−07−30 − 2 BvR 1242/20 − Kammer
2020−08−18 − 1 BvR 1918/20 − Kammer
2020−08−25 − 1 BvR 1981/20 − Kammer
2020−08−29 − 1 BvR 2039/20 − Kammer
2020−08−30 − 1 BvQ 0094/20 − Kammer
2020−09−16 − 1 BvR 1977/20 − Kammer
2020−09−28 − 1 BvR 1948/20 − Kammer
2020−10−14 − 1 BvQ 0037/20 − Kammer
2020−10−14 − 1 BvR 0828/20 − Kammer
2020−10−22 − 1 BvQ 0116/20 − Kammer
2020−10−22 − 1 BvR 1949/20 − Kammer
2020−10−23 − 1 BvQ 0120/20 − Kammer
2020−11−11 − 1 BvR 2530/20 − Kammer
2020−11−16 − 2 BvQ 0087/20 − Kammer
2020−11−21 − 1 BvQ 0135/20 − Kammer
2020−12−12 − 1 BvQ 0147/20 − Kammer
2020−12−15 − 2 BvR 2187/20 − Kammer
2020−12−17 − 2 BvR 1787/20 − Kammer
2020−12−28 − 1 BvR 2692/20 − Kammer
2021−01−11 − 1 BvR 2582/20 − Kammer
2021−01−12 − 2 BvR 2006/15 − Senat
2021−01−22 − 1 BvR 2793/20 − Kammer
2021−02−08 − 1 BvR 0242/21 − Kammer
2021−02−09 − 2 BvQ 0008/21 − Kammer
2021−02−22 − 1 BvQ 0015/21 − Kammer
2021−03−22 − 2 BvR 0353/21 − Kammer
2021−03−24 − 1 BvR 2656/18 − Senat
2021−03−31 − 1 BvQ 0022/21 − Kammer
2021−04−07 − 2 BvR 0572/21 − Kammer
2021−04−09 − 1 BvQ 0039/21 − Kammer
2021−04−13 − 2 BvE 0001/21 − Senat
2021−04−15 − 2 BvR 0547/21 − Senat
2021−04−21 − 1 BvR 0683/21 − Kammer
2021−05−05 − 1 BvR 0781/21 − Senat
2021−05−19 − 2 BvQ 0014/21 − Senat
2021−05−20 − 1 BvR 0900/21 − Kammer
2021−05−20 − 1 BvR 0928/21 − Kammer
2021−05−20 − 1 BvR 0968/21 − Kammer
2021−05−31 − 1 BvR 0794/21 − Kammer
2021−05−31 − 1 BvR 1211/21 − Kammer
2021−06−01 − 1 BvR 0927/21 − Kammer
2021−06−02 − 2 BvR 0899/20 − Kammer
2021−06−07 − 1 BvR 1260/21 − Kammer
2021−06−08 − 2 BvE 0004/21 − Senat
2021−07−20 − 1 BvR 2756/20 − Senat
2021−07−22 − 2 BvC 0005/21 − Senat
2021−07−22 − 2 BvC 0006/21 − Senat
2021−07−22 − 2 BvC 0007/21 − Senat
2021−07−22 − 2 BvC 0008/21 − Senat
2021−07−28 − 1 BvR 1172/21 − Kammer
2021−08−08 − 2 BvR 2000/20 − Kammer
2021−10−07 − 1 BvR 0609/21 − Kammer
2021−10−12 − 1 BvR 0781/21 − Senat
2021−10−29 − 1 BvQ 0147/20 − Kammer
2021−11−19 − 1 BvR 0781/21 − Senat
2021−11−19 − 1 BvR 0971/21 − Senat
2021−12−06 − 2 BvR 2164/21 − Kammer
2021−12−09 − 2 BvR 1333/21 − Kammer
2021−12−16 − 1 BvR 1541/20 − Senat
2021−12−22 − 2 BvR 1432/21 − Kammer
2022−01−11 − 1 BvR 0123/21 − Kammer
2022−01−17 − 1 BvR 2727/21 − Kammer
2022−01−20 − 1 BvR 2604/21 − Kammer
2022−01−26 − 2 BvE 0001/22 − Senat
2022−02−09 − 1 BvQ 0011/22 − Kammer
2022−02−10 − 1 BvR 1073/21 − Kammer
2022−02−10 − 1 BvR 2649/21 − Senat
2022−02−12 − 1 BvR 1576/20 − Kammer
2022−03−08 − 2 BvE 0001/22 − Senat
2022−03−09 − 2 BvR 0091/22 − Kammer
2022−03−15 − 1 BvR 2622/21 − Kammer
2022−03−23 − 1 BvR 1295/21 − Kammer
2022−04−06 − 1 BvR 2649/21 − Senat
2022−04−07 − 2 BvR 2194/21 − Kammer
2022−04−07 − 1 BvL 0003/18 − Senat
2022−04−27 − 1 BvR 2649/21 − Senat
2022−06−29 − 2 BvR 0447/22 − Kammer
2022−07−21 − 1 BvR 0469/20 − Senat
2022−09−29 − 1 BvR 2380/21 − Senat
2022−10−13 − 2 BvR 1111/21 − Senat
2022−10−19 − 1 BvN 0001/21 − Senat
2022−11−09 − 1 BvR 2263/21 − Kammer
2022−11−22 − 2 BvF 0001/22 − Senat
2022−12−06 − 2 BvR 0547/21 − Senat
2023−01−25 − 2 BvR 2189/22 − Senat
2023−02−01 − 1 BvL 0007/18 − Senat
2023−02−22 − 2 BvE 0003/19 − Senat
2023−09−21 − 2 BvR 0825/23 − Kammer
2023−10−16 − 2 BvR 1330/23 − Kammer
2023−10−31 − 2 BvE 0004/21 − Senat
2023−11−07 − 2 BvL 0012/20 − Kammer
2023−11−15 − 2 BvF 0001/22 − Senat
2023−11−17 − 1 BvR 1037/23 − Kammer
2023−12−11 − 1 BvR 1188/23 − Kammer
2023−12−19 − 2 BvC 0004/23 − Senat
2024−01−23 − 2 BvB 0001/19 − Senat
2024−02−09 − 2 BvR 0017/24 − Kammer
2024−04−09 − 2 BvQ 0026/24 − Senat
0.00
0.25
0.50
0.75
1.00
Relative token index
Fobbe | DOI: 10.5281/zenodo.13765529

