

**Corpus der
Amtlichen Entscheidungssammlung
des
Bundesverfassungsgerichts
(C-BVerfGE-Source)**

COMPILATION REPORT

Version 2021-01-03

License MIT-0

DOI: 10.5281/zenodo.4265934

Titel	Source Code des »Corpus der amtlichen Entscheidungssammlung des Bundesverfassungsgerichts«
Abkürzung	C-BVerfGE-Source
Autor	Seán Fobbe
Version	2021-01-03
Download	https://doi.org/10.5281/zenodo.4265934
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2021). Source Code des »Corpus der amtlichen Entscheidungssammlung des Bundesverfassungsgerichts« (C-BVerfGE-Source). Version 2021-01-03. Zenodo. DOI: 10.5281/zenodo.4265934.

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 2021-01-03. Für das Gesamtkonzept der Software steht eine »Concept DOI« zur Verfügung, die auf der Zenodo-Seite jeder Version unter »Cite all versions?« zu finden ist. Die »Concept DOI« verlinkt immer die aktuellste Version.

Lizenz: MIT No Attribution (MIT-0)

Copyright — 2021— 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

1	Einleitung	7
1.1	Überblick	7
1.2	Endprodukte	7
1.3	Systemanforderungen	7
1.4	Kompilierung	8
1.4.1	Datensatz	8
1.4.2	Codebook	8
2	Parameter	9
2.1	Name des Datensatzes	9
2.2	DOI des Datensatz-Konzeptes	9
2.3	DOI der konkreten Version	9
2.4	Verzeichnis für Analyse-Ergebnisse	9
2.5	Optionen: Quanteda	9
2.6	Optionen: Knitr	9
2.6.1	Ausgabe-Format	9
2.6.2	DPI für Raster-Grafiken	9
2.6.3	Ausrichtung von Grafiken im Compilation Report	9
2.7	Frequenztabellen: Ignorierte Variablen	10
3	Vorbereitung	11
3.1	Datumsstempel	11
3.2	Datum und Uhrzeit (Beginn)	11
3.3	Ordner für Analyse-Ergebnisse erstellen	11
3.4	Packages Laden	11
3.5	Zusätzliche Funktionen einlesen	12
3.6	Quanteda-Optionen setzen	12
3.7	Knitr Optionen setzen	13
3.8	Vollzitate statistischer Software	13
3.9	Parallelisierung aktivieren	13
3.9.1	Logische Kerne	13
3.9.2	Quanteda	13
3.9.3	Data.table	13
3.9.4	DoParallel	14
4	Download	15
4.1	Zeitstempel: Linksammlung Beginn	15
4.2	Download vorbereiten	15
4.2.1	Funktion zeigen	15
4.2.2	Link zur amtlichen Sammlung definieren	15
4.2.3	Links zu HTML-Übersichten extrahieren	15
4.2.4	Links zu Entscheidungen extrahieren	16
4.3	Zeitstempel: Linksammlung Ende	17
4.4	Dauer: Linksammlung	18
4.5	Dateinamen erstellen	18
4.5.1	Extrahieren relevanter Metadaten	18
4.5.2	Formatierung von Registerzeichen anpassen	18
4.5.3	Formatierung von Spruchkoerper-Typ anpassen	19

4.5.4	Strenge REGEX-Validierung der Dateinamen	19
4.5.5	Zusätzliche Variablen einfügen	19
4.5.6	Nicht benannte Entscheidungen anzeigen	20
4.5.7	NAs einfügen für nicht benannte Entscheidungen	20
4.5.8	Strenge REGEX-Validierung der Dateinamen	20
4.6	Data Table für Download erstellen	20
4.7	Zeitstempel: Download Beginn	21
4.8	Download durchführen	21
4.9	Zeitstempel: Download Ende	21
4.10	Dauer: Download	22
4.11	Download: Ergebnis	22
4.11.1	Anzahl herunterzuladender Dateien	22
4.11.2	Anzahl heruntergeladener Dateien	22
4.11.3	Fehlbetrag	22
4.11.4	Fehlende Dateien	22
4.12	Wiederholungsversuch	23
4.13	Download: Gesamtergebnis	23
4.13.1	Anzahl herunterzuladender Dateien	23
4.13.2	Anzahl heruntergeladener Dateien	23
4.13.3	Fehlbetrag	24
4.13.4	Fehlende Dateien	24
4.14	Abschließende Hinweise	24
5	Text-Extraktion	25
5.1	Vektor der zu extrahierenden Dateien erstellen	25
5.2	Anzahl zu extrahierender Dateien	25
5.3	Seiten zählen: Funktion anzeigen	25
5.4	Anzahl zu extrahierender Seiten	25
5.5	PDF extrahieren: Funktion anzeigen	26
5.6	Text Extrahieren	26
6	Korpus Erstellen	27
6.1	TXT-Dateien Einlesen	27
6.2	In Data Table umwandeln	27
6.3	Variable »datum« als Datentyp »IDate« kennzeichnen	27
6.4	Datensatz nach Datum sortieren	27
6.5	Variable »entscheidungsjahr« hinzufügen	27
6.6	Variable »eingangsjahr_iso« hinzufügen	28
6.7	Variable »praesi« hinzufügen	28
6.7.1	Lebensdaten einlesen	28
6.7.2	Lebensdaten anzeigen	28
6.7.3	Hypothetisches Amtsende für PräsidentIn	28
6.7.4	Schleife vorbereiten	29
6.7.5	Vektor erstellen	29
6.7.6	Vektor einfügen	29
6.8	Variable »v_praesi« hinzufügen	29
6.8.1	Lebensdaten einlesen	29
6.8.2	Lebensdaten anzeigen	29
6.8.3	Hypothetisches Amtsende für Vize-PräsidentIn	30
6.8.4	Schleife vorbereiten	30

6.8.5	Vektor erstellen	30
6.8.6	Vektor einfügen	31
6.9	Variable »aktenzeichen« hinzufügen	31
6.10	Variable »ecli« hinzufügen	31
6.11	Variable »doi_concept« hinzufügen	32
6.12	Variable »doi_version« hinzufügen	32
6.13	Variable »version« hinzufügen	32
7	Frequenztabellen erstellen	33
7.1	Funktion anzeigen	33
7.2	Ignorierte Variablen	34
7.3	Liste zu prüfender Variablen	34
7.4	Frequenztabellen erstellen	35
8	Frequenztabellen visualisieren	46
8.1	Präfix erstellen	46
8.2	Tabellen einlesen	46
8.3	Spruchkörper Typ	47
8.4	Spruchkörper AZ	48
8.5	Registerzeichen	49
8.6	PräsidentIn	51
8.7	Vize-PräsidentIn	53
8.8	Entscheidungsjahr	55
8.9	Eingangsjahr (ISO)	56
9	Korpus-Analytik	57
9.1	Berechnung linguistischer Kennwerte	57
9.2	Variablen-Namen anpassen	57
9.3	Kennwerte dem Korpus hinzufügen	57
9.4	Anzahl Variablen im Korpus	57
9.5	Alle Variablen-Namen im Korpus	58
9.6	Linguistische Kennwerte	58
9.6.1	Zusammenfassungen berechnen	58
9.6.2	Zusammenfassungen anzeigen	59
9.6.3	Zusammenfassungen speichern	59
9.7	Quantitative Variablen	59
9.7.1	Entscheidungsdatum	59
9.7.2	Zusammenfassungen berechnen	59
9.7.3	Zusammenfassungen anzeigen	60
9.7.4	Zusammenfassungen speichern	61
9.8	Density	62
9.8.1	Density Tokens	62
9.8.2	Density Typen	63
9.8.3	Density Sätze	64
10	Beispiel-Werte für alle Metadaten anzeigen	65
11	CSV-Dateien erstellen	67
11.1	CSV mit vollem Datensatz speichern	67
11.2	CSV mit Metadaten speichern	67

12 Dateigrößen analysieren	68
12.1 Gesamtgröße	68
12.1.1 Korpus-Objekt in RAM (MB)	68
12.1.2 CSV Korpus (MB)	68
12.1.3 CSV Metadaten (MB)	68
12.1.4 PDF-Dateien (MB)	68
12.1.5 TXT-Dateien (MB)	69
12.2 Verteilung der Dateigrößen (PDF)	70
13 Erstellen der ZIP-Archive	72
13.1 Verpacken der CSV-Dateien	72
13.2 Verpacken der PDF-Dateien	72
13.3 Verpacken der TXT-Dateien	72
13.4 Verpacken der Analyse-Dateien	73
14 Kryptographische Hashes	74
14.1 Liste der ZIP-Archive erstellen	74
14.2 Funktion anzeigen	74
14.3 Hashes berechnen	75
14.4 In Data Table umwandeln	75
14.5 Index hinzufügen	75
14.6 In Datei schreiben	75
14.7 Leerzeichen hinzufügen um Zeilenumbruch zu ermöglichen	75
14.8 In Bericht anzeigen	76
15 Abschluss	78
15.1 Cluster stoppen	78
15.2 Datum und Uhrzeit (Ende)	78
15.3 Laufzeit des gesamten Skriptes	78
15.4 Warnungen	78
16 Parameter für strenge Replikationen	79
Literaturverzeichnis	80

1 Einleitung

1.1 Überblick

Dieses R-Skript lädt alle auf www.bundesverfassungsgericht.de veröffentlichten Entscheidungen der amtlichen Entscheidungssammlung des Bundesverfassungsgerichts (BVerfG) herunter und kompiliert sie in einen reichhaltigen menschen- und maschinenlesbaren Korpus. Es ist die Grundlage für den **Corpus der amtlichen Entscheidungssammlung des Bundesverfassungsgerichts (C-BVerfGE)**.

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.3831111>

1.2 Endprodukte

Primäre Endprodukte des Skripts sind fünf ZIP-Archive:

1. Der volle Datensatz im CSV-Format
2. Die reinen Metadaten im CSV-Format (wie unter 1, nur ohne Entscheidungstexte)
3. Der volle Datensatz im TXT-Format (reduzierter Umfang an Metadaten)
4. Der volle Datensatz im PDF-Format (reduzierter Umfang an Metadaten)
5. Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)

Zusätzliche werden für alle ZIP-Archive kryptographische Signaturen (SHA2-256 und SHA3-512) berechnet und in einer CSV-Datei hinterlegt. Die Analyse-Ergebnisse werden zum Ende hin nicht gelöscht, damit sie für die Codebook-Erstellung verwendet werden können. Weiterhin kann optional ein PDF-Bericht erstellt werden (siehe unter »Kompilierung«).

1.3 Systemanforderungen

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 dieses Berichts.

In der Standard-Einstellung wird das Skript vollautomatisch die maximale Anzahl an Rechenkernen/Threads auf dem System zu nutzen. Wenn die Anzahl Threads (Variable »fullCores«) auf 1 gesetzt wird, ist die Parallelisierung deaktiviert.

Auf der Festplatte sollten 2 GB Speicherplatz vorhanden sein.

Um die PDF-Berichte kompilieren zu können benötigen Sie das R package **rmarkdown**, eine vollständige Installation von \LaTeX und alle in der Präambel-TEX-Datei angegebenen \LaTeX Packages.

1.4 Kompilierung

Mit der Funktion `render()` von `rmarkdown` können der **vollständige Datensatz** und das **Codebook** kompiliert und die Skripte mitsamt ihrer Rechenergebnisse in ein gut lesbares PDF-Format überführt werden.

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.

1.4.1 Datensatz

Um den vollständigen Datensatz zu kompilieren und einen PDF-Bericht zu erstellen, kopieren Sie bitte alle im Source-Archiv bereitgestellten Dateien in einen leeren Ordner und führen mit R diesen Befehl aus:

```
rmarkdown::render(input = "C-BVerfGE_Source_CorpusCreation.R",
                  output_file = paste0("C-BVerfGE_",
                                      Sys.Date(),
                                      "_CompilationReport.pdf"),
                  envir = new.env())
```

1.4.2 Codebook

Um das **Codebook** zu kompilieren und einen PDF-Bericht zu erstellen, kopieren Sie bitte alle im Source-Archiv bereitgestellten Dateien in einen leeren Ordner und führen im Anschluss an die Kompilierung des Datensatzes (!) untenstehenden Befehl mit R aus.

Bei der Prüfung der GPG-Signatur wird ein Fehler auftreten und im Codebook dokumentiert, weil die Daten nicht mit meiner Original-Signatur versehen sind. Dieser Fehler hat jedoch keine Auswirkungen auf die Funktionalität und hindert die Kompilierung nicht.

```
rmarkdown::render(input = "C-BVerfGE_Source_CodebookCreation.R",
                  output_file = paste0("C-BVerfGE_",
                                      Sys.Date(),
                                      "_Codebook.pdf"),
                  envir = new.env())
```


2 Parameter

2.1 Name des Datensatzes

```
datasetname <- "C-BVerfGE"
```

2.2 DOI des Datensatz-Konzeptes

```
doi.concept <- "10.5281/zenodo.3831111"
```

2.3 DOI der konkreten Version

```
doi.version <- "10.5281/zenodo.4265943"
```

2.4 Verzeichnis für Analyse-Ergebnisse

Hinweis: Muss mit einem Schrägstrich enden!

```
outputdir <- paste0(getwd(), "/ANALYSE/")
```

2.5 Optionen: Quanteda

```
tokens_locale <- "de_DE"
```

2.6 Optionen: Knitr

2.6.1 Ausgabe-Format

```
dev <- c("pdf", "png")
```

2.6.2 DPI für Raster-Grafiken

```
dpi <- 150
```

2.6.3 Ausrichtung von Grafiken im Compilation Report

```
fig.align <- "center"
```

2.7 Frequenztabellen: Ignorierte Variablen

Diese Variablen werden bei der Erstellung der Frequenztabellen nicht berücksichtigt.

```
varremove <- c("text",  
               "eingangsnummer",  
               "datum",  
               "doc_id",  
               "seite",  
               "name",  
               "ecli",  
               "aktenzeichen")
```

3 Vorbereitung

3.1 Datumsstempel

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

```
datestamp <- Sys.Date()
print(datestamp)
```

```
## [1] "2021-01-03"
```

3.2 Datum und Uhrzeit (Beginn)

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

```
## [1] "2021-01-03 00:15:06 CET"
```

3.3 Ordner für Analyse-Ergebnisse erstellen

```
dir.create(outputdir)
```

3.4 Packages Laden

```
library(httr)          # HTTP-Werkzeuge
library(rvest)         # HTML/XML-Extraktion
```

```
## Loading required package: xml2
```

```
library(knitr)          # Professionelles Reporting
library(kableExtra)     # Verbesserte Kable Tabellen
library(pdftools)       # Verarbeitung von PDF-Dateien
```

```
## Using poppler version 0.84.0
```

```
library(doParallel)  # Parallelisierung
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
library(ggplot2)      # Fortgeschrittene Datenvisualisierung  
library(scales)       # Skalierung von Diagrammen  
library(data.table)   # Fortgeschrittene Datenverarbeitung  
library(readtext)     # TXT-Dateien einlesen  
library(quanteda)     # Fortgeschrittene Computerlinguistik
```

```
## Package version: 2.1.2
```

```
## Parallel computing: 2 of 16 threads used.
```

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

```
##  
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':  
##  
##      View
```

3.5 Zusätzliche Funktionen einlesen

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

```
source("General_Source_Functions.R")
```

3.6 Quanteda-Optionen setzen

```
quanteda_options(tokens_locale = tokens_locale)
```

3.7 Knitr Optionen setzen

```
knitr::opts_chunk$set(fig.path = outputdir,  
  dev = dev,  
  dpi = dpi,  
  fig.align = fig.align)
```

3.8 Vollzitate statistischer Software

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

```
## tweaking foreach
```

3.9 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.

