

# Corona-Rechtsprechung des Bundesverfassungsgerichts (BVerfG-Corona-Source)

COMPILATION REPORT

Version 2022-02-01

License MIT-0

DOI: 10.5281/zenodo.5959980

<b>Titel</b>	Source Code der »Corona-Rechtsprechung des Bundesverfassungsgerichts«
<b>Abkürzung</b>	BVerfG-Corona-Source
<b>Autor</b>	Seán Fobbe
<b>Version</b>	2022-02-01
<b>Download</b>	<a href="https://doi.org/10.5281/zenodo.5959980">https://doi.org/10.5281/zenodo.5959980</a>
<b>Lizenz</b>	MIT No Attribution (MIT-0)

### Zitiervorschlag

*Seán Fobbe* (2022). Source Code der »Corona-Rechtsprechung des Bundesverfassungsgerichts« (BVerfG-Corona-Source). Version 2022-02-01. Zenodo. DOI: 10.5281/zenodo.5959980.

### 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 2022-02-01. 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.

### Lizenz: MIT No Attribution (MIT-0)

Copyright — 2022 — Seán Fobbe

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

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

### Disclaimer

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

# Inhaltsverzeichnis

<b>1</b>	<b>README</b>	<b>6</b>
1.1	Überblick . . . . .	6
1.2	Funktionsweise . . . . .	6
1.3	Systemanforderungen . . . . .	6
1.3.1	Betriebssystem . . . . .	6
1.3.2	Software . . . . .	6
1.3.3	Parallelisierung . . . . .	7
1.3.4	Speicherplatz . . . . .	7
1.4	Kompilierung . . . . .	7
<b>2</b>	<b>Vorbereitung</b>	<b>8</b>
2.1	Datumsstempel . . . . .	8
2.2	Datum und Uhrzeit (Beginn) . . . . .	8
2.3	Packages Laden . . . . .	8
2.4	Zusätzliche Funktionen einlesen . . . . .	9
2.5	Verzeichnis für Analyse-Ergebnisse und Diagramme definieren . . . . .	9
2.6	Weitere Verzeichnisse definieren . . . . .	10
2.7	Dateien aus vorherigen Runs bereinigen . . . . .	10
2.8	Verzeichnisse anlegen . . . . .	10
2.9	Vollzitate statistischer Software schreiben . . . . .	10
2.10	Allgemeine Konfiguration . . . . .	11
2.10.1	Konfiguration einlesen . . . . .	11
2.10.2	Konfiguration anzeigen . . . . .	11
2.10.3	Knitr Optionen setzen . . . . .	11
2.10.4	Download Timeout setzen . . . . .	12
2.10.5	Quellenangabe für Diagramme definieren . . . . .	12
2.10.6	Präfix für Dateien definieren . . . . .	12
2.10.7	Präfix für Diagramme definieren . . . . .	12
2.10.8	Quanteda-Optionen setzen . . . . .	12
2.11	LaTeX Konfiguration . . . . .	13
2.11.1	LaTeX Parameter definieren . . . . .	13
2.11.2	LaTeX Parameter schreiben . . . . .	14
2.12	Parallelisierung aktivieren . . . . .	14
2.12.1	Anzahl logischer Kerne festlegen . . . . .	14
2.12.2	Quanteda . . . . .	14
2.12.3	Data.table . . . . .	14
<b>3</b>	<b>Stamm-Datensatz einlesen (CE-BVerfG)</b>	<b>15</b>
3.1	Download der CSV-Datei . . . . .	15
3.2	CSV-Datei einlesen . . . . .	15
3.3	Korpus-Objekt erstellen . . . . .	16
<b>4</b>	<b>Keywords in Context (KWIC)</b>	<b>17</b>
4.1	Tokenisierung . . . . .	17
4.2	KWIC-Analyse durchführen . . . . .	17
4.3	KWIC-Tabelle speichern . . . . .	17
<b>5</b>	<b>Lexical Dispersion Plot</b>	<b>18</b>