5.3 A4-Format

```
textplot_xray(kwic.display,
              scale = "relative")+
    labs(
        title = paste(prefix.figuretitle,
                      "| Lexical Dispersion Plot"),
        caption = caption)+
    theme(
        text = element_text(size = 9),
        plot.title = element_text(size = 14,
                                  face = "bold"),
        legend.position = "none",
        plot.margin = margin(10, 20, 10, 10)
    )
```

BVerfG−Corona | Version 2024−07−24 | Lexical Dispersion Plot



Fobbe | DOI: 10.5281/zenodo.13765529

6 TXT-Datensatz erstellen

6.1 Namen der Corona-Entscheidungen definieren

```
keep.txt <- unique(kwic$docname)
```

6.2 Anzahl der TXT-Dateien

```
length(keep.txt)
```

```
## [1] 142
```

6.3 TXT-Datensatz herunterladen

```
zip.txt <- paste0("CE-BVerfG_",
                  config$cebverfg$date,
                  "_DE_TXT_Datensatz.zip")


link.txt <- paste0("https://zenodo.org/record/",
                   gsub("10\\.5281/zenodo\\.([0-9]+)",
                        "\\1",
                        config$cebverfg$doi$data$version),
                   "/files/",
                   zip.txt,
                   "?download=1")

zip.txt.rel <- file.path("files", zip.txt)



if(file.exists(zip.txt.rel) == FALSE){

    download.file(link.txt,
                  zip.txt.rel)

}
```