Der Download der Dateien von <https://www.bundesverfassungsgericht.de> ist absichtlich nicht parallelisiert, damit das Skript nicht versehentlich als DoS-Tool verwendet wird.

Die hier verwendete Funktion **makeForkCluster()** ist viel schneller als die Alternativen, funktioniert aber nur auf Unix-basierten Systemen (Linux, MacOS).

3.9.1 Logische Kerne

```
fullCores <- detectCores()  
print(fullCores)
```

```
## [1] 16
```

3.9.2 Quanteda

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

3.9.3 Data.table

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

3.9.4 DoParallel

```
cl <- makeForkCluster(fullCores)
registerDoParallel(cl)
```

4 Download

4.1 Zeitstempel: Linksammlung Beginn

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

```
## [1] "2021-01-03 00:15:07 CET"
```

4.2 Download vorbereiten

4.2.1 Funktion zeigen

```
print(f.linkextract)
```

```
## function(URL){
##   tryCatch({
##     read_html(URL) %>%
##       html_nodes("a")%>%
##       html_attr('href'),
##     error=function(cond) {
##       return(NA)}
##   )
## }
```

4.2.2 Link zur amtlichen Sammlung definieren

```
URL <- "https://www.bundesverfassungsgericht.de/DE/Entscheidungen/Entscheidungen/
Amtliche%20Sammlung%20BVerfGE.html"
```

4.2.3 Links zu HTML-Übersichten extrahieren

```
links1 <- f.linkextract(URL)
links2 <- grep ("Entscheidungen/Liste",
               links1,
               ignore.case = TRUE,
               value = TRUE)

links2 <- paste0("https://www.bundesverfassungsgericht.de/",
               links2)
```

4.2.4 Links zu Entscheidungen extrahieren

Es gibt zwei verschiedene URL-Varianten mit denen Entscheidungen verlinkt sind. Diese werden als Variante A und B separat ausgewertet und danach zusammengefügt.

```
links3 <- lapply(links2,
                  f.linkextract)

links4 <- unlist(links3)
```

Variante A

```
links5a <- grep ("SharedDocs/Entscheidungen",
                 links4,
                 ignore.case = TRUE,
                 value = TRUE)

links5a <- paste0("https://www.bundesverfassungsgericht.de/",
                 links5a)

links6a <- gsub("Entscheidungen",
               "Downloads",
               links5a)

links.pdf.a <- gsub("\\\\.html.*",
                  "\\..pdf\\?__blob=publicationFile\\&v\\=1",
                  links6a)
```

Variante B

```
links5b <- grep ("https://www.bverfg.de/e/",
                 links4,
                 ignore.case = TRUE,
                 value = TRUE)

links6b <- gsub("https://www.bverfg.de/e",
               "",
               links5b)

links6b <- gsub("(/[a-z]{2})([0-9]{4})([0-9]{2})(.*)",
               "\\2/\\3\\1\\2\\3\\4",
               links6b)

links.pdf.b <- paste0("https://www.bundesverfassungsgericht.de/SharedDocs/
                     Downloads/DE/",
                     links6b,
                     ".pdf?__blob=publicationFile&v=1")
```

Links manuell hinzufügen

Diese Entscheidungen sind in der offiziellen Liste nicht verlinkt und müssen daher manuell der Liste hinzugefügt werden.

```
links.add <- c("https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/
2003/04/up20030430_1pbvu000102.pdf?__blob=publicationFile&v=1",
              "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/
2004/03/ks20040330_2bvk000101.pdf?__blob=publicationFile&v=1",
              "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/
2007/03/es20070329_2bve000207.pdf?__blob=publicationFile&v=1",
              "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/
2015/07/qk20150720_1bvq002515.pdf?__blob=publicationFile&v=2",
              "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/DE/
2015/12/rs20151216_2bvr195813.pdf?__blob=publicationFile&v=5")
```

Links manuell entfernen

Diese Entscheidungen sind in der offiziellen Liste irrtümlicherweise verlinkt obwohl nicht in der amtlichen Sammlung enthalten und müssen entfernt werden.

```
links.remove <- c("https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/
DE/2004/03/ks20040323_2bvk000101.pdf?__blob=publicationFile&v=1",
                  "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/
DE/2015/07/qk20150718_1bvq002515.pdf?__blob=publicationFile&v=1",
                  "https://www.bundesverfassungsgericht.de/SharedDocs/Downloads/
DE/2013/12/rk20131216_2bvr195813.pdf?__blob=publicationFile&v=1")
```

Varianten Zusammenfügen

Hinweis: In der Auflistung der Entscheidungen der amtlichen Sammlung sind abweichende Meinungen separat aufgeführt. Diese sind aber zusammen mit dem ursprünglichen Urteil in derselben PDF-Datei dokumentiert. Daher führen für manche Urteile mehrere Links zur selben PDF-Datei. Durch `setdiff()` werden nicht nur die oben definierten Urteile entfernt, sondern auch alle Duplikate.

```
links.pdf <- c(links.pdf.a,
               links.pdf.b,
               links.add)

links.pdf <- setdiff(links.pdf,
                    links.remove)
```

4.3 Zeitstempel: Linksammlung Ende

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

```
## [1] "2021-01-03 00:15:15 CET"
```

4.4 Dauer: Linksammlung

```
end.links-begin.links
```

```
## Time difference of 8.433434 secs
```

4.5 Dateinamen erstellen

4.5.1 Extrahieren relevanter Metadaten

Die Links zu jeder Entscheidung enthalten das Ordinalzahl-Element ihres jeweiligen ECLI-Codes. Struktur und Inhalt der ECLI für deutsche Gerichte sind auf dem Europäischen Justizportal näher erläutert.¹

```
filenames <- basename(links.pdf)

filenames <- gsub("[?].*",
                 "",
                 filenames)
```

Normale Struktur

```
filenames1 <- gsub("[a-z]([a-z])([0-9]{4})([0-9]{2})([0-9]{2})_([0-9])([a-z]*)
                  ([0-9]{4})([0-9]{2}).*",
                  "BVerfG_\\2-\\3-\\4_\\1_\\5_\\6_\\7_\\8_NA",
                  filenames)
```

Struktur von Entscheidungen mit Kollisions-Variable

```
filenames1 <- gsub("[a-z]([a-z])([0-9]{4})([0-9]{2})([0-9]{2})([a-z])_([0-9])([a-
z]*) ([0-9]{4})([0-9]{2}).*",
                  "BVerfG_\\2-\\3-\\4_\\1_\\6_\\7_\\8_\\9_\\5",
                  filenames1)
```

4.5.2 Formatierung von Registerzeichen anpassen

```
filenames1 <- gsub("_bv([a-z])_",
                  "_Bv\\U\\1_",
                  perl = TRUE,
                  filenames1)

filenames1 <- gsub("pbvu",
                  "PBvU",
                  filenames1)
```

¹ https://e-justice.europa.eu/content_european_case_law_identifier_ecli-175-de-de.do?member=1

4.5.3 Formatierung von Spruchkoerper-Typ anpassen

```
filenames1 <- gsub("_([kps])_",  
                  "_\\U\\1_",  
                  perl = TRUE,  
                  filenames1)
```

4.5.4 Strenge REGEX-Validierung der Dateinamen

Das Ergebnis sollte ein leerer Vektor sein!

```
grep("BVerfG_[0-9]{4}-[0-9]{2}-[0-9]{2}_[A-Z]_[0-9NA]*_[A-Za-z]*_[0-9]{4}_[0-9]{2}_[0-9a-zA]*$",  
     filenames1,  
     invert = TRUE,  
     value = TRUE)
```

```
## character(0)
```

4.5.5 Zusätzliche Variablen einfügen

```
extravariablen <- fread("C-BVerfGE_Source_Variablen_NameBandSeite.csv")  
  
extravariablen$newname <- paste(extravariablen$oldname,  
                                extravariablen$name,  
                                extravariablen$band,  
                                extravariablen$seite,  
                                sep = "_")  
  
extravariablen$newname <- paste0(extravariablen$newname,  
                                ".pdf")  
  
filenames2 <- filenames1  
  
targetindices <- match(extravariablen$oldname,  
                       filenames2)  
  
newname <- extravariablen$newname  
  
dt <- data.table(targetindices, newname)[complete.cases(targetindices)]  
  
if(dt[,.N] > 0){  
  filenames2 <- replace(filenames2,
```

```

        dt$targetindices,
        dt$newname)
}

```

4.5.6 Nicht benannte Entscheidungen anzeigen

Für alle Entscheidungen im C-BVerGE sollten per Hand ein Name vergeben werden sein. Ist dies nicht der Fall, werden noch zu benennende Entscheidungen hier angezeigt.

```

values <- grep(".pdf",
               filenames2,
               invert = TRUE,
               value = TRUE)

indices <- grep(".pdf",
               filenames2,
               invert = TRUE)

print(values)

```

```
## character(0)
```

4.5.7 NAs einfügen für nicht benannte Entscheidungen

```

filenames2[indices] <- paste0(values,
                              "_NA_NA_NA.pdf")

```

4.5.8 Strenge REGEX-Validierung der Dateinamen

Das Ergebnis sollte ein leerer Vektor sein!

```

grep("^BVerfG_[0-9]{4}-[0-9]{2}-[0-9]{2}_[SPKB]_[0-9NA]*_[A-Za-z]*_[0-9]{4}_[0-9]{2}_[0-9a-zA]*_[0-9ÄÜÖäüöA-Za-z\\-]*_[NA0-9]*_[NA0-9]*\\.pdf$",
     filenames2,
     value=TRUE,
     invert=TRUE)

```

```
## character(0)
```

4.6 Data Table für Download erstellen

```

dt <- data.table(links.pdf,
                 filenames2)

```

4.7 Zeitstempel: Download Beginn

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

```
## [1] "2021-01-03 00:15:16 CET"
```

4.8 Download durchführen

Hinweis: Es ist nötig jeden Link auf das Vorhandensein einer PDF-Datei zu prüfen, weil für manche Entscheidungen zwar HTML-Seiten vorhanden sind, aber keine korrespondierende PDF-Datei.

```
for (i in seq_len(dt[,.N])){
  response <- GET(dt$links.pdf[i])
  Sys.sleep(runif(1, 0.25, 0.75))
  if (response$headers$content-type == "application/pdf;charset=UTF-8" &
      response$status_code == 200){
    tryCatch({download.file(url = dt$links.pdf[i],
                           destfile = dt$filenames2[i])
    },
    error=function(cond) {
      return(NA)}
    )
  }else{
    print(paste0(dt$filenames2[i],
                 " : kein PDF vorhanden"))
  }
  Sys.sleep(runif(1, 0.5, 1.5))
}
```

```
## [1] "BVerfG_2001-10-01_S_2_BvB_0001_01_NA_NPD-Verbot-EA-RückgabeEDV-2_104_61.
pdf : kein PDF vorhanden"
```

4.9 Zeitstempel: Download Ende

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

```
## [1] "2021-01-03 00:40:15 CET"
```

4.10 Dauer: Download

```
end.download - begin.download
```

```
## Time difference of 24.98673 mins
```

4.11 Download: Ergebnis

4.11.1 Anzahl herunterzuladender Dateien

```
dt[, .N]
```

```
## [1] 711
```

4.11.2 Anzahl heruntergeladener Dateien

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

```
## [1] 710
```

4.11.3 Fehlbetrag

```
N.missing <- dt[, .N] - length(files.pdf)  
print(N.missing)
```

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

4.11.4 Fehlende Dateien

```
missing <- setdiff(dt$filenames2,  
                  files.pdf)  
print(missing)
```

```
## [1] "BVerfG_2001-10-01_S_2_BvB_0001_01_NA_NPD-Verbot-EA-RückgabeEDV-2_104_61.  
pdf"
```

4.12 Wiederholungsversuch

Download für fehlende Dokumente wiederholen.

```
if(N.missing > 0){  
  
  dt.retry <- dt[filenames2 %in% missing]  
  
  for (i in seq_len(dt.retry[,.N])){  
    response <- GET(dt.retry$links.pdf[i])  
    Sys.sleep(runif(1, 0.25, 0.75))  
    if (response$headers$"content-type" == "application/pdf;charset=UTF-8" &  
response$status_code == 200){  
      tryCatch({download.file(url = dt.retry$links.pdf[i],  
                             destfile = dt.retry$filenames2[i])  
      },  
      error=function(cond) {  
        return(NA)}  
      )  
    }else{  
      print(paste0(dt.retry$filenames2[i],  
                   " : kein PDF vorhanden"))  
    }  
    Sys.sleep(runif(1, 0.5, 1.5))  
  }  
}
```

```
## [1] "BVerfG_2001-10-01_S_2_BvB_0001_01_NA_NPD-Verbot-EA-RückgabeEDV-2_104_61.  
pdf : kein PDF vorhanden"
```

4.13 Download: Gesamtergebnis

4.13.1 Anzahl herunterzuladender Dateien

```
dt[,.N]
```

```
## [1] 711
```

4.13.2 Anzahl heruntergeladener Dateien

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

```
## [1] 710
```

4.13.3 Fehlbetrag

```
N.missing <- dt[,.N] - length(files.pdf)
print(N.missing)
```

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

4.13.4 Fehlende Dateien

```
missing <- setdiff(dt$filenames2, files.pdf)
print(missing)
```

```
## [1] "BVerfG_2001-10-01_S_2_BvB_0001_01_NA_NPD-Verbot-EA-RückgabeEDV-2_104_61.
pdf"
```

4.14 Abschließende Hinweise

Hinweis: Für die Entscheidung vom 1.10.2001 zur Rückgabe von EDV-Anlagen im Rahmen des NPD-Verfahrens war auch nach manueller Suche keine PDF-Datei auffindbar.

