Corpus der Entscheidungen des Bundesgerichtshofs

(CE-BGH-Source)

Compilation Report

Version 2021-04-27

DOI: 10.5281/zenodo.4705865

Titel	Source Code des »Corpus der Entscheidungen des Bundesgerichtshofs«
Abkürzung	CE-BGH-Source
Autor	Seán Fobbe
Version	2021-04-27
Download	https://doi.org/10.5281/zenodo.4705865
Lizenz	MIT No Attribution (MIT-0)

Zitiervorschlag

Seán Fobbe (2021). Source Code des »Corpus der Entscheidungen des Bundesgerichtshofs« (CE-BGH-Source). Version 2021-04-27. Zenodo. DOI: 10.5281/zenodo.4705865.

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 2021-04-27. Für das Gesamtkonzept der Software steht eine »Concept DOI« zur Verfügung, die auf der Zenodo-Seite jeder Version unter »Cite all versions?« zu finden ist. Die »Concept DOI« verlinkt immer die aktuellste Version.

Lizenz: MIT No Attribution (MIT-0)

Disclaimer

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

Inhaltsverzeichnis

1 Einleitung

1.1 Überblick

Dieses R-Skript lädt alle in der amtlichen Datenbank des Bundesgerichtshofs[1] veröffentlichten Entscheidungen des des Bundesgerichtshofs (BGH) herunter und kompiliert sie in einen reichhaltigen menschen- und maschinenlesbaren Korpus. Es ist die Basis für den **Corpus der Entscheidungen des Bundesgerichtshofs (CE-BGH)**.

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 diesen Link erreichbar: https://doi.org/10.5281/zenodo.3942742

1.2 Funktionsweise

Primäre Endprodukte des Skripts sind folgende ZIP-Archive:

1. Der volle Datensatz im CSV-Format (mit zusätzlichen Metadaten)
2. Die reinen Metadaten im CSV-Format (wie unter 1, nur ohne Entscheidungsinhalte)
3. Alle Entscheidungen im TXT-Format
4. Alle Entscheidungen im PDF-Format
5. Nur Leitsatz-Entscheidungen im PDF-Format
6. Nur benannte Entscheidungen im PDF-Format
7. Alle Analyse-Ergebnisse (Tabellen als CSV, Grafiken als PDF und PNG)

Zusätzlich werden für alle ZIP-Archive kryptographische Signaturen (SHA2-256 und SHA3-512) berechnet und in einer CSV-Datei hinterlegt. 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 versuchen 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 20 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] https://www.bundesgerichtshof.de

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 = "CE-BGH_Source_CorpusCreation.R",
                  output_file = paste0("CE-BGH_",
                                       Sys.Date(),
                                       "_CompilationReport.pdf"),
                  envir = new.env())
```

1.4.2 Codebook

Um das **Codebook** zu kompilieren und einen PDF-Bericht zu erstellen führen Sie bitte 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 = "CE-BGH_Source_CodebookCreation.R",
                  output_file = paste0("CE-BGH_",
                                       Sys.Date(),
                                       "_Codebook.pdf"),
                  envir = new.env())
```

2 Parameter

2.1 Name des Datensatzes

```
datasetname <- "CE-BGH"
```

2.2 DOI des Datensatz-Konzeptes

```
doi.concept <- "10.5281/zenodo.3942742" # checked
```

2.3 DOI der konkreten Version

```
doi.version <- "10.5281/zenodo.4705855" # checked
```

2.4 Lizenz

```
license <- "Creative Commons Zero 1.0 Universal"
```

2.5 Verzeichnis für Analyse-Ergebnisse

Muss mit einem Schrägstrich enden!

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

2.6 Debugging-Modus

Der Debugging-Modus reduziert den Such-Umfang auf den in der Variable »debug.scope« angegebenen Umfang Seiten (jede Seite enthält idR 30 Entscheidungen), den Download-Umfang auf den in der Variable »debug.sample« definierten Umfang zufällig ausgewählter Entscheidungen und löscht im Anschluss fünf zufällig ausgewählte Entscheidungen um den Wiederholungsversuch zu testen. Nur für Test- und Demonstrationszwecke.

```
mode.debug <- FALSE
debug.scope <- 50
debug.sample <- 500
```

2.7 Optionen: Quanteda

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

2.8 Optionen: Knitr

2.8.1 Ausgabe-Format

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

2.8.2 DPI für Raster-Grafiken

```
dpi <- 300
```

2.8.3 Ausrichtung von Grafiken im Compilation Report

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

2.9 Frequenztabellen: Ignorierte Variablen

Diese Variablen werden bei der Erstellung der Frequenztabellen nicht berücksichtigt.

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

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] "2021-04-27"
```

3.2 Datum und Uhrzeit (Beginn)

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

```
## [1] "2021-04-27 03:49:07 CEST"
```

3.3 Ordner für Analyse-Ergebnisse erstellen

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

3.4 Packages Laden

```
library(fs)           # Verbessertes File Handling
library(mgsub)        # Vektorisiertes Gsub
library(httr)         # HTTP-Werkzeuge
library(rvest)        # HTML/XML-Extraktion
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
```

```
## Loading required package: parallel
```

```
library(ggplot2)      # Fortgeschrittene Datenvisualisierung
library(scales)       # Skalierung von Diagrammen
library(data.table)   # Fortgeschrittene Datenverarbeitung
```

```
## data.table 1.14.0 using 8 threads (see ?getDTthreads).  Latest news: r-
    datatable.com
```

```
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 Daten ist bewusst 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 (Anzahl)

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

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

3.9.2 Quanteda

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

3.9.3 Data.table

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

4 Download: Weitere Datensätze

4.1 Registerzeichen und Verfahrensarten

Die Registerzeichen werden im Laufe des Skripts mit ihren detaillierten Bedeutungen aus dem folgenden Datensatz abgeglichen: »Seán Fobbe (2021). Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD). Version 1.0.1. Zenodo. DOI: 10.5281/zenodo.4569564.« Das Ergebnis des Abgleichs wird in der Variable »verfahrensart« in den Datensatz eingefügt.

```
if (file.exists("AZ-BRD_1-0-1_DE_Registerzeichen_Datensatz.csv") == FALSE){
    download.file("https://zenodo.org/record/4569564/files/AZ-BRD_1-0-1_DE_
    Registerzeichen_Datensatz.csv?download=1",
 "AZ-BRD_1-0-1_DE_Registerzeichen_Datensatz.csv")
    }
```

4.2 Personendaten zu Präsident:innen

Die Personendaten stammen aus folgendem Datensatz: »Seán Fobbe and Tilko Swalve (2021). Presidents and Vice-Presidents of the Federal Courts of Germany (PVP-FCG). Version 2021-04-08. Zenodo. DOI: 10.5281/zenodo.4568682«.

```
if (file.exists("PVP-FCG_2021-04-08_GermanFederalCourts_Presidents.csv") == FALSE
    ){
    download.file("https://zenodo.org/record/4568682/files/PVP-FCG_2021-04-08_
    GermanFederalCourts_Presidents.csv?download=1",
                  "PVP-FCG_2021-04-08_GermanFederalCourts_Presidents.csv")
}
```

4.3 Personendaten zu Vize-Präsident:innen

Die Personendaten stammen aus folgendem Datensatz: »Seán Fobbe and Tilko Swalve (2021). Presidents and Vice-Presidents of the Federal Courts of Germany (PVP-FCG). Version 2021-04-08. Zenodo. DOI: 10.5281/zenodo.4568682«.

```
if (file.exists("PVP-FCG_2021-04-08_GermanFederalCourts_VicePresidents.csv") ==
    FALSE){
    download.file("https://zenodo.org/record/4568682/files/PVP-FCG_2021-04-08_
    GermanFederalCourts_VicePresidents.csv?download=1",
                  "PVP-FCG_2021-04-08_GermanFederalCourts_VicePresidents.csv")
}
```

5 Links suchen

5.1 Maximalen Such-Umfang einlesen

```
scope.source <- fread("CE-BGH_Source_Scope.csv")
```

5.2 Maximalen Such-Umfang anzeigen

Die Variable »pagemax1« ist die maximale Anzahl Seiten wenn der Index mit 1 beginnt. Die Variable »pagemax0« ist die maximale Anzahl Seiten wenn der Index mit 0 beginnt. Die URL beginn mit dem Index 0.

```
print(scope.source)
```

```
##     year pagemax1 pagemax0
##  1: 2000       74       73
##  2: 2001       81       80
##  3: 2002       92       91
##  4: 2003       97       96
##  5: 2004      103      102
##  6: 2005      104      103
##  7: 2006      104      103
##  8: 2007      111      110
##  9: 2008      122      121
## 10: 2009      114      113
## 11: 2010      122      121
## 12: 2011      124      123
## 13: 2012      118      117
## 14: 2013      107      106
## 15: 2014      103      102
## 16: 2015      103      102
## 17: 2016      110      109
## 18: 2017      105      104
## 19: 2018      100       99
## 20: 2019       99       98
## 21: 2020      109      108
## 22: 2021       27       26
##     year pagemax1 pagemax0
```

5.3 Funktion definieren

Diese Funktion nimmt eine ganzzahlige y-Variable als Maximum einer Sequenz von 1 bis y und weist ihr in einer data.table jeweils immer die gleiche x-Variable zu.

```
f.extend <- function(x, y, begin = 0){
    y.ext <- begin:y
    x.ext <- rep(x, length(y.ext))
```

```
    dt.out <- list(data.table(x.ext, y.ext))
    return(dt.out)
}

f.extend <- Vectorize(f.extend)
```

5.4 Genauen Such-Umfang berechnen

```
scope <- f.extend(scope.source$year,
                  scope.source$pagemax0)


scope <- rbindlist(scope)

setnames(scope,
         c("year",
           "page"))
```

5.5 Locator einfügen

```
scope[, loc := {
    loc <- paste0(year, "-", page)
    list(loc)
    }]
```

5.6 [Debugging Modus] Reduzierung des Such-Umfangs

```
if (mode.debug == TRUE){
    scope <- scope[sample(scope[,.N], debug.scope)][order(year, page)]
    }
```

5.7 Geschätzte Such-Dauer in Minuten

```
(scope[,.N]* 2.5) / 60
```

```
## [1] 92.875
```

5.8 Zeitstempel: Linksammlung Beginn

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

```
## [1] "2021-04-27 03:49:15 CEST"
```

5.9 Metadaten extrahieren

```
meta.all.list <- vector("list",
                        scope[,.N])

scope.random <- sample(scope[,.N])

for (i in seq_along(scope.random)){

    year <- scope$year[scope.random[i]]
    page <- scope$page[scope.random[i]]

    URL  <- paste0("https://juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/
    list.py?Gericht=bgh&Art=en&Datum=",
                   year,
                   "&Seite=",
                   page)

    html <- read_html(URL)

    link <-  html_nodes(html, "a" )%>% html_attr('href')

    link <- grep ("Blank=1.pdf",
                   link,
                   ignore.case = TRUE,
                   value = TRUE)

    link <- sprintf("https://juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/%s
    ",
                    link)


    datum <- html_nodes(html, "[class='EDatum']") %>% html_text(trim = TRUE)

    spruch <- html_nodes(html, "[class='ESpruchk']") %>% html_text(trim = TRUE)

    az <- html_nodes(html, "[class='doklink']") %>% html_text(trim = TRUE)

    comment <- html_nodes(html, "[class='ETitel']") %>% html_text(trim = TRUE)

    meta.all.list[[scope.random[i]]] <- data.table(year,
                                    page,
                                    link,
```

```
                                    datum,
                                    spruch,
                                    az,
                                    comment)

    remaining <- length(scope.random) - i

    if ((remaining %% 10^2) == 0){
        print(paste(Sys.time(), "| Noch", remaining , "verbleibend."))
    }

    if((i %% 100) == 0){
        Sys.sleep(runif(1, 5, 15))
    }else{
        Sys.sleep(runif(1, 1.5, 2.5))
        }
}
```

```
## [1] "2021-04-27 03:50:25 | Noch 2200 verbleibend."
## [1] "2021-04-27 03:54:26 | Noch 2100 verbleibend."
## [1] "2021-04-27 03:58:41 | Noch 2000 verbleibend."
## [1] "2021-04-27 04:02:54 | Noch 1900 verbleibend."
## [1] "2021-04-27 04:07:09 | Noch 1800 verbleibend."
## [1] "2021-04-27 04:11:15 | Noch 1700 verbleibend."
## [1] "2021-04-27 04:15:18 | Noch 1600 verbleibend."
## [1] "2021-04-27 04:19:42 | Noch 1500 verbleibend."
## [1] "2021-04-27 04:24:02 | Noch 1400 verbleibend."
## [1] "2021-04-27 04:28:15 | Noch 1300 verbleibend."
## [1] "2021-04-27 04:32:33 | Noch 1200 verbleibend."
## [1] "2021-04-27 04:36:49 | Noch 1100 verbleibend."
## [1] "2021-04-27 04:40:55 | Noch 1000 verbleibend."
## [1] "2021-04-27 04:45:17 | Noch 900 verbleibend."
## [1] "2021-04-27 04:49:30 | Noch 800 verbleibend."
## [1] "2021-04-27 04:53:47 | Noch 700 verbleibend."
## [1] "2021-04-27 04:58:04 | Noch 600 verbleibend."
## [1] "2021-04-27 05:02:17 | Noch 500 verbleibend."
## [1] "2021-04-27 05:06:33 | Noch 400 verbleibend."
## [1] "2021-04-27 05:10:55 | Noch 300 verbleibend."
## [1] "2021-04-27 05:15:17 | Noch 200 verbleibend."
## [1] "2021-04-27 05:19:35 | Noch 100 verbleibend."
## [1] "2021-04-27 05:23:51 | Noch 0 verbleibend."
```

5.10 Zeitstempel: Linksammlung Ende

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

```
## [1] "2021-04-27 05:23:53 CEST"
```

5.11 Dauer Linksammlung

```
end.linkcollect - begin.linkcollect
```

```
## Time difference of 1.577259 hours
```

5.12 Zusammenfügen

```
dt.download <- rbindlist(meta.all.list)
```

6 Test-Reihe: Vollständigkeit der Auswertung

6.1 Locator einfügen

```
dt.download[, loc := {
    loc <- paste0(year,
                  "-",
                  page)
    list(loc)
    }]
```

6.2 Theoretischer Fehlbetrag

```
SOLL <- scope[,.N] * 30
IST <- dt.download[,.N]

missing.N <- SOLL - IST

print(missing.N)
```

```
## [1] 244
```

6.3 Seiten mit weniger als 30 Entscheidungen anzeigen

```
less30 <- dt.download[, .N,  keyby = "loc"][N < 30]

print(less30)
```

```
##          loc  N
##  1:  2000-73 16
##  2:  2001-80  8
##  3:  2002-91 23
##  4:  2003-96 12
##  5: 2004-102 27
##  6: 2005-103 13
##  7: 2006-103 23
##  8: 2007-110 18
##  9: 2008-121  4
## 10: 2009-113 28
## 11: 2010-121 18
## 12: 2011-123 22
## 13: 2012-117 21
## 14: 2013-106 24
## 15: 2014-102 20
## 16: 2015-102 19
```

```
## 17: 2016-109 15
## 18: 2017-104  7
## 19:  2018-99 24
## 20:  2019-98 16
## 21: 2020-108 28
##          loc  N
```

6.4 Fehlbetrag durch Seiten mit weniger als 30 Entscheidungen

```
less30.N <- (length(less30$N) * 30) - sum(less30$N)
print(less30.N)
```

```
## [1] 244
```

6.5 Tatsächlicher Fehlbetrag

Test: Ist der Fehlbetrag vollständig durch Seiten mit weniger als 30 Entscheidungen zu erklären? Falls ja, weisen beide sub-Tests maximal ein Ergebnis von 0 auf.

6.5.1 Fehlbetrag der NICHT durch Seiten mit weniger als 30 Entscheidungen erklärbar ist

```
print(missing.N - less30.N)
```

```
## [1] 0
```

6.5.2 Gegenüberstellung: Anzahl Jahre und Anzahl Seiten mit weniger als 30 Entscheidungen

Für jedes Jahr sollte es eine letzte Seite mit weniger als 30 Entscheidungen geben. Falls zufällig die letzte Seite exakt 30 Entscheidungen hat, wäre das Ergebnis negativ. Ein Ergebnis von 0 oder kleiner bedeutet, dass der Test bestanden wurde. Der Test ist nur aussagekräftig wenn der gesamte Such-Umfang abgefragt wurde.

```
if (mode.debug == FALSE){
    less30[,.N] - uniqueN(scope$year)
    }
```

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

6.6 Vorhandensein aller Jahr/Seiten-Kombinationen

Dieser Test zeigt an, ob alle Jahr/Seiten-Kombinationen auch in den Daten vorhanden sind. Falls nicht, zeigt er die fehlenden Kombinationen an.

```
setdiff(scope$loc,
        dt.download$loc)
```

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

7 Bereinigung der Metadaten

7.1 Datum bereinigen

```
dt.download[, datum := {
    datum <- as.character(datum)
    datum <- as.IDate(datum, "%d.%m.%Y")
    list(datum)}]
```

7.2 Aktenzeichen bereinigen

7.2.1 Reguläre Substitutionen

```
az.out <- dt.download$az
```

Hinweis: An dieser Stelle wird eine mysteriöse Unterstrich-Variante mit einem regulären Unterstrich ersetzt. Es ist mir aktuell unklar um was für eine Art von Zeichen es sich handelt und wieso es in den Daten des Bundesgerichtshofs auftaucht. Weil die Code-Zeile zu einem Anzeigefehler in der LaTeX-Kompilierung führt, ist sie im Compilation Report nicht abgedruckt. Sie finden die Zeile im eigentlichen Source Code. Ich arbeite daran, die Anzeige zu vervollständigen.

```
az.out1 <- gsub("\\/", "_", az.out1)

az.out1 <- gsub("_und.*$", "", az.out1)

az.out1 <- gsub("StB_", "NA_StB_", az.out1)
az.out1 <- gsub("StbSt_\\(R\\)", "NA_StbStR", az.out1)
az.out1 <- gsub("StbSt_\\(B\\)", "NA_StbStB", az.out1)

az.out1 <- gsub("PatAnwZ_", "NA_PatAnwZ_", az.out1)

az.out1 <- gsub("AnwZ_\\(Brf[gG]\\)", "NA_AnwZBrfg", az.out1)
az.out1 <- gsub("AK", "NA_AK", az.out1)
az.out1 <- gsub("ARAnw_", "NA_ARAnw_", az.out1)
az.out1 <- gsub("ARs_\\(Voll[zZ]\\)_", "ARsVollz_", az.out1)
az.out1 <- gsub("AR_\\(VZ\\)", "ARVZ", az.out1)
az.out1 <- gsub("AR_\\(VS\\)", "ARVS", az.out1)
az.out1 <- gsub("AR_\\(Ri\\)", "NA_ARRi", az.out1)
az.out1 <- gsub("^AnwSt_\\(B\\)", "NA_AnwStB", az.out1)
az.out1 <- gsub("^AnwSt_\\(R\\)", "NA_AnwStR", az.out1)
az.out1 <- gsub("^PatAnwSt_\\(B\\)", "NA_PatAnwStB", az.out1)
az.out1 <- gsub("^PatAnwSt_\\(R\\)", "NA_PatAnwStR", az.out1)
az.out1 <- gsub("AnwZ_\\(B\\)", "NA_AnwZB", az.out1)
az.out1 <- gsub("AnwZ_\\(P\\)_", "NA_AnwZP_", az.out1)
az.out1 <- gsub("^AnwZ_", "NA_AnwZ_", az.out1)
az.out1 <- gsub("ARNot_", "NA_ARNot_", az.out1)

az.out1 <- gsub("BLw", "NA_BLw", az.out1)
```

```
az.out1 <- gsub("En[vV]R_", "NA_EnVR_", az.out1)
az.out1 <- gsub("EnZR_", "NA_EnZR_", az.out1)
az.out1 <- gsub("EnVZ_", "NA_EnVZ_", az.out1)
az.out1 <- gsub("EnZB_", "NA_EnZB_", az.out1)

az.out1 <- gsub("GmS-OGB_", "NA_GMSOGB_", az.out1)
az.out1 <- gsub("GSSt_", "NA_GSSt_", az.out1)
az.out1 <- gsub("GSZ_", "NA_GSZ_", az.out1)

az.out1 <- gsub("KZR_", "NA_KZR_", az.out1)
az.out1 <- gsub("KVZ_", "NA_KVZ_", az.out1)
az.out1 <- gsub("KRB_", "NA_KRB_", az.out1)
az.out1 <- gsub("KZB_", "NA_KZB_", az.out1)
az.out1 <- gsub("KVR_", "NA_KVR_", az.out1)

az.out1 <- gsub("LwZA_", "NA_LwZA_", az.out1)
az.out1 <- gsub("LwZR_", "NA_LwZR_", az.out1)
az.out1 <- gsub("LwZB", "NA_LwZB", az.out1)

az.out1 <- gsub("NotSt_\\(B\\)", "NA_NotStB", az.out1)
az.out1 <- gsub("NotSt_\\(Brfg\\)", "NA_NotStBrfg", az.out1)
az.out1 <- gsub("NotZ_\\(Brfg\\)", "NA_NotZBrfg", az.out1)
az.out1 <- gsub("NotZ_", "NA_NotZ_", az.out1)

az.out1 <- gsub("RiZ_\\(R\\)", "NA_RiZR", az.out1)
az.out1 <- gsub("RiZ\\(R\\)", "NA_RiZR", az.out1)
az.out1 <- gsub("RiZ_\\(B\\)", "NA_RiZB", az.out1)
az.out1 <- gsub("RiZ_", "NA_RiZ_", az.out1)
az.out1 <- gsub("RiSt_\\(R\\)_", "NA_RiStR_", az.out1)
az.out1 <- gsub("RiSt_\\(B\\)_", "NA_RiStB_", az.out1)


az.out1 <- gsub("WpSt_\\(R\\)_", "NA_WpStR_", az.out1)
az.out1 <- gsub("WpSt_\\(B\\)_", "NA_WpStB_", az.out1)

az.out1 <- gsub("VGS_", "NA_VGS_", az.out1)

az.out1 <- gsub("V_I_", "VI_", az.out1)
```