6.4 ZIP-Archiv entpacken

```
unzip(zip.txt.rel,
      files = keep.txt,
      exdir = "txt")
```

6.5 Corona-Entscheidungen verpacken

```
zip(paste0("output/",
           prefix.files,
           "_DE_TXT_Datensatz.zip"),
    list.files("txt", full.names = TRUE),
    mode = "cherry-pick")
```

7 PDF-Datensatz erstellen

7.1 Namen der Corona-Entscheidungen definieren

```
keep.pdf <- gsub(".txt",
                 ".pdf",
                 keep.txt)
```

7.2 Anzahl der PDF-Dateien

```
length(keep.pdf)
```

```
## [1] 142
```

7.3 PDF-Datensatz herunterladen

```
zip.pdf <- paste0("CE-BVerfG_",
                  config$cebverfg$date,
                  "_DE_PDF_Datensatz.zip")


link.pdf <- paste0("https://zenodo.org/record/",
                   gsub("10\\.5281/zenodo\\.([0-9]+)",
                        "\\1",
                        config$cebverfg$doi$data$version),
                   "/files/",
                   zip.pdf,
                   "?download=1")

zip.pdf.rel <- file.path("files", zip.pdf)


if(file.exists(zip.pdf.rel) == FALSE){

    download.file(link.pdf,
                  zip.pdf.rel)

}
```

7.4 ZIP-Archiv entpacken

```
unzip(zip.pdf.rel,
      files = keep.pdf,
      exdir = "pdf")
```

7.5 Corona-Entscheidungen verpacken

```
zip(paste0("output/",
           prefix.files,
           "_DE_PDF_Datensatz.zip"),
    list.files("pdf", full.names = TRUE),
    mode = "cherry-pick")
```

8 Frequenztabellen erstellen

8.1 CE-BVerfG auf Corona-Entscheidungen reduzieren

```
dt.corona <- dt.bverfg[doc_id %in% keep.txt]
```

8.2 Ignorierte Variablen

```
print(config$freqtable$ignore)
```

```
##  [1] "text"              "eingangsnummer"    "datum"
##  [4] "doc_id"            "seite"             "name"
##  [7] "ecli"              "aktenzeichen"      "aktenzeichen_alle"
## [10] "zeichen"           "tokens"            "typen"
## [13] "saetze"            "version"           "pressemitteilung"
## [16] "zitiervorschlag"   "kurzbeschreibung"  "url_pdf"
## [19] "url_html"
```

8.3 Liste zu prüfender Variablen

```
varlist <- names(dt.corona)

varlist <- setdiff(varlist,
                   config$freqtable$ignore)

print(varlist)
```

```
##  [1] "gericht"           "entscheidungsjahr" "entscheidung_typ"
##  [4] "spruchkoerper_typ" "spruchkoerper_az"  "registerzeichen"
##  [7] "verfahrensart"     "eingangsjahr_az"   "eingangsjahr_iso"
## [10] "kollision"         "bverfge"           "band"
## [13] "praesi"            "v_praesi"          "richter"
## [16] "doi_concept"       "doi_version"       "lizenz"
```

8.4 Frequenztabellen erstellen

```
prefix <- paste0(config$project$shortname,
                 "_00_Frequenztabelle_var-")
```

```
f.fast.freqtable(dt.corona,
                 varlist = varlist,
                 sumrow = TRUE,
                 output.list = FALSE,
                 output.kable = TRUE,
                 output.csv = TRUE,
                 outputdir = dir.analysis,
                 prefix = prefix,
                 align = c("p{5cm}",
                           rep("r", 4)))
```

Frequency Table for Variable: gericht

1 unique value(s) detected.

gericht	N	exactpercent	roundedpercent	cumulpercent
BVerfG	142	100	100	100
Total	142	100	100	100

Frequency Table for Variable: entscheidungsjahr

5 unique value(s) detected.

entscheidungsjahr	N	exactpercent	roundedpercent	cumulpercent
2020	63	44.366197	44.37	44.37
2021	41	28.873239	28.87	73.24
2022	24	16.901408	16.90	90.14
2023	11	7.746479	7.75	97.89
2024	3	2.112676	2.11	100.00
Total	142	100.000000	100.00	100.00