5 Text-Extraktion

5.1 Vektor der zu extrahierenden Dateien erstellen

```
files.pdf <- list.files(pattern = "\\\\.pdf$",  
                        ignore.case = TRUE)
```

5.2 Anzahl zu extrahierender Dateien

```
length(files.pdf)
```

```
## [1] 710
```

5.3 Seiten zählen: Funktion anzeigen

```
print(f.dopar.pagenums)
```

```
function(x){
```

```
  pagenums <- foreach(filename = x,  
                      .combine = 'c',  
                      .errorhandling = 'remove',  
                      .inorder = FALSE) %dopar% {  
    pdf_length(filename)  
  }  
  return(pagenums)
```

```
}
```

5.4 Anzahl zu extrahierender Seiten

```
sum(f.dopar.pagenums(files.pdf))
```

```
## [1] 18278
```

5.5 PDF extrahieren: Funktion anzeigen

```
print(f.dopar.pdfextract)
```

```
function(x){
```

```
  newnames <- gsub("\\\\.pdf",
                  "\\..txt",
                  x)

  out <- foreach(i = seq_along(x),
                .errorhandling = 'pass') %dopar% {

    ## Extract text layer from PDF
    pdf.extracted <- pdf_text(x[i])

    ## Write TXT to Disk
    write.table(pdf.extracted,
                newnames[i],
                quote = FALSE,
                row.names = FALSE,
                col.names = FALSE)

  }
  return(length(out))
}
```

5.6 Text Extrahieren

```
f.dopar.pdfextract(files.pdf)
```

```
## [1] 710
```

6 Korpus Erstellen

6.1 TXT-Dateien Einlesen

```
txt.bverfg <- readtext("./*.txt",
  docvarsfrom = "filenames",
  docvarnames = c("gericht",
    "datum",
    "spruchkoerper_typ",
    "spruchkoerper_az",
    "registerzeichen",
    "eingangsnummer",
    "eingangsjahr_az",
    "kollision",
    "name",
    "band",
    "seite"),
  dvsep = "_",
  encoding = "UTF-8")
```

6.2 In Data Table umwandeln

```
setDT(txt.bverfg)
```

6.3 Variable »datum« als Datentyp »IDate« kennzeichnen

```
txt.bverfg$datum <- as.IDate(txt.bverfg$datum)
```

6.4 Datensatz nach Datum sortieren

Aufgrund der Position der Datums-Variable ist der Datensatz vermutlich schon von Linux nach Datum sortiert worden. Die Erstellung der Variablen für Präsidenten und Vize-Präsidenten trifft allerdings die starke Annahme, dass eine aufsteigende Sortierung nach Datum besteht. Wäre das nicht der Fall, würden dort Fehler auftreten. Diese Sortierung ist als fail-safe gedacht.

```
setorder(txt.bverfg, datum)
```

6.5 Variable »entscheidungsjahr« hinzufügen

```
txt.bverfg$entscheidungsjahr <- year(txt.bverfg$datum)
```

6.6 Variable »eingangsjahr_iso« hinzufügen

```
txt.bverfg$eingangsjahr_iso <- f.year.iso(txt.bverfg$eingangsjahr_az)
```

6.7 Variable »praesi« hinzufügen

Diese Variable dokumentiert für jede Entscheidung welche/r PräsidentIn am Tag der Entscheidung im Amt war.

6.7.1 Lebensdaten einlesen

```
praesi <- fread("C-BVerfGE_Source_Variablen_Praesident.csv")
```

6.7.2 Lebensdaten anzeigen

```
kable(praesi,
      format = "latex",
      align = "r",
      booktabs=TRUE,
      longtable=TRUE)
```

Vorname	Nachname	AmtszeitBeginn	AmtszeitEnde	Geboren	Gestorben
Hermann	Höpker-Aschoff	1951-09-07	1954-01-15	1883-01-31	1954-01-15
Josef	Wintrich	1954-03-23	1958-10-19	1891-02-15	1958-10-19
Gebhard	Müller	1959-01-08	1971-12-08	1900-04-17	1990-08-07
Ernst	Benda	1971-12-08	1983-12-20	1925-01-15	2009-03-02
Wolfgang	Zeidler	1983-12-20	1987-11-16	1924-09-02	1987-12-31
Roman	Herzog	1987-11-16	1994-09-13	1934-04-05	2017-01-10
Jutta	Limbach	1994-09-14	2002-04-10	1934-03-27	2016-09-10
Hans-Jürgen	Papier	2002-04-10	2010-03-16	1943-07-06	NA
Andreas	Voßkuhle	2010-03-16	2020-06-22	1963-12-21	NA
Stephan	Harbarth	2020-06-22	NA	1971-12-19	NA

6.7.3 Hypothetisches Amtsende für PräsidentIn

Weil der/die aktuelle PräsidentIn noch im Amt ist, ist der Wert für das Amtsende »NA«. Dieser ist aber für die verwendete Logik nicht greifbar, weshalb an dieser Stelle ein hypothetisches Amtsende in einem Jahr ab dem Tag der Datensatzerstellung fingiert wird.

Es wird nur an dieser Stelle verwendet und danach verworfen.

```
praesi[is.na(AmtszeitEnde)]$AmtszeitEnde <- Sys.Date() + 365
```

6.7.4 Schleife vorbereiten

```
N <- praesi[,.N]

praesi.list <- vector("list", N)
```

6.7.5 Vektor erstellen

```
for (i in seq_len(N)){
  praesi.N <- txt.bverfg[datum >= praesi$AmtszeitBeginn[i] & datum < praesi$
    AmtszeitEnde[i], .N]
  praesi.list[[i]] <- rep(praesie$Nachname[i],
    praesi.N)
}
```

6.7.6 Vektor einfügen

```
txt.bverfg$praesi <- unlist(praesie.list)
```

6.8 Variable »v_praesi« hinzufügen

Diese Variable dokumentiert für jede Entscheidung welche/r Vize-PräsidentIn am Tag der Entscheidung im Amt war.

6.8.1 Lebensdaten einlesen

```
vpraesi <- fread("C-BVerfGE_Source_Variablen_VizePraesident.csv")
```

6.8.2 Lebensdaten anzeigen

```
kable(vpraesi,
  format = "latex",
  align = "r",
  booktabs=TRUE,
  longtable=TRUE)
```

Vorname	Nachname	AmtszeitBeginn	AmtszeitEnde	Geboren	Gestorben
Rudolf	Katz	1951-09-07	1961-07-23	1895-11-23	1961-07-23
Friedrich Wilhelm	Wagner	1961-12-19	1967-10-18	1894-02-28	1971-03-27
Walter	Seuffert	1967-10-18	1975-11-07	1907-02-04	1989-12-28
Wolfgang	Zeidler	1975-11-07	1983-12-20	1924-09-02	1987-12-31
Roman	Herzog	1983-12-20	1987-11-16	1934-04-05	2017-01-10
Ernst Gottfried	Mahrenholz	1987-11-16	1994-03-24	1929-06-18	NA
Jutta	Limbach	1994-03-24	1994-09-14	1934-03-27	2016-09-10
Johann Friedrich	Henschel	1994-09-29	1995-10-13	1931-06-10	2007-03-19
Otto	Seidl	1995-10-13	1998-02-27	1931-12-11	NA
Hans-Jürgen	Papier	1998-02-27	2002-04-10	1943-07-06	NA
Winfried	Hassemer	2002-04-10	2008-05-07	1940-02-17	2014-01-09
Andreas	Voßkuhle	2008-05-07	2010-03-16	1963-12-21	NA
Ferdinand	Kirchhof	2010-03-16	2018-11-30	1950-06-21	NA
Stephan	Harbarth	2018-11-30	2020-06-22	1971-12-19	NA
Doris	König	2020-06-22	NA	1957-06-25	NA

6.8.3 Hypothetisches Amtsende für Vize-PräsidentIn

Weil der/die aktuelle Vize-PräsidentIn noch im Amt ist, ist der Wert für das Amtsende »NA«. Dieser ist aber für die verwendete Logik nicht greifbar, weshalb an dieser Stelle ein hypothetisches Amtsende in einem Jahr ab dem Tag der Datensatzerstellung fingiert wird. Es wird nur an dieser Stelle verwendet und danach verworfen.

```
vpraesi[is.na(AmtszeitEnde)]$AmtszeitEnde <- Sys.Date() + 365
```

6.8.4 Schleife vorbereiten

```
N <- vpraesi[, .N]
vpraesi.list <- vector("list", N)
```

6.8.5 Vektor erstellen

```
for (i in seq_len(N)){
```

```

vpraesi.N <- txt.bverfg[datum >= vpraesi$AmtszeitBeginn[i] & datum < vpraesi$
AmtszeitEnde[i], .N]
vpraesi.list[[i]] <- rep(vpraesi$Nachname[i],
                        vpraesi.N)
}

```

6.8.6 Vektor einfügen

```

txt.bverfg$v_praesi <- unlist(vpraesi.list)

```

6.9 Variable »aktenzeichen« hinzufügen

```

txt.bverfg$aktenzeichen <- paste0(txt.bverfg$spruchkoerper_az,
                                " ",
                                txt.bverfg$registerzeichen,
                                " ",
                                txt.bverfg$eingangsnummer,
                                "/",
                                txt.bverfg$eingangsjahr_az)

```

Bei Entscheidungen der Verzögerungskammer fehlt das Spruchkörper-Element des Aktenzeichens. Diese Zeile entfernt die »NA«-Angabe um ein korrektes Aktenzeichen herzustellen.

```

txt.bverfg$aktenzeichen <- gsub("NA ", "", txt.bverfg$aktenzeichen)

```

6.10 Variable »ecli« hinzufügen

Struktur und Inhalt der ECLI für deutsche Gerichte sind auf dem Europäischen Justizportal näher erläutert.²

Sofern die Variablen korrekt extrahiert wurden lässt sich die ECLI vollständig rekonstruieren.

```

ecli.ordinalzahl <- paste0(gsub("Bv([A-Z])",
                                "\\1",
                                txt.bverfg$registerzeichen),
                          txt.bverfg$spruchkoerper_typ,
                          txt.bverfg$datum,
                          txt.bverfg$kollision,
                          ". ",
                          txt.bverfg$spruchkoerper_az,
                          txt.bverfg$registerzeichen,
                          formatC(txt.bverfg$eingangsnummer,
                                width = 4,
                                flag = "0"),

```

² https://e-justice.europa.eu/content_european_case_law_identifier_ecli-175-de-de.do?member=1

```

        formatC(txt.bverfg$eingangsjahr_az,
                width = 2,
                flag = "0"))

ecli.ordinalzahl <- gsub("NA",
                        "",
                        ecli.ordinalzahl)

ecli.ordinalzahl <- gsub("-",
                        "",
                        ecli.ordinalzahl)

ecli.ordinalzahl <- gsub("vzb",
                        "vb",
                        ecli.ordinalzahl)

ecli.ordinalzahl <- gsub("pup",
                        "up",
                        ecli.ordinalzahl)

ecli.ordinalzahl <- tolower(ecli.ordinalzahl)

txt.bverfg$ecli <- paste0("ECLI:DE:BVerfG:",
                        txt.bverfg$entscheidungsjahr,
                        ":",
                        ecli.ordinalzahl)

```

6.11 Variable »doi_concept« hinzufügen

```

txt.bverfg$doi_concept <- rep(doi.concept,
                             txt.bverfg[,.N])

```

6.12 Variable »doi_version« hinzufügen

```

txt.bverfg$doi_version <- rep(doi.version,
                             txt.bverfg[,.N])

```

6.13 Variable »version« hinzufügen

```

txt.bverfg$version <- as.character(rep(datestamp,
                                       txt.bverfg[,.N]))

```


7 Frequenztabellen erstellen

7.1 Funktion anzeigen

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

```
function(x, varlist = names(x), sumrow = TRUE, output.list = TRUE, output.kable = FALSE, output.csv = FALSE, outputdir = »./«, prefix = ){
```

```
## 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,
           paste0(outputdir,
                  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 = "r",
             booktabs=TRUE,
             longtable=TRUE) %>% kable_styling(latex_options = "repeat_
header"))
}
}

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

}

```

7.2 Ignorierte Variablen

```
print(varremove)
```

```
## [1] "text"          "eingangsnummer" "datum"          "doc_id"
## [5] "seite"         "name"           "ecli"           "aktenzeichen"
```

7.3 Liste zu prüfender Variablen

```
varlist <- names(txt.bverfg)
varlist <- grep(paste(varremove,
                      collapse="|"),
               varlist,
               invert = TRUE,
               value = TRUE)
print(varlist)
```

```
## [1] "gericht"      "spruchkoerper_typ" "spruchkoerper_az"
## [4] "registerzeichen" "eingangsjahr_az"   "kollision"
## [7] "band"         "entscheidungsjahr" "eingangsjahr_iso"
## [10] "praesi"       "v_praesi"         "doi_concept"
## [13] "doi_version"   "version"
```

7.4 Frequenztabelle erstellen

```
prefix <- paste0(datasetname,
  "_01_Frequenztabelle_var-")
```

```
f.fast.freqtable(txt.bverfg,
  varlist = varlist,
  sumrow = TRUE,
  output.list = FALSE,
  output.kable = TRUE,
  output.csv = TRUE,
  outputdir = outputdir,
  prefix = prefix)
```

Frequency Table for Variable: gericht

1 unique value(s) detected.

gericht	N	exactpercent	roundedpercent	cumulpercent
BVerfG	710	100	100	100
Total	710	100	100	100

Frequency Table for Variable: spruchkoerper_typ

3 unique value(s) detected.

spruchkoerper_typ	N	exactpercent	roundedpercent	cumulpercent
K	3	0.4225352	0.42	0.42
P	2	0.2816901	0.28	0.70
S	705	99.2957746	99.30	100.00
Total	710	100.0000000	100.00	100.00

(continued)

spruchkoerper_typ	N	exactpercent	roundedpercent	cumulpercent
-------------------	---	--------------	----------------	--------------

Frequency Table for Variable: spruchkoerper_az

2 unique value(s) detected.

spruchkoerper_az	N	exactpercent	roundedpercent	cumulpercent
1	325	45.77465	45.77	45.77
2	385	54.22535	54.23	100.00
Total	710	100.00000	100.00	100.00

Frequency Table for Variable: registerzeichen

14 unique value(s) detected.