5.1	Rechteckiges Format . . . . .	18
5.2	A4-Format . . . . .	20
<b>6</b>	<b>TXT-Datensatz erstellen</b>	<b>22</b>
6.1	Namen der Corona-Entscheidungen definieren . . . . .	22
6.2	Anzahl der TXT-Dateien . . . . .	22
6.3	TXT-Datensatz herunterladen . . . . .	22
6.4	ZIP-Archiv entpacken . . . . .	22
6.5	Corona-Entscheidungen verpacken . . . . .	23
6.6	TXT-Dateien löschen . . . . .	23
<b>7</b>	<b>PDF-Datensatz erstellen</b>	<b>24</b>
7.1	Namen der Corona-Entscheidungen definieren . . . . .	24
7.2	Anzahl der PDF-Dateien . . . . .	24
7.3	PDF-Datensatz herunterladen . . . . .	24
7.4	ZIP-Archiv entpacken . . . . .	24
7.5	Corona-Entscheidungen verpacken . . . . .	25
7.6	PDF-Dateien löschen . . . . .	25
<b>8</b>	<b>Frequenztabellen erstellen</b>	<b>26</b>
8.1	CE-BVerfG auf Corona-Entscheidungen reduzieren . . . . .	26
8.2	Funktion anzeigen: f.fast.freqtable . . . . .	26
8.3	Ignorierte Variablen . . . . .	27
8.4	Liste zu prüfender Variablen . . . . .	27
8.5	Frequenztabellen erstellen . . . . .	28
<b>9</b>	<b>Diagramm Kopieren</b>	<b>35</b>
<b>10</b>	<b>Erstellen der ZIP-Archive</b>	<b>36</b>
10.1	Verpacken der Analyse-Dateien . . . . .	36
10.2	Verpacken der Source-Dateien . . . . .	36
<b>11</b>	<b>Kryptographische Hashes</b>	<b>37</b>
11.1	Liste der ZIP-Archive erstellen . . . . .	37
11.2	Funktion anzeigen: future_multihashes . . . . .	37
11.3	Hashes berechnen . . . . .	38
11.4	In Data Table umwandeln . . . . .	38
11.5	Index hinzufügen . . . . .	38
11.6	In Datei schreiben . . . . .	39
11.7	Leerzeichen hinzufügen um Zeilenumbruch zu ermöglichen . . . . .	39
11.8	In Bericht anzeigen . . . . .	39
<b>12</b>	<b>Aufräumen</b>	<b>41</b>
<b>13</b>	<b>Abschluss</b>	<b>42</b>
13.1	Datumsstempel . . . . .	42
13.2	Datum und Uhrzeit (Anfang) . . . . .	42
13.3	Datum und Uhrzeit (Ende) . . . . .	42
13.4	Laufzeit des gesamten Skriptes . . . . .	42
13.5	Warnungen . . . . .	42

<b>14 Parameter für strenge Replikationen</b>	<b>43</b>
<b>15 Changelog</b>	<b>44</b>
15.1 Version 2022-02-01 . . . . .	44
15.2 Version 2021-09-19 . . . . .	44
15.3 Version 2021-05-20 . . . . .	44
15.4 Version 2021-01-08 . . . . .	44
<b>Literaturverzeichnis</b>	<b>45</b>

# 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

### 1.3.1 Betriebssystem

Das Skript in seiner veröffentlichten Form kann nur unter **Linux** ausgeführt werden, da es Linux-spezifische Optimierungen (z.B. Fork Cluster) und Shell-Kommandos (z.B. OpenSSL) nutzt. Das Skript wurde unter Fedora Linux entwickelt und getestet. Die zur Kompilierung benutzte Version entnehmen Sie bitte dem **sessionInfo()**-Ausdruck am Ende des jeweiligen Compilation Reports.

### 1.3.2 Software

Sie müssen die Programmiersprache R installiert haben. Starten Sie danach eine Session im Ordner des Projekts, Sie sollten automatisch zur Installation aller packages in der empfohlenen Version aufgefordert werden. Andernfalls führen Sie bitte folgenden Befehl aus:

```
renv::restore()
```

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

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

Alternativ können sie das R package **tinytex** installieren.

### 1.3.3 Parallelisierung

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

### 1.3.4 Speicherplatz

Auf der Festplatte sollten 4 GB Speicherplatz vorhanden sein.

## 1.4 Kompilierung

Alle Kommentare sind im roxygen2-Stil gehalten. Die beiden Skripte können daher auch **ohne render()** regulär als R-Skripte ausgeführt werden. Es wird in diesem Fall kein PDF-Bericht erstellt und Diagramme werden nicht abgespeichert.

Um den **vollständigen Datensatz** zu kompilieren, sowie Compilation Report und Codebook zu erstellen, kopieren Sie bitte alle im Source-Archiv bereitgestellten Dateien in einen leeren Ordner (!) und führen mit R diesen Befehl aus:

```
source("00_BVerfG-Corona_FullCompile.R")
```

## 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] "2022-02-03 15:40:09 CET"
```

### 2.3 Packages Laden

```
library(magick)      # Cropping von PNG-Dateien
```

```
## Linking to ImageMagick 6.9.12.32
## Enabled features: cairo, fontconfig, freetype, ghostscript, lcms, pango, raw,
    rsvg, webp, x11
## Disabled features: fftw, heic
```

```
## 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
```

```
## data.table 1.14.2 using 8 threads (see ?getDTthreads). Latest news: r-
    datatable.com
```

```
library(quanteda)    # Fortgeschrittenes Natural Language Processing
```



```
## Package version: 3.2.0
## Unicode version: 13.0
## ICU version: 67.1
```

```
## Parallel computing: 16 of 16 threads used.
```

```
## 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
```

## 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("R-fobbe-proto-package/f.fast.freqtable.R")
source("R-fobbe-proto-package/f.future_multihashes.R")
```