Frequency Table for Variable: entscheidung_typ

2 unique value(s) detected.

entscheidung_typ	N	exactpercent	roundedpercent	cumulpercent
B	137	96.478873	96.48	96.48
U	5	3.521127	3.52	100.00
Total	142	100.000000	100.00	100.00

Frequency Table for Variable: spruchkoerper_typ

2 unique value(s) detected.

spruchkoerper_typ	N	exactpercent	roundedpercent	cumulpercent
K	106	74.64789	74.65	74.65
S	36	25.35211	25.35	100.00
Total	142	100.00000	100.00	100.00

Frequency Table for Variable: spruchkoerper_az

2 unique value(s) detected.

spruchkoerper_az	N	exactpercent	roundedpercent	cumulpercent
1	97	68.30986	68.31	68.31
2	45	31.69014	31.69	100.00
Total	142	100.00000	100.00	100.00

Frequency Table for Variable: registerzeichen

8 unique value(s) detected.

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
BvB	1	0.7042254	0.70	0.70
BvC	5	3.5211268	3.52	4.23
BvE	6	4.2253521	4.23	8.45

(continued)

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
BvF	2	1.4084507	1.41	9.86
BvL	3	2.1126761	2.11	11.97
BvN	1	0.7042254	0.70	12.68
BvQ	32	22.5352113	22.54	35.21
BvR	92	64.7887324	64.79	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: verfahrensart

8 unique value(s) detected.

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Abstrakte Normenkontrolle	2	1.4084507	1.41	1.41
Divergenzvorlagen eines Landesverfassungsgerichts zur Auslegung des Grundgesetzes	1	0.7042254	0.70	2.11
Einstweilige Anordnungen	32	22.5352113	22.54	24.65
Konkrete Normenkontrolle	3	2.1126761	2.11	26.76
Organstreitverfahren	6	4.2253521	4.23	30.99
Verfassungsbeschwerden; Kommunalverfassungsbeschwerden	92	64.7887324	64.79	95.77
Verfassungswidrigkeit von Parteien	1	0.7042254	0.70	96.48
Wahlprüfungsverfahren	5	3.5211268	3.52	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_az

8 unique value(s) detected.

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
15	1	0.7042254	0.70	0.70
18	3	2.1126761	2.11	2.82
19	2	1.4084507	1.41	4.23
20	73	51.4084507	51.41	55.63
21	48	33.8028169	33.80	89.44
22	8	5.6338028	5.63	95.07
23	5	3.5211268	3.52	98.59
24	2	1.4084507	1.41	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_iso

8 unique value(s) detected.

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
2015	1	0.7042254	0.70	0.70
2018	3	2.1126761	2.11	2.82
2019	2	1.4084507	1.41	4.23
2020	73	51.4084507	51.41	55.63
2021	48	33.8028169	33.80	89.44
2022	8	5.6338028	5.63	95.07
2023	5	3.5211268	3.52	98.59
2024	2	1.4084507	1.41	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: kollision

2 unique value(s) detected.

kollision	N	exactpercent	roundedpercent	cumulpercent
NA	141	99.2957746	99.3	99.3
a	1	0.7042254	0.7	100.0
Total	142	100.0000000	100.0	100.0

Frequency Table for Variable: bverfge

2 unique value(s) detected.

bverfge	N	exactpercent	roundedpercent	cumulpercent
FALSE	23	16.19718	16.2	16.2
TRUE	119	83.80282	83.8	100.0
Total	142	100.00000	100.0	100.0

Frequency Table for Variable: band

10 unique value(s) detected.

band	N	exactpercent	roundedpercent	cumulpercent
NA	119	83.8028169	83.80	83.80
156	1	0.7042254	0.70	84.51
157	4	2.8169014	2.82	87.32
158	2	1.4084507	1.41	88.73
159	4	2.8169014	2.82	91.55
160	4	2.8169014	2.82	94.37
161	2	1.4084507	1.41	95.77
162	1	0.7042254	0.70	96.48
163	3	2.1126761	2.11	98.59
164	2	1.4084507	1.41	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: praesi

2 unique value(s) detected.

praesi	N	exactpercent	roundedpercent	cumulpercent
Harbarth	102	71.83099	71.83	71.83
Voßkuhle	40	28.16901	28.17	100.00
Total	142	100.00000	100.00	100.00

Frequency Table for Variable: v_praesi

2 unique value(s) detected.