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
BvB	15	2.1126761	2.11	2.11
BvC	19	2.6760563	2.68	4.79
BvE	69	9.7183099	9.72	14.51
BvF	48	6.7605634	6.76	21.27
BvG	8	1.1267606	1.13	22.39
BvH	4	0.5633803	0.56	22.96
BvK	9	1.2676056	1.27	24.23
BvL	144	20.2816901	20.28	44.51
BvM	4	0.5633803	0.56	45.07
BvN	1	0.1408451	0.14	45.21
BvP	1	0.1408451	0.14	45.35
BvQ	22	3.0985915	3.10	48.45
BvR	364	51.2676056	51.27	99.72
PBvU	2	0.2816901	0.28	100.00

(continued)

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_az

37 unique value(s) detected.

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
0	27	3.8028169	3.80	3.80
1	36	5.0704225	5.07	8.87
2	33	4.6478873	4.65	13.52
3	31	4.3661972	4.37	17.89
4	28	3.9436620	3.94	21.83
5	29	4.0845070	4.08	25.92
6	19	2.6760563	2.68	28.59
7	39	5.4929577	5.49	34.08
8	23	3.2394366	3.24	37.32
9	29	4.0845070	4.08	41.41
10	27	3.8028169	3.80	45.21
11	37	5.2112676	5.21	50.42
12	36	5.0704225	5.07	55.49
13	35	4.9295775	4.93	60.42
14	22	3.0985915	3.10	63.52
15	25	3.5211268	3.52	67.04
16	14	1.9718310	1.97	69.01
17	8	1.1267606	1.13	70.14
18	8	1.1267606	1.13	71.27
19	3	0.4225352	0.42	71.69
51	1	0.1408451	0.14	71.83
52	2	0.2816901	0.28	72.11

(continued)

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
83	2	0.2816901	0.28	72.39
86	1	0.1408451	0.14	72.54
87	1	0.1408451	0.14	72.68
88	1	0.1408451	0.14	72.82
89	3	0.4225352	0.42	73.24
90	7	0.9859155	0.99	74.23
91	13	1.8309859	1.83	76.06
92	6	0.8450704	0.85	76.90
93	17	2.3943662	2.39	79.30
94	20	2.8169014	2.82	82.11
95	23	3.2394366	3.24	85.35
96	28	3.9436620	3.94	89.30
97	21	2.9577465	2.96	92.25
98	31	4.3661972	4.37	96.62
99	24	3.3802817	3.38	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: kollision

4 unique value(s) detected.

kollision	N	exactpercent	roundedpercent	cumulpercent
NA	704	99.1549296	99.15	99.15
a	3	0.4225352	0.42	99.58
b	2	0.2816901	0.28	99.86
c	1	0.1408451	0.14	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: band

64 unique value(s) detected.

band	N	exactpercent	roundedpercent	cumulpercent
1	2	0.2816901	0.28	0.28
7	1	0.1408451	0.14	0.42
62	1	0.1408451	0.14	0.56
65	1	0.1408451	0.14	0.70
90	1	0.1408451	0.14	0.85
91	1	0.1408451	0.14	0.99
93	1	0.1408451	0.14	1.13
96	5	0.7042254	0.70	1.83
97	18	2.5352113	2.54	4.37
98	14	1.9718310	1.97	6.34
99	26	3.6619718	3.66	10.00
100	14	1.9718310	1.97	11.97
101	15	2.1126761	2.11	14.08
102	17	2.3943662	2.39	16.48
103	19	2.6760563	2.68	19.15
104	26	3.6619718	3.66	22.82
105	14	1.9718310	1.97	24.79
106	17	2.3943662	2.39	27.18
107	14	1.9718310	1.97	29.15
108	18	2.5352113	2.54	31.69
109	12	1.6901408	1.69	33.38
110	14	1.9718310	1.97	35.35
111	15	2.1126761	2.11	37.46
112	18	2.5352113	2.54	40.00
113	10	1.4084507	1.41	41.41
114	13	1.8309859	1.83	43.24

(continued)

band	N	exactpercent	roundedpercent	cumulpercent
115	12	1.6901408	1.69	44.93
116	11	1.5492958	1.55	46.48
117	14	1.9718310	1.97	48.45
118	9	1.2676056	1.27	49.72
119	10	1.4084507	1.41	51.13
120	10	1.4084507	1.41	52.54
121	10	1.4084507	1.41	53.94
122	13	1.8309859	1.83	55.77
123	9	1.2676056	1.27	57.04
124	12	1.6901408	1.69	58.73
125	7	0.9859155	0.99	59.72
126	13	1.8309859	1.83	61.55
127	10	1.4084507	1.41	62.96
128	13	1.8309859	1.83	64.79
129	12	1.6901408	1.69	66.48
130	11	1.5492958	1.55	68.03
131	11	1.5492958	1.55	69.58
132	13	1.8309859	1.83	71.41
133	14	1.9718310	1.97	73.38
134	14	1.9718310	1.97	75.35
135	11	1.5492958	1.55	76.90
136	11	1.5492958	1.55	78.45
137	9	1.2676056	1.27	79.72
138	12	1.6901408	1.69	81.41
139	10	1.4084507	1.41	82.82
140	12	1.6901408	1.69	84.51
141	8	1.1267606	1.13	85.63
142	13	1.8309859	1.83	87.46
143	8	1.1267606	1.13	88.59

(continued)

band	N	exactpercent	roundedpercent	cumulpercent
144	4	0.5633803	0.56	89.15
145	9	1.2676056	1.27	90.42
146	7	0.9859155	0.99	91.41
147	9	1.2676056	1.27	92.68
148	10	1.4084507	1.41	94.08
149	13	1.8309859	1.83	95.92
150	8	1.1267606	1.13	97.04
151	9	1.2676056	1.27	98.31
152	12	1.6901408	1.69	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: entscheidungsjahr

29 unique value(s) detected.

entscheidungsjahr	N	exactpercent	roundedpercent	cumulpercent
1952	2	0.2816901	0.28	0.28
1958	1	0.1408451	0.14	0.42
1983	2	0.2816901	0.28	0.70
1994	2	0.2816901	0.28	0.99
1996	1	0.1408451	0.14	1.13
1997	9	1.2676056	1.27	2.39
1998	51	7.1830986	7.18	9.58
1999	31	4.3661972	4.37	13.94
2000	20	2.8169014	2.82	16.76
2001	37	5.2112676	5.21	21.97
2002	37	5.2112676	5.21	27.18
2003	38	5.3521127	5.35	32.54
2004	41	5.7746479	5.77	38.31

(continued)

entscheidungsjahr	N	exactpercent	roundedpercent	cumulpercent
2005	38	5.3521127	5.35	43.66
2006	25	3.5211268	3.52	47.18
2007	27	3.8028169	3.80	50.99
2008	29	4.0845070	4.08	55.07
2009	29	4.0845070	4.08	59.15
2010	29	4.0845070	4.08	63.24
2011	24	3.3802817	3.38	66.62
2012	35	4.9295775	4.93	71.55
2013	28	3.9436620	3.94	75.49
2014	37	5.2112676	5.21	80.70
2015	29	4.0845070	4.08	84.79
2016	29	4.0845070	4.08	88.87
2017	27	3.8028169	3.80	92.68
2018	30	4.2253521	4.23	96.90
2019	21	2.9577465	2.96	99.86
2020	1	0.1408451	0.14	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_iso

37 unique value(s) detected.

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
1951	1	0.1408451	0.14	0.14
1952	2	0.2816901	0.28	0.42
1983	2	0.2816901	0.28	0.70
1986	1	0.1408451	0.14	0.85
1987	1	0.1408451	0.14	0.99
1988	1	0.1408451	0.14	1.13

(continued)

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
1989	3	0.4225352	0.42	1.55
1990	7	0.9859155	0.99	2.54
1991	13	1.8309859	1.83	4.37
1992	6	0.8450704	0.85	5.21
1993	17	2.3943662	2.39	7.61
1994	20	2.8169014	2.82	10.42
1995	23	3.2394366	3.24	13.66
1996	28	3.9436620	3.94	17.61
1997	21	2.9577465	2.96	20.56
1998	31	4.3661972	4.37	24.93
1999	24	3.3802817	3.38	28.31
2000	27	3.8028169	3.80	32.11
2001	36	5.0704225	5.07	37.18
2002	33	4.6478873	4.65	41.83
2003	31	4.3661972	4.37	46.20
2004	28	3.9436620	3.94	50.14
2005	29	4.0845070	4.08	54.23
2006	19	2.6760563	2.68	56.90
2007	39	5.4929577	5.49	62.39
2008	23	3.2394366	3.24	65.63
2009	29	4.0845070	4.08	69.72
2010	27	3.8028169	3.80	73.52
2011	37	5.2112676	5.21	78.73
2012	36	5.0704225	5.07	83.80
2013	35	4.9295775	4.93	88.73
2014	22	3.0985915	3.10	91.83
2015	25	3.5211268	3.52	95.35
2016	14	1.9718310	1.97	97.32
2017	8	1.1267606	1.13	98.45

(continued)

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
2018	8	1.1267606	1.13	99.58
2019	3	0.4225352	0.42	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: praesi

7 unique value(s) detected.

praesi	N	exactpercent	roundedpercent	cumulpercent
Benda	2	0.2816901	0.28	0.28
Herzog	1	0.1408451	0.14	0.42
Höpker-Aschoff	2	0.2816901	0.28	0.70
Limbach	164	23.0985915	23.10	23.80
Papier	253	35.6338028	35.63	59.44
Voßkuhle	287	40.4225352	40.42	99.86
Wintrich	1	0.1408451	0.14	100.00
Total	710	100.0000000	100.00	100.00

Frequency Table for Variable: v_praesi

10 unique value(s) detected.

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Harbarth	26	3.6619718	3.66	3.66
Hassemer	207	29.1549296	29.15	32.82
Henschel	1	0.1408451	0.14	32.96
Katz	3	0.4225352	0.42	33.38
Kirchhof	261	36.7605634	36.76	70.14
Mahrenholz	1	0.1408451	0.14	70.28

(continued)

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Papier	144	20.2816901	20.28	90.56
Seidl	19	2.6760563	2.68	93.24
Voßkuhle	46	6.4788732	6.48	99.72
Zeidler	2	0.2816901	0.28	100.00
Total	710	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.3831111	710	100	100	100
Total	710	100	100	100

Frequency Table for Variable: doi_version

1 unique value(s) detected.

doi_version	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.4265943	710	100	100	100
Total	710	100	100	100

Frequency Table for Variable: version

1 unique value(s) detected.

version	N	exactpercent	roundedpercent	cumulpercent
2021-01-03	710	100	100	100
Total	710	100	100	100

8 Frequenztabellen visualisieren

8.1 Präfix erstellen

```
prefix <- paste0("ANALYSE/",  
                 datasetname,  
                 "_01_Frequenztafel_var-")
```

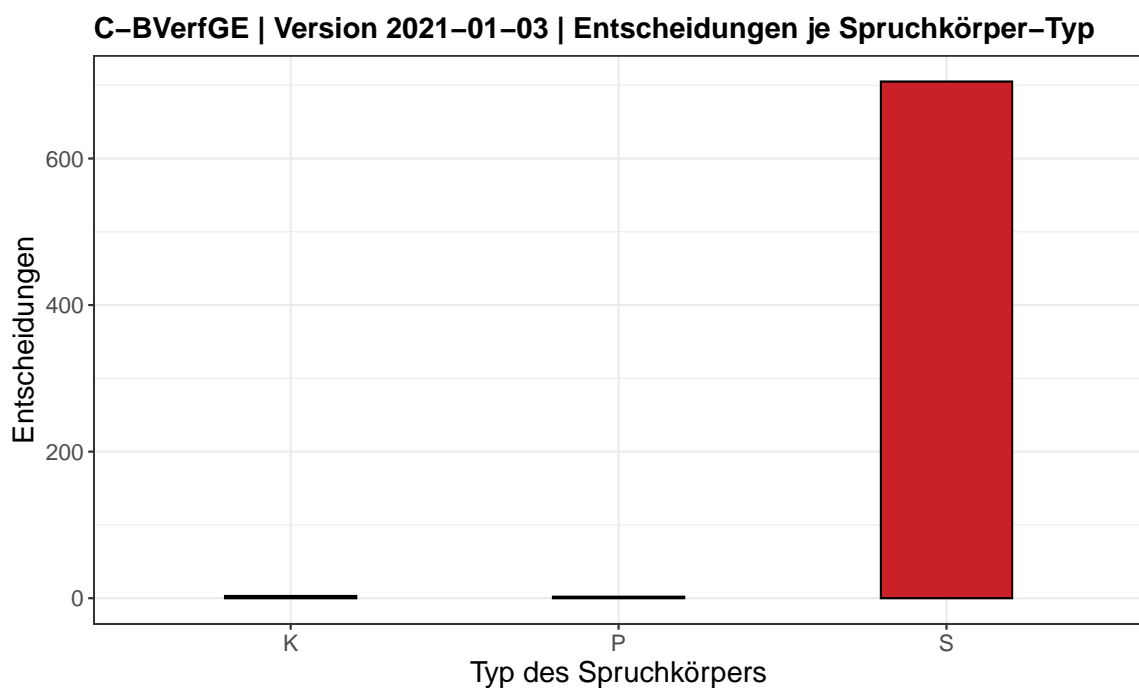
8.2 Tabellen einlesen

```
table.spruch.typ <- fread(paste0(prefix,  
                                 "spruchkoerper_typ.csv"))  
  
table.spruch.az <- fread(paste0(prefix,  
                                 "spruchkoerper_az.csv"))  
  
table.regz <- fread(paste0(prefix,  
                            "registerzeichen.csv"))  
  
table.jahr.eingangISO <- fread(paste0(prefix,  
                                     "eingangsjahr_iso.csv"))  
  
table.jahr.entscheid <- fread(paste0(prefix,  
                                     "entscheidungsjahr.csv"))  
  
table.output.praesi <- fread(paste0(prefix,  
                                    "praesi.csv"))  
  
table.output.vpraesi <- fread(paste0(prefix,  
                                    "v_praesi.csv"))
```