7.2.2 Einzelne Fehler bereinigen

```
az.out1 <- gsub("_ZR_\\(Ü\\)_", "_ZRÜ_", az.out1)
az.out1 <- gsub("_\\+_48", "", az.out1)
az.out1 <- gsub("u\\._25_", "", az.out1)
az.out1 <- gsub("_-_28_07", "", az.out1)



az.out1 <- gsub("-[0-9]*_", "_", az.out1)

az.out1 <- gsub("_\\(a\\)", "_a", az.out1)
```

7.2.3 Variable »zusatz_az« einfügen

```
indices <- grep("[0-9]*_[A-Za-zÜ]*_[0-9]*_[0-9]*_[a-z]*",
                az.out1,
                invert = TRUE)

values <- grep("[0-9]*_[A-Za-zÜ]*_[0-9]*_[0-9]*_[a-z]*",
               az.out1,
               invert = TRUE,
               value = TRUE)

az.out1[indices] <- paste0(values,
                  "_NA")
```

7.2.4 Finale Korrekturen

Bei der Entscheidung 1 BGs 29/2009 ist als einzige unter allen Entscheidungen das Eingangsjahr vierstellig und nicht zweistellig, auch im Text der Entscheidung selber. Ich gehe dennoch davon aus, das es sich hier um einen Schreibfehler handelt und nehme eine Korrektur vor.

```
az.out1 <- gsub("1_BGs_29_2009_NA",
                "1_BGs_29_09_NA",
                az.out1)
```

7.2.5 Strenge REGEX-Validierung des Aktenzeichens

```
regex.test1 <- grep("[0-9A-Za]+_[A-Za-zÜ]+_[0-9]+_[0-9]{2}_[A-Za-z]+",
                    az.out1,
                    invert = TRUE,
                    value = TRUE)
```

7.2.6 Ergebnis der REGEX-Validierung

```
print(regex.test1)
```

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

7.2.7 Skript stoppen falls REGEX-Validierung gescheitert

```
if (length(regex.test1) != 0){
    stop("REGEX VALIDIERUNG GESCHEITERT: AKTENZEICHEN ENTSPRECHEN NICHT DEM
    CODEBOOK-SCHEMA!")
    }
```

7.2.8 Aktenzeichen-Vektor in Download Table einfügen

```
dt.download$az <- az.out1
```

7.3 Spruchkörper bereinigen

7.3.1 Reguläre Substitutionen

Hinweis: An dieser Stelle wird eine mysteriöse Unterstrich-Variante mit einem regulären Unterstrich ersetzt. Es ist mir aktuell unklar um was für eine Art von Zeichen es sich handelt und wieso es in den Daten des Bundesgerichtshofs auftaucht. Weil die Code-Zeile zu einem Anzeigefehler in der LaTeX-Kompilierung führt, ist sie im Compilation Report nicht abgedruckt. Sie finden die Zeile im eigentlichen Source Code. Ich arbeite daran, die Anzeige zu vervollständigen.

```
spruch1 <- gsub("\\._Zivilsenat",
                "",
                spruch1)

spruch1 <- gsub("\\._Strafsenat",
                "",
                spruch1)

spruch1 <- gsub("Senat_für Anwaltssachen",
                "Anwaltsenat",
                spruch1)

spruch1 <- gsub("Senat_für Landwirtschaftssachen",
                "Landwirtschaftsenat",
                spruch1)

spruch1 <- gsub("Senat_für Notarsachen",
                "Notarsenat",
                spruch1)

spruch1 <- gsub("Dienstgericht des_Bundes",
                "DienstgerichtBund",
                spruch1)

spruch1 <- gsub("Senat_für Wirtschaftsprüfersachen",
                "Wirtschaftsprüfersenat",
                spruch1)

spruch1 <- gsub("Senat_für Steuerberater-_und Steuerbevollmächtigtensachen",
                "Steuerberatersenat",
                spruch1)

spruch1 <- gsub("Gemeinsamer Senat der obersten Gerichtshöfe des Bundes",
                "GemeinsamerSenatObersteGerichtshöfeBund",
                spruch1)

spruch1 <- gsub("Senat_für Patentanwaltssachen",
```

```
                "Patentanwaltsenat",
                spruch1)

spruch1 <- gsub("Großer_Senat für_Strafsachen",
                "GrosserStrafsenat",
                spruch1)

spruch1 <- gsub("Großer_Senat für_Zivilsachen",
                "GrosserZivilsenat",
                spruch1)

spruch1 <- gsub("-_Zivilsenat",
                "",
                spruch1)

spruch1 <- gsub("Vereinigte Große_Senate",
                "VereinigteGrosseSenate",
                spruch1)
```

7.3.2 Alle Spruchkörper anzeigen

```
print(unique(spruch1))
```

```
##  [1] "5"
##  [2] "2"
##  [3] "3"
##  [4] "VII"
##  [5] "4"
##  [6] "III"
##  [7] "V"
##  [8] "VIII"
##  [9] "XII"
## [10] "VI"
## [11] "X"
## [12] "XI"
## [13] "II"
## [14] "I"
## [15] "IX"
## [16] "1"
## [17] "IV"
## [18] "Anwaltsenat"
## [19] "Landwirtschaftsenat"
## [20] "Kartellsenat"
## [21] "Notarsenat"
## [22] "DienstgerichtBund"
## [23] "Wirtschaftsprüfersenat"
## [24] "Steuerberatersenat"
## [25] "GemeinsamerSenatObersteGerichtshöfeBund"
## [26] "Patentanwaltsenat"
## [27] "Ermittlungsrichter"
## [28] "GrosserStrafsenat"
## [29] "IXa"
```

```
## [30] "GrosserZivilsenat"
## [31] "Xa"
## [32] "VereinigteGrosseSenate"
## [33] "XIII"
## [34] "6"
```

7.3.3 Spruchkörper-Vektor in Download Table einfügen

```
dt.download$spruch <- spruch1
```

7.4 Bemerkungen bereinigen

```
dt.download$comment <- gsub("Leitsaetz",
                            "Leitsatz",
                            dt.download$comment)

dt.download$comment <- gsub("Leitsaz",
                            "Leitsatz",
                            dt.download$comment)

dt.download$comment <- gsub("Leitsazt",
                            "Leitsatz",
                            dt.download$comment)
```

7.5 Variable »leitsatz« erstellen

```
dt.download$leitsatz <- ifelse(grepl("Leitsatz",
                                     dt.download$comment),
                               "LE",
                               "NA")
```

7.6 Variable »name« erstellen

```
name <- sub("(.*)\r\n.*",
            "\\1",
            dt.download$comment)

name[grepl("Leitsatz|Pressemitteilung|Berichtigung",
           name,
           ignore.case = TRUE)] <- NA

name[grepl("^$", name)] <- NA

dt.download$name <- name
```

7.7 Variable »name_datei« erstellen

```
name_datei <- name

name_datei <- trimws(name_datei)

name_datei <- path_sanitize(name_datei,
                            "")

name_datei <- gsub(" +",
                   "_",
                   name_datei)

name_datei <- gsub("'",
                   "",
                   name_datei)

name_datei <- gsub(",",
                   "",
                   name_datei)

dt.download$name_datei <- name_datei
```

7.8 Dateinamen erstellen

```
filename <- paste("BGH",
                  dt.download$spruch,
                  dt.download$leitsatz,
                  dt.download$datum,
                  dt.download$az,
                  dt.download$name_datei,
                  sep="_")
```

7.9 Einzelkorrektur vornehmen

```
filename <- gsub("BGH_3_([NALE]{2})_NA_NA_AK_13_19_NA",
                  "BGH_3_\\1_2019-05-07_NA_AK_13_19_NA",
                  filename)
```

7.10 KollisionsID einfügen

7.10.1 Anzahl Duplikate

```
length(filename[duplicated(filename)])
```

```
## [1] 794
```

7.10.2 Kollisions-IDs vergeben

```
filenames1 <- make.unique(filename, sep = "-----")


indices <- grep("-----",
                filenames1,
                invert = TRUE)

values <- grep("-----",
               filenames1,
               invert = TRUE,
               value = TRUE)

filenames1[indices] <- paste0(values,
                              "_0")

filenames1 <- gsub("-----",
                   "_",
                   filenames1)
```

7.11 Zufällige Auswahl zur Prüfung anzeigen

```
filenames1[sample(length(filenames1), 50)]
```

```
##  [1] "BGH_1_NA_2001-11-07_1_StR_455_01_NA_NA_0"
##  [2] "BGH_1_NA_2002-10-08_1_StR_326_02_NA_NA_0"
##  [3] "BGH_IX_NA_2015-07-15_IX_ZR_121_14_NA_NA_0"
##  [4] "BGH_IX_NA_2012-01-12_IX_ZB_97_11_NA_NA_0"
##  [5] "BGH_XII_LE_2011-06-08_XII_ZR_17_09_NA_NA_0"
##  [6] "BGH_XI_LE_2008-05-27_XI_ZR_132_07_NA_NA_0"
##  [7] "BGH_VI_NA_2006-10-31_VI_ZR_280_05_NA_NA_0"
##  [8] "BGH_II_LE_2015-02-24_II_ZB_17_14_NA_NA_0"
##  [9] "BGH_5_NA_2017-11-29_5_StR_335_17_NA_NA_0"
## [10] "BGH_V_LE_2010-03-18_V_ZB_117_09_NA_NA_0"
## [11] "BGH_XI_NA_2013-01-17_XI_ZR_512_11_NA_NA_0"
## [12] "BGH_II_NA_2006-10-25_II_ZR_289_05_NA_NA_0"
## [13] "BGH_V_NA_2018-02-08_V_ZR_87_17_NA_NA_0"
## [14] "BGH_IX_NA_2004-06-29_IX_ZR_96_03_NA_NA_0"
## [15] "BGH_Landwirtschaftsenat_LE_2014-11-28_NA_BLw_4_13_NA_NA_0"
## [16] "BGH_XI_NA_2012-05-08_XI_ZR_317_10_NA_NA_0"
## [17] "BGH_VI_NA_2015-09-15_VI_ZR_306_15_NA_NA_0"
## [18] "BGH_5_NA_2007-09-13_5_StR_291_07_NA_NA_0"
## [19] "BGH_I_LE_2008-07-17_I_ZR_160_05_NA_Sammelaktion-für-Schoko-Riegel_0"
## [20] "BGH_1_NA_2000-01-12_1_StR_636_99_NA_NA_0"
## [21] "BGH_6_NA_2020-12-16_6_StR_224_20_NA_NA_0"
## [22] "BGH_IX_NA_2017-01-12_IX_ZA_30_16_NA_NA_0"
## [23] "BGH_Kartellsenat_NA_2017-04-24_NA_EnVR_35_15_NA_NA_0"
## [24] "BGH_I_LE_2005-01-27_I_ZR_146_02_NA_Sammelmitgliedschaft-III_0"
## [25] "BGH_V_LE_2018-06-08_V_ZR_195_17_NA_NA_0"
```

```
## [26] "BGH_XII_LE_2012-06-27_XII_ZB_492_11_NA_NA_0"
## [27] "BGH_XII_NA_2006-06-28_XII_ZR_82_04_NA_NA_0"
## [28] "BGH_X_LE_2005-02-01_X_ZB_27_04_NA_NA_0"
## [29] "BGH_III_NA_2017-03-30_III_ZB_44_16_NA_NA_0"
## [30] "BGH_VII_LE_2011-02-24_VII_ZB_108_08_NA_NA_0"
## [31] "BGH_2_NA_2020-03-10_2_StR_473_19_NA_NA_0"
## [32] "BGH_VIII_LE_2013-07-17_VIII_ZR_163_12_NA_NA_0"
## [33] "BGH_III_LE_2010-02-18_III_ZR_295_09_NA_NA_0"
## [34] "BGH_4_NA_2002-05-16_4_StR_105_02_NA_NA_0"
## [35] "BGH_4_NA_2018-09-13_4_StR_154_18_NA_NA_0"
## [36] "BGH_4_NA_2006-09-21_4_StR_323_06_NA_NA_0"
## [37] "BGH_IX_NA_2007-10-11_IX_ZB_126_04_NA_NA_0"
## [38] "BGH_2_NA_2017-09-21_2_StR_275_17_NA_NA_0"
## [39] "BGH_2_NA_2009-11-25_2_ARs_455_09_NA_NA_0"
## [40] "BGH_VI_LE_2008-06-03_VI_ZR_235_07_NA_NA_0"
## [41] "BGH_I_NA_2018-10-11_I_ZR_156_16_NA_NA_0"
## [42] "BGH_Notarsenat_NA_2007-07-23_NA_NotZ_88_07_NA_NA_0"
## [43] "BGH_II_LE_2002-02-25_II_ZR_374_00_NA_NA_0"
## [44] "BGH_3_NA_2000-11-24_3_StR_367_00_NA_NA_0"
## [45] "BGH_IX_LE_2008-07-17_IX_ZR_148_07_NA_NA_0"
## [46] "BGH_XI_NA_2014-09-16_XI_ZR_77_13_NA_NA_0"
## [47] "BGH_2_NA_2012-04-19_2_StR_5_12_NA_NA_0"
## [48] "BGH_4_NA_2019-03-26_4_StR_381_18_NA_NA_0"
## [49] "BGH_XI_NA_2009-11-10_XI_ZB_16_09_NA_NA_0"
## [50] "BGH_VII_LE_2014-10-16_VII_ZR_152_12_NA_NA_0"
```

7.12 PDF-Endung anfügen

```
filenames2 <- paste0(filenames1,
                     ".pdf")
```

7.13 Strenge REGEX-Validierung: Gesamter Dateiname

```
regex.test2 <-grep("BGH_.*_[NALE]{2}_[0-9]{4}-[0-9]{2}-[0-9]{2}_[A-Za-z0-9]*_[A-
    Za-zÜ]*_[0-9-]*_[0-9]{2}_[A-Za-z]*_.*_[NA0-9]*.pdf",
     filenames2,
     value = TRUE,
     invert = TRUE)
```

7.14 Ergebnis der REGEX-Validierung

```
print(regex.test2)
```

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

7.15 Skript stoppen falls REGEX-Validierung gescheitert

```
if (length(regex.test2) != 0){
    stop("REGEX VALIDIERUNG GESCHEITERT: DATEINAMEN ENTSPRECHEN NICHT DEM
    CODEBOOK-SCHEMA!")
    }
```

7.16 Vollen Dateinamen in Download Table einfügen

```
dt.download$filenames.final <- filenames2
```

8 Download der Entscheidungen im PDF-Format

8.1 [Debugging Modus] Reduzierung des Download-Umfangs

```
if (mode.debug == TRUE){
    dt.download <- dt.download[sample(.N,
                                      debug.sample)]
    }
```

8.2 Zeitstempel: Download Beginn

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

```
## [1] "2021-04-27 05:24:00 CEST"
```

8.3 Download durchführen

```
for (i in sample(dt.download[,.N])){
        tryCatch({download.file(url = dt.download$link[i],
                                destfile = dt.download$filenames.final[i])
        },
        error = function(cond) {
            return(NA)}
        )
        Sys.sleep(runif(1, 0, 0.1))
}
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2006&Seite=79&nr=35820&pos=2394&anz=3113&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2010&Seite=18&nr=54882&pos=559&anz=3648&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2019&Seite=88&nr=94224&pos=2667&anz=2956&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2006&Seite=71&nr=36182&pos=2153&anz=3113&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2017&Seite=74&nr=78157&pos=2238&anz=3127&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2010&Seite=1&nr=54791&pos=55&anz=3648&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2017&Seite=99&nr=77314&pos=2973&anz=3127&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2008&Seite=74&nr=44290&pos=2227&anz=3634&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
```

```
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2017&Seite=6&nr=80527&pos=189&anz=3127&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2017&Seite=23&nr=82176&pos=695&anz=3127&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2018&Seite=38&nr=87700&pos=1142&anz=2994&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2013&Seite=40&nr=65153&pos=1200&anz=3204&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2014&Seite=52&nr=68883&pos=1589&anz=3080&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2009&Seite=101&nr=47136&pos=3037&anz=3418&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2005&Seite=62&nr=33767&pos=1886&anz=3103&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2020&Seite=87&nr=105405&pos=2619&anz=3268&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2000&Seite=11&nr=23308&pos=346&anz=2206&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2018&Seite=46&nr=87730&pos=1383&anz=2994&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2021&Seite=7&nr=117202&pos=233&anz=810&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2017&Seite=24&nr=80574&pos=744&anz=3127&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2020&Seite=47&nr=109817&pos=1426&anz=3268&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2000&Seite=2&nr=21052&pos=87&anz=2206&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2008&Seite=92&nr=43687&pos=2760&anz=3634&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2018&Seite=26&nr=88699&pos=786&anz=2994&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2015&Seite=100&nr=70149&pos=3029&anz=3079&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2005&Seite=72&nr=32804&pos=2185&anz=3103&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2009&Seite=109&nr=46765&pos=3290&anz=3418&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2012&Seite=60&nr=60980&pos=1819&anz=3531&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
```

```
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2020&Seite=35&nr=110726&pos=1067&anz=3268&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2009&Seite=64&nr=48748&pos=1933&anz=3418&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2005&Seite=68&nr=32718&pos=2058&anz=3103&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2014&Seite=77&nr=67749&pos=2320&anz=3080&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2002&Seite=42&nr=21587&pos=1278&anz=2753&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2001&Seite=25&nr=18058&pos=751&anz=2408&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2015&Seite=37&nr=72208&pos=1121&anz=3079&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2004&Seite=64&nr=29684&pos=1921&anz=3087&Blank=1.pdf
    ':
## status was 'Stream error in the HTTP/2 framing layer'
```

```
## Warning in download.file(url = dt.download$link[i],
## destfile = dt.download$filenames.final[i]): URL 'https://
## juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?
## Gericht=bgh&Art=en&Datum=2018&Seite=7&nr=91281&pos=223&anz=2994&Blank=1.pdf':
## status was 'Stream error in the HTTP/2 framing layer'
```