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Harbarth	40	28.16901	28.17	28.17
König	102	71.83099	71.83	100.00
Total	142	100.00000	100.00	100.00

Frequency Table for Variable: richter

28 unique value(s) detected.

richter	N	exactpercent	roundedpercent	cumulpercent
Baer|Ott|Radtke	17	11.9718310	11.97	11.97
Christ|Wolff|Meßling	1	0.7042254	0.70	12.68
Harbarth|Baer|Britz|Ott|Christ|Radtke|Härtel	3	2.1126761	2.11	14.79
Harbarth|Baer|Britz|Ott|Christ|Radtke|Härtel|Wolff	1	0.7042254	0.70	15.49
Harbarth|Baer|Ott	5	3.5211268	3.52	19.01
Harbarth|Britz|Ott	2	1.4084507	1.41	20.42
Harbarth|Britz|Radtke	25	17.6056338	17.61	38.03
Harbarth|Ott|Härtel	1	0.7042254	0.70	38.73

(continued)

richter	N	exactpercent	roundedpercent	cumulpercent
Harbarth|Paulus|Baer|Britz|Ott|Christ|Radtke|Härtel	10	7.0422535	7.04	45.77
Hermanns|Maidowski|Langenfeld	2	1.4084507	1.41	47.18
Huber|Kessal-Wulf|König	5	3.5211268	3.52	50.70
Huber|Kessal-Wulf|Wallrabenstein	7	4.9295775	4.93	55.63
Kessal-Wulf|Wallrabenstein|Offenloch	2	1.4084507	1.41	57.04
König|Huber|Hermanns|Müller|Kessal-Wulf|Langenfeld|Wallrabenstein	4	2.8169014	2.82	59.86
König|Huber|Hermanns|Müller|Kessal-Wulf|Maidowski|Langenfeld	1	0.7042254	0.70	60.56
König|Huber|Hermanns|Müller|Kessal-Wulf|Maidowski|Langenfeld|Wallrabenstein	10	7.0422535	7.04	67.61
König|Maidowski|Langenfeld|Wallrabenstein|Fetzer|Offenloch|Franke|Wöckel	1	0.7042254	0.70	68.31
König|Maidowski|Offenloch	1	0.7042254	0.70	69.01
König|Maidowski|Wallrabenstein	1	0.7042254	0.70	69.72
König|Müller|Kessal-Wulf|Langenfeld|Wallrabenstein|Fetzer|Offenloch	3	2.1126761	2.11	71.83
König|Müller|Kessal-Wulf|Maidowski|Langenfeld|Wallrabenstein|Fetzer|Offenloch	2	1.4084507	1.41	73.24
König|Müller|Maidowski	5	3.5211268	3.52	76.76
Langenfeld|Fetzer|Offenloch	1	0.7042254	0.70	77.46
Masing|Paulus|Christ	15	10.5633803	10.56	88.03
Ott|Radtke|Wolff	1	0.7042254	0.70	88.73
Paulus|Britz|Ott|Christ|Radtke|Härtel	1	0.7042254	0.70	89.44
Paulus|Christ|Härtel	14	9.8591549	9.86	99.30
Paulus|Christ|Radtke	1	0.7042254	0.70	100.00
Total	142	100.0000000	100.00	100.00

Frequency Table for Variable: doi_concept

1 unique value(s) detected.

doi_concept	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.3902658	142	100	100	100
Total	142	100	100	100

Frequency Table for Variable: doi_version

1 unique value(s) detected.

doi_version	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.12705674	142	100	100	100
Total	142	100	100	100