8.3 Spruchkörper Typ

```
freqtable <- table.spruch.typ[-.N]
```

```
ggplot(data = freqtable) +  
  geom_bar(aes(x = spruchkoerper_typ,  
              y = N),  
          stat = "identity",  
          fill = "#ca2129",  
          color = "black",  
          width = 0.4) +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Spruchkörper-Typ"),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "Typ des Spruchkörpers",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```

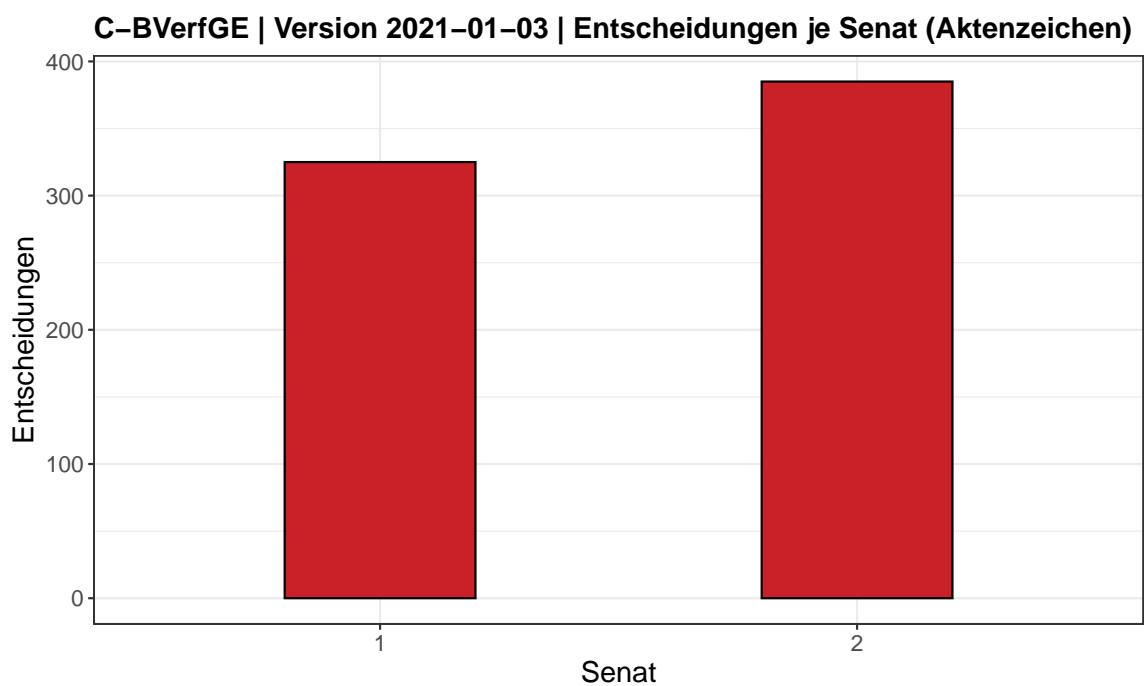


DOI: 10.5281/zenodo.4265943

8.4 Spruchkörper AZ

```
freqtable <- table.spruch.az[-.N]
```

```
ggplot(data = freqtable) +  
  geom_bar(aes(x = spruchkoerper_az,  
               y = N),  
           stat = "identity",  
           fill = "#ca2129",  
           color = "black",  
           width = 0.4) +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Senat (Aktenzeichen)" ),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "Senat",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```



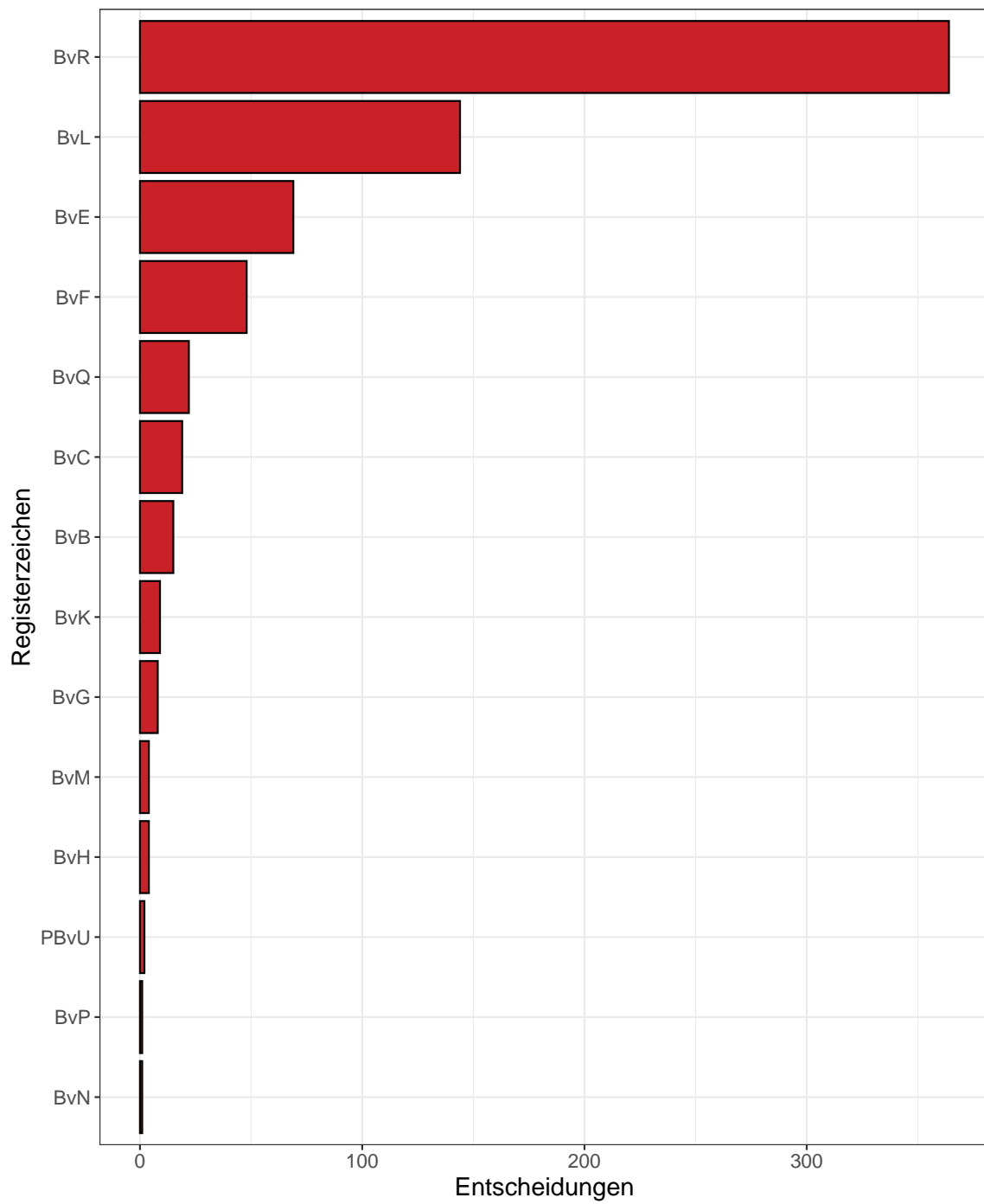
DOI: 10.5281/zenodo.4265943

8.5 Registerzeichen

```
frequetable <- table.regz[-.N]
```

```
ggplot(data = frequetable) +  
  geom_bar(aes(x = reorder(registerzeichen, N),  
               y = N),  
           stat="identity",  
           fill = "#ca2129",  
           color = "black") +  
  coord_flip() +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Registerzeichen"),  
    caption = paste("DOI:",  
                   doi.version),  
    x = "Registerzeichen",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```

C-BVerfGE | Version 2021-01-03 | Entscheidungen je Registerzeichen



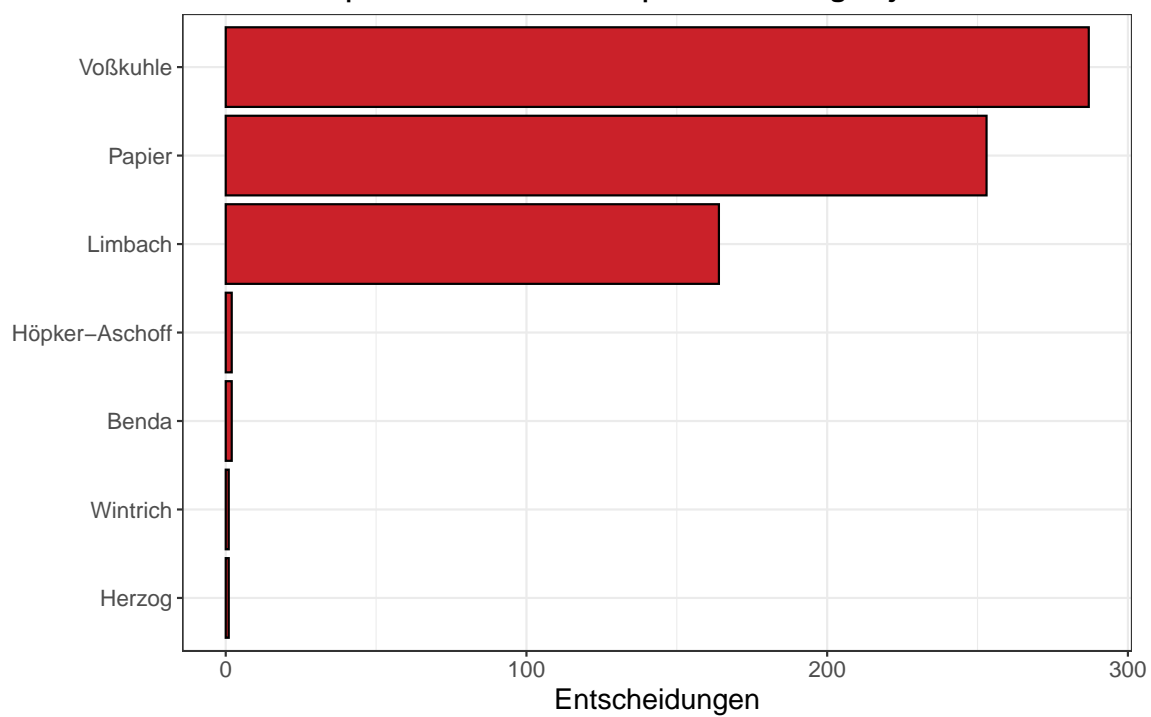
DOI: 10.5281/zenodo.4265943

8.6 PräsidentIn

```
freqtable <- table.output.praesi[-.N]
```

```
ggplot(data = freqtable) +  
  geom_bar(aes(x = reorder(praesi, N),  
                 y = N),  
           stat="identity",  
           fill = "#ca2129",  
           color = "black") +  
  coord_flip() +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je PräsidentIn"),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "PräsidentIn",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    axis.title.y = element_blank(),  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position = "none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```

C-BVerfGE | Version 2021-01-03 | Entscheidungen je PräsidentIn



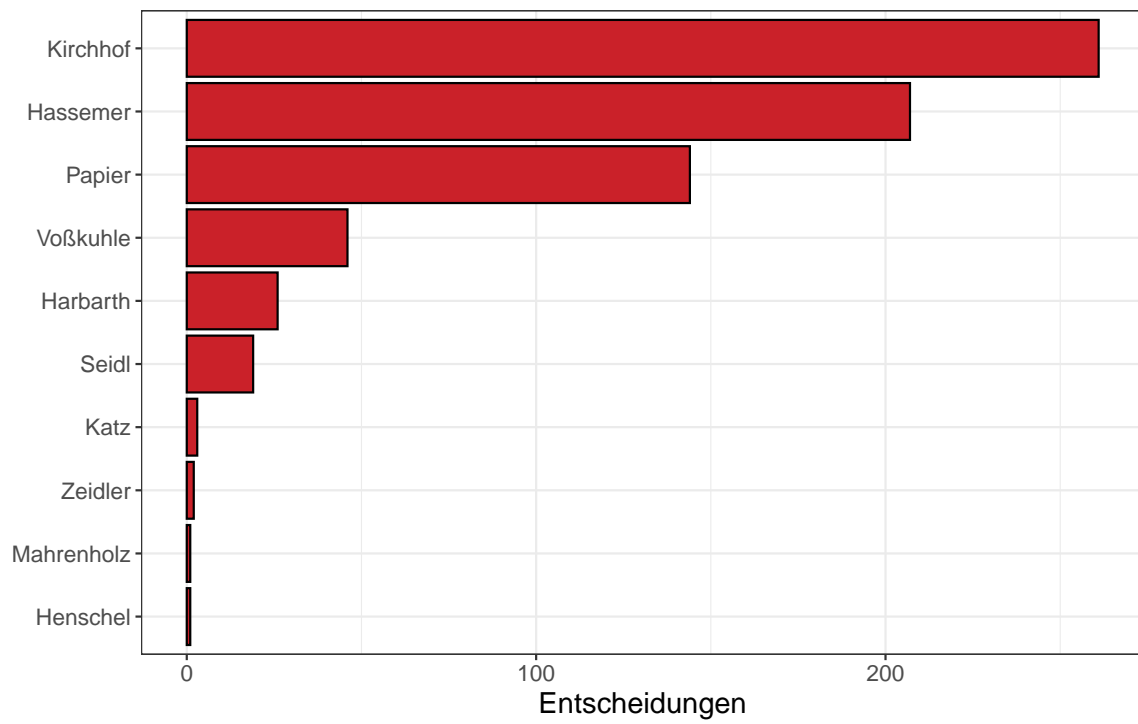
DOI: 10.5281/zenodo.4265943

8.7 Vize-PräsidentIn

```
freqtable <- table.output.vpraesi[-.N]
```

```
ggplot(data = freqtable) +  
  geom_bar(aes(x = reorder(v_praesi, N),  
                 y = N),  
           stat="identity",  
           fill = "#ca2129",  
           color = "black") +  
  coord_flip() +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Vize-PräsidentIn"),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "Vize-PräsidentIn",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    axis.title.y = element_blank(),  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position = "none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```

C-BVerfGE | Version 2021-01-03 | Entscheidungen je Vize-PräsidentIn

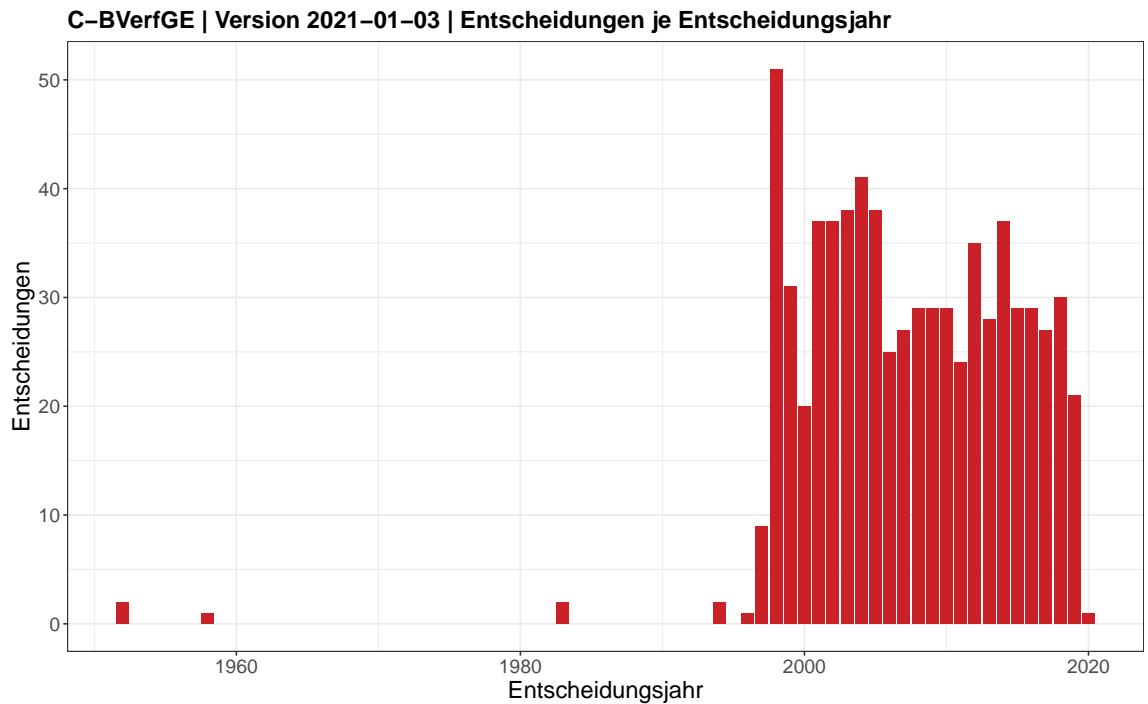


DOI: 10.5281/zenodo.4265943

8.8 Entscheidungsjahr