8.4 Zeitstempel: Download Ende

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

```
## [1] "2021-04-27 12:00:25 CEST"
```

8.5 Dauer: Download

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

```
## Time difference of 6.607117 hours
```

8.6 [Debugging Modus] Löschen zufälliger Dateien

Dient dazu den Wiederholungsversuch zu testen.

```
if (mode.debug == TRUE){
    files.pdf <- list.files(pattern = "\\.pdf")
    unlink(sample(files.pdf, 5))
    }
```

8.7 Download: Zwischenergebnis

8.7.1 Anzahl herunterzuladender Dateien

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

```
## [1] 66626
```

8.7.2 Anzahl heruntergeladener Dateien

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

```
## [1] 66589
```

8.7.3 Fehlbetrag

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

```
## [1] 37
```

8.7.4 Fehlende Dateien

```
missing <- setdiff(dt.download$filenames.final, files.pdf)
print(missing)
```

```
##  [1] "BGH_2_NA_2000-12-13_2_StR_485_00_NA_NA_0.pdf"
##  [2] "BGH_IX_NA_2000-11-09_IX_ZR_403_99_NA_NA_0.pdf"
##  [3] "BGH_1_NA_2001-09-25_1_StR_270_01_NA_NA_0.pdf"
##  [4] "BGH_VII_LE_2002-07-11_VII_ZR_261_00_NA_NA_0.pdf"
##  [5] "BGH_X_LE_2004-05-18_X_ZB_7_04_NA_NA_0.pdf"
##  [6] "BGH_XII_LE_2005-05-25_XII_ZR_204_02_NA_NA_0.pdf"
##  [7] "BGH_Anwaltsenat_NA_2005-04-26_NA_AnwZB_98_04_NA_NA_0.pdf"
##  [8] "BGH_III_NA_2005-04-14_III_ZR_287_04_NA_NA_0.pdf"
##  [9] "BGH_IX_NA_2006-04-13_IX_ZR_173_02_NA_NA_0.pdf"
## [10] "BGH_5_NA_2006-03-21_5_StR_78_06_NA_NA_0.pdf"
## [11] "BGH_IX_NA_2008-05-29_IX_ZB_103_07_NA_NA_0.pdf"
## [12] "BGH_5_NA_2008-04-01_5_StR_80_08_NA_NA_0.pdf"
## [13] "BGH_Anwaltsenat_NA_2009-06-08_NA_AnwZB_23_08_NA_NA_0.pdf"
```

```
## [14] "BGH_5_NA_2009-02-11_5_StR_11_09_NA_NA_0.pdf"
## [15] "BGH_XI_NA_2009-01-20_XI_ZR_450_07_NA_NA_0.pdf"
## [16] "BGH_VII_NA_2010-12-20_VII_ZR_100_10_NA_NA_0.pdf"
## [17] "BGH_Kartellsenat_LE_2010-11-09_NA_EnVR_1_10_NA_Bahnstromfernleitungen_0.
    pdf"
## [18] "BGH_5_NA_2012-06-20_5_StR_134_12_NA_NA_0.pdf"
## [19] "BGH_4_NA_2013-08-15_4_StR_196_13_NA_NA_0.pdf"
## [20] "BGH_III_NA_2014-06-18_III_ZB_89_13_NA_NA_0.pdf"
## [21] "BGH_3_NA_2014-03-20_3_StR_353_13_NA_NA_0.pdf"
## [22] "BGH_X_NA_2015-08-25_X_ZB_6_14_NA_NA_0.pdf"
## [23] "BGH_II_NA_2015-01-13_II_ZR_312_13_NA_NA_0.pdf"
## [24] "BGH_4_NA_2017-12-05_4_StR_513_17_NA_NA_0.pdf"
## [25] "BGH_1_NA_2017-10-10_1_StR_496_16_NA_NA_0.pdf"
## [26] "BGH_2_NA_2017-10-04_2_StR_219_15_NA_NA_0.pdf"
## [27] "BGH_5_NA_2017-04-10_5_StR_493_16_NA_NA_0.pdf"
## [28] "BGH_III_NA_2017-01-19_III_ZR_296_16_NA_NA_0.pdf"
## [29] "BGH_1_NA_2018-12-04_1_StR_519_18_NA_NA_0.pdf"
## [30] "BGH_IX_NA_2018-10-01_IX_ZB_49_18_NA_NA_0.pdf"
## [31] "BGH_3_NA_2018-08-21_3_StR_298_18_NA_NA_0.pdf"
## [32] "BGH_II_NA_2018-07-17_II_ZB_5_18_NA_NA_0.pdf"
## [33] "BGH_II_NA_2019-02-05_II_ZB_10_18_NA_NA_0.pdf"
## [34] "BGH_2_NA_2020-09-09_2_StR_253_20_NA_NA_0.pdf"
## [35] "BGH_4_NA_2020-07-28_4_StR_97_20_NA_NA_0.pdf"
## [36] "BGH_Kartellsenat_NA_2020-03-24_NA_EnVR_45_18_NA_NA_0.pdf"
## [37] "BGH_III_NA_2021-02-25_III_ZB_72_20_NA_NA_0.pdf"
```

8.8 Wiederholungsversuch: Download

Download für fehlende Dokumente wiederholen.

```
if(N.missing > 0){

    dt.retry <- dt.download[filenames.final %in% missing]

    for (i in 1:dt.retry[,.N]){
        response <- GET(dt.retry$link[i])
        Sys.sleep(runif(1, 0.25, 0.75))
        if (response$headers$"content-type" == "application/pdf" & response$
    status_code == 200){
            tryCatch({download.file(url = dt.retry$link[i],
                                    destfile = dt.retry$filenames.final[i])
            },
            error=function(cond) {
                return(NA)}
            )
        }else{
            print(paste0(dt.retry$filenames.final[i], " : kein PDF vorhanden"))
        }
        Sys.sleep(runif(1, 2, 5))
    }
}
```

8.9 Download: Gesamtergebnis

8.9.1 Anzahl herunterzuladender Dateien

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

```
## [1] 66626
```

8.9.2 Anzahl heruntergeladener Dateien

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

```
## [1] 66626
```

8.9.3 Fehlbetrag

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

```
## [1] 0
```

8.9.4 Fehlende Dateien

```
missing <- setdiff(dt.download$filenames.final, files.pdf)
print(missing)
```

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

9 Text-Extraktion

9.1 Vektor der zu extrahierenden Dateien erstellen

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

9.2 Anzahl zu extrahierender Dateien

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

```
## [1] 66626
```

9.3 Seiten zählen: Funktion anzeigen

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

function(x, sum = FALSE, threads = detectCores()){

```
print(paste("Parallel processing using", threads, "threads."))

cl <- makeForkCluster(threads)
registerDoParallel(cl)

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

if (sum == TRUE){
    sum.out <- sum(pagenums)
    print(paste("Total number of pages:", sum.out))
    return(sum.out)
}else{
    return(pagenums)
}
```

}

9.4 Anzahl zu extrahierender Seiten

```
f.dopar.pagenums(files.pdf,
                 sum = TRUE,
                 threads = fullCores)
```

```
## [1] "Parallel processing using 16 threads."
## [1] "Total number of pages: 512781"
```

```
## [1] 512781
```

9.5 PDF extrahieren: Funktion anzeigen

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

function(x, threads = detectCores()){

```
begin.extract <- Sys.time()

print(paste("Parallel processing using", threads, "threads. Begin at", begin.
    extract))


cl <- makeForkCluster(threads)
registerDoParallel(cl)

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

result <- 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)
               }
stopCluster(cl)

end.extract <- Sys.time()
```

```
duration.extract <- end.extract - begin.extract

print(paste0("Processed ",
             length(result),
             " files. Runtime was ",
             round(duration.extract,
                   digits = 2),
             " ",
             attributes(duration.extract)$units,
             ". Ended at ",
            end.extract, "."))

return(result)
```

}

9.6 Text Extrahieren

```
result <- f.dopar.pdfextract(files.pdf,
                             threads = fullCores)
```

```
## [1] "Parallel processing using 16 threads. Begin at 2021-04-27 12:03:28"
## [1] "Processed 66626 files. Runtime was 1.54 mins. Ended at 2021-04-27
    12:05:01."
```

10 Korpus Erstellen

10.1 TXT-Dateien Einlesen

```
txt.bgh <- readtext("./*.txt",
                    docvarsfrom = "filenames",
                    docvarnames = c("gericht",
                                    "spruchkoerper_db",
                                    "leitsatz",
                                    "datum",
                                    "spruchkoerper_az",
                                    "registerzeichen",
                                    "eingangsnummer",
                                    "eingangsjahr_az",
                                    "zusatz_az",
                                    "name",
                                    "kollision"),
                    dvsep = "_",
                    encoding = "UTF-8")
```

10.2 In Data Table umwandeln

```
setDT(txt.bgh)
```

10.3 Durch Zeilenumbruch getrennte Wörter zusammenfügen

Durch Zeilenumbrüche getrennte Wörter stellen bei aus PDF-Dateien gewonnene Text-Korpora ein erhebliches Problem dar. Wörter werden dadurch in zwei sinnentleerte Tokens getrennt, statt ein einzelnes und sinnvolles Token zu bilden. Dieser Schritt entfernt die Bindestriche, den Zeilenumbruch und ggf. dazwischenliegende Leerzeichen.

10.3.1 Funktion anzeigen

```
print(f.hyphen.remove)
```

```
## function(text){
##     ## Examples: Ham-\nburg, Mei-\n   nungsäußerung
##     text.out <- gsub("([a-zöäüß])-[[:blank:]]*\n[[:blank:]]*([a-zöäüß])",
##                      "\\1\\2",
##                      text)
##     ## Examples: SARS-CoV-\n2
##     text.out <- gsub("([a-zA-ZöäüÖÄÜß])-[[:blank:]]*\n[[:blank:]]*([A-Z0-9ÖÄÜß
    ])",
##                      "\\1-\\2",
##                      text.out)
##     ## Example: hat-    2\nte, Unsterb-    6\nliche
```

```
##     text.out <- gsub("([a-zöäüß])-[[:blank:]]*[0-9]+[[:blank:]]*\n[[:blank:]]*
    ([a-zöäüß])",
##                      "\\1\\2",
##                      text.out)
##
##     ## Example: hat-  \n  2 te, Unsterb-  \n  6 liche
##     text.out <- gsub("([a-zöäüß])-[[:space:]]*[0-9]+[[:blank:]]*([a-zöäüß])",
##                      "\\1\\2",
##                      text.out)
##
##     return(text.out)
## }
```

10.3.2 Funktion ausführen

```
txt.bgh[, text := lapply(.(text), f.hyphen.remove)]
```

10.4 Variable »datum« als Datentyp »IDate« kennzeichnen

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

10.5 Variable »entscheidungsjahr« hinzufügen

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

10.6 Variable »eingangsjahr_iso« hinzufügen

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

10.7 Datensatz nach Datum sortieren

Die Erstellung der Variablen für Präsident:innen und Vize-Präsident:innen trifft die starke Annahme, dass eine aufsteigende Sortierung nach Datum besteht. Wäre das nicht der Fall, würden dort Fehler auftreten.

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

10.8 Variable »praesi« hinzufügen

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

10.8.1 Personaldaten einlesen

```
praesi <- fread("PVP-FCG_2021-04-08_GermanFederalCourts_Presidents.csv")
praesi <- praesi[court == "BGH", c(1:3, 5:6)]
```

10.8.2 Personaldaten anzeigen

```
kable(praesi,
      format = "latex",
      align = "r",
      booktabs = TRUE,
      longtable = TRUE) %>% kable_styling(latex_options = "repeat_header")
```

court	name_last	name_first	term_begin_date	term_end_date
BGH	Weinkauff	Hermann	1950-10-01	1960-03-31
BGH	Heusinger	Bruno	1960-04-01	1968-03-31
BGH	Fischer	Robert	1968-04-01	1977-09-30
BGH	Pfeiffer	Gerd	1977-10-01	1987-12-31
BGH	Odersky	Walter	1988-01-01	1996-07-31
BGH	Geiss	Karlmann	1996-08-01	2000-05-31
BGH	VACANCY-1	VACANCY-1	2000-06-01	2000-07-14
BGH	Hirsch	Günter	2000-07-15	2008-01-31
BGH	Tolksdorf	Klaus	2008-02-01	2014-01-31
BGH	VACANCY-2	VACANCY-2	2014-02-01	2014-06-30
BGH	Limberg	Bettina	2014-07-01	NA

10.8.3 Hypothethisches Amtsende für Präsident:in

Weil der/die aktuelle Präsident:in 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(term_end_date)]$term_end_date <- Sys.Date() + 365
```

10.8.4 Schleife vorbereiten

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

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

10.8.5 Vektor erstellen

```
for (i in seq_len(N)){
    praesi.N <- txt.bgh[datum >= praesi$term_begin_date[i] & datum <= praesi$term
    _end_date[i], .N]
    praesi.list[[i]] <- rep(praesi$name_last[i],
                            praesi.N)
}
```

10.8.6 Vektor einfügen

```
txt.bgh$praesi <- unlist(praesi.list)
```

10.9 Variable »v_praesi« hinzufügen

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

10.9.1 Personaldaten einlesen

```
vpraesi <- fread("PVP-FCG_2021-04-08_GermanFederalCourts_VicePresidents.csv")
vpraesi <- vpraesi[court == "BGH", c(1:3, 5:6)]
```

10.9.2 Personaldaten anzeigen

```
kable(vpraesi,
      format = "latex",
      align = "r",
      booktabs = TRUE,
      longtable = TRUE) %>% kable_styling(latex_options = "repeat_header")
```

court	name_last	name_first	term_begin_date	term_end_date
BGH	Glanzmann	Roderich	1965-05-17	1972-04-30
BGH	VACANCY-1	VACANCY-1	1972-05-01	1972-05-22
BGH	Hauß	Fritz	1972-05-23	1976-10-31
BGH	VACANCY-2	VACANCY-2	1976-11-01	1976-11-02

(continued)

court	name_last	name_first	term_begin_date	term_end_date
BGH	Pfeiffer	Gerd	1976-11-03	1977-09-30
BGH	Stimpel	Walter	1977-10-01	1985-11-30
BGH	VACANCY-3	VACANCY-3	1985-12-01	1985-12-01
BGH	Thumm	Ludwig	1985-12-02	1988-04-30
BGH	Salger	Hannskarl	1988-05-01	1994-11-30
BGH	Hagen	Horst	1994-12-01	1999-02-28
BGH	Jähnke	Burkhard	1999-03-01	2002-05-31
BGH	Wenzel	Joachim	2002-06-01	2005-06-30
BGH	Müller	Gerda	2005-07-01	2009-06-30
BGH	Schlick	Wolfgang	2009-07-01	2015-07-31
BGH	VACANCY-4	VACANCY-4	2015-08-01	2016-12-01
BGH	Ellenberger	Jürgen	2016-12-02	NA

10.9.3 Hypothethisches Amtsende für Vize-Präsident:in

Weil der/die aktuelle Vize-Präsident:in 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(term_end_date)]$term_end_date <- Sys.Date() + 365
```

10.9.4 Schleife vorbereiten

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

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

10.9.5 Vektor erstellen

```
for (i in seq_len(N)){
    vpraesi.N <- txt.bgh[datum >= vpraesi$term_begin_date[i] & datum <= vpraesi$
    term_end_date[i], .N]
    vpraesi.list[[i]] <- rep(vpraesi$name_last[i],
                             vpraesi.N)
}
```

10.9.6 Vektor einfügen

```
txt.bgh$v_praesi <- unlist(vpraesi.list)
```

10.10 Variable »verfahrensart« hinzufügen

Die Registerzeichen werden an dieser Stelle mit ihren detaillierten Bedeutungen aus dem folgenden Datensatz abgeglichen: »Seán Fobbe (2021). Aktenzeichen der Bundesrepublik Deutschland (AZ-BRD). Version 1.0.1. Zenodo. DOI: 10.5281/zenodo.4569564.« Das Ergebnis des Abgleichs wird in der Variable »verfahrensart« in den Datensatz eingefügt.

10.10.1 Datensatz einlesen

```
az.source <- fread("AZ-BRD_1-0-1_DE_Registerzeichen_Datensatz.csv")
```

10.10.2 Datensatz auf relevante Daten reduzieren

```
az.bgh <- az.source[stelle == "BGH" & position == "hauptzeichen"]
```

10.10.3 Indizes bestimmen

```
targetindices <- match(txt.bgh$registerzeichen,
                       az.bgh$zeichen_code)
```

10.10.4 Vektor der Verfahrensarten erstellen und einfügen

```
txt.bgh$verfahrensart <- az.bgh$bedeutung[targetindices]
```

10.11 Variable »aktenzeichen« hinzufügen

```
txt.bgh$aktenzeichen <- paste0(txt.bgh$spruchkoerper_az,
                               " ",
                               mgsub(txt.bgh$registerzeichen,
                                     az.bgh$zeichen_code,
                                     az.bgh$zeichen_original),
                               " ",
                               txt.bgh$eingangsnummer,
                               "/",
                               txt.bgh$eingangsjahr_az)


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

10.12 Variable »entscheidung_typ« hinzufügen

10.12.1 Entscheidungen Parsen

```
matches <- regexpr("BESCHLUSS|URTEIL|VERFÜGUNG",
                   txt.bgh$text,
                   ignore.case = TRUE)
```

10.12.2 Indizes bestimmen

```
matches.logical <- ifelse(matches > 0,
                          TRUE,
                          FALSE)

matches.indices <- which(matches.logical)
```

10.12.3 Leeren Vektor erstellen

```
entscheidung_typ <- rep(NA,
                        txt.bgh[,.N])
```

10.12.4 Typen bei Indizes platzieren

```
entscheidung_typ[matches.indices] <- regmatches(txt.bgh$text,
                                                matches)
```

10.12.5 Typen auf Kurzform reduzieren

```
entscheidung_typ  <- gsub("URTEIL",
                          "U",
                          entscheidung_typ,
                          ignore.case = TRUE)

entscheidung_typ  <- gsub("BESCHLUSS",
                          "B",
                          entscheidung_typ,
                          ignore.case = TRUE)


entscheidung_typ  <- gsub("VERFÜGUNG",
                          "V",
                          entscheidung_typ,
                          ignore.case = TRUE)
```

10.12.6 Vektor in Datensatz einfügen

```
txt.bgh$entscheidung_typ <- entscheidung_typ
```

10.13 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.

ACHTUNG: diese ECLIs sind experimentell. Der BGH vergibt offiziell ECLI-Identifikatoren nur für Entscheidungen ab 2016. Ausserdem steht die originale Kollisions-ID nicht zur Verfügung. Alle Entscheidungen mit einer Kollisions-ID größer 0 und ihre korrespondierenden Entscheidungen mit der ID 0 haben potenziell syntaktisch korrekte, aber semantisch fehlerhafte ECLIs.

10.13.1 Formatieren der Registerzeichen für ECLI

```
ecli.registerzeichen <- az.bgh$zeichen_original[targetindices]

ecli.registerzeichen <- gsub("\\(",
                             "\\.",
                             ecli.registerzeichen)

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

ecli.registerzeichen <- toupper(ecli.registerzeichen)
```

10.13.2 Erstellen der ECLI-Ordinalzahl