Frequency Table for Variable: lizenz

1 unique value(s) detected.

lizenz	N	exactpercent	roundedpercent	cumulpercent
Creative Commons Zero 1.0 Universal (CC Zero 1.0)	142	100	100	100
Total	142	100	100	100

9 Diagramm Kopieren

```
rechteckig.source <- list.files(dir.analysis,
                                pattern = "Rechteckig.*\\.pdf",
                                full.names = TRUE)

rechteckig.destination <- file.path("output",
                                    gsub("BVerfG-Corona",
                                         paste0(prefix.files, "_ANALYSE"),
                                         basename(rechteckig.source)))

rechteckig.destination <- gsub("-1\\.pdf",
                               "\\.pdf",
                               rechteckig.destination)

file.copy(rechteckig.source,
          rechteckig.destination)
```

```
## [1] TRUE
```

10 Erstellen der ZIP-Archive

10.1 Verpacken der Analyse-Dateien

```
zip(paste0("output/",
           prefix.files,
           "_DE_ANALYSE.zip"),
    basename(dir.analysis))
```

10.2 Verpacken der Source-Dateien

```
files.source <- c(system2("git", "ls-files", stdout = TRUE),
                  ".git")



zip(paste0("output/",
          prefix.files,
          "_Source_Files.zip"),
    files.source)
```

11 Kryptographische Hashes

Dieses Modul berechnet für jedes ZIP-Archiv zwei Arten von Hashes: SHA2-256 und SHA3-512. Mit diesen kann die Authentizität der Dateien geprüft werden und es wird dokumentiert, dass sie aus diesem Source Code hervorgegangen sind. Die SHA-2 und SHA-3 Algorithmen sind äußerst resistent gegenüber *collision* und *pre-imaging* Angriffen, sie gelten derzeit als kryptographisch sicher. Ein SHA3-Hash mit 512 bit Länge ist nach Stand von Wissenschaft und Technik auch gegenüber quantenkryptoanalytischen Verfahren unter Einsatz des *Grover-Algorithmus* hinreichend resistent.

11.1 Liste der ZIP-Archive erstellen

```
files.zip <- list.files("output",
                        pattern = "\\.zip$",
                        full.names = TRUE,
                        ignore.case = TRUE)
```

11.2 Funktion anzeigen: future_multihashes

```
print(f.future_multihashes)
```

```
## function (x)
## {
##     begin <- Sys.time()
##     message(paste("Processing", length(x), "files. Begin at:",
##         begin))
##     hashes.list <- future.apply::future_lapply(x, f.multihashes)
##     hashes.table <- data.table::rbindlist(hashes.list)
##     data.table::setDF(hashes.table)
##     end <- Sys.time()
##     duration <- end - begin
##     message(paste0("Processed ", length(x), " files. Runtime was ",
##         round(duration, digits = 2), " ", attributes(duration)$units,
##         "."))
##     return(hashes.table)
## }
```

11.3 Hashes berechnen

```
if(config$parallel$multihashes == TRUE){

    plan("multicore",
         workers = fullCores)

}else{
```

```
    plan("sequential")

     }


multihashes <- f.future_multihashes(files.zip)
```

```
## Processing 4 files. Begin at: 2024-09-15 19:34:24.648066
```

```
## Processed 4 files. Runtime was 0.15 secs.
```

11.4 In Data Table umwandeln

```
setDT(multihashes)

setnames(multihashes,
         old = "x",
         new = "filename")
```

11.5 Index hinzufügen

```
multihashes$index <- seq_len(multihashes[,.N])
```

11.6 In Datei schreiben

```
fwrite(multihashes,
       file.path("output",
                 paste(prefix.files,
                       "KryptographischeHashes.csv",
                       sep = "_")),
       na = "NA")
```

11.7 Leerzeichen hinzufügen um Zeilenumbruch zu ermöglichen

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

11.8 In Bericht anzeigen

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

index	filename
1	output/BVerfG-Corona_2024-07-24_DE_ANALYSE.zip
2	output/BVerfG-Corona_2024-07-24_DE_PDF_Datensatz.zip
3	output/BVerfG-Corona_2024-07-24_DE_TXT_Datensatz.zip
4	output/BVerfG-Corona_2024-07-24_Source_Files.zip