## 2.5 Verzeichnis für Analyse-Ergebnisse und Diagramme definieren

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

## 2.6 Weitere Verzeichnisse definieren

```
dirs <- c("output",  
         "temp",  
         "data")
```

## 2.7 Dateien aus vorherigen Runs bereinigen

```
unlink(dir.analysis,  
       recursive = TRUE)  
  
unlink(dirs,  
       recursive = TRUE)  
  
files.delete <- list.files(pattern = "\\\\.zip|\\.jpe?g|\\.png|\\.gif|\\.pdf|\\.txt  
|\\.bib|\\.csv|\\.spin\\.|\\.log|\\.html?",  
                           ignore.case = TRUE)  
  
unlink(files.delete)
```

## 2.8 Verzeichnisse anlegen

```
dir.create(dir.analysis)  
  
lapply(dirs, dir.create)
```

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

## 2.9 Vollzitate statistischer Software schreiben

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

## 2.10 Allgemeine Konfiguration

### 2.10.1 Konfiguration einlesen

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

### 2.10.2 Konfiguration anzeigen

```
print(config)
```

```
## List of 10
## $ cebverfg :List of 2
## ..$ date: chr "2022-02-01"
## ..$ doi :List of 1
## .. ..$ data:List of 1
## .. .. ..$ version: chr "10.5281/zenodo.5910152"
## $ cores :List of 2
## ..$ max : logi TRUE
## ..$ number: int 8
## $ doi :List of 2
## ..$ data :List of 2
## .. ..$ concept: chr "10.5281/zenodo.4459405"
## .. ..$ version: chr "10.5281/zenodo.5959979"
## ..$ software:List of 2
## .. ..$ concept: chr "10.5281/zenodo.4459415"
## .. ..$ version: chr "10.5281/zenodo.5959980"
## $ 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 [1:17] "text" "eingangsnummer" "datum" "doc_id" ...
## $ license :List of 2
## ..$ code: chr "MIT-0"
## ..$ 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"
```

### 2.10.3 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)
```

#### 2.10.4 Download Timeout setzen

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

#### 2.10.5 Quellenangabe für Diagramme definieren

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

```
## [1] "Fobbe | DOI: 10.5281/zenodo.5959979"
```

#### 2.10.6 Präfix für Dateien definieren

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

```
## [1] "BVerfG-Corona_2022-02-01"
```

#### 2.10.7 Präfix für Diagramme definieren

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

#### 2.10.8 Quanteda-Optionen setzen

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

## 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)
```

### 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 langzeit-stabilen 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_2022-02-01_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/5910152/files/CE-BVerfG_2022-02-01_DE_CSV_
    Datensatz.zip?download=1"
```

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

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

}
```