```
ecli.ordinalzahl <- paste0(format(txt.bgh$datum,
                                  "%d%m%y"),
                           txt.bgh$entscheidung_typ,
                           ifelse(is.na(txt.bgh$spruchkoerper_az),
                                  "",
                                  txt.bgh$spruchkoerper_az),
                           ecli.registerzeichen,
                           txt.bgh$eingangsnummer,
                           ".",
                           formatC(txt.bgh$eingangsjahr_az,
                                   width = 2,
                                   flag = "0"),
                           ".",
                           txt.bgh$kollision)
```

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

10.13.3 Vollständige ECLI erstellen

```
txt.bgh$ecli <- paste0("ECLI:DE:BGH:",
                       txt.bgh$entscheidungsjahr,
                       ":",
                       ecli.ordinalzahl)
```

10.13.4 Zufällige ECLI-Beispiele zur manuellen Nachprüfung

```
sample(txt.bgh$ecli,
       20)
```

```
##  [1] "ECLI:DE:BGH:2012:151112U2STR190.12.0"
##  [2] "ECLI:DE:BGH:2001:080801U1STR139.01.0"
##  [3] "ECLI:DE:BGH:2006:141206UIZR34.04.0"
##  [4] "ECLI:DE:BGH:2007:230707BNOTZ7.07.0"
##  [5] "ECLI:DE:BGH:2009:280409BIXZR112.08.0"
##  [6] "ECLI:DE:BGH:2007:150507BXZR20.05.0"
##  [7] "ECLI:DE:BGH:2015:140115UIVZR43.14.0"
##  [8] "ECLI:DE:BGH:2011:270411U2STR631.10.0"
##  [9] "ECLI:DE:BGH:2016:271016BIXZA17.16.0"
## [10] "ECLI:DE:BGH:2016:301116BXIIZB167.15.0"
## [11] "ECLI:DE:BGH:2008:290408BKVR28.07.0"
## [12] "ECLI:DE:BGH:2003:130203U3STR440.02.0"
## [13] "ECLI:DE:BGH:2011:271011BVIIZB88.10.0"
## [14] "ECLI:DE:BGH:2004:290404BIIIZB72.03.0"
## [15] "ECLI:DE:BGH:2012:220312BIIIZR135.11.0"
## [16] "ECLI:DE:BGH:2009:290409UVIIIZR142.08.0"
## [17] "ECLI:DE:BGH:2010:040810B5AR.VS22.10.0"
## [18] "ECLI:DE:BGH:2015:240315B4STR24.15.0"
## [19] "ECLI:DE:BGH:2017:190917U5STR593.16.0"
## [20] "ECLI:DE:BGH:2011:201211BVIZB25.11.0"
```

10.14 Variable »bemerkung« hinzufügen

```
dt.names.txt <- gsub("\\.pdf",
                     "\\.txt",
                     dt.download$filenames.final)

targetindices <- match(txt.bgh$doc_id,
                       dt.names.txt)

txt.bgh$bemerkung <- dt.download[targetindices]$comment
```

10.15 Variable »berichtigung« hinzufügen

```
txt.bgh$berichtigung <- ifelse(grepl("Berichtigung",
                                     txt.bgh$bemerkung,
                                     ignore.case = TRUE),
                               "Berichtigung",
                               NA)
```

10.16 Variable »leitsatz« hinzufügen

```
txt.bgh$leitsatz <- dt.download[targetindices]$leitsatz
```

10.17 Variable »name« hinzufügen

```
txt.bgh$name <- dt.download[targetindices]$name
```

10.18 Variable »doi_concept« hinzufügen

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

10.19 Variable »doi_version« hinzufügen

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

10.20 Variable »version« hinzufügen

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

10.21 Variable »lizenz« hinzufügen

```
txt.bgh$lizenz <- as.character(rep(license,
                                   txt.bgh[,.N]))
```

10.22 Entfernen von Dokumenten ohne Typ/Name/Berichtigung

Dokumente ohne Typ, Name oder Berichtigung sind fast immer Platzhalter-Dokumente, die auf den Push-Service der Universität des Saarlandes hinweisen. Diese werden am Ende des Variablen-Abschnitts aus der CSV-Datei entfernt und die PDF- und TXT-Dateien zur weiteren Prüfung separat gespeichert.

10.22.1 Platzhalter-Dokumente definieren

Dokumente ohne Typ, Name und Berichtigung sind fast immer Platzhalter-Dokumente, die keine Begründung enthalten und/oder auf den Push-Service der Universität des Saarlandes hinweisen. Diese werden am Ende dieses Abschnitts entfernt.

```
placeholder.txt <- txt.bgh[is.na(entscheidung_typ) == TRUE & is.na(name) == TRUE
    & is.na(berichtigung) == TRUE]$doc_id
```

10.22.2 Einzelkorrektur

Das folgende Dokument ist nach Extraktion ein leeres Text-Dokument, im originalen PDF aber ein funktionaler Scan. Es wird temporär vom Datensatz ausgeschlossen damit keine Fehler in der Zählung linguistischer Kennzahlen auftreten. In Zukunft wird ein OCR-Modul hierfür eingerichtet.

```
if (file.exists("BGH_I_LE_2006-07-13_I_ZR_241_03_NA_Kontaktanzeigen_0.txt") ==
    TRUE){

placeholder.txt <- c(placeholder.txt,
                     "BGH_I_LE_2006-07-13_I_ZR_241_03_NA_Kontaktanzeigen_0.txt")

}
```

10.22.3 Dokumente ohne Typ, Name und Berichtigung anzeigen

```
print(placeholder.txt)
```

```
##   [1] "BGH_1_NA_2000-01-26_1_StR_616_99_NA_NA_0.txt"
##   [2] "BGH_4_NA_2000-02-15_4_StR_507_99_NA_NA_0.txt"
##   [3] "BGH_GemeinsamerSenatObersteGerichtshöfeBund_LE_2000-04-05_NA_GMSOGB
    _1_98_NA_NA_0.txt"
##   [4] "BGH_4_NA_2000-08-03_4_StR_280_00_NA_NA_0.txt"
##   [5] "BGH_4_NA_2000-08-03_4_StR_302_00_NA_NA_0.txt"
##   [6] "BGH_4_NA_2000-08-10_4_StR_188_00_NA_NA_0.txt"
##   [7] "BGH_2_NA_2000-08-25_2_StR_295_00_NA_NA_0.txt"
##   [8] "BGH_4_NA_2000-12-21_4_StR_431_00_NA_NA_0.txt"
##   [9] "BGH_5_NA_2000-12-23_5_StR_377_00_NA_NA_0.txt"
##  [10] "BGH_3_NA_2001-02-22_3_StR_587_00_NA_NA_0.txt"
##  [11] "BGH_1_NA_2001-05-10_1_StR_505_00_NA_NA_0.txt"
```

```
##  [12] "BGH_4_NA_2001-07-10_4_StR_175_01_NA_NA_0.txt"
##  [13] "BGH_5_NA_2001-12-11_5_StR_372_01_NA_NA_0.txt"
##  [14] "BGH_1_NA_2002-02-05_1_StR_403_01_NA_NA_0.txt"
##  [15] "BGH_1_NA_2002-02-05_1_StR_570_01_NA_NA_0.txt"
##  [16] "BGH_IX_NA_2002-03-21_IX_ZB_57_02_NA_NA_0.txt"
##  [17] "BGH_2_NA_2002-06-26_2_StR_175_02_NA_NA_0.txt"
##  [18] "BGH_1_NA_2002-09-10_1_StR_318_02_NA_NA_0.txt"
##  [19] "BGH_4_NA_2002-11-05_4_StR_312_02_NA_NA_0.txt"
##  [20] "BGH_5_NA_2002-11-06_5_StR_421_02_NA_NA_0.txt"
##  [21] "BGH_2_NA_2002-12-11_2_StR_400_02_NA_NA_0.txt"
##  [22] "BGH_1_NA_2003-01-14_1_StR_502_02_NA_NA_0.txt"
##  [23] "BGH_1_NA_2003-01-29_1_StR_494_02_NA_NA_0.txt"
##  [24] "BGH_1_NA_2003-01-29_1_StR_519_02_NA_NA_0.txt"
##  [25] "BGH_3_NA_2003-01-30_3_StR_428_02_NA_NA_0.txt"
##  [26] "BGH_1_NA_2003-03-02_1_StR_25_04_NA_NA_0.txt"
##  [27] "BGH_I_LE_2003-05-08_I_ZR_287_02_NA_NA_0.txt"
##  [28] "BGH_5_NA_2003-05-20_5_StR_66_03_NA_NA_0.txt"
##  [29] "BGH_IV_NA_2003-06-25_IV_ZB_32_02_NA_NA_0.txt"
##  [30] "BGH_4_NA_2003-06-26_4_StR_168_03_NA_NA_0.txt"
##  [31] "BGH_I_LE_2003-06-26_I_ZR_269_00_NA_NA_0.txt"
##  [32] "BGH_1_NA_2003-09-09_1_StR_278_03_NA_NA_0.txt"
##  [33] "BGH_3_NA_2003-10-08_3_StR_342_03_NA_NA_0.txt"
##  [34] "BGH_4_NA_2003-12-16_4_StR_482_03_NA_NA_0.txt"
##  [35] "BGH_1_NA_2004-02-04_1_StR_545_03_NA_NA_0.txt"
##  [36] "BGH_2_NA_2004-02-13_2_StR_489_03_NA_NA_0.txt"
##  [37] "BGH_5_NA_2004-03-31_5_StR_498_03_NA_NA_0.txt"
##  [38] "BGH_XII_LE_2004-04-07_XII_ZB_79_04_NA_NA_0.txt"
##  [39] "BGH_2_NA_2004-05-21_2_StR_35_04_NA_NA_0.txt"
##  [40] "BGH_4_NA_2004-05-27_4_StR_564_03_NA_NA_0.txt"
##  [41] "BGH_2_NA_2004-07-02_2_StR_32_04_NA_NA_0.txt"
##  [42] "BGH_X_LE_2004-07-06_X_ZR_171_02_NA_NA_1.txt"
##  [43] "BGH_5_NA_2004-09-30_5_StR_266_04_NA_NA_0.txt"
##  [44] "BGH_3_NA_2005-01-25_3_StR_486_04_NA_NA_0.txt"
##  [45] "BGH_2_NA_2005-03-01_2_StR_18_05_NA_NA_0.txt"
##  [46] "BGH_1_NA_2005-03-30_1_StR_537_04_NA_NA_0.txt"
##  [47] "BGH_1_NA_2005-09-08_1_StR_323_05_NA_NA_0.txt"
##  [48] "BGH_1_NA_2005-10-26_1_StR_412_05_NA_NA_0.txt"
##  [49] "BGH_1_NA_2006-02-07_1_StR_555_05_NA_NA_0.txt"
##  [50] "BGH_1_NA_2006-05-09_1_StR_142_06_NA_NA_0.txt"
##  [51] "BGH_4_NA_2006-05-11_4_StR_10_06_NA_NA_0.txt"
##  [52] "BGH_4_NA_2006-05-11_4_StR_90_06_NA_NA_0.txt"
##  [53] "BGH_5_NA_2006-10-10_5_StR_212_06_NA_NA_0.txt"
##  [54] "BGH_4_NA_2006-10-19_4_StR_359_06_NA_NA_0.txt"
##  [55] "BGH_X_LE_2006-11-23_X_ZR_16_05_NA_NA_0.txt"
##  [56] "BGH_2_NA_2006-12-13_2_StR_403_06_NA_NA_0.txt"
##  [57] "BGH_1_NA_2007-01-17_1_StR_539_06_NA_NA_0.txt"
##  [58] "BGH_3_NA_2007-04-03_3_StR_107_07_NA_NA_0.txt"
##  [59] "BGH_2_NA_2007-04-13_2_StR_519_06_NA_NA_0.txt"
##  [60] "BGH_1_NA_2007-04-25_1_StR_130_07_NA_NA_0.txt"
##  [61] "BGH_1_NA_2007-04-25_1_StR_152_07_NA_NA_0.txt"
##  [62] "BGH_3_NA_2007-05-02_3_StR_145_07_NA_NA_0.txt"
##  [63] "BGH_3_NA_2007-10-01_3_StR_379_07_NA_NA_0.txt"
##  [64] "BGH_2_NA_2007-10-24_2_StR_421_07_NA_NA_0.txt"
##  [65] "BGH_2_NA_2008-01-09_2_StR_504_07_NA_NA_0.txt"
##  [66] "BGH_2_NA_2008-01-16_2_StR_542_07_NA_NA_0.txt"
##  [67] "BGH_5_NA_2008-01-23_5_StR_606_07_NA_NA_0.txt"
```

```
##  [68] "BGH_2_NA_2008-02-01_2_StR_539_07_NA_NA_0.txt"
##  [69] "BGH_2_NA_2008-02-27_2_StR_520_07_NA_NA_0.txt"
##  [70] "BGH_5_NA_2008-03-03_5_StR_9_08_NA_NA_0.txt"
##  [71] "BGH_4_NA_2008-03-04_4_StR_589_07_NA_NA_0.txt"
##  [72] "BGH_X_LE_2008-04-22_X_ZR_76_07_NA_NA_0.txt"
##  [73] "BGH_2_NA_2008-06-11_2_StR_30_08_NA_NA_0.txt"
##  [74] "BGH_1_NA_2008-07-09_1_StR_316_08_NA_NA_0.txt"
##  [75] "BGH_1_NA_2008-07-16_1_StR_259_08_NA_NA_0.txt"
##  [76] "BGH_3_NA_2008-08-12_3_StR_110_08_NA_NA_0.txt"
##  [77] "BGH_1_NA_2008-08-26_1_StR_391_08_NA_NA_0.txt"
##  [78] "BGH_1_NA_2008-08-26_1_StR_398_08_NA_NA_0.txt"
##  [79] "BGH_1_NA_2008-08-27_1_StR_433_08_NA_NA_0.txt"
##  [80] "BGH_1_NA_2008-09-09_1_StR_459_08_NA_NA_0.txt"
##  [81] "BGH_2_NA_2008-09-10_2_StR_320_08_NA_NA_0.txt"
##  [82] "BGH_2_NA_2008-10-15_2_StR_430_08_NA_NA_0.txt"
##  [83] "BGH_2_NA_2008-10-31_2_StR_378_08_NA_NA_0.txt"
##  [84] "BGH_3_NA_2008-11-25_3_StR_444_08_NA_NA_0.txt"
##  [85] "BGH_2_NA_2008-11-28_2_StR_471_08_NA_NA_0.txt"
##  [86] "BGH_5_NA_2009-01-07_5_StR_539_08_NA_NA_0.txt"
##  [87] "BGH_3_NA_2009-02-10_3_StR_546_08_NA_NA_0.txt"
##  [88] "BGH_2_NA_2009-02-11_2_StR_558_08_NA_NA_0.txt"
##  [89] "BGH_3_NA_2009-02-17_3_StR_37_09_NA_NA_0.txt"
##  [90] "BGH_3_NA_2009-03-17_3_StR_18_09_NA_NA_0.txt"
##  [91] "BGH_1_NA_2009-04-28_1_StR_148_09_NA_NA_0.txt"
##  [92] "BGH_1_NA_2009-04-28_1_StR_176_09_NA_NA_0.txt"
##  [93] "BGH_5_NA_2009-05-05_5_StR_118_09_NA_NA_0.txt"
##  [94] "BGH_5_NA_2009-05-06_5_StR_131_09_NA_NA_0.txt"
##  [95] "BGH_2_NA_2009-05-13_2_StR_142_09_NA_NA_0.txt"
##  [96] "BGH_5_NA_2009-05-26_5_StR_139_09_NA_NA_0.txt"
##  [97] "BGH_5_NA_2009-05-26_5_StR_180_09_NA_NA_0.txt"
##  [98] "BGH_2_NA_2009-06-03_2_StR_162_09_NA_NA_0.txt"
##  [99] "BGH_5_NA_2009-06-09_5_StR_190_09_NA_NA_0.txt"
## [100] "BGH_2_NA_2009-06-10_2_StR_145_09_NA_NA_0.txt"
## [101] "BGH_2_NA_2009-06-17_2_StR_190_09_NA_NA_0.txt"
## [102] "BGH_4_NA_2009-07-28_4_StR_57_09_NA_NA_0.txt"
## [103] "BGH_1_NA_2009-08-04_1_StR_349_09_NA_NA_0.txt"
## [104] "BGH_1_NA_2009-08-05_1_StR_366_09_NA_NA_0.txt"
## [105] "BGH_5_NA_2009-09-01_5_StR_309_09_NA_NA_0.txt"
## [106] "BGH_3_NA_2009-09-08_3_StR_356_09_NA_NA_0.txt"
## [107] "BGH_3_NA_2009-09-22_3_StR_203_09_NA_NA_0.txt"
## [108] "BGH_3_NA_2009-09-22_3_StR_262_09_NA_NA_0.txt"
## [109] "BGH_4_NA_2009-09-22_4_StR_657_08_NA_NA_0.txt"
## [110] "BGH_2_NA_2009-09-23_2_StR_293_09_NA_NA_0.txt"
## [111] "BGH_3_NA_2009-10-06_3_StR_375_09_NA_NA_0.txt"
## [112] "BGH_2_NA_2009-10-21_2_StR_287_09_NA_NA_0.txt"
## [113] "BGH_2_NA_2009-10-23_2_StR_317_09_NA_NA_0.txt"
## [114] "BGH_5_NA_2009-12-08_5_StR_441_09_NA_NA_0.txt"
## [115] "BGH_2_NA_2010-01-27_2_StR_555_09_NA_NA_0.txt"
## [116] "BGH_GemeinsamerSenatObersteGerichtshöfeBund_NA_2010-03-08_NA_GMSOGB
    _2_07_NA_NA_0.txt"
## [117] "BGH_2_NA_2010-03-24_2_StR_579_09_NA_NA_0.txt"
## [118] "BGH_5_NA_2010-03-24_5_StR_29_10_NA_NA_0.txt"
## [119] "BGH_5_NA_2010-03-24_5_StR_31_10_NA_NA_0.txt"
## [120] "BGH_1_NA_2010-05-11_1_StR_103_10_NA_NA_0.txt"
## [121] "BGH_1_NA_2010-05-11_1_StR_188_10_NA_NA_0.txt"
## [122] "BGH_3_NA_2010-05-18_3_StR_149_10_NA_NA_0.txt"
```