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

index	sha2.256
1	7ac927ecde3a952e9bf577a8bf5faecacdac1a92967ec3e8d53b243ae2e91388
2	65a43e3379288dfda8c652df8993cf9186529356f75003d3cae7d34d81b61d38
3	857a83f92f43b8708598d8ebffdd80662a047837888fa75488a6d0f791014402
4	d0db4a564fa86bd389cea37c04ae0189e00c3b2e9b085d32b856a95804166d2c

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

index	sha3.512
1	a35cd2b71bbc3707e17f289fde6eac0e49594f2b95ef6bae36081706726cb9ec 870219aa1ed9388cea0b3a43483a9b62f203a1106d77c8fe5b76f6a0d2c6e1d9
2	c20e9784cd3ae8e6808589e2675333427609300be9a8b6f78c97b6a23aa82461 78ef- caa5ab9aac848805b30d78c59a9fdac1087fb688c6647d5e30d19609d395
3	80824b487ed86cf66802373227l8b0f2698f8ced3b3360d9efae8f2ca4ad0c1f ee3cf8bf9357e1068a27b31e63a990e8ac885a310c2afec92fa2c301bdbe4fa3
4	c063ec3fe1e5f1313e66d674010a3acb3348879458b3581c8f719df42f632d50 5da0344f2b4860c974a4252709466a7aeb75f5b042faa791dce46fb161b23bd7

12 Abschluss

12.1 Datumsstempel

Hinweis: der Datumsstempel weicht vom Zeitpunkt der tatsächlichen Erstellung des Datensatzes ab, weil sich der Datumsstempel nach dem Tag des Abrufs des CE-BVerfG richtet.

```
print(config$cebverfg$date)
```

```
## [1] "2024-07-24"
```

12.2 Datum und Uhrzeit (Anfang)

```
print(begin.script)
```

```
## [1] "2024-09-15 19:32:08 CEST"
```

12.3 Datum und Uhrzeit (Ende)

```
end.script <- Sys.time()
print(end.script)
```

```
## [1] "2024-09-15 19:34:24 CEST"
```

12.4 Laufzeit des gesamten Skriptes

```
print(end.script - begin.script)
```

```
## Time difference of 2.280486 mins
```

12.5 Warnungen

```
warnings()
```

13 Parameter für strenge Replikationen

```
system2("openssl", "version", stdout = TRUE)
```

```
## [1] "OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)"
```

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 22.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so;
    LAPACK version 3.10.0
##
## locale:
##  [1] LC_CTYPE=en_US.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
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] zip_2.3.1                 future.apply_1.11.2
##  [3] future_1.33.2             quanteda.textplots_0.94.4
##  [5] quanteda_4.0.2            data.table_1.15.4
##  [7] kableExtra_1.4.0          knitr_1.48
##  [9] ggplot2_3.5.1             magick_2.8.3
## [11] stringr_1.5.1             RcppTOML_0.2.2
## [13] rmarkdown_2.27
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    xml2_1.3.6        stringi_1.8.4
##  [5] lattice_0.22-6    listenv_0.9.1     digest_0.6.36     magrittr_2.0.3
##  [9] evaluate_0.24.0   grid_4.4.0        fastmap_1.2.0     Matrix_1.7-0
## [13] stopwords_2.3     fansi_1.0.6       viridisLite_0.4.2 scales_1.3.0
## [17] codetools_0.2-20  cli_3.6.3         rlang_1.1.4       parallelly_1.37.1
## [21] munsell_0.5.1     withr_3.0.0       yaml_2.3.9        parallel_4.4.0
## [25] tools_4.4.0       dplyr_1.1.4       colorspace_2.1-0  fastmatch_1.1-4
## [29] globals_0.16.3    vctrs_0.6.5       R6_2.5.1          lifecycle_1.0.4
```