#### 3.2 CSV-Datei einlesen

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

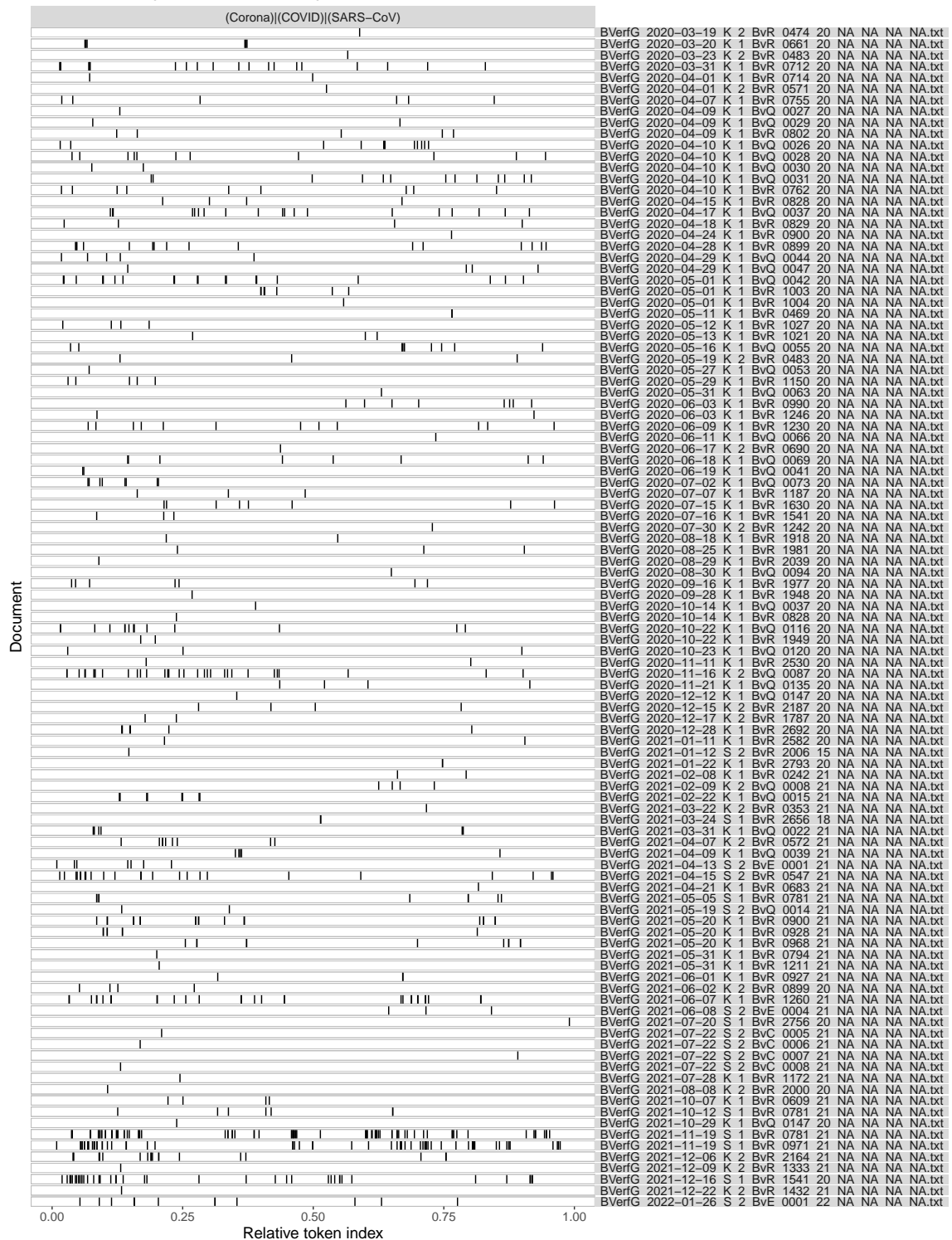
## 5 Lexical Dispersion Plot

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 Rechteckiges Format

```
textplot_xray(kwic,
              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)
  )
```

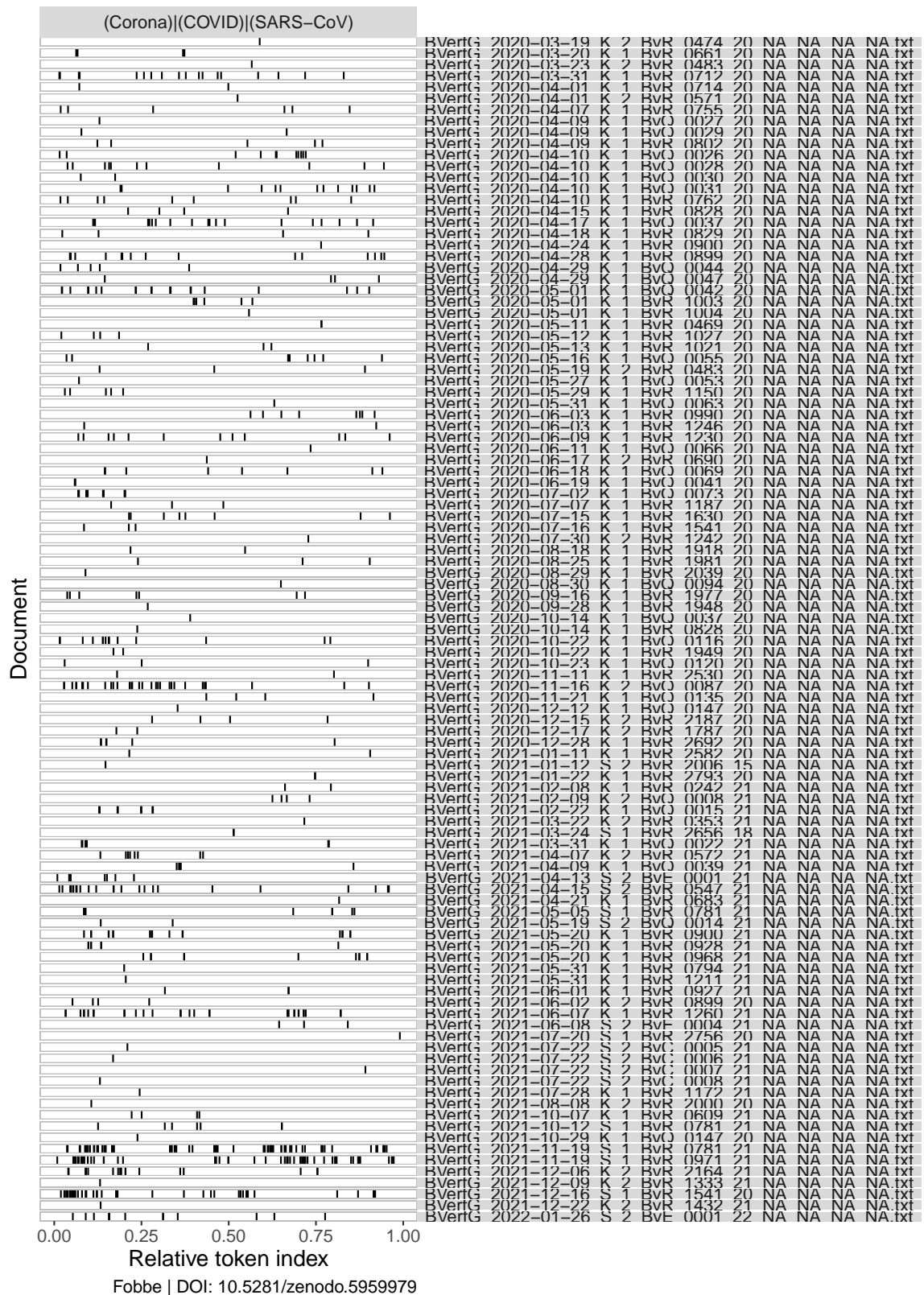
# BVerfG–Corona | Version 2022–02–01 | Lexical Dispersion Plot