```
## [123] "BGH_5_NA_2010-06-14_5_StR_207_10_NA_NA_0.txt"
## [124] "BGH_II_NA_2010-06-21_II_ZR_166_09_NA_NA_0.txt"
## [125] "BGH_2_NA_2010-07-14_2_StR_240_10_NA_NA_0.txt"
## [126] "BGH_1_NA_2010-08-24_1_StR_414_10_NA_NA_0.txt"
## [127] "BGH_5_NA_2010-08-31_5_StR_321_10_NA_NA_0.txt"
## [128] "BGH_2_NA_2010-09-01_2_StR_347_10_NA_NA_0.txt"
## [129] "BGH_2_NA_2010-09-08_2_StR_316_10_NA_NA_0.txt"
## [130] "BGH_2_NA_2010-09-08_2_StR_389_10_NA_NA_0.txt"
## [131] "BGH_1_NA_2010-09-09_1_StR_376_10_NA_NA_0.txt"
## [132] "BGH_GemeinsamerSenatObersteGerichtshöfeBund_LE_2010-09-27_NA_GMSOGB
    _1_09_NA_NA_0.txt"
## [133] "BGH_5_NA_2010-09-29_5_StR_280_10_NA_NA_0.txt"
## [134] "BGH_3_NA_2010-10-12_3_StR_330_10_NA_NA_0.txt"
## [135] "BGH_1_NA_2010-11-03_1_StR_432_10_NA_NA_0.txt"
## [136] "BGH_5_NA_2010-11-23_5_StR_380_10_NA_NA_0.txt"
## [137] "BGH_2_NA_2010-12-15_2_StR_196_10_NA_NA_0.txt"
## [138] "BGH_5_NA_2011-01-12_5_StR_538_10_NA_NA_0.txt"
## [139] "BGH_1_NA_2011-01-19_1_StR_569_10_NA_NA_0.txt"
## [140] "BGH_5_NA_2011-02-21_5_StR_27_11_NA_NA_0.txt"
## [141] "BGH_5_NA_2011-02-22_5_StR_11_11_NA_NA_0.txt"
## [142] "BGH_5_NA_2011-02-24_5_StR_534_10_NA_NA_0.txt"
## [143] "BGH_1_NA_2011-03-17_1_StR_11_11_NA_NA_0.txt"
## [144] "BGH_1_NA_2011-05-03_1_StR_140_11_NA_NA_0.txt"
## [145] "BGH_5_NA_2011-06-08_5_StR_181_11_NA_NA_0.txt"
## [146] "BGH_1_NA_2011-07-13_1_StR_692_10_NA_NA_0.txt"
## [147] "BGH_2_NA_2011-09-22_2_StR_391_11_NA_NA_0.txt"
## [148] "BGH_5_NA_2011-10-11_5_StR_390_11_NA_NA_0.txt"
## [149] "BGH_2_NA_2011-11-23_2_StR_292_11_NA_NA_0.txt"
## [150] "BGH_5_NA_2011-11-30_5_StR_470_11_NA_NA_0.txt"
## [151] "BGH_2_NA_2011-12-13_2_StR_521_11_NA_NA_0.txt"
## [152] "BGH_1_NA_2011-12-21_1_StR_400_11_NA_NA_0.txt"
## [153] "BGH_5_NA_2012-01-10_5_StR_490_11_NA_NA_0.txt"
## [154] "BGH_5_NA_2012-01-11_5_StR_491_11_NA_NA_0.txt"
## [155] "BGH_2_NA_2012-02-09_2_StR_534_11_NA_NA_0.txt"
## [156] "BGH_II_NA_2012-03-06_II_ZR_89_11_NA_NA_0.txt"
## [157] "BGH_XII_NA_2012-03-07_XII_ZR_23_11_NA_NA_0.txt"
## [158] "BGH_2_NA_2012-03-15_2_StR_436_11_NA_NA_0.txt"
## [159] "BGH_IV_NA_2012-03-21_IV_ZR_50_10_NA_NA_0.txt"
## [160] "BGH_3_NA_2012-03-27_3_StR_38_12_NA_NA_0.txt"
## [161] "BGH_5_NA_2012-03-28_5_StR_81_12_NA_NA_0.txt"
## [162] "BGH_5_NA_2012-05-07_5_StR_164_12_NA_NA_0.txt"
## [163] "BGH_2_NA_2012-05-16_2_StR_107_12_NA_NA_0.txt"
## [164] "BGH_5_NA_2012-06-05_5_StR_235_12_NA_NA_0.txt"
## [165] "BGH_5_NA_2012-06-19_5_StR_257_12_NA_NA_0.txt"
## [166] "BGH_2_NA_2012-07-10_2_StR_26_12_NA_NA_0.txt"
## [167] "BGH_1_NA_2012-07-24_1_StR_221_12_NA_NA_0.txt"
## [168] "BGH_5_NA_2012-08-16_5_StR_321_12_NA_NA_0.txt"
## [169] "BGH_GemeinsamerSenatObersteGerichtshöfeBund_LE_2012-08-22_NA_GMSOGB
    _1_10_NA_NA_0.txt"
## [170] "BGH_1_NA_2012-08-23_1_StR_356_12_NA_NA_0.txt"
## [171] "BGH_5_NA_2012-09-13_5_StR_244_12_NA_NA_0.txt"
## [172] "BGH_1_NA_2012-09-25_1_StR_412_12_NA_NA_0.txt"
## [173] "BGH_3_NA_2012-10-02_3_StR_202_12_NA_NA_0.txt"
## [174] "BGH_1_NA_2012-10-24_1_StR_483_12_NA_NA_0.txt"
## [175] "BGH_5_NA_2012-11-06_5_StR_473_12_NA_NA_0.txt"
## [176] "BGH_5_NA_2012-11-27_5_StR_559_12_NA_NA_0.txt"
```

```
## [177] "BGH_1_NA_2012-12-05_1_StR_546_12_NA_NA_0.txt"
## [178] "BGH_1_NA_2013-01-22_1_StR_232_12_NA_NA_0.txt"
## [179] "BGH_1_NA_2013-01-23_1_StR_596_12_NA_NA_0.txt"
## [180] "BGH_1_NA_2013-02-19_1_StR_275_12_NA_NA_0.txt"
## [181] "BGH_1_NA_2013-03-05_1_StR_37_13_NA_NA_0.txt"
## [182] "BGH_5_NA_2013-03-19_5_StR_80_13_NA_NA_0.txt"
## [183] "BGH_VI_NA_2013-03-19_VI_ZR_106_12_NA_NA_0.txt"
## [184] "BGH_VI_NA_2013-03-19_VI_ZR_107_12_NA_NA_0.txt"
## [185] "BGH_VI_NA_2013-03-19_VI_ZR_108_12_NA_NA_0.txt"
## [186] "BGH_5_NA_2013-03-20_5_StR_28_13_NA_NA_0.txt"
## [187] "BGH_1_NA_2013-05-02_1_StR_96_13_NA_NA_0.txt"
## [188] "BGH_4_NA_2013-07-16_4_StR_66_13_NA_NA_0.txt"
## [189] "BGH_5_NA_2013-09-03_5_StR_187_13_NA_NA_0.txt"
## [190] "BGH_3_NA_2013-09-17_3_StR_227_13_NA_NA_0.txt"
## [191] "BGH_3_NA_2013-11-14_3_StR_92_13_NA_NA_0.txt"
## [192] "BGH_5_NA_2014-01-22_5_StR_561_13_NA_NA_0.txt"
## [193] "BGH_1_NA_2014-01-30_1_StR_616_13_NA_NA_0.txt"
## [194] "BGH_5_NA_2014-03-10_5_StR_51_14_NA_NA_0.txt"
## [195] "BGH_5_NA_2014-04-08_5_StR_97_14_NA_NA_0.txt"
## [196] "BGH_2_NA_2014-07-08_2_StR_195_14_NA_NA_0.txt"
## [197] "BGH_2_NA_2014-08-05_2_StR_172_14_NA_NA_0.txt"
## [198] "BGH_3_NA_2014-10-01_3_StR_150_14_NA_NA_0.txt"
## [199] "BGH_2_NA_2014-10-14_2_StR_44_14_NA_NA_0.txt"
## [200] "BGH_2_NA_2014-10-22_2_StR_62_14_NA_NA_0.txt"
## [201] "BGH_3_NA_2015-03-31_3_StR_527_14_NA_NA_0.txt"
## [202] "BGH_4_NA_2015-05-06_4_StR_87_15_NA_NA_0.txt"
## [203] "BGH_5_NA_2015-06-16_5_StR_184_15_NA_NA_0.txt"
## [204] "BGH_5_NA_2015-09-30_5_StR_347_15_NA_NA_0.txt"
## [205] "BGH_4_NA_2015-12-16_4_StR_226_15_NA_NA_0.txt"
## [206] "BGH_5_NA_2016-01-13_5_StR_460_15_NA_NA_0.txt"
## [207] "BGH_2_NA_2016-01-14_2_StR_449_15_NA_NA_0.txt"
## [208] "BGH_1_NA_2016-01-19_1_StR_603_15_NA_NA_0.txt"
## [209] "BGH_2_NA_2016-02-04_2_StR_527_15_NA_NA_0.txt"
## [210] "BGH_1_NA_2016-02-17_1_StR_209_15_NA_NA_0.txt"
## [211] "BGH_2_NA_2016-02-24_2_StR_533_15_NA_NA_0.txt"
## [212] "BGH_2_NA_2016-03-15_2_StR_157_15_NA_NA_0.txt"
## [213] "BGH_5_NA_2016-04-05_5_StR_18_16_NA_NA_0.txt"
## [214] "BGH_4_NA_2016-04-28_4_StR_88_16_NA_NA_0.txt"
## [215] "BGH_2_NA_2016-05-19_2_StR_482_15_NA_NA_0.txt"
## [216] "BGH_1_NA_2016-06-29_1_StR_110_16_NA_NA_0.txt"
## [217] "BGH_4_NA_2016-09-13_4_StR_371_16_NA_NA_0.txt"
## [218] "BGH_1_NA_2016-09-20_1_StR_349_16_NA_NA_0.txt"
## [219] "BGH_3_NA_2016-12-21_3_StR_454_16_NA_NA_0.txt"
## [220] "BGH_4_NA_2017-07-04_4_StR_149_17_NA_NA_0.txt"
## [221] "BGH_2_NA_2017-08-03_2_StR_265_17_NA_NA_0.txt"
## [222] "BGH_2_NA_2017-08-15_2_StR_222_17_NA_NA_0.txt"
## [223] "BGH_2_NA_2017-08-22_2_StR_97_17_NA_NA_0.txt"
## [224] "BGH_1_NA_2017-10-12_1_StR_324_17_NA_NA_0.txt"
## [225] "BGH_2_NA_2017-11-15_2_StR_261_17_NA_NA_0.txt"
## [226] "BGH_1_NA_2017-12-18_1_StR_547_17_NA_NA_0.txt"
## [227] "BGH_2_NA_2018-02-21_2_StR_511_17_NA_NA_0.txt"
## [228] "BGH_1_NA_2018-03-20_1_StR_401_17_NA_NA_0.txt"
## [229] "BGH_1_NA_2018-03-22_1_StR_412_17_NA_NA_0.txt"
## [230] "BGH_2_NA_2018-09-19_2_StR_455_17_NA_NA_0.txt"
## [231] "BGH_5_NA_2018-09-24_5_StR_471_18_NA_NA_0.txt"
## [232] "BGH_2_NA_2018-11-20_2_StR_325_18_NA_NA_0.txt"
```

```
## [233] "BGH_5_NA_2019-01-22_5_StR_583_18_NA_NA_0.txt"
## [234] "BGH_2_NA_2019-03-12_2_StR_22_19_NA_NA_0.txt"
## [235] "BGH_5_NA_2019-04-16_5_StR_558_18_NA_NA_0.txt"
## [236] "BGH_5_NA_2019-05-08_5_StR_182_19_NA_NA_0.txt"
## [237] "BGH_1_NA_2019-08-06_1_StR_188_19_NA_NA_0.txt"
## [238] "BGH_2_NA_2019-09-24_2_StR_222_19_NA_NA_0.txt"
## [239] "BGH_3_NA_2019-10-02_3_StR_200_19_NA_NA_0.txt"
## [240] "BGH_V_NA_2019-12-19_V_ZR_85_19_NA_NA_0.txt"
## [241] "BGH_3_NA_2020-01-08_3_StR_288_19_NA_NA_0.txt"
## [242] "BGH_5_NA_2020-04-14_5_StR_473_19_NA_NA_0.txt"
## [243] "BGH_6_NA_2020-04-21_6_StR_42_20_NA_NA_0.txt"
## [244] "BGH_4_NA_2020-04-22_4_StR_492_19_NA_NA_0.txt"
## [245] "BGH_5_NA_2020-04-27_5_StR_74_20_NA_NA_0.txt"
## [246] "BGH_2_NA_2020-04-28_2_StR_25_20_NA_NA_0.txt"
## [247] "BGH_4_NA_2020-05-07_4_StR_633_19_NA_NA_0.txt"
## [248] "BGH_5_NA_2020-07-23_5_StR_251_20_NA_NA_0.txt"
## [249] "BGH_4_NA_2020-07-28_4_StR_97_20_NA_NA_0.txt"
## [250] "BGH_4_NA_2020-07-29_4_StR_598_19_NA_NA_0.txt"
## [251] "BGH_6_NA_2020-07-29_6_StR_215_20_NA_NA_0.txt"
## [252] "BGH_6_NA_2020-07-30_6_StR_182_20_NA_NA_0.txt"
## [253] "BGH_3_NA_2020-09-01_3_StR_624_19_NA_NA_0.txt"
## [254] "BGH_3_NA_2020-10-01_3_StR_265_20_NA_NA_0.txt"
## [255] "BGH_2_NA_2020-10-14_2_StR_270_20_NA_NA_0.txt"
## [256] "BGH_3_NA_2020-10-27_3_StR_260_20_NA_NA_0.txt"
## [257] "BGH_2_NA_2020-11-04_2_StR_130_20_NA_NA_0.txt"
## [258] "BGH_3_NA_2020-11-10_3_StR_311_20_NA_NA_0.txt"
## [259] "BGH_6_NA_2020-11-16_6_StR_332_20_NA_NA_0.txt"
## [260] "BGH_5_NA_2020-12-08_5_StR_437_20_NA_NA_0.txt"
## [261] "BGH_6_NA_2020-12-16_6_StR_314_20_NA_NA_0.txt"
## [262] "BGH_5_NA_2021-01-05_5_StR_530_20_NA_NA_0.txt"
## [263] "BGH_V_NA_2021-01-14_V_ZR_107_20_NA_NA_0.txt"
## [264] "BGH_5_NA_2021-01-19_5_StR_471_20_NA_NA_0.txt"
## [265] "BGH_5_NA_2021-01-19_5_StR_492_20_NA_NA_0.txt"
## [266] "BGH_4_NA_2021-01-21_4_StR_83_20_NA_NA_0.txt"
## [267] "BGH_6_NA_2021-02-23_6_StR_11_21_NA_NA_0.txt"
## [268] "BGH_3_NA_2021-02-25_3_StR_204_20_NA_NA_0.txt"
## [269] "BGH_3_NA_2021-03-09_3_StR_26_21_NA_NA_0.txt"
## [270] "BGH_Kartellsenat_NA_2021-03-09_NA_KZR_55_19_NA_NA_0.txt"
## [271] "BGH_2_NA_2021-03-16_2_StR_36_21_NA_NA_0.txt"
## [272] "BGH_VIII_NA_2021-03-18_VIII_ZR_305_19_NA_NA_0.txt"
## [273] "BGH_Anwaltsenat_NA_2021-03-22_NA_AnwZBrfg_2_20_NA_NA_0.txt"
## [274] "BGH_6_NA_2021-03-23_6_StR_100_21_NA_NA_0.txt"
## [275] "BGH_I_NA_2021-03-25_I_ZR_203_19_NA_NA_0.txt"
## [276] "BGH_3_NA_2021-03-30_3_StR_474_19_NA_NA_0.txt"
## [277] "BGH_2_NA_2021-03-31_2_StR_109_20_NA_NA_0.txt"
## [278] "BGH_I_NA_2021-04-01_I_ZR_9_18_NA_NA_0.txt"
## [279] "BGH_1_NA_2021-04-08_1_StR_69_21_NA_NA_0.txt"
## [280] "BGH_5_NA_2021-04-13_5_StR_47_21_NA_NA_0.txt"
## [281] "BGH_I_LE_2006-07-13_I_ZR_241_03_NA_Kontaktanzeigen_0.txt"
```

10.22.4 PDF-Namen definieren

```
placeholder.pdf <- gsub("\\.txt",
                        "\\.pdf",
```

```
                          placeholder.txt)
```

10.22.5 Platzhalter PDF/TXT speichern

```
dir.create("PlatzhalterDokumente")

file_move(placeholder.txt,
          "PlatzhalterDokumente")

file_move(placeholder.pdf,
          "PlatzhalterDokumente")
```

10.22.6 Platzhalter aus Datensatz entfernen

```
txt.bgh <- txt.bgh[!(doc_id %in% placeholder.txt)]
```

11 Frequenztabellen erstellen

11.1 Funktion anzeigen

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

```
## function(x,
##                               varlist = names(x),
##                               sumrow = TRUE,
##                               output.list = TRUE,
##                               output.kable = FALSE,
##                               output.csv = FALSE,
##                               outputdir = "./",
##                               prefix = "",
##                               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,
##                    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 = align,
##                         booktabs = TRUE,
##                         longtable = TRUE) %>% kable_styling(latex_options = "
    repeat_header"))
##         }
##     }
##
##     ## Return List of Frequency Tables
##     if (output.list == TRUE){
##         return(freqtable.list)
##     }
## }
```

11.2 Ignorierte Variablen

```
print(varremove)
```

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

11.3 Liste zu prüfender Variablen

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