```
## [33] pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.5      glue_1.7.0
## [37] Rcpp_1.0.12       systemfonts_1.1.0 xfun_0.45         tibble_3.2.1
## [41] tidyselect_1.2.1  rstudioapi_0.16.0 farver_2.1.2      htmltools_0.5.8.1
## [45] labeling_0.4.3    svglite_2.1.3     compiler_4.4.0
```

14 Changelog

14.1 Version 2024-07-24

- Vollständige Aktualisierung der Daten
- LIZENZÄNDERUNG: Source Code jetzt unter GNU General Public License Version 3 (GPLv3) oder später lizenziert
- R-Version und Rocker Image auf 4.4.0 aktualisiert (wegen CVE-2024-27322)
- Vereinfachung der Repository-Struktur mit Ordner etc/ für Konfigurationsdateien
- Anpassung der Docker Compose-Konfiguration an Debian 11

14.2 Version 2023-02-06

- Vollständige Aktualisierung der Daten
- Gesamte Laufzeitumgebung nun mit Docker versionskontrolliert
- Funktionen werden nun nicht mehr aus einem Submodule bezogen, sondern sind direkt im Projekt verankert
- Vereinfachung der Konfigurations-Datei
- ZIP-Archiv mit Source Code wird nun aus dem Git-Manifest generiert
- README im Hinblick auf Docker überarbeitet
- Speichern von temporären Dateien nun in speziellen Ordnern in files/, pdf/ und txt/
- Option für automatische Löschung der Dateien aus vorherigen Runs zu Konfiguration hinzugefügt
- Delete-Skript hinzugefügt

14.3 Version 2022-08-24

- Vollständige Aktualisierung der Daten
- Diagramme sind deutlich überarbeitet und die Labels verschönert worden
- Umbenennung des run scripts und der Konfigurations-Datei

14.4 Version 2022-02-01

- Vollständige Aktualisierung der Daten
- Strenge Versionskontrolle von R packages mit **renv**
- Kompilierung jetzt detailliert konfigurierbar, insbesondere die Parallelisierung
- Parallelisierung nun vollständig mit *future* statt mit *foreach* und *doParallel*
- Fehlerhafte Kompilierungen werden vor der nächsten Kompilierung vollautomatisch aufgeräumt
- Alle Ergebnisse werden automatisch fertig verpackt in den Ordner ‘output’ sortiert
- README und CHANGELOG sind jetzt externe Markdown-Dateien, die bei der Kompilierung automatisiert eingebunden werden

14.5 Version 2021-09-19

- Vollständige Aktualisierung der Daten

14.6 Version 2021-05-20

- Vollständige Aktualisierung der Daten

14.7 Version 2021-01-08

- Erstveröffentlichung

Literaturverzeichnis

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

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

Bengtsson, Henrik. 2021. "A Unifying Framework for Parallel and Distributed Processing in R Using Futures." *The R Journal* 13 (2): 208–27. https://doi.org/10.32614/RJ-2021-048.

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

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

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018a. "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.

———. 2018b. "Quanteda: An R Package for the Quantitative Analysis of Textual Data." *Journal of Open Source Software* 3 (30): 774. https://doi.org/10.21105/joss.00774.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, Akitaka Matsuo, and William Lowe. 2024. *Quanteda: Quantitative Analysis of Textual Data*. https://quanteda.io.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2024. *Quanteda.textplots: Plots for the Quantitative Analysis of Textual Data*.

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

Eddelbuettel, Dirk. 2023. *RcppTOML: Rcpp Bindings to Parser for "Tom's Obvious Markup Language"*. http://dirk.eddelbuettel.com/code/rcpp.toml.html.

Ooms, Jeroen. 2024. *Magick: Advanced Graphics and Image-Processing in R*. https://docs.ropensci.org/magick/.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

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

———. 2023. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://stringr.tidyverse.org.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, and Teun van den Brand. 2024. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://ggplot2.tidyverse.org.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich

Leisch, and Roger D. Peng. Chapman; Hall/CRC.

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

———. 2024. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

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

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

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