## 5.2 A4-Format

```
textplot_xray(kwic,
               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 2022-02-01 | Lexical Dispersion Plot



## 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] 105
```

### 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("data", 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,  
      exdir = ".")
```

## 6.5 Corona-Entscheidungen verpacken

```
zip(paste(prefix.files,  
          "DE_TXT_Datensatz.zip",  
          sep = "_"),  
    keep.txt)
```

## 6.6 TXT-Dateien löschen

```
files.txt <- list.files(pattern = ".txt")  
  
unlink(files.txt)
```

## 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] 105
```

### 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("data", 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,  
      exdir = ".")
```



## 7.5 Corona-Entscheidungen verpacken

```
zip(paste(prefix.files,  
          "DE_PDF_Datensatz.zip",  
          sep = "_"),  
    keep.pdf)
```

## 7.6 PDF-Dateien löschen

```
files.pdf <- list.files(pattern = ".pdf")  
unlink(files.pdf)
```

## 8 Frequenztabellen erstellen

### 8.1 CE-BVerfG auf Corona-Entscheidungen reduzieren

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

### 8.2 Funktion anzeigen: f.fast.freqtable

```
print(f.fast.freqtable)
```

```
function(x, varlist = names(x), sumrow = TRUE, output.list = TRUE, output.kable = FALSE, output.csv = FALSE, outputdir = "./", prefix = „“, align = "r"){
```

```
## Begin List
freqtable.list <- vector("list", length(varlist))

## Calculate Frequency Table
for (i in seq_along(varlist)){

  varname <- varlist[i]

  freqtable <- x[, .N, keyby=c(paste0(varname))]

  freqtable[, c("exactpercent",
               "roundedpercent",
               "cumulpercent")] := {
    exactpercent <- N/sum(N)*100
    roundedpercent <- round(exactpercent, 2)
    cumulpercent <- round(cumsum(exactpercent), 2)
    list(exactpercent,
         roundedpercent,
         cumulpercent)}}

## Calculate Summary Row
if (sumrow == TRUE){
  colsums <- cbind("Total",
                  freqtable[, lapply(.SD, function(x){round(sum(x))}),
                  .SDcols = c("N",
                              "exactpercent",
                              "roundedpercent")
                  ], round(max(freqtable$cumulpercent)))

  colnames(colsums)[c(1,5)] <- c(varname, "cumulpercent")
  freqtable <- rbind(freqtable, colsums)
}

## Add Frequency Table to List
freqtable.list[[i]] <- freqtable

## Write CSV
```

```

if (output.csv == TRUE){
  fwrite(freqtable,
        file.path(outputdir,
                  paste0(prefix,
                        varname,
                        ".csv")),
        na = "NA")
}

## Output Kable
if (output.kable == TRUE){
  cat("\n-----\n")
  cat(paste0("Frequency Table for Variable:  ", varname, "\n"))
  cat("-----\n")
  cat(paste0("\n ",
            x[, .N, keyby=c(paste0(varname))] [, .N],
            " unique value(s) detected.\n\n"))

  print(kable(freqtable,
              format = "latex",
              align = align,
              booktabs = TRUE,
              longtable = TRUE) %>% kable_styling(latex_options = "repeat_
header"))
}
}

## Return List of Frequency Tables
if (output.list == TRUE){
  return(freqtable.list)
}
}
}

```

### 8.3 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] "zitativorschlag" "kurzbeschreibung"