```
##  [1] "gericht"           "spruchkoerper_db"  "leitsatz"
##  [4] "spruchkoerper_az"  "registerzeichen"   "eingangsjahr_az"
##  [7] "zusatz_az"         "kollision"         "entscheidungsjahr"
## [10] "eingangsjahr_iso"  "praesi"            "v_praesi"
## [13] "verfahrensart"     "entscheidung_typ"  "berichtigung"
## [16] "doi_concept"       "doi_version"       "version"
## [19] "lizenz"
```

11.4 Präfix definieren

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

11.5 Frequenztabellen berechnen

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

Frequency Table for Variable: gericht

1 unique value(s) detected.

gericht	N	exactpercent	roundedpercent	cumulpercent
BGH	66344	100	100	100
Total	66344	100	100	100

Frequency Table for Variable: spruchkoerper_db

33 unique value(s) detected.

spruchkoerper_db	N	exactpercent	roundedpercent	cumulpercent
1	3867	5.8287110	5.83	5.83
2	5552	8.3685036	8.37	14.20
3	5193	7.8273845	7.83	22.02
4	4619	6.9621970	6.96	28.99
5	4631	6.9802846	6.98	35.97
6	227	0.3421560	0.34	36.31
Anwaltsenat	2029	3.0583022	3.06	39.37
DienstgerichtBund	111	0.1673098	0.17	39.53
Ermittlungsrichter	32	0.0482334	0.05	39.58
GrosserStrafsenat	19	0.0286386	0.03	39.61
GrosserZivilsenat	5	0.0075365	0.01	39.62
I	3677	5.5423249	5.54	45.16
II	2479	3.7365851	3.74	48.90
III	2985	4.4992765	4.50	53.40
IV	2658	4.0063909	4.01	57.40
IX	6413	9.6662848	9.67	67.07
IXa	200	0.3014591	0.30	67.37
Kartellsenat	890	1.3414928	1.34	68.71
Landwirtschaftsenat	392	0.5908598	0.59	69.30
Notarsenat	568	0.8561437	0.86	70.16
Patentanwaltsenat	28	0.0422043	0.04	70.20
Steuerberatersenat	16	0.0241167	0.02	70.23
V	3925	5.9161341	5.92	76.14
VI	2689	4.0531171	4.05	80.20
VII	1924	2.9000362	2.90	83.10
VIII	2807	4.2309779	4.23	87.33

(continued)

spruchkoerper_db	N	exactpercent	roundedpercent	cumulpercent
VereinigteGrosseSenate	1	0.0015073	0.00	87.33
Wirtschaftsprüfersenat	5	0.0075365	0.01	87.34
X	2083	3.1396961	3.14	90.48
XI	2589	3.9023876	3.90	94.38
XII	3454	5.2061980	5.21	99.58
XIII	114	0.1718317	0.17	99.76
Xa	162	0.2441818	0.24	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: leitsatz

2 unique value(s) detected.

leitsatz	N	exactpercent	roundedpercent	cumulpercent
LE	18994	28.62957	28.63	28.63
NA	47350	71.37043	71.37	100.00
Total	66344	100.00000	100.00	100.00

Frequency Table for Variable: spruchkoerper_az

23 unique value(s) detected.

spruchkoerper_az	N	exactpercent	roundedpercent	cumulpercent
NA	4675	7.0466056	7.05	7.05
1	3882	5.8513204	5.85	12.90
2	5558	8.3775473	8.38	21.28
3	4589	6.9169782	6.92	28.19
4	4621	6.9652116	6.97	35.16
5	4632	6.9817919	6.98	42.14

(continued)

spruchkoerper_az	N	exactpercent	roundedpercent	cumulpercent
6	227	0.3421560	0.34	42.48
I	3677	5.5423249	5.54	48.02
II	2479	3.7365851	3.74	51.76
III	2985	4.4992765	4.50	56.26
IV	2658	4.0063909	4.01	60.27
IX	6414	9.6677921	9.67	69.93
IXa	201	0.3029664	0.30	70.24
V	3925	5.9161341	5.92	76.15
VI	2688	4.0516098	4.05	80.20
VII	1925	2.9015435	2.90	83.11
VIII	2807	4.2309779	4.23	87.34
X	2083	3.1396961	3.14	90.48
XA	1	0.0015073	0.00	90.48
XI	2588	3.9008803	3.90	94.38
XII	3453	5.2046907	5.20	99.58
XIII	115	0.1733390	0.17	99.76
Xa	161	0.2426745	0.24	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: registerzeichen

54 unique value(s) detected.

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
AK	317	0.4778126	0.48	0.48
ARAnw	6	0.0090438	0.01	0.49
ARNot	1	0.0015073	0.00	0.49
ARRi	7	0.0105511	0.01	0.50
ARVS	42	0.0633064	0.06	0.56

(continued)

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
ARVZ	42	0.0633064	0.06	0.63
ARZ	131	0.1974557	0.20	0.82
ARs	955	1.4394670	1.44	2.26
ARsVollz	2	0.0030146	0.00	2.27
AnwStB	65	0.0979742	0.10	2.36
AnwStR	35	0.0527553	0.05	2.42
AnwZ	27	0.0406970	0.04	2.46
AnwZB	1122	1.6911853	1.69	4.15
AnwZBrfg	773	1.1651393	1.17	5.31
AnwZP	1	0.0015073	0.00	5.31
BGs	32	0.0482334	0.05	5.36
BLw	288	0.4341010	0.43	5.80
EnVR	329	0.4959002	0.50	6.29
EnVZ	39	0.0587845	0.06	6.35
EnZB	2	0.0030146	0.00	6.35
EnZR	38	0.0572772	0.06	6.41
GSSt	19	0.0286386	0.03	6.44
GSZ	4	0.0060292	0.01	6.45
KRB	38	0.0572772	0.06	6.50
KVR	109	0.1642952	0.16	6.67
KVZ	56	0.0844085	0.08	6.75
KZB	12	0.0180875	0.02	6.77
KZR	264	0.3979260	0.40	7.17
LwZA	2	0.0030146	0.00	7.17
LwZB	15	0.0226094	0.02	7.19
LwZR	85	0.1281201	0.13	7.32
NotStB	19	0.0286386	0.03	7.35
NotStBrfg	54	0.0813939	0.08	7.43
NotZ	370	0.5576993	0.56	7.99

(continued)

registerzeichen	N	exactpercent	roundedpercent	cumulpercent
NotZBrfg	123	0.1853973	0.19	8.18
PatAnwStB	4	0.0060292	0.01	8.18
PatAnwStR	3	0.0045219	0.00	8.19
PatAnwZ	21	0.0316532	0.03	8.22
RiStB	1	0.0015073	0.00	8.22
RiStR	5	0.0075365	0.01	8.23
RiZ	10	0.0150730	0.02	8.24
RiZB	7	0.0105511	0.01	8.25
RiZR	87	0.1311347	0.13	8.38
StB	295	0.4446521	0.44	8.83
StR	22476	33.8779694	33.88	42.71
StbStB	6	0.0090438	0.01	42.72
StbStR	10	0.0150730	0.02	42.73
VGS	1	0.0015073	0.00	42.73
WpStB	2	0.0030146	0.00	42.73
WpStR	3	0.0045219	0.00	42.74
ZA	1272	1.9172796	1.92	44.66
ZB	11308	17.0444954	17.04	61.70
ZR	25406	38.2943446	38.29	100.00
ZRÜ	3	0.0045219	0.00	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_az

33 unique value(s) detected.

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
0	2368	3.5692753	3.57	3.57
1	2530	3.8134571	3.81	7.38

(continued)

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
2	3231	4.8700711	4.87	12.25
3	3279	4.9424213	4.94	17.20
4	3037	4.5776559	4.58	21.77
5	3054	4.6032799	4.60	26.38
6	3161	4.7645605	4.76	31.14
7	3446	5.1941396	5.19	36.33
8	3464	5.2212710	5.22	41.56
9	3384	5.1006873	5.10	46.66
10	3465	5.2227782	5.22	51.88
11	3563	5.3704932	5.37	57.25
12	3257	4.9092608	4.91	62.16
13	3086	4.6515133	4.65	66.81
14	3143	4.7374292	4.74	71.55
15	3268	4.9258411	4.93	76.47
16	2954	4.4525503	4.45	80.93
17	2912	4.3892439	4.39	85.32
18	2788	4.2023393	4.20	89.52
19	2933	4.4208971	4.42	93.94
20	1953	2.9437477	2.94	96.88
21	89	0.1341493	0.13	97.02
80	1	0.0015073	0.00	97.02
86	1	0.0015073	0.00	97.02
88	1	0.0015073	0.00	97.02
91	1	0.0015073	0.00	97.02
93	2	0.0030146	0.00	97.03
94	2	0.0030146	0.00	97.03
95	4	0.0060292	0.01	97.04
96	9	0.0135657	0.01	97.05
97	100	0.1507295	0.15	97.20

(continued)

eingangsjahr_az	N	exactpercent	roundedpercent	cumulpercent
98	515	0.7762571	0.78	97.98
99	1343	2.0242976	2.02	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: zusatz_az

2 unique value(s) detected.

zusatz_az	N	exactpercent	roundedpercent	cumulpercent
NA	66343	99.9984927	100	100
a	1	0.0015073	0	100
Total	66344	100.0000000	100	100

Frequency Table for Variable: kollision

6 unique value(s) detected.

kollision	N	exactpercent	roundedpercent	cumulpercent
0	65551	98.8047148	98.80	98.80
1	724	1.0912818	1.09	99.90
2	51	0.0768721	0.08	99.97
3	11	0.0165802	0.02	99.99
4	5	0.0075365	0.01	100.00
5	2	0.0030146	0.00	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: entscheidungsjahr

22 unique value(s) detected.

entscheidungsjahr	N	exactpercent	roundedpercent	cumulpercent
2000	2197	3.311528	3.31	3.31
2001	2404	3.623538	3.62	6.94
2002	2745	4.137526	4.14	11.07
2003	2879	4.339503	4.34	15.41
2004	3078	4.639455	4.64	20.05
2005	3098	4.669601	4.67	24.72
2006	3104	4.678645	4.68	29.40
2007	3310	4.989147	4.99	34.39
2008	3613	5.445858	5.45	39.83
2009	3389	5.108224	5.11	44.94
2010	3625	5.463946	5.46	50.41
2011	3697	5.572471	5.57	55.98
2012	3506	5.284577	5.28	61.26
2013	3190	4.808272	4.81	66.07
2014	3071	4.628904	4.63	70.70
2015	3074	4.633426	4.63	75.33
2016	3271	4.930363	4.93	80.26
2017	3120	4.702761	4.70	84.97
2018	2988	4.503798	4.50	89.47
2019	2947	4.441999	4.44	93.91
2020	3247	4.894188	4.89	98.81
2021	791	1.192271	1.19	100.00
Total	66344	100.000000	100.00	100.00

Frequency Table for Variable: eingangsjahr_iso

33 unique value(s) detected.

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
1980	1	0.0015073	0.00	0.00

(continued)

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
1986	1	0.0015073	0.00	0.00
1988	1	0.0015073	0.00	0.00
1991	1	0.0015073	0.00	0.01
1993	2	0.0030146	0.00	0.01
1994	2	0.0030146	0.00	0.01
1995	4	0.0060292	0.01	0.02
1996	9	0.0135657	0.01	0.03
1997	100	0.1507295	0.15	0.18
1998	515	0.7762571	0.78	0.96
1999	1343	2.0242976	2.02	2.98
2000	2368	3.5692753	3.57	6.55
2001	2530	3.8134571	3.81	10.37
2002	3231	4.8700711	4.87	15.24
2003	3279	4.9424213	4.94	20.18
2004	3037	4.5776559	4.58	24.76
2005	3054	4.6032799	4.60	29.36
2006	3161	4.7645605	4.76	34.12
2007	3446	5.1941396	5.19	39.32
2008	3464	5.2212710	5.22	44.54
2009	3384	5.1006873	5.10	49.64
2010	3465	5.2227782	5.22	54.86
2011	3563	5.3704932	5.37	60.23
2012	3257	4.9092608	4.91	65.14
2013	3086	4.6515133	4.65	69.79
2014	3143	4.7374292	4.74	74.53
2015	3268	4.9258411	4.93	79.46
2016	2954	4.4525503	4.45	83.91
2017	2912	4.3892439	4.39	88.30
2018	2788	4.2023393	4.20	92.50

(continued)

eingangsjahr_iso	N	exactpercent	roundedpercent	cumulpercent
2019	2933	4.4208971	4.42	96.92
2020	1953	2.9437477	2.94	99.87
2021	89	0.1341493	0.13	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: praesi

6 unique value(s) detected.

praesi	N	exactpercent	roundedpercent	cumulpercent
Geiss	935	1.409321	1.41	1.41
Hirsch	21887	32.990172	32.99	34.40
Limberg	20932	31.550705	31.55	65.95
Tolksdorf	21013	31.672796	31.67	97.62
VACANCY-1	288	0.434101	0.43	98.06
VACANCY-2	1289	1.942904	1.94	100.00
Total	66344	100.000000	100.00	100.00

Frequency Table for Variable: v_praesi

6 unique value(s) detected.

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Ellenberger	13325	20.084710	20.08	20.08
Jähnke	5716	8.615700	8.62	28.70
Müller	13227	19.936995	19.94	48.64
Schlick	20623	31.084951	31.08	79.72
VACANCY-4	4284	6.457253	6.46	86.18
Wenzel	9169	13.820391	13.82	100.00

(continued)

v_praesi	N	exactpercent	roundedpercent	cumulpercent
Total	66344	100.000000	100.00	100.00

Frequency Table for Variable: verfahrensart

54 unique value(s) detected.

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
NA	7	0.0105511	0.01	0.01
Aktenkontrolle für Haftprüfungsverfahren	317	0.4778126	0.48	0.49
Allgemeines Register (Anwaltssachen)	6	0.0090438	0.01	0.50
Allgemeines Register (Dienstgericht des Bundes)	7	0.0105511	0.01	0.51
Allgemeines Register (Notarsachen)	1	0.0015073	0.00	0.51
Allgemeines Register und Gerichtsstandsbestimmungen (Strafsachen)	955	1.4394670	1.44	1.95
Allgemeines Register und Gerichtsstandsbestimmungen (Zivilsachen)	131	0.1974557	0.20	2.15
Anträge außerhalb eines in der Rechtsmittelinstanz anhängigen Verfahrens (Zivilsachen)	1272	1.9172796	1.92	4.06
Anträge außerhalb eines in der Revisionsinstanz für Landwirtschaftssachen anhängigen Verfahrens	2	0.0030146	0.00	4.07
Anträge betreffend Richter im Bundesdienst und Mitglieder des Bundesrechnungshofes auf gerichtliche Entscheidung im Versetzungs- und Prüfungsverfahren sowie auf vorläufige Untersagung der Amtsgeschäfte	10	0.0150730	0.02	4.08

(continued)

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Berufungen und Anträge auf Zulassung der Berufung gegen Entscheidungen der Oberlandesgerichte in Notarsachen	123	0.1853973	0.19	4.27
Berufungen und Anträge auf Zulassung der Berufung gegen Entscheidungen eines Anwaltsgerichtshofes	773	1.1651393	1.17	5.43
Berufungen und Anträge auf Zulassung der Berufung gegen Urteile der Oberlandesgerichte in Disziplinarsachen gegen Notare	54	0.0813939	0.08	5.51
Beschwerden (Strafsachen)	295	0.4446521	0.44	5.96
Beschwerden gegen Beschlüsse der Oberlandesgerichte in Disziplinarsachen gegen Notare	19	0.0286386	0.03	5.99
Beschwerden gegen Entscheidungen eines Anwaltsgerichtshofes	1122	1.6911853	1.69	7.68
Beschwerden gegen die Nichtzulassung der Revision und Beschwerden der Richter, Staatsanwälte und Mitglieder des Bundesrechnungshofes gegen Disziplinarverfügungen	1	0.0015073	0.00	7.68
Beschwerden gegen die Nichtzulassung der Revision und Beschwerden gegen Entscheidungen eines Anwaltsgerichtshofes	65	0.0979742	0.10	7.78
Beschwerden gegen die Nichtzulassung der Revision und Beschwerden in berufsgerichtlichen Verfahren (Steuerberater- und Steuerbevollmächtigtensachen)	6	0.0090438	0.01	7.79

(continued)

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Beschwerden gegen die Nichtzulassung der Revision und Beschwerden in berufsgerichtlichen Verfahren (Wirtschaftsprüfersachen)	2	0.0030146	0.00	7.79
Beschwerden gegen die Nichtzulassung der Revision und Beschwerden nach der PatAO	4	0.0060292	0.01	7.80
Beschwerden und Rechtsbeschwerden in Landwirtschaftssachen der streitigen bürgerlichen Gerichtsbarkeit	15	0.0226094	0.02	7.82
Beschwerden, Rechtsbeschwerden, weitere Beschwerden, Beschwerden gegen die Nichtzulassung der Revision nach dem BEG (Zivilsachen)	11308	17.0444954	17.04	24.86
Einzelne richterliche Anordnungen des Ermittlungsrichters (Strafsachen)	32	0.0482334	0.05	24.91
Entscheidungen über Justizverwaltungsakte (Strafsachen)	42	0.0633064	0.06	24.97
Entscheidungen über Justizverwaltungsakte (Zivilsachen)	42	0.0633064	0.06	25.04
Erstinstanzliche Klagen auf Entschädigung wegen überlanger Gerichtsverfahren und strafrechtlicher Ermittlungsverfahren (Zivilsachen)	3	0.0045219	0.00	25.04
Großer Senat (Strafsachen)	19	0.0286386	0.03	25.07
Großer Senat (Zivilsachen)	4	0.0060292	0.01	25.08
Klagen über Entscheidungen in Zulassungssachen oder gegen sonstige Verwaltungsakte betreffend Rechtsanwälte beim Bundesgerichtshof	27	0.0406970	0.04	25.12

(continued)

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Klagen über die Anfechtung von Wahlen und Beschlüssen der Rechtsanwaltskammer beim BGH und der Bundesrechtsanwaltskammer	1	0.0015073	0.00	25.12
Nichtzulassungsbeschwerden (Kartellverwaltungssachen)	56	0.0844085	0.08	25.20
Nichtzulassungsbeschwerden in energiewirtschaftsrechtlichen Verwaltungssachen nach dem EnWG	39	0.0587845	0.06	25.26
Rechtsbeschwerden (Kartellverwaltungssachen)	109	0.1642952	0.16	25.43
Rechtsbeschwerden in Kartellbußgeldverfahren	38	0.0572772	0.06	25.48
Rechtsbeschwerden in Landwirtschaftssachen der freiwilligen Gerichtsbarkeit	288	0.4341010	0.43	25.92
Rechtsbeschwerden in energiewirtschaftsrechtlichen Verwaltungssachen nach dem EnWG	329	0.4959002	0.50	26.41
Rechtsbeschwerden und Beschwerden in bürgerlichen Rechtsstreitigkeiten (Kartellsachen)	12	0.0180875	0.02	26.43
Rechtsbeschwerden und Beschwerden in bürgerlichen Rechtsstreitigkeiten nach dem EnWG	2	0.0030146	0.00	26.43
Revisionen gegen Urteile eines Anwaltsgerichtshofes	35	0.0527553	0.05	26.49
Revisionen in Disziplinarsachen nach dem Deutschen Richtergesetz	5	0.0075365	0.01	26.50
Revisionen in Versetzungs- und Prüfungsverfahren nach dem Deutschen Richtergesetz	87	0.1311347	0.13	26.63

(continued)

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Revisionen in berufsgerichtlichen Verfahren (Steuerberater- und Steuerbevollmächtigtensachen)	10	0.0150730	0.02	26.64
Revisionen in berufsgerichtlichen Verfahren (Wirtschaftsprüfersachen)	3	0.0045219	0.00	26.65
Revisionen nach der PatAO	3	0.0045219	0.00	26.65
Revisionen und Vorlegungssachen nach § 121 Abs.1 Nr.1, Abs. 2 GVG, § 79 Abs. 3 OWiG, §§ 13 Abs. 4, 25 StrRehaG (Strafsachen)	22476	33.8779694	33.88	60.53
Revisionen, Beschwerden gegen die Nichtzulassung der Revision und Anträge auf Zulassung der Sprungrevision (Landwirtschaftssachen)	85	0.1281201	0.13	60.66
Revisionen, Beschwerden gegen die Nichtzulassung der Revision und Anträge auf Zulassung der Sprungrevision in bürgerlichen Rechtsstreitigkeiten (Kartellsachen)	264	0.3979260	0.40	61.05
Revisionen, Beschwerden gegen die Nichtzulassung der Revision und Anträge auf Zulassung der Sprungrevision in bürgerlichen Rechtsstreitigkeiten nach dem EnWG	38	0.0572772	0.06	61.11
Revisionen, Beschwerden gegen die Nichtzulassung der Revision, Anträge auf Zulassung der Sprungrevision, Berufungen in Patentsachen (Zivilsachen)	25406	38.2943446	38.29	99.41
Vereinigte Große Senate	1	0.0015073	0.00	99.41

(continued)

verfahrensart	N	exactpercent	roundedpercent	cumulpercent
Verwaltungsstreitverfahren in Notarsachen und Beschwerden gegen Entscheidungen der Oberlandesgerichte	370	0.5576993	0.56	99.97
Verwaltungsstreitverfahren in Patentanwaltssachen; Berufungen und Anträge auf Zulassung der Berufung und Beschwerden gegen Entscheidungen der Oberlandesgerichte	21	0.0316532	0.03	100.00
Vorlegungssachen (Strafvollzugssachen)	2	0.0030146	0.00	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: entscheidung_typ

4 unique value(s) detected.

entscheidung_typ	N	exactpercent	roundedpercent	cumulpercent
NA	22	0.0331605	0.03	0.03
B	47677	71.8633185	71.86	71.90
U	18634	28.0869408	28.09	99.98
V	11	0.0165802	0.02	100.00
Total	66344	100.0000000	100.00	100.00

Frequency Table for Variable: berichtigung

2 unique value(s) detected.

berichtigung	N	exactpercent	roundedpercent	cumulpercent
NA	65580	98.848426	98.85	98.85
Berichtigung	764	1.151574	1.15	100.00
Total	66344	100.000000	100.00	100.00

(continued)

berichtigung	N	exactpercent	roundedpercent	cumulpercent

Frequency Table for Variable: doi_concept

1 unique value(s) detected.

doi_concept	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.3942742	66344	100	100	100
Total	66344	100	100	100

Frequency Table for Variable: doi_version

1 unique value(s) detected.

doi_version	N	exactpercent	roundedpercent	cumulpercent
10.5281/zenodo.4705855	66344	100	100	100
Total	66344	100	100	100

Frequency Table for Variable: version

1 unique value(s) detected.

version	N	exactpercent	roundedpercent	cumulpercent
2021-04-27	66344	100	100	100
Total	66344	100	100	100

Frequency Table for Variable: lizenz

1 unique value(s) detected.

lizenz	N	exactpercent	roundedpercent	cumulpercent
Creative Commons Zero 1.0 Universal	66344	100	100	100
Total	66344	100	100	100

12 Frequenztabellen visualisieren

12.1 Präfix erstellen

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

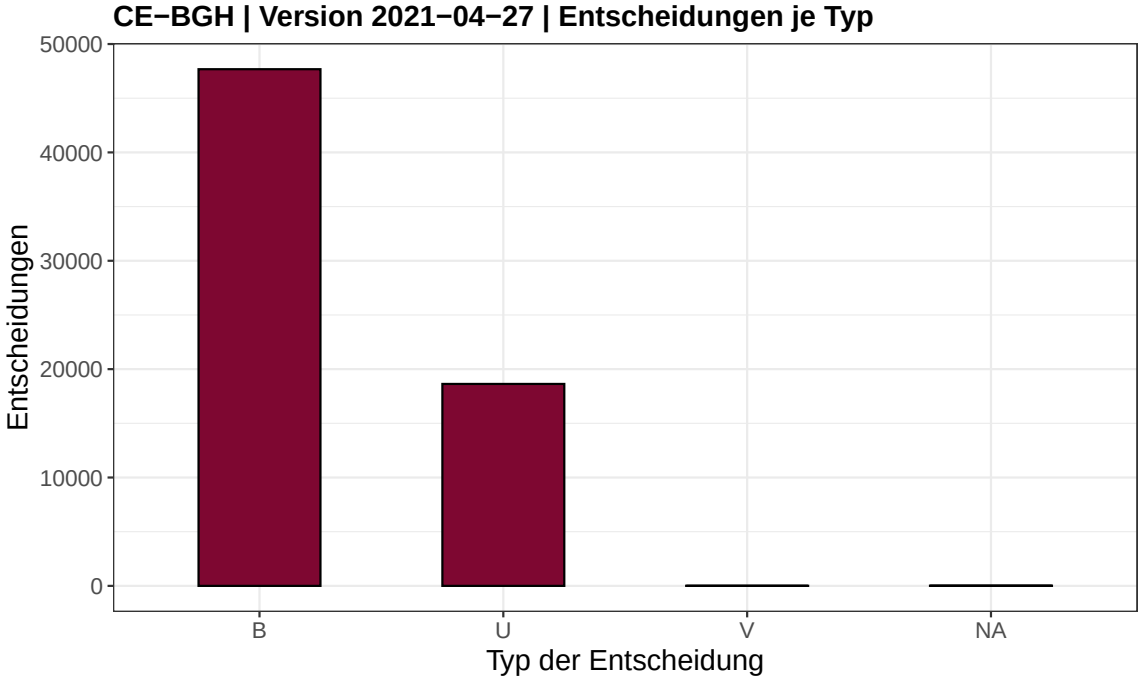
12.2 Tabellen einlesen