```
frequetable <- table.jahr.entscheid[-.N][,lapply(.SD, as.numeric)]
```

```
ggplot(data = frequetable) +  
  geom_bar(aes(x = entscheidungsjahr,  
              y = N),  
          stat="identity",  
          fill = "#ca2129") +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Entscheidungsjahr"),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "Entscheidungsjahr",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    text = element_text(size=16),  
    plot.title = element_text(size=16, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```

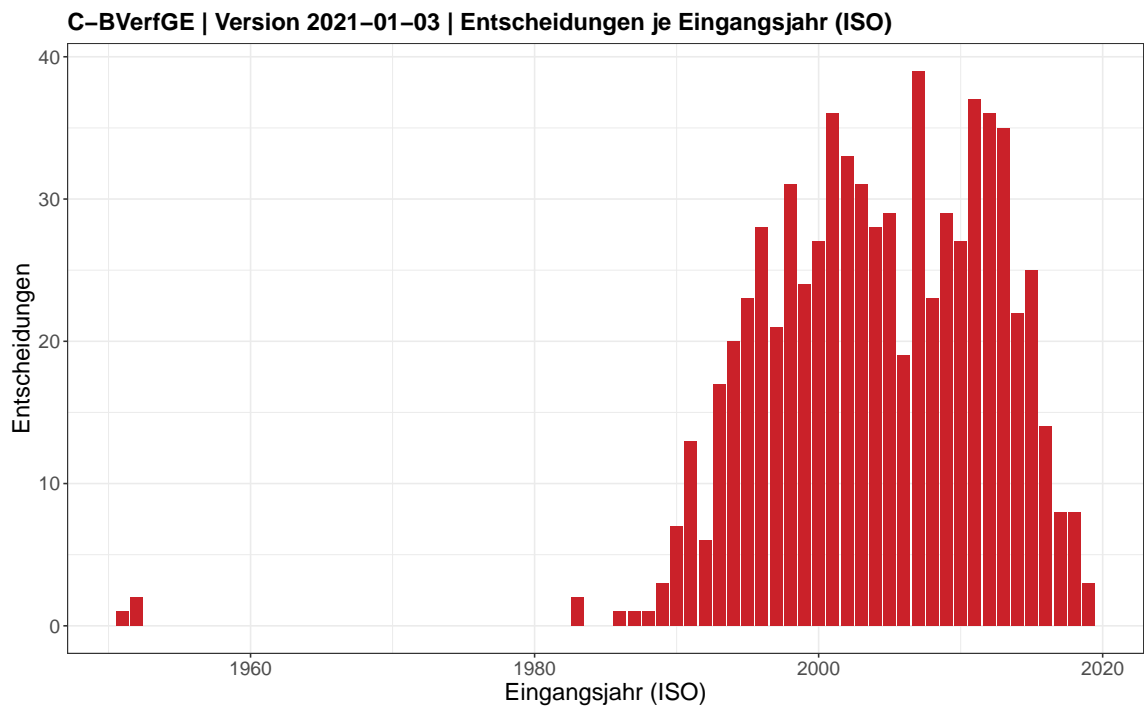


DOI: 10.5281/zenodo.4265943

8.9 Eingangsjahr (ISO)

```
frequetable <- table.jahr.eingangISO[-.N][,lapply(.SD, as.numeric)]
```

```
ggplot(data = frequetable) +  
  geom_bar(aes(x = eingangsjahr_iso,  
               y = N),  
           stat="identity",  
           fill = "#ca2129") +  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
                  "| Version",  
                  datestamp,  
                  "| Entscheidungen je Eingangsjahr (ISO)'),  
    caption = paste("DOI:",  
                    doi.version),  
    x = "Eingangsjahr (ISO)",  
    y = "Entscheidungen"  
  ) +  
  theme(  
    text = element_text(size=16),  
    plot.title = element_text(size=16, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```



DOI: 10.5281/zenodo.4265943

9 Korpus-Analytik

9.1 Berechnung linguistischer Kennwerte

An dieser Stelle werden für jedes Dokument die Anzahl Tokens, Typen und Sätze berechnet und mit den jeweiligen Metadaten verknüpft. Das Ergebnis ist grundsätzlich identisch mit dem eigentlichen Datensatz, nur ohne den Text der Entscheidungen.

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

scope <- seq_len(ndoc(corpus))
result <- foreach(i = scope,
                  .errorhandling = 'pass') %dopar% {
  temp <- summary(corpus[i])
  return(temp)
}

summary.corpus <- rbindlist(result)
```

9.2 Variablen-Namen anpassen

```
setnames(summary.corpus,
         old = c("Text",
                 "Tokens",
                 "Types",
                 "Sentences"),
         new = c("doc_id",
                 "tokens",
                 "typen",
                 "saetze"))
```

9.3 Kennwerte dem Korpus hinzufügen

```
txt.bverfg$tokens <- summary.corpus$tokens
txt.bverfg$typen <- summary.corpus$typen
txt.bverfg$saetze <- summary.corpus$saetze
```

9.4 Anzahl Variablen im Korpus

```
length(txt.bverfg)
```

```
## [1] 25
```

9.5 Alle Variablen-Namen im Korpus

```
names(txt.bverfg)
```

```
## [1] "doc_id"      "text"        "gericht"
## [4] "datum"      "spruchkoerper_typ" "spruchkoerper_az"
## [7] "registerzeichen" "eingangsnummer" "eingangsjahr_az"
## [10] "kollision"   "name"        "band"
## [13] "seite"      "entscheidungs_jahr" "eingangsjahr_iso"
## [16] "praesi"     "v_praesi"    "aktenzeichen"
## [19] "ecli"       "doi_concept" "doi_version"
## [22] "version"    "tokens"      "typen"
## [25] "saetze"
```

9.6 Linguistische Kennwerte

Hinweis: Typen sind definiert als einzigartige Tokens und werden für jedes Dokument gesondert berechnet. Daher ergibt es an dieser Stelle auch keinen Sinn die Typen zu summieren, denn bezogen auf den Korpus wäre der Kennwert ein anderer. Der Wert wird daher manuell auf »NA« gesetzt.

9.6.1 Zusammenfassungen berechnen

```
dt.summary.ling <- summary.corpus[, lapply(.SD,
                                           function(x) unclass(summary(x))),
                                   .SDcols = c("tokens",
                                               "saetze",
                                               "typen")]

dt.sums.ling <- summary.corpus[,
                               lapply(.SD, sum),
                               .SDcols = c("tokens",
                                             "saetze",
                                             "typen")]

dt.sums.ling$typen <- NA

dt.stats.ling <- rbind(dt.sums.ling,
                      dt.summary.ling)

dt.stats.ling <- transpose(dt.stats.ling,
                           keep.names = "names")

setnames(dt.stats.ling, c("Variable",
```

```
"Sum",
"Min",
"Quart1",
"Median",
"Mean",
"Quart3",
"Max"))
```

9.6.2 Zusammenfassungen anzeigen

```
kable(dt.stats.ling,
      format.args = list(big.mark = ","),
      format = "latex",
      booktabs=TRUE,
      longtable=TRUE)
```

Variable	Sum	Min	Quart1	Median	Mean	Quart3	Max
tokens	8,120,097	268	4,872.00	9,329.5	11,436.7563	14,792.75	119,512
saetze	421,471	10	261.25	488.0	593.6211	771.25	5,282
typen	NA	127	1,376.75	2,239.0	2,420.1535	3,096.75	15,930

9.6.3 Zusammenfassungen speichern

```
fwrite(dt.stats.ling,
      paste0(outputdir,
            datasetname,
            "_00_KorpusStatistik_ZusammenfassungLinguistisch.csv"),
      na = "NA")
```

9.7 Quantitative Variablen

9.7.1 Entscheidungsdatum

```
summary(as.IDate(summary.corpus$datum))
```

```
##           Min.         1st Qu.         Median         Mean         3rd Qu.         Max.
## "1952-09-10" "2002-10-09" "2007-07-06" "2007-11-09" "2013-12-03" "2020-01-14"
```

9.7.2 Zusammenfassungen berechnen

```

dt.summary.docvars <- summary.corpus[, lapply(.SD,
                                             function(x)unclass(summary(x))),
                                       .SDcols = c("entscheidungsjahr",
                                                  "eingangsjahr_iso",
                                                  "band",
                                                  "eingangsnummer")]

dt.unique.docvars <- summary.corpus[, lapply(.SD,
                                             function(x)length(unique(x))),
                                       .SDcols = c("entscheidungsjahr",
                                                  "eingangsjahr_iso",
                                                  "band",
                                                  "eingangsnummer")]

dt.stats.docvars <- rbind(dt.unique.docvars,
                          dt.summary.docvars)

dt.stats.docvars <- transpose(dt.stats.docvars,
                              keep.names = "names")

setnames(dt.stats.docvars, c("Variable",
                             "Unique",
                             "Min",
                             "Quart1",
                             "Median",
                             "Mean",
                             "Quart3",
                             "Max"))

```

9.7.3 Zusammenfassungen anzeigen

```

kable(dt.stats.docvars,
      format = "latex",
      booktabs=TRUE,
      longtable=TRUE)

```

Variable	Unique	Min	Quart1	Median	Mean	Quart3	Max
entscheidungsjahr	29	1952	2002	2007.0	2007.3746	2013.00	2020
eingangsjahr_iso	37	1951	1999	2004.0	2004.2127	2011.00	2019
band	64	1	106	119.0	120.1239	134.00	152
eingangsnummer	357	1	4	108.5	696.1817	1310.25	3588

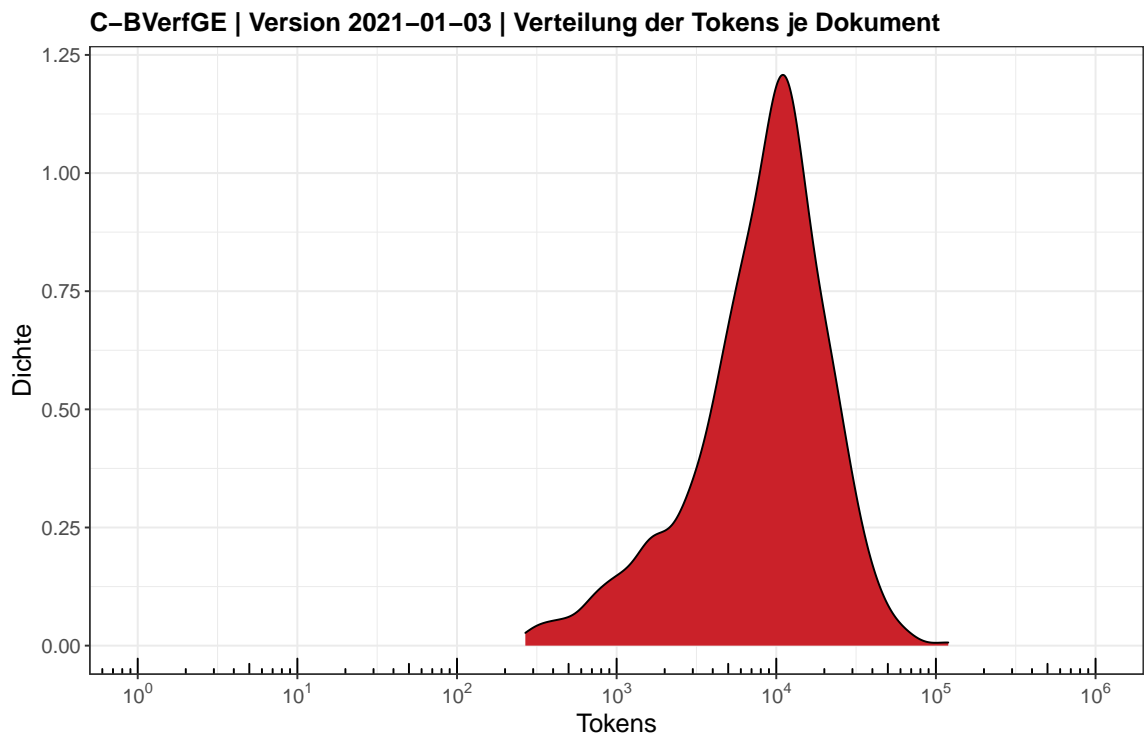
9.7.4 Zusammenfassungen speichern

```
fwrite(dt.stats.docvars,  
      paste0(outputdir,  
             datasetname,  
             "_00_KorpusStatistik_ZusammenfassungDocvarsQuantitativ.csv"),  
      na = "NA")
```