```

### 8.4 Liste zu prüfender Variablen

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

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

print(varlist)
```

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

## 8.5 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	105	100	100	100
Total	105	100	100	100

---

Frequency Table for Variable: entscheidung\_typ

---

1 unique value(s) detected.

entscheidung_typ	N	exactpercent	roundedpercent	cumulpercent
B	105	100	100	100
Total	105	100	100	100

---

Frequency Table for Variable: spruchkoerper\_typ

---

2 unique value(s) detected.

spruchkoerper_typ	N	exactpercent	roundedpercent	cumulpercent
K	88	83.80952	83.81	83.81
S	17	16.19048	16.19	100.00
Total	105	100.00000	100.00	100.00

---

Frequency Table for Variable: spruchkoerper\_az

---

2 unique value(s) detected.

spruchkoerper_az	N	exactpercent	roundedpercent	cumulpercent
1	78	74.28571	74.29	74.29
2	27	25.71429	25.71	100.00
Total	105	100.00000	100.00	100.00

---

Frequency Table for Variable: registerzeichen

---

4 unique value(s) detected.

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
BvC	4	3.809524	3.81	3.81
BvE	3	2.857143	2.86	6.67
BvQ	30	28.571429	28.57	35.24
BvR	68	64.761905	64.76	100.00
Total	105	100.000000	100.00	100.00

---

Frequency Table for Variable: verfahrensart

---

4 unique value(s) detected.

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Einstweilige Anordnungen	30	28.571429	28.57	28.57
Organstreitverfahren	3	2.857143	2.86	31.43
Verfassungsbeschwerden; Kommunalverfassungsbe- schwerden	68	64.761905	64.76	96.19
Wahlprüfungsverfahren	4	3.809524	3.81	100.00
Total	105	100.000000	100.00	100.00

---

Frequency Table for Variable: eingangsjahr\_az

---

5 unique value(s) detected.

ingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
15	1	0.952381	0.95	0.95
18	1	0.952381	0.95	1.90
20	70	66.666667	66.67	68.57
21	32	30.476191	30.48	99.05
22	1	0.952381	0.95	100.00
Total	105	100.000000	100.00	100.00

---

Frequency Table for Variable: eingangsjahr\_iso

---

5 unique value(s) detected.

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
2015	1	0.952381	0.95	0.95
2018	1	0.952381	0.95	1.90
2020	70	66.666667	66.67	68.57
2021	32	30.476191	30.48	99.05
2022	1	0.952381	0.95	100.00
Total	105	100.000000	100.00	100.00

---

Frequency Table for Variable: entscheidungsjahr

---

3 unique value(s) detected.

entscheidungsjahr	N	exactpercent	roundedpercent	cumulpercent
2020	63	60.000000	60.00	60.00
2021	41	39.047619	39.05	99.05
2022	1	0.952381	0.95	100.00
Total	105	100.000000	100.00	100.00

---

Frequency Table for Variable: kollision

---

1 unique value(s) detected.

kollision	N	exactpercent	roundedpercent	cumulpercent
NA	105	100	100	100
Total	105	100	100	100

---

Frequency Table for Variable: band

---

1 unique value(s) detected.

band	N	exactpercent	roundedpercent	cumulpercent
NA	105	100	100	100
Total	105	100	100	100

---

Frequency Table for Variable: praesi

---

2 unique value(s) detected.

praesi	N	exactpercent	roundedpercent	cumulpercent
Harbarth	65	61.90476	61.9	61.9
Voßkuhle	40	38.09524	38.1	100.0
Total	105	100.00000	100.0	100.0

---

Frequency Table for Variable: v\_praesi

---

2 unique value(s) detected.

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Harbarth	40	38.09524	38.1	38.1
König	65	61.90476	61.9	100.0
Total	105	100.00000	100.0	100.0

---

Frequency Table for Variable: richter

---

15 unique value(s) detected.



richter	N	exactpercent	roundedpercent	cumulpercent
Baer Ott Radtke	12	11.428571	11.43	11.43
Harbarth Baer Ott	5	4.761905	4.76	16.19
Harbarth Britz Ott	2	1.904762	1.90	18.10
Harbarth Britz Radtke	25	23.809524	23.81	41.90
Harbarth Paulus Baer Britz Ott Christ Radtke Härtel	6	5.714286	5.71	47.62
Huber Kessal-Wulf König	5	4.761905	4.76	52.38
Huber Kessal-Wulf Wallrabenstein	6	5.714286	5.71	58.10
König Huber Hermanns Müller Kessal-Wulf Maidowski Langenfeld	9	0.952381	0.95	59.05
König Huber Hermanns Müller Kessal-Wulf Maidowski Langenfeld Wallrabenstein	9	8.571429	8.57	67.62
König Maidowski Wallrabenstein	1	0.952381	0.95	68.57
König Müller Maidowski	5	4.761905	4.76	73.33
Masing Paulus Christ	15	14.285714	14.29	87.62
Paulus Britz Ott Christ Radtke Härtel	1	0.952381	0.95	88.57
Paulus Christ Härtel	11	10.476190	10.48	99.05
Paulus Christ Radtke	1	0.952381	0.95	100.00
Total	105	100.000000	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	105	100	100	100
Total	105	100	100	100

Frequency Table for Variable: doi\_version

1 unique value(s) detected.

doi_version	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.5910152	105	100	100	100
Total	105	100	100	100

---

Frequency Table for Variable: lizenz

---

1 unique value(s) detected.

lizenz	N	exactpercent	roundedpercent	cumulpercent
Creative Commons Zero 1.0 Universal	105	100	100	100
Total	105	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(prefix.files,  
           "_DE_ANALYSE.zip"),  
    basename(dir.analysis))
```