```
table.entsch.typ <- fread(paste0(prefix,
                                 "entscheidung_typ.csv"))

table.spruch.db <- fread(paste0(prefix,
                                "spruchkoerper_db.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"))
```

12.3 Diagramm: Typ der Entscheidung

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(entscheidung_typ,
                             -N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             color = "black",
             width = 0.5) +
    theme_bw() +
    labs(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| Entscheidungen je Typ"),
        caption = paste("DOI:",
                        doi.version),
        x = "Typ der Entscheidung",
        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)
    )
```

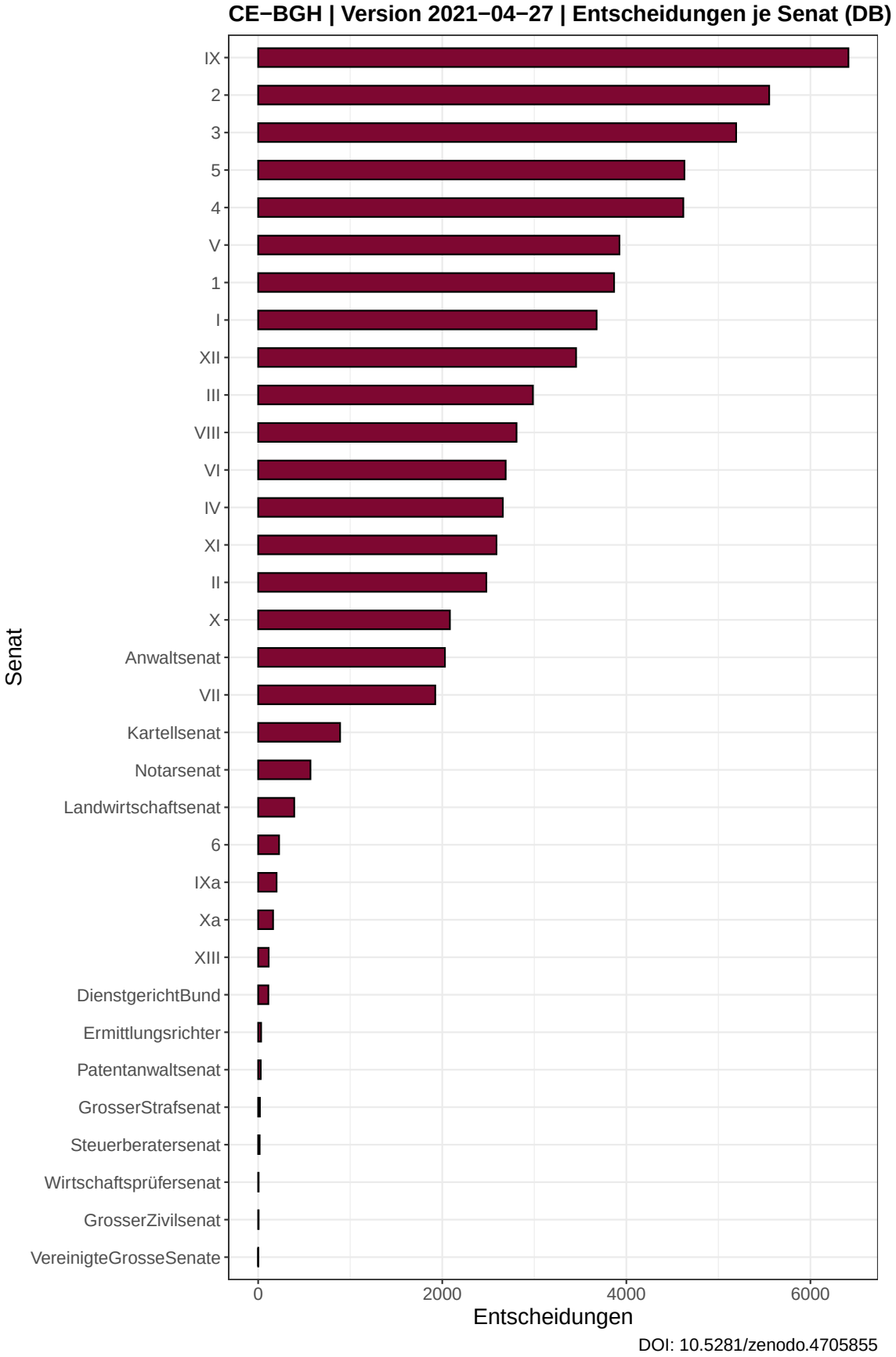


DOI: 10.5281/zenodo.4705855

12.4 Diagramm: Spruchkörper nach Datenbank

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(spruchkoerper_db,
                             N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             color = "black",
             width = 0.5) +
    coord_flip()+
    theme_bw() +
    labs(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| Entscheidungen je Senat (DB)"),
        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)
    )
```

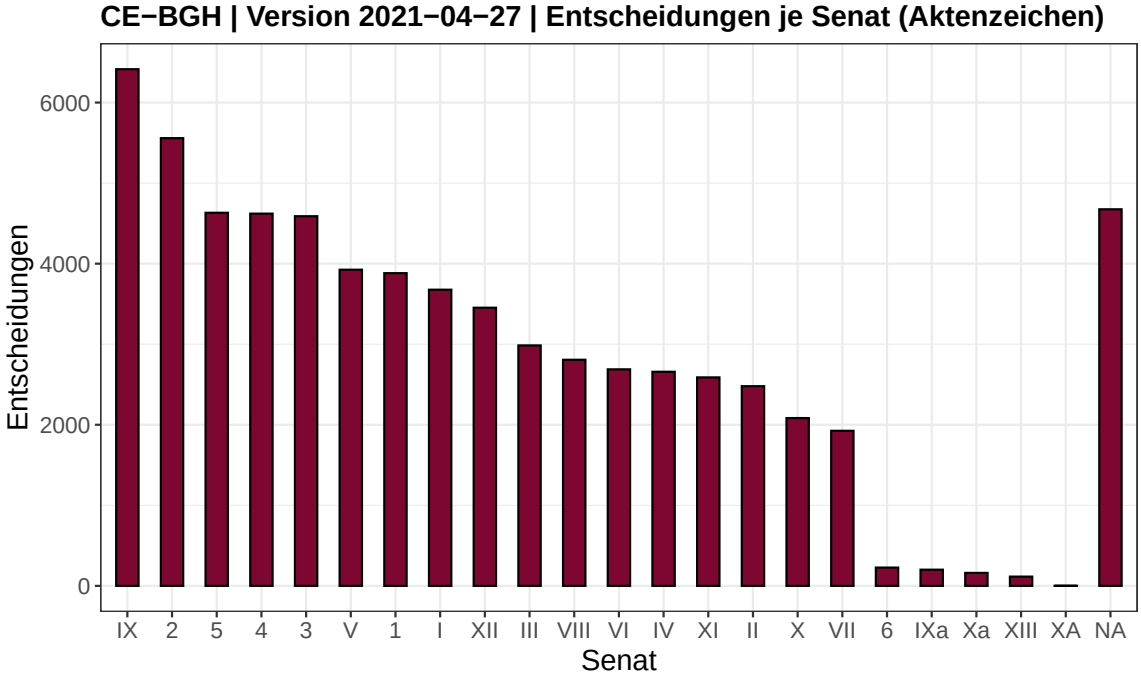


DOI: 10.5281/zenodo.4705855

12.5 Diagramm: Spruchkörper nach Aktenzeichen

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(spruchkoerper_az,
                             -N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             color = "black",
             width = 0.5) +
    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)
    )
```

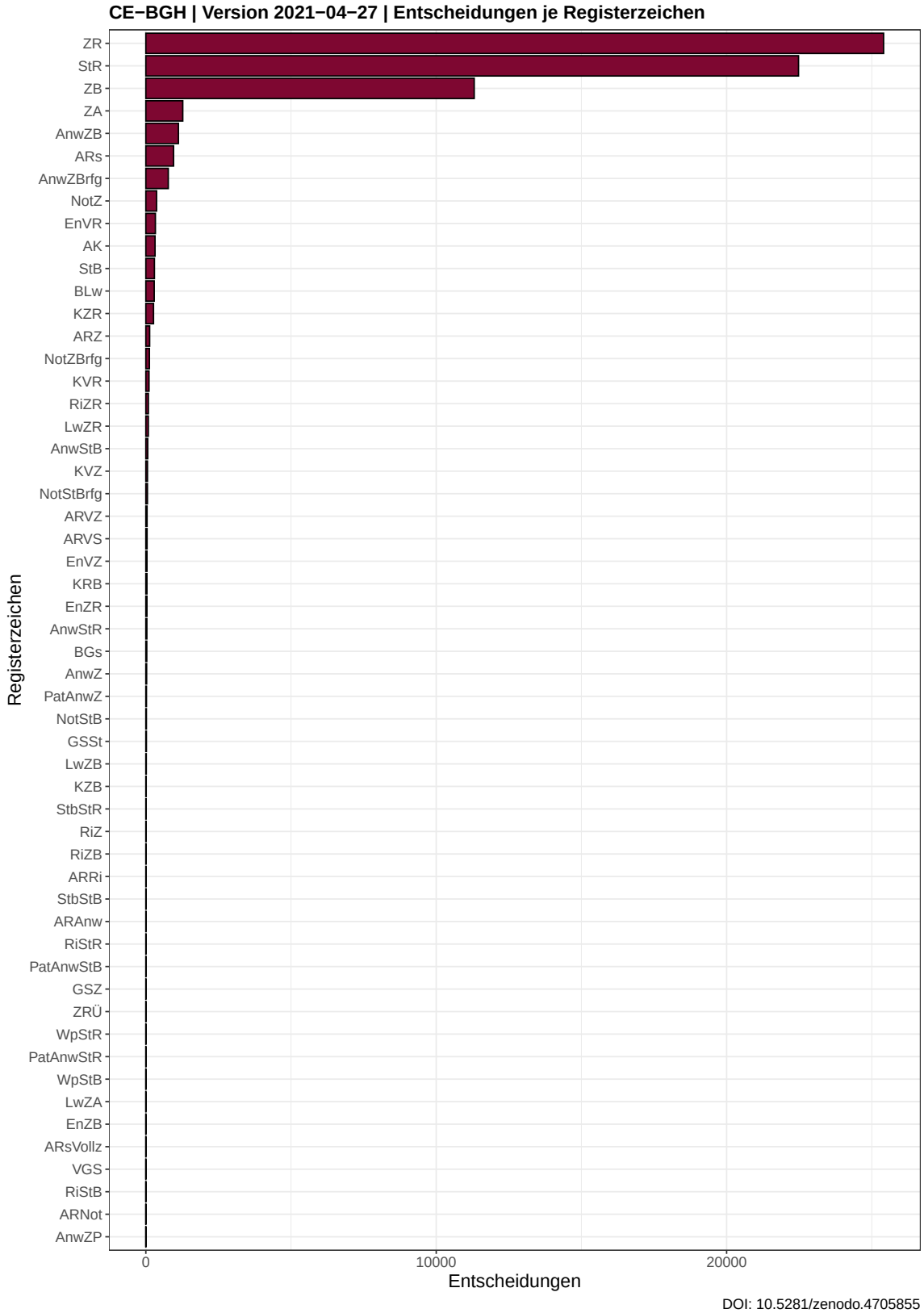


DOI: 10.5281/zenodo.4705855

12.6 Diagramm: Registerzeichen

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(registerzeichen,
                             N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             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)
    )
```

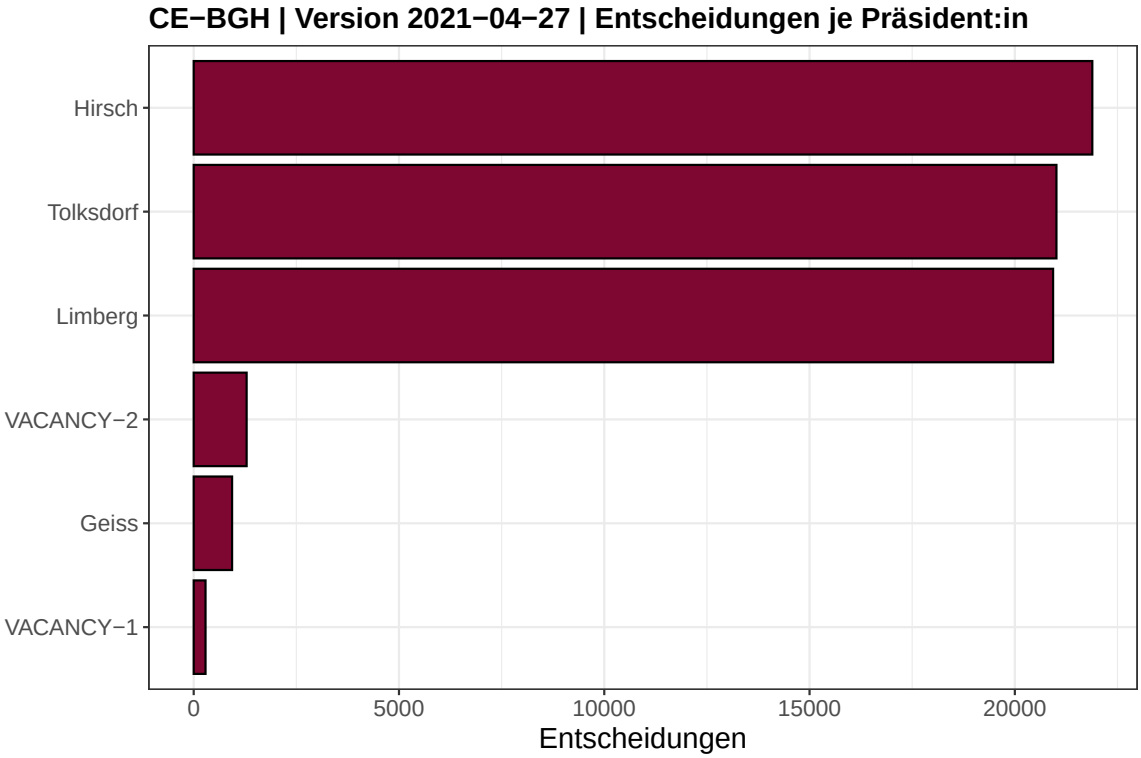


DOI: 10.5281/zenodo.4705855

12.7 Diagramm: Präsident:in

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(praesi,
                             N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             color = "black") +
    coord_flip()+
    theme_bw() +
    labs(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| Entscheidungen je Präsident:in"),
        caption = paste("DOI:",
                        doi.version),
        x = "Präsident:in",
        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)
    )
```