9.8 Density

9.8.1 Density Tokens

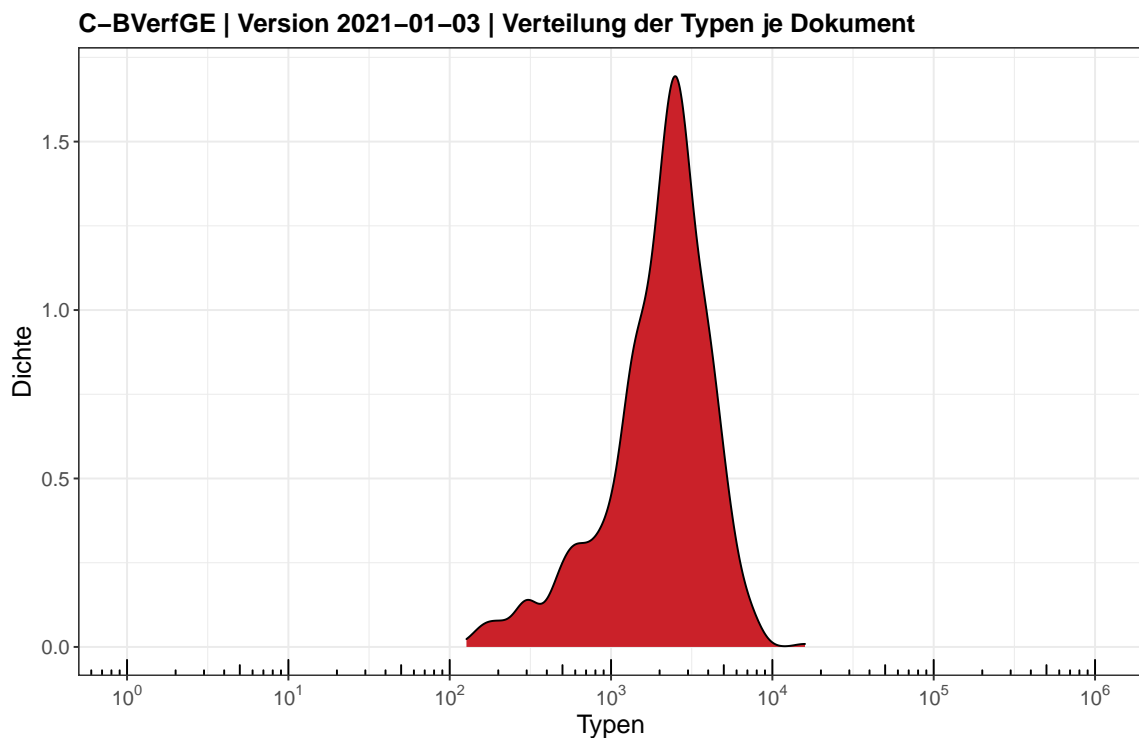
```
ggplot(data = summary.corpus) +  
  geom_density(aes(x = tokens),  
    fill = "#ca2129") +  
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),  
    labels = trans_format("log10", math_format(10^.x)))+  
  annotation_logticks(sides = "b")+  
  coord_cartesian(xlim = c(1, 10^6))+  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
      "| Version",  
      datestamp,  
      "| Verteilung der Tokens je Dokument"),  
    caption = paste("DOI:",  
      doi.version),  
    x = "Tokens",  
    y = "Dichte"  
  )+  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```



DOI: 10.5281/zenodo.4265943

9.8.2 Density Typen

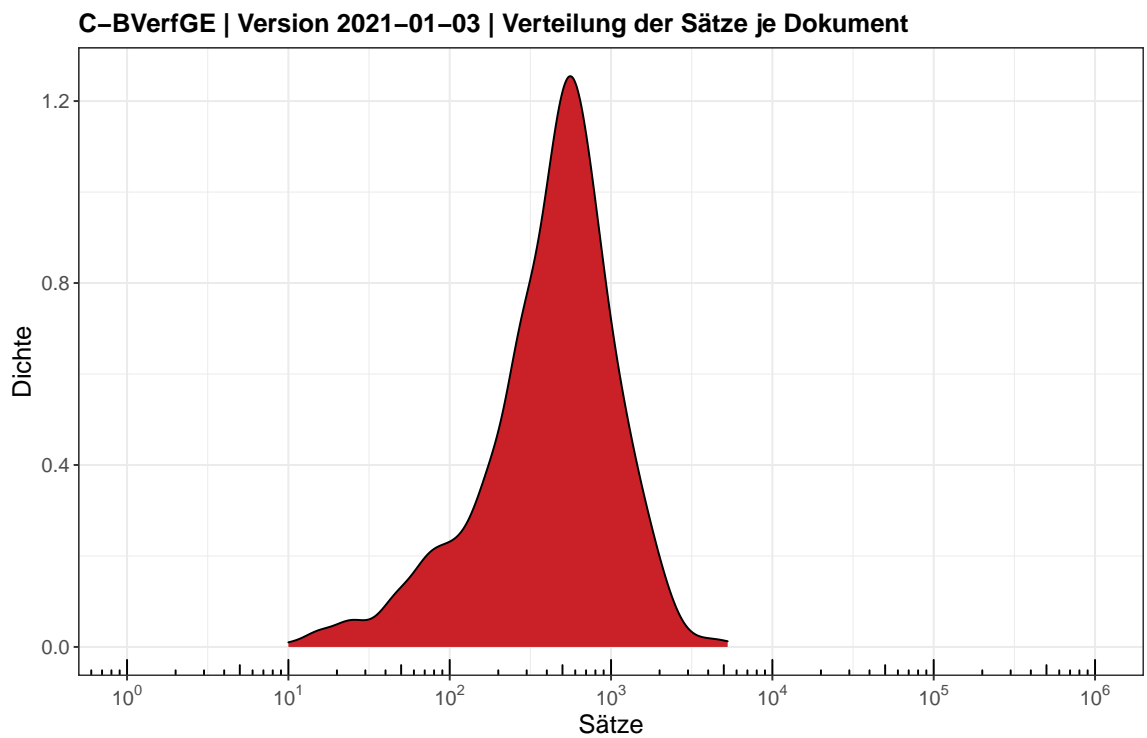
```
ggplot(data = summary.corpus) +  
  geom_density(aes(x = typen),  
    fill = "#ca2129") +  
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),  
    labels = trans_format("log10", math_format(10^.x)))+  
  annotation_logticks(sides = "b")+  
  coord_cartesian(xlim = c(1, 10^6))+  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
      "| Version",  
      datestamp,  
      "| Verteilung der Typen je Dokument"),  
    caption = paste("DOI:",  
      doi.version),  
    x = "Typen",  
    y = "Dichte"  
  )+  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```



DOI: 10.5281/zenodo.4265943

9.8.3 Density Sätze

```
ggplot(data = summary.corpus) +  
  geom_density(aes(x = saetze),  
    fill = "#ca2129") +  
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),  
    labels = trans_format("log10", math_format(10^.x)))+  
  annotation_logticks(sides = "b")+  
  coord_cartesian(xlim = c(1, 10^6))+  
  theme_bw() +  
  labs(  
    title = paste(datasetname,  
      "| Version",  
      datestamp,  
      "| Verteilung der Sätze je Dokument"),  
    caption = paste("DOI:",  
      doi.version),  
    x = "Sätze",  
    y = "Dichte"  
  )+  
  theme(  
    text = element_text(size=14),  
    plot.title = element_text(size=14, face="bold"),  
    legend.position="none",  
    plot.margin = margin(10, 20, 10, 10)  
  )
```



DOI: 10.5281/zenodo.4265943

10 Beispiel-Werte für alle Metadaten anzeigen

```
print(summary.corpus)
```

```
##
##          doc_id
## 1: BVerfG_1952-09-10_S_1_BvR_0379_52_NA_FristlaufVerfBeschwerde-Rü
ckwirkendesGesetz_1_415.txt
## 2: BVerfG_1952-10-10_S_1_BvR_0511_52_NA_BGH-Strafsenat
-Berlin_1_439.txt
## 3: BVerfG_1958-01-15_S_1_BvR_0400_51_
NA_Lüth_7_198.txt
## 4: BVerfG_1983-02-16_S_2_BvE_0001_83_NA_
Bundestagsauflösung-1_62_1.txt
## 5: BVerfG_1983-12-15_S_1_BvR_0209_83_NA_Volkszä
hlung1983_65_1.txt
## ---
## 706: BVerfG_2019-11-06_S_1_BvR_0016_13_NA_
RechtAufVergessen-1_152_152.txt
## 707: BVerfG_2019-11-06_S_1_BvR_0276_17_NA_
RechtAufVergessen-2_152_216.txt
## 708: BVerfG_2019-11-19_S_2_BvL_0022_14_NA_
Erstausbildungskosten_152_274.txt
## 709: BVerfG_2019-12-05_S_1_BvL_0007_18_NA_NichtigkeitKinderehe-
BefangenheitHarbarth_152_332.txt
## 710: BVerfG_2020-01-14_S_2_BvR_2055_16_NA_EntfernungBeamtenverhä
ltnisVerwaltungsakt_152_345.txt
##      typen tokens saetze gericht datum spruchkoerper_typ spruchkoerper_az
## 1: 454 1102 81 BVerfG 1952-09-10 S 1
## 2: 296 715 57 BVerfG 1952-10-10 S 1
## 3: 2930 11249 394 BVerfG 1958-01-15 S 1
## 4: 6343 39850 2092 BVerfG 1983-02-16 S 2
## 5: 4732 23430 1093 BVerfG 1983-12-15 S 1
## ---
## 706: 4440 22345 1063 BVerfG 2019-11-06 S 1
## 707: 3865 21726 1134 BVerfG 2019-11-06 S 1
## 708: 3904 20988 989 BVerfG 2019-11-19 S 2
## 709: 1318 4778 262 BVerfG 2019-12-05 S 1
## 710: 4344 20691 1106 BVerfG 2020-01-14 S 2
##      registerzeichen eingangsnummer eingangsjahr_az kollision
## 1: BvR 379 52 <NA>
## 2: BvR 511 52 <NA>
## 3: BvR 400 51 <NA>
## 4: BvE 1 83 <NA>
## 5: BvR 209 83 <NA>
## ---
## 706: BvR 16 13 <NA>
## 707: BvR 276 17 <NA>
## 708: BvL 22 14 <NA>
## 709: BvL 7 18 <NA>
## 710: BvR 2055 16 <NA>
##
##      name band seite entscheidungsjahr
## 1: FristlaufVerfBeschwerde-RückwirkendesGesetz 1 415 1952
```

```

## 2: BGH-Strafsenat-Berlin 1 439 1952
## 3: Lüth 7 198 1958
## 4: Bundestagsauflösung-1 62 1 1983
## 5: Volkszählung1983 65 1 1983
## ---
## 706: RechtAufVergessen-1 152 152 2019
## 707: RechtAufVergessen-2 152 216 2019
## 708: Erstausbildungskosten 152 274 2019
## 709: NichtigkeitKinderehe-BefangenheitHarbarth 152 332 2019
## 710: EntfernungBeamtenverhältnisVerwaltungsakt 152 345 2020
## eingangsjahr_iso praesi v_praesi aktenzeichen
## 1: 1952 Höpker-Aschoff Katz 1 BvR 379/52
## 2: 1952 Höpker-Aschoff Katz 1 BvR 511/52
## 3: 1951 Wintrich Katz 1 BvR 400/51
## 4: 1983 Benda Zeidler 2 BvE 1/83
## 5: 1983 Benda Zeidler 1 BvR 209/83
## ---
## 706: 2013 Voßkuhle Harbarth 1 BvR 16/13
## 707: 2017 Voßkuhle Harbarth 1 BvR 276/17
## 708: 2014 Voßkuhle Harbarth 2 BvL 22/14
## 709: 2018 Voßkuhle Harbarth 1 BvL 7/18
## 710: 2016 Voßkuhle Harbarth 2 BvR 2055/16
## ecli doi_concept
## 1: ECLI:DE:BVerfG:1952:rs19520910.1bvr037952 10.5281/zenodo.3831111
## 2: ECLI:DE:BVerfG:1952:rs19521010.1bvr051152 10.5281/zenodo.3831111
## 3: ECLI:DE:BVerfG:1958:rs19580115.1bvr040051 10.5281/zenodo.3831111
## 4: ECLI:DE:BVerfG:1983:es19830216.2bve000183 10.5281/zenodo.3831111
## 5: ECLI:DE:BVerfG:1983:rs19831215.1bvr020983 10.5281/zenodo.3831111
## ---
## 706: ECLI:DE:BVerfG:2019:rs20191106.1bvr001613 10.5281/zenodo.3831111
## 707: ECLI:DE:BVerfG:2019:rs20191106.1bvr027617 10.5281/zenodo.3831111
## 708: ECLI:DE:BVerfG:2019:ls20191119.2bvl002214 10.5281/zenodo.3831111
## 709: ECLI:DE:BVerfG:2019:ls20191205.1bvl000718 10.5281/zenodo.3831111
## 710: ECLI:DE:BVerfG:2020:rs20200114.2bvr205516 10.5281/zenodo.3831111
## doi_version version
## 1: 10.5281/zenodo.4265943 2021-01-03
## 2: 10.5281/zenodo.4265943 2021-01-03
## 3: 10.5281/zenodo.4265943 2021-01-03
## 4: 10.5281/zenodo.4265943 2021-01-03
## 5: 10.5281/zenodo.4265943 2021-01-03
## ---
## 706: 10.5281/zenodo.4265943 2021-01-03
## 707: 10.5281/zenodo.4265943 2021-01-03
## 708: 10.5281/zenodo.4265943 2021-01-03
## 709: 10.5281/zenodo.4265943 2021-01-03
## 710: 10.5281/zenodo.4265943 2021-01-03

```

11 CSV-Dateien erstellen

11.1 CSV mit vollem Datensatz speichern

```
csvname.full <- paste(datasetname,  
                      datestamp,  
                      "DE_CSV_Datensatz.csv",  
                      sep = "_")  
  
fwrite(txt.bverfg,  
       csvname.full,  
       na = "NA")
```

11.2 CSV mit Metadaten speichern

Diese Datei ist grundsätzlich identisch mit dem eigentlichen Datensatz, nur ohne den Text der Entscheidungen.

```
csvname.meta <- paste(datasetname,  
                     datestamp,  
                     "DE_CSV_Metadaten.csv",  
                     sep = "_")  
  
fwrite(summary.corpus,  
       csvname.meta,  
       na = "NA")
```

12 Dateigrößen analysieren

12.1 Gesamtgröße

12.1.1 Korpus-Objekt in RAM (MB)

```
print(object.size(corpus),  
      standard = "SI",  
      humanReadable = TRUE,  
      units = "MB")
```

```
## 53.2 MB
```