### 10.2 Verpacken der Source-Dateien

```
files.source <- c(list.files(pattern = "\\R$|\\.toml$"),  
                  "CHANGELOG.md",  
                  "README.md",  
                  "R-fobbe-proto-package",  
                  "buttons",  
                  "tex",  
                  "gpg",  
                  list.files(pattern = "renv\\.lock|\\.Rprofile",  
                             all.files = TRUE),  
                  list.files("renv",  
                             pattern = "activate\\.R",  
                             full.names = TRUE))  
  
files.source <- grep("spin",  
                    files.source,  
                    value = TRUE,  
                    ignore.case = TRUE,  
                    invert = TRUE)  
  
zip(paste(prefix.files,  
          "Source_Files.zip",  
          sep = "_"),  
    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(pattern = "\\\\.zip$",  
                        ignore.case = TRUE)
```

### 11.2 Funktion anzeigen: future\_multihashes

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

```
## function(x){  
##  
##   ## Timestamp: Begin  
##   begin <- Sys.time()  
##  
##   ## Intro Message  
##   message(paste("Processing",  
##                 length(x),  
##                 "files. Begin at:",  
##                 begin))  
##  
##   ## Compute Hashes  
##   hashes.list <- future.apply::future_lapply(x,  
##                                             f.multihashes)  
##  
##   ## Coerce List to data.table  
##   hashes.table <- data.table::rbindlist(hashes.list)  
##  
##   ## Coerce data.table to data.frame  
##   data.table::setDF(hashes.table)  
##  
##   ## Timestamp: End  
##   end <- Sys.time()  
##  
##   ## Duration  
##   duration <- end - begin  
##  
##   ## Result Message
```

```
##      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: 2022-02-03 15:44:00
```

```
## Processed 4 files. Runtime was 0.2 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	BVerfG-Corona_2022-02-01_DE_ANALYSE.zip
2	BVerfG-Corona_2022-02-01_DE_PDF_Datensatz.zip
3	BVerfG-Corona_2022-02-01_DE_TXT_Datensatz.zip
4	BVerfG-Corona_2022-02-01_Source_Files.zip

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

---

index	sha2.256
<hr/>	
1	a31f669cbaff3645dc676fb90c6f037c951cdd9d1350dfa0124c4f9681031931
2	73a6255ebb304320e23ca126e0740dff12e9b28c6a97ffafc7b511c639caa8ef
3	ee870a54807213e9055ee565bf8eb13d1dcce66e0a33a730355a7597b31108df
4	315a72968c70641d4caeb884ae3f95fc97c38418f55e3a8b57e166d60f347323

---

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

---

index	sha3.512
<hr/>	
1	66217f06aadcebbe71a98dae8c3b1c5406ffe3598f94e874a422c28b2f7c0bbcb 5f12e47d98d7dc58dc09003a142f93439cb83523b5f3b185d1e2178978b370a2
2	b24022734d55dc8c908ea47ba4e9c4ee53e12aa73ad4a9308e66645832169699 b821db4a7b55cb9beea509826bb71691af0f884d43e93e5939fdd75c3c1189ae
3	92e8092ad2f55d8e21fd43a12cdda5c658e94ecd69aafd1675721e783b40feab c9d84ca9e0492237d924c7084dd547a6c4ffea3087fc7eed6521cae612360b8f
4	c5b8dc3328e04fbb3c0c0cc4dc9dfd4969b88ef8714ce6b7e6820399f3fc089f 02625dbc8f29e0fed22fbc78032aebac82beecea8cbc00e63f60285ba7857cda

---



## 12 Aufräumen

```
files.output <- list.files(pattern = "\\\\.zip")

output.destination <- file.path("output",
                                files.output)

print(files.output)
```

```
## [1] "BVerfG-Corona_2022-02-01_DE_ANALYSE.zip"
## [2] "BVerfG-Corona_2022-02-01_DE_PDF_Datensatz.zip"
## [3] "BVerfG-Corona_2022-02-01_DE_TXT_Datensatz.zip"
## [4] "BVerfG-Corona_2022-02-01_Source_Files.zip"
```

```
file.rename(files.output,
             output.destination)
```

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

## 13 Abschluss

### 13.1 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] "2022-02-01"
```

### 13.2 Datum und Uhrzeit (Anfang)

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

```
## [1] "2022-02-03 15:40:09 CET"
```

### 13.3 Datum und Uhrzeit (Ende)

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

```
## [1] "2022-02-03 15:44:00 CET"
```

### 13.4 Laufzeit des gesamten Skriptes

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

```
## Time difference of 3.861022 mins
```

### 13.5 Warnungen

```
warnings()
```

## 14 Parameter für strenge Replikationen

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

```
## [1] "OpenSSL 1.1.1l FIPS 24 Aug 2021"
```