12.1.2 CSV Korpus (MB)

```
file.size(csvname.full) / 10 ^ 6
```

```
## [1] 52.77971
```

12.1.3 CSV Metadaten (MB)

```
file.size(csvname.meta) / 10 ^ 6
```

```
## [1] 0.229394
```

12.1.4 PDF-Dateien (MB)

```
files.pdf <- list.files(pattern = "\\\\.pdf$",  
                        ignore.case = TRUE)  
  
pdf.MB <- file.size(files.pdf) / 10^6  
sum(pdf.MB)
```

```
## [1] 122.426
```

12.1.5 TXT-Dateien (MB)

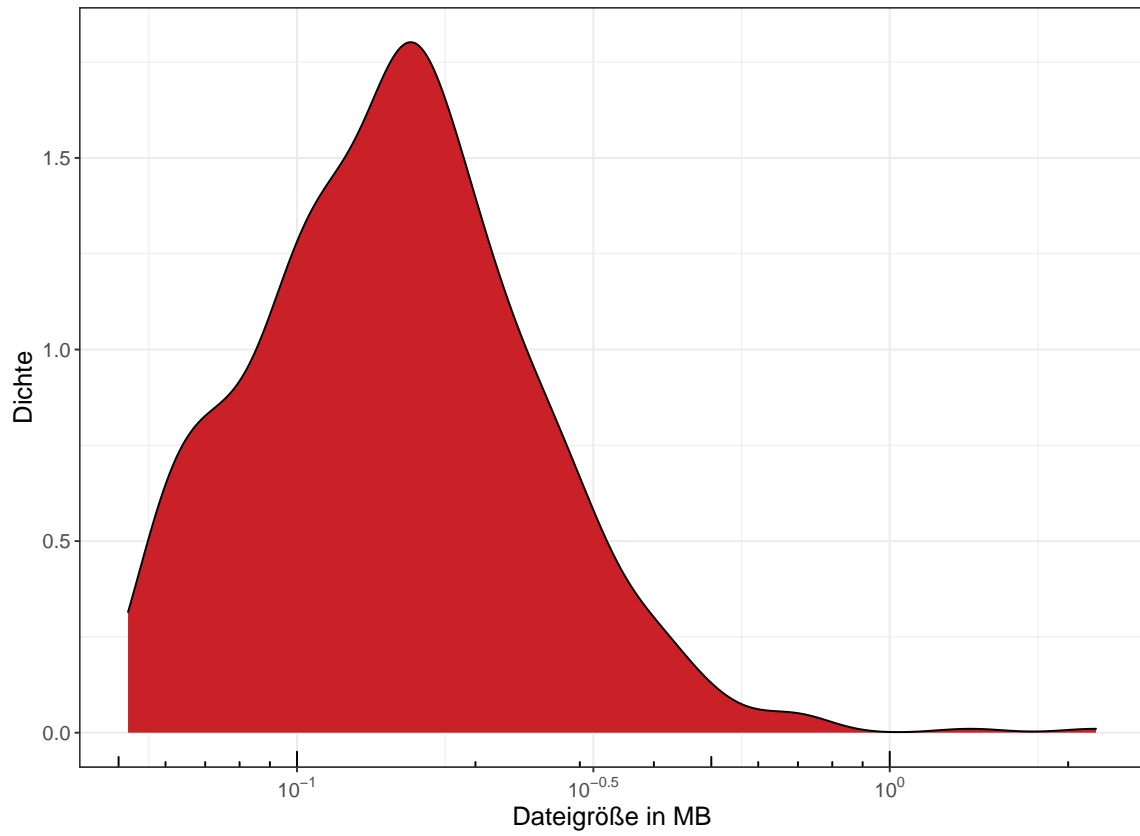
```
files.txt <- list.files(pattern = "\\..txt$",  
                        ignore.case = TRUE)  
  
txt.MB <- file.size(files.txt) / 10^6  
sum(txt.MB)
```

```
## [1] 52.53761
```

12.2 Verteilung der Dateigrößen (PDF)

```
dt.plot <- data.table(pdf.MB)
```

```
ggplot(data = dt.plot,
  aes(x = pdf.MB)) +
  geom_density(fill = "#ca2129") +
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),
    labels = trans_format("log10", math_format(10^.x)))+
  annotation_logticks(sides = "b")+
  theme_bw() +
  labs(
    title = paste(datasetname,
      "| Version",
      datestamp,
      "| Verteilung der Dateigrößen (PDF)",
    caption = paste("DOI:",
      doi.version),
    x = "Dateigröße in MB",
    y = "Dichte"
  )+
  theme(
    text = element_text(size=14),
    plot.title = element_text(size=14, face="bold"),
    legend.position="none",
    panel.spacing = unit(0.1, "lines"),
    plot.margin = margin(10, 20, 10, 10)
  )
```



DOI: 10.5281/zenodo.4265943

13 Erstellen der ZIP-Archive

13.1 Verpacken der CSV-Dateien

```
csvname.full.zip <- gsub(".csv",  
                        ".zip",  
                        csvname.full)  
  
zip(csvname.full.zip,  
    csvname.full)  
  
unlink(csvname.full)
```

```
csvname.meta.zip <- gsub(".csv",  
                        ".zip",  
                        csvname.meta)  
  
zip(csvname.meta.zip,  
    csvname.meta)
```

13.2 Verpacken der PDF-Dateien

```
zip(paste(datasetname,  
          datestamp,  
          "DE_PDF_Datensatz.zip",  
          sep = "_"),  
    files.pdf)  
  
unlink(files.pdf)
```

13.3 Verpacken der TXT-Dateien

```
files.txt <- list.files(pattern="\\.txt",  
                       ignore.case = TRUE)  
  
zip(paste(datasetname,  
          datestamp,  
          "DE_TXT_Datensatz.zip",  
          sep = "_"),  
    files.txt)  
  
unlink(files.txt)
```


13.4 Verpacken der Analyse-Dateien

```
zip(paste0(datasetname,  
          "_",  
          datestamp,  
          "_DE_",  
          basename(outputdir),  
          ".zip"),  
    basename(outputdir))
```

14 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 gelten derzeit als sicher und ein SHA3-Hash mit 512 bit Länge ist nach derzeitigem Wissen auch gegenüber quantenkryptoanalytischen Verfahren hinreichend resistent.

14.1 Liste der ZIP-Archive erstellen

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

14.2 Funktion anzeigen

```
print(f.dopar.multihashes)
```

```
function(x){
```

```
  multihashes <- foreach(filename = x,  
    .errorhandling = 'pass',  
    .combine = 'rbind') %dopar% {  
  
    sha2.256 <- system2("openssl",  
                      paste("sha256",  
                            filename),  
                      stdout = TRUE)  
  
    sha2.256 <- gsub("^.*\\|= ",  
                  "",  
                  sha2.256)  
  
    sha3.512 <- system2("openssl",  
                      paste("sha3-512",  
                            filename),  
                      stdout = TRUE)  
  
    sha3.512 <- gsub("^.*\\|= ",  
                  "",  
                  sha3.512)  
  
    out <- data.frame(filename,  
                      sha2.256,  
                      sha3.512)  
  
    return(out)  
  }  
  return(multihashes)
```

```
}
```

14.3 Hashes berechnen

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

14.4 In Data Table umwandeln

```
setDT(multihashes)
```

14.5 Index hinzufügen

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

14.6 In Datei schreiben

```
fwrite(multihashes,  
      paste(datasetname,  
            datestamp,  
            "KryptographischeHashes.csv",  
            sep = "_"),  
      na = "NA")
```

14.7 Leerzeichen hinzufügen um Zeilenumbruch zu ermöglichen

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

14.8 In Bericht anzeigen

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

index	filename
1	C-BVerfGE_2021-01-03_DE_ANALYSE.zip
2	C-BVerfGE_2021-01-03_DE_CSV_Datensatz.zip
3	C-BVerfGE_2021-01-03_DE_CSV_Metadaten.zip
4	C-BVerfGE_2021-01-03_DE_PDF_Datensatz.zip
5	C-BVerfGE_2021-01-03_DE_TXT_Datensatz.zip

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

index	sha2.256
1	7b7b5a4fb063a96d69b303a20b859662d4e14859ae641be917b85ee95e2f425f
2	1a4559e917085db4734e772ad746f8cd19b7d29805a6c91af0a9a88373c5953f
3	229e24000e87e530365ce7e03ca9c666f7cbb1ecdede6af230afa57c6e284cfc
4	85d077cd074074c350c273755a55d155e9212d1beb9d50e20a49f1be7c1090b7
5	10f9662ceb6427e1b3d054d5e31bf47c0845d477d25d93f8dc227f52815b237d

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

index	sha3.512
1	58e7cd06de2424d11676a01ce5e3a5f384fea61edc4a28c9cf763a825e8fddf537e9a272631b7cb15b83dea6eb8a2d9195737ff5bbe52f1fedcd7592c46b422e
2	2bbe93acc9ed59182ef7cc2b6f3adfce663fb959fa5db2ef70a800ec22d6ced7f386d05a4617e734f589f8917e6fc3346e127542092c24d13085658c9b7e5c3c
3	2e83b28949fa37367b2fdc8e1f032f21c3c0c12954be523c07d82824ba4a5f54356323b9a5b645a6332d36a13459172677e7d61c3ea7de0b45d1634a6dcb736d
4	59a90ba5d710b8e6441ccba9626c31181d6314aea1d71f95fedd2aeaeabff49ce4a03e7d95f63a6efbff096a5a8490715a09caa69726d53d5ae28f3e273838d7
5	b8d4ac2c648fcbc1b58d2ff3ea224cdb81fdb46c31217752a0ef9ed8447dc547c27ea3ecada32eba34f6b3e215436d3efa28aade1a5bc2577717418e5437300f

15 Abschluss

15.1 Cluster stoppen

```
stopCluster(cl)
```

15.2 Datum und Uhrzeit (Ende)

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

```
## [1] "2021-01-03 00:40:55 CET"
```

15.3 Laufzeit des gesamten Skriptes

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

```
## Time difference of 25.81045 mins
```

15.4 Warnungen

```
warnings()
```

```
## Warning messages:  
## 1: In grep("(LaTeX|Package [[:alnum:]]+)" Warning:", x) :  
##   input string 991 is invalid in this locale  
## 2: In grep("(LaTeX|Package [[:alnum:]]+)" Warning:", x) :  
##   input string 1018 is invalid in this locale
```

16 Parameter für strenge Replikationen

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

```
## [1] "OpenSSL 1.1.1i FIPS 8 Dec 2020"
```

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 32 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libopenblas-r0.3.12.so
##
## 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] parallel stats      graphics grDevices utils      datasets methods
## [8] base
##
## other attached packages:
##  [1] quanteda_2.1.2      readtext_0.80      doParallel_1.0.16 iterators_1.0.13
##  [5] foreach_1.5.1      pdftools_2.3.1     rvest_0.3.6        xml2_1.3.2
##  [9] httr_1.4.2          data.table_1.13.4  scales_1.1.1       ggplot2_3.3.2
## [13] magick_2.5.2        kableExtra_1.3.1  knitr_1.30
##
## loaded via a namespace (and not attached):
##  [1] tinytex_0.28        qpdf_1.1           tidyselect_1.1.0    xfun_0.19
##  [5] purrr_0.3.4         lattice_0.20-41    colorspace_2.0-0    vctrs_0.3.6
##  [9] generics_0.1.0      usethis_2.0.0      htmltools_0.5.0     viridisLite
## [13] yaml_2.2.1          rlang_0.4.9        pillar_1.4.7        glue_1.4.2
## [17] withr_2.3.0         selectr_0.4-2      lifecycle_0.2.0     stringr_1.4.0
## [21] munsell_0.5.0       gtable_0.3.0       codetools_0.2-18    evaluate_0.14
## [25] labeling_0.4.2      curl_4.3           highr_0.8           Rcpp_1.0.5
## [29] RcppParallel_5.0.2  webshot_0.5.2      fs_1.5.0            farver_2.0.3
## [33] fastmatch_1.1-0     stopwords_2.1      askpass_1.1         digest_0.6.27
## [37] stringi_1.5.3       dplyr_1.0.2        grid_4.0.3          tools_4.0.3
## [41] magrittr_2.0.1      tibble_3.0.4       crayon_1.3.4        pkgconfig_2.0.3
## [45] Matrix_1.2-18       ellipsis_0.3.1     rmarkdown_2.5       rstudioapi_0.13
## [49] R6_2.5.0            compiler_4.0.3
```

Literaturverzeichnis

- Analytics, Revolution, and Steve Weston. 2020. *Iterators: Provides Iterator Construct*. <https://github.com/RevolutionAnalytics/iterators>.
- Benoit, Kenneth, and Adam Obeng. 2020. *Readtext: Import and Handling for Plain and Formatted Text Files*. <https://github.com/quanteda/readtext>.
- Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. “Quanteda: An R Package for the Quantitative Analysis of Textual Data.” *Journal of Open Source Software* 3 (30): 774. <https://doi.org/10.21105/joss.00774>.
- Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, Akitaka Matsuo, Jiong Wei Lua, Jouni Kuha, and William Lowe. 2020. *Quanteda: Quantitative Analysis of Textual Data*. <https://quanteda.io>.
- Corporation, Microsoft, and Steve Weston. 2020. *DoParallel: Foreach Parallel Adaptor for the Parallel Package*. <https://CRAN.R-project.org/package=doParallel>.
- Dowle, Matt, and Arun Srinivasan. 2020. *Data.table: Extension of ‘Data.frame’*. <https://CRAN.R-project.org/package=data.table>.
- Ooms, Jeroen. 2020a. *Magick: Advanced Graphics and Image-Processing in R*. <https://CRAN.R-project.org/package=magick>.
- . 2020b. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents*. <https://CRAN.R-project.org/package=pdfutils>.
- R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Revolution Analytics, and Steve Weston. n.d. *Foreach: Provides Foreach Looping Construct*.
- Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.
- . 2020a. *Htttr: Tools for Working with Urls and Http*. <https://CRAN.R-project.org/package=htttr>.
- . 2020b. *Rvest: Easily Harvest (Scrape) Web Pages*. <https://CRAN.R-project.org/package=rvest>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Wickham, Hadley, Jim Hester, and Jeroen Ooms. 2020. *Xml2: Parse Xml*. <https://CRAN.R-project.org/package=xml2>.
- Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization*. <https://CRAN.R-project.org/package=scales>.
- Xie, Yihui. 2014. “Knitr: A Comprehensive Tool for Reproducible Research in R.” In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich

Leisch, and Roger D. Peng. Chapman; Hall/CRC. <http://www.crcpress.com/product/isbn/9781466561595>.

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

———. 2020. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. <https://yihui.org/knitr/>.

Zhu, Hao. 2020. *KableExtra: Construct Complex Table with Kable and Pipe Syntax*. <https://CRAN.R-project.org/package=kableExtra>.