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 34 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libflexiblas.so.3.0
##
## locale:
##  [1] LC_CTYPE=en_US.utf8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.utf8      LC_COLLATE=en_US.utf8
##  [5] LC_MONETARY=en_US.utf8  LC_MESSAGES=en_US.utf8
##  [7] LC_PAPER=en_US.utf8     LC_NAME=C
##  [9] LC_ADDRESS=C            LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.utf8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
##  [1] future.apply_1.8.1      future_1.23.0           quanteda.textplots_0.94
##  [4] quanteda_3.2.0          data.table_1.14.2       kableExtra_1.3.4
##  [7] knitr_1.37              ggplot2_3.3.5           magick_2.7.3
## [10] RcppTOML_0.1.7          rmarkdown_2.11
##
## loaded via a namespace (and not attached):
##  [1] xfun_0.29               listenv_0.8.0           lattice_0.20-45         colorspace_2.0-2
##  [5] vctr_0.3.8             htmltools_0.5.2         viridisLite_0.4.0      yaml_2.2.2
##  [9] utf8_1.2.2             rlang_1.0.0            pillar_1.7.0           glue_1.6.1
## [13] withr_2.4.3            lifecycle_1.0.1         stringr_1.4.0          munsell_0.5.0
## [17] gtable_0.3.0           rvest_1.0.2            codetools_0.2-18       evaluate_0.14
## [21] labeling_0.4.2         fastmap_1.1.0           parallel_4.0.5         fansi_1.0.2
## [25] highr_0.9              Rcpp_1.0.8             renv_0.15.0            scales_1.1.1
## [29] RcppParallel_5.1.5     webshot_0.5.2          farver_2.1.0           parallelly
## [33] systemfonts_1.0.3      fastmatch_1.1-3         stopwords_2.3           digest_0.6.29
## [37] stringi_1.7.6          grid_4.0.5             cli_3.1.1              tools_4.0.5
## [41] magrittr_2.0.2         tibble_3.1.6           crayon_1.4.2           pkgconfig_2.0.3
## [45] ellipsis_0.3.2         Matrix_1.4-0           xml2_1.3.3             svglite_2.0.0
## [49] httr_1.4.2            rstudioapi_0.13        globals_0.14.0         R6_2.5.1
## [53] compiler_4.0.5
```

## 15 Changelog

### 15.1 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

### 15.2 Version 2021-09-19

- Vollständige Aktualisierung der Daten

### 15.3 Version 2021-05-20

- Vollständige Aktualisierung der Daten

### 15.4 Version 2021-01-08

- Erstveröffentlichung

## Literaturverzeichnis

- Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2021. *Rmarkdown: Dynamic Documents for R*. <https://CRAN.R-project.org/package=rmarkdown>.
- Bengtsson, Henrik. 2021a. “A Unifying Framework for Parallel and Distributed Processing in R Using Futures.” <https://journal.r-project.org/archive/2021/RJ-2021-048/index.html>.
- . 2021b. “A Unifying Framework for Parallel and Distributed Processing in R Using Futures.” <https://journal.r-project.org/archive/2021/RJ-2021-048/index.html>.
- . 2021c. *Future.apply: Apply Function to Elements in Parallel Using Futures*. <https://CRAN.R-project.org/package=future.apply>.
- . 2021d. *Future: Unified Parallel and Distributed Processing in R for Everyone*. <https://CRAN.R-project.org/package=future>.
- 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. 2021. *Quanteda: Quantitative Analysis of Textual Data*. <https://quanteda.io>.
- Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2021. *Quanteda.textplots: Plots for the Quantitative Analysis of Textual Data*. <https://CRAN.R-project.org/package=quanteda.textplots>.
- Dowle, Matt, and Arun Srinivasan. 2021. *Data.table: Extension of ‘Data.frame’*. <https://CRAN.R-project.org/package=data.table>.
- Eddelbuettel, Dirk. 2020. *RcppTOML: Rcpp Bindings to Parser for Tom’s Obvious Markup Language*. <http://dirk.eddelbuettel.com/code/rcpp.toml.html>.
- Ooms, Jeroen. 2021. *Magick: Advanced Graphics and Image-Processing in R*. <https://CRAN.R-project.org/package=magick>.
- R Core Team. 2021. *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>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2021. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Xie, Yihui. 2014. “Knitr: A Comprehensive Tool for Reproducible Research in R.” In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich

- Leisch, and Roger D. Peng. Chapman; Hall/CRC. <http://www.crcpress.com/product/isbn/9781466561595>.
- . 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <https://yihui.org/knitr/>.
- . 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. <https://yihui.org/knitr/>.
- Xie, Yihui, J. J. Allaire, and Garrett Golemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.
- Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown-cookbook>.
- Zhu, Hao. 2021. *KableExtra: Construct Complex Table with Kable and Pipe Syntax*. <https://CRAN.R-project.org/package=kableExtra>.