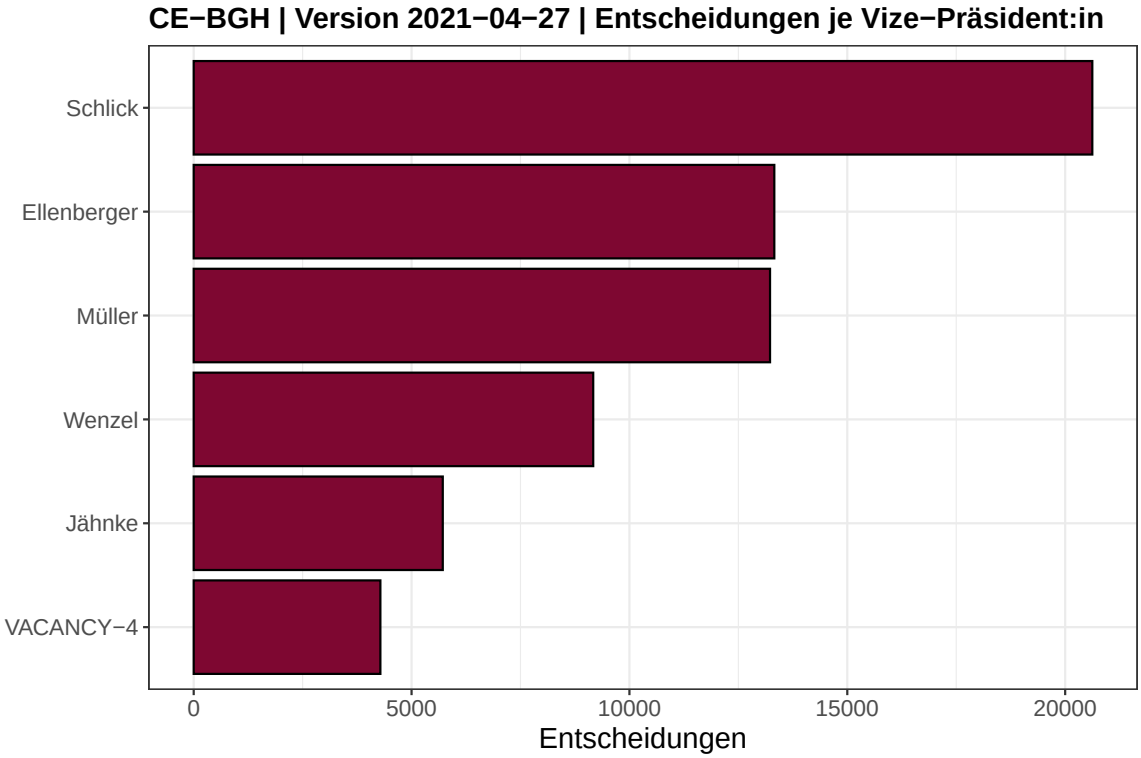


DOI: 10.5281/zenodo.4705855

12.8 Diagramm: Vize-Präsident:in

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = reorder(v_praesi,
                             N),
                 y = N),
             stat = "identity",
             fill = "#7e0731",
             color = "black") +
    coord_flip()+
    theme_bw() +
    labs(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| Entscheidungen je Vize-Präsident:in"),
        caption = paste("DOI:",
                        doi.version),
        x = "Vize-Präsident:in",
        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)
    )
```

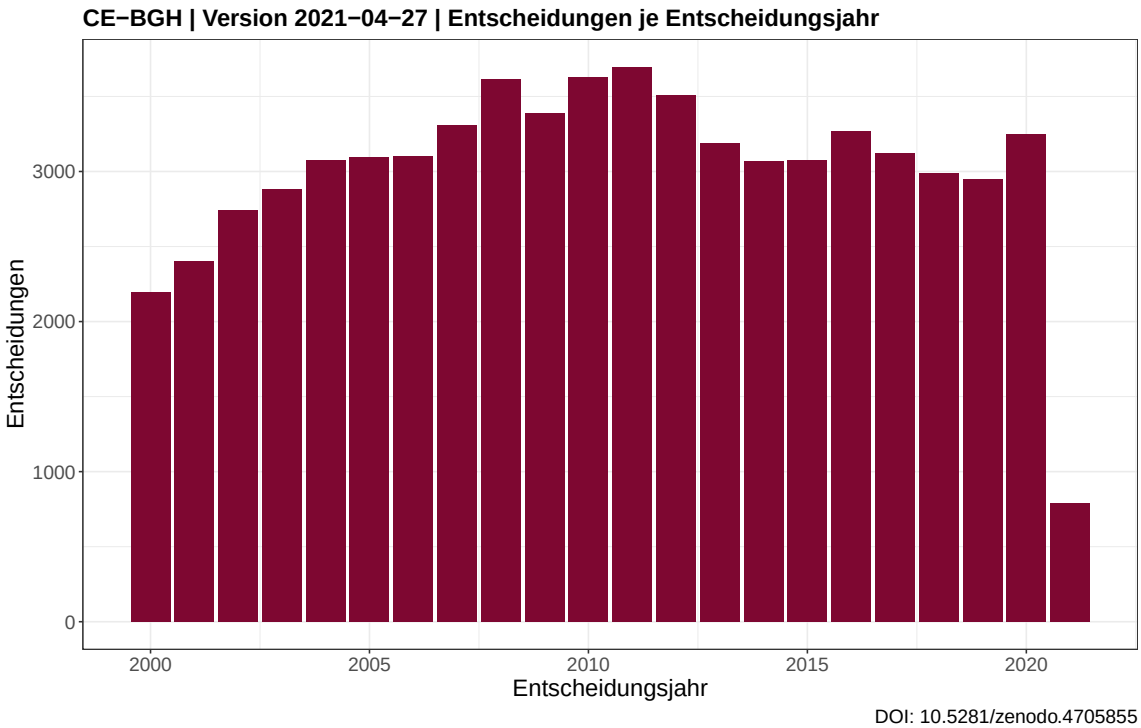


DOI: 10.5281/zenodo.4705855

12.9 Diagramm: Entscheidungsjahr

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

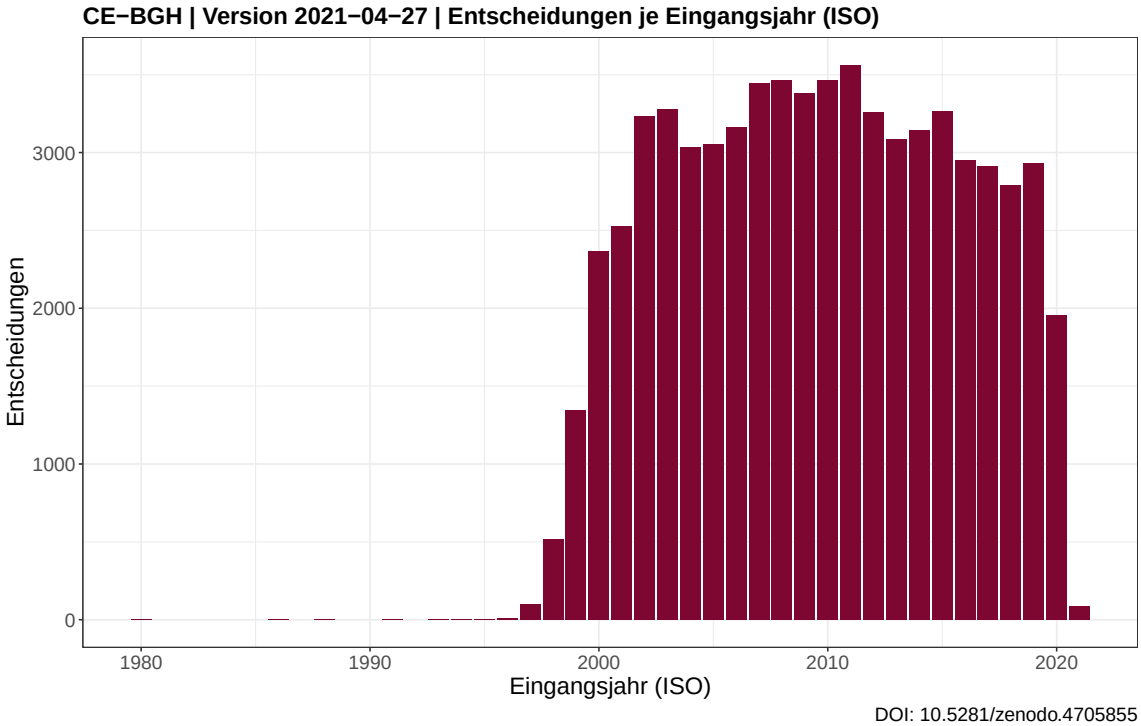
```
ggplot(data = freqtable) +
    geom_bar(aes(x = entscheidungsjahr,
                 y = N),
             stat = "identity",
             fill = "#7e0731") +
    theme_bw() +
    labs(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| 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)
    )
```



12.10 Diagramm: Eingangsjahr (ISO)

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

```
ggplot(data = freqtable) +
    geom_bar(aes(x = eingangsjahr_iso,
                 y = N),
             stat = "identity",
             fill = "#7e0731") +
    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)
    )
```



13 Korpus-Analytik

13.1 Berechnung linguistischer Kennwerte

An dieser Stelle werden für jedes Dokument die Anzahl Zeichen, 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.

13.1.1 Funktion anzeigen

```
print(f.summarize.iterator,
      threads = fullCores,
      chunksize = 1)
```

```
## function(dt,
##                                  threads = detectCores(),
##                                  chunksize = 1){
##
##
##     begin.dopar <- Sys.time()
##
##     dt[, nchars := lapply(.(text), nchar)]
##
##     print(paste0("Parallel processing using ",
##                  threads,
##                  " threads. Begin at ",
##                  begin.dopar,
##                  ". Processing ",
##                  dt[,.N],
##                  " documents with a total length of ",
##                  dt[,sum(nchars)],
##                  " characters."))
##
##
##     ord <- order(-dt$nchars)
##     dt <- dt[ord]
##
##     cl <- makeForkCluster(threads)
##     registerDoParallel(cl)
##
##     itx <- iter(dt[nchars > 0],
##                 by = "row",
##                 chunksize = chunksize)
##
##     result <- foreach(i = itx,
##                       .errorhandling = 'pass') %dopar% {
##                           temp <- summary(corpus(i))
##                           return(temp)
##                       }
##
##     stopCluster(cl)
##
```

```
##
##     end.dopar <- Sys.time()
##     duration.dopar <- end.dopar - begin.dopar
##
##     summary.corpus <- rbindlist(result)
##
##     setnames(summary.corpus,
##              old = c("Text",
##                      "Tokens",
##                      "Types",
##                      "Sentences"),
##              new = c("doc_id",
##                      "ntokens",
##                      "ntypes",
##                      "nsentences"))
##
##     if(dt[nchars == 0, .N] > 0){
##
##         dt.charnull <- dt[nchars == 0]
##         dt.charnull$text <- NULL
##         dt.charnull$ntokens <- rep(0, dt.charnull[,.N])
##         dt.charnull$ntypes <- rep(0, dt.charnull[,.N])
##         dt.charnull$nsentences <- rep(0, dt.charnull[,.N])
##
##         summary.corpus <- rbind(summary.corpus,
##                                 dt.charnull)
##     }
##
##
##     summary.corpus <- summary.corpus[order(ord)]
##
##
##     print(paste0("Runtime was ",
##                  round(duration.dopar,
##                        digits = 2),
##                  " ",
##                  attributes(duration.dopar)$units,
##                  ". Ended at ",
##                  end.dopar, "."))
##
##     return(summary.corpus)
##
## }
```

13.1.2 Berechnung durchführen

```
summary.corpus <- f.summarize.iterator(txt.bgh)
```

```
## [1] "Parallel processing using 16 threads. Begin at 2021-04-27 12:08:20.
    Processing 66344 documents with a total length of 742090149 characters."
## [1] "Runtime was 7.02 mins. Ended at 2021-04-27 12:15:21."
```

13.1.3 Variablen-Namen anpassen

```
setnames(summary.corpus,
         old = c("nchars",
                 "ntokens",
                 "ntypes",
                 "nsentences"),
         new = c("zeichen",
                 "tokens",
                 "typen",
                 "saetze"))

setnames(txt.bgh,
         old = "nchars",
         new = "zeichen")
```

13.1.4 Kennwerte dem Korpus hinzufügen

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

13.2 Zusammenfasssungen: 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.

13.2.1 Zusammenfassungen berechnen

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


dt.sums.ling <- summary.corpus[,
                               lapply(.SD, sum),
                               .SDcols = c("zeichen",
                                           "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"))
```

13.2.2 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
zeichen	742,090,149	176	2,962.75	7,283	11,185.49001	15,664.75	339,765
tokens	113,513,695	30	427.00	1,096	1,710.98660	2,395.25	53,108
saetze	6,033,607	2	28.00	63	90.94428	126.00	2,683
typen	NA	25	214.00	437	540.69406	769.00	5,594

13.2.3 Zusammenfassungen speichern

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

13.3 Zusammenfassungen: Quantitative Variablen

13.3.1 Entscheidungsdatum

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

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2000-01-04" "2006-01-24" "2010-12-02" "2010-12-03" "2015-12-03" "2021-04-15"
```

13.3.2 Zusammenfassungen berechnen

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


dt.unique.docvars <- summary.corpus[,
                                    lapply(.SD, function(x)length(unique(na.omit(
    x)))),
                                    .SDcols = c("entscheidungsjahr",
                                                "eingangsjahr_iso",
                                                "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",
                             "Einzigartig",
                             "Min",
                             "Quart1",
                             "Median",
                             "Mean",
                             "Quart3",
                             "Max"))
```

13.3.3 Zusammenfassungen anzeigen

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

Variable	Einzigartig	Min	Quart1	Median	Mean	Quart3	Max
entscheidungsjahr	22	2000	2006	2010	2010.4275	2015	2021
eingangsjahr_iso	33	1980	2005	2010	2009.5618	2015	2021
eingangsnummer	752	1	56	156	198.2656	302	1304

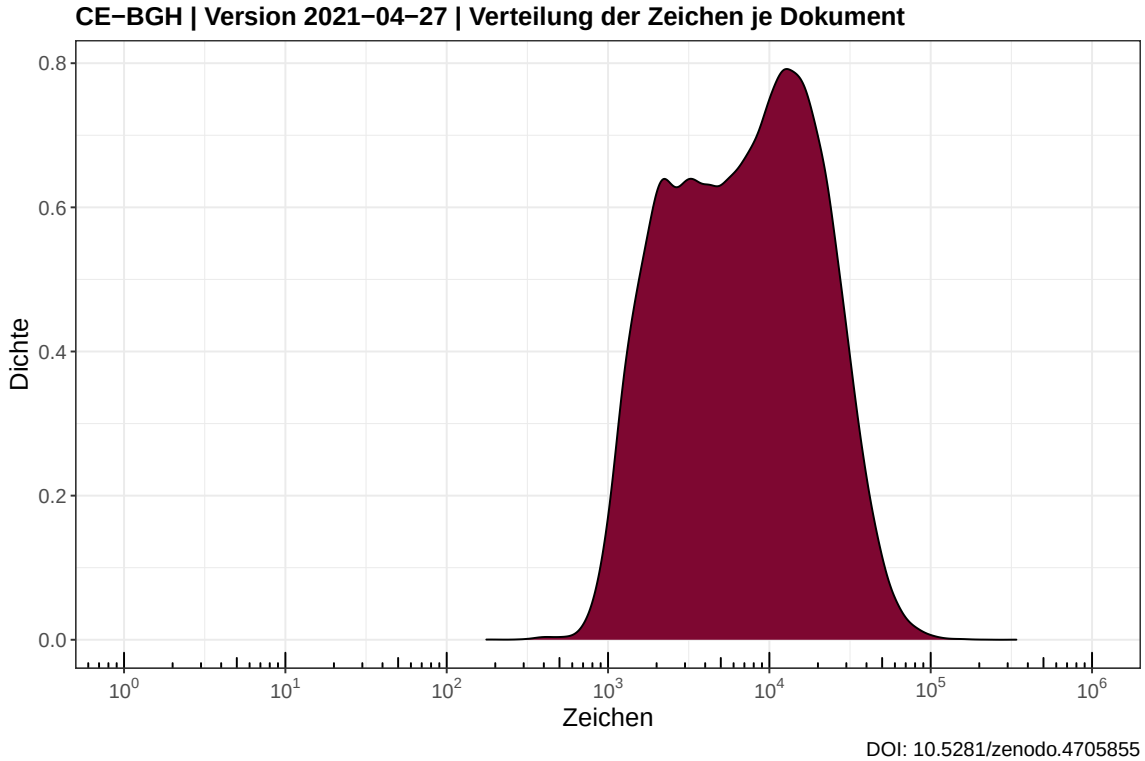
13.3.4 Zusammenfassungen speichern

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

13.4 Verteilungen linguistischer Kennwerte

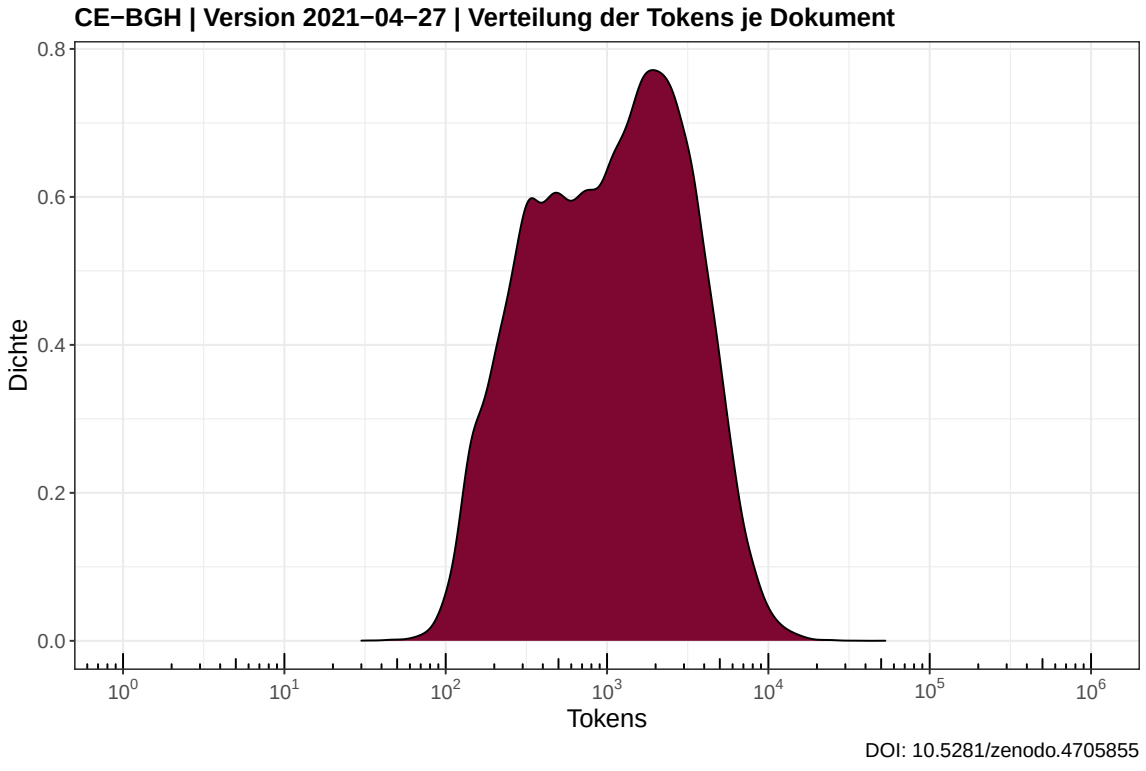
13.4.1 Diagramm: Verteilung Zeichen

```
ggplot(data = summary.corpus)+
    geom_density(aes(x = zeichen),
                 fill = "#7e0731")+
    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()+
    labels(
        title = paste(datasetname,
                      "| Version",
                      datestamp,
                      "| Verteilung der Zeichen je Dokument"),
        caption = paste("DOI:",
                        doi.version),
        x = "Zeichen",
        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)
    )
```



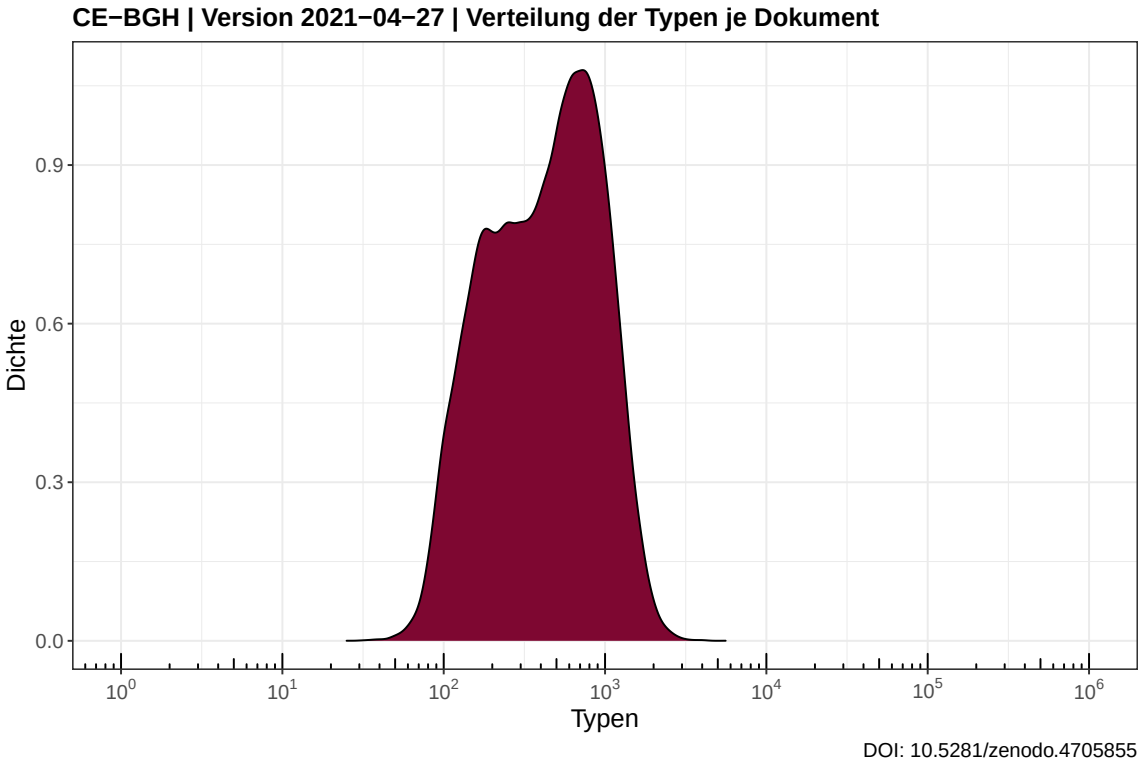
13.4.2 Diagramm: Verteilung Tokens

```
ggplot(data = summary.corpus)+
    geom_density(aes(x = tokens),
                 fill = "#7e0731")+
    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)
    )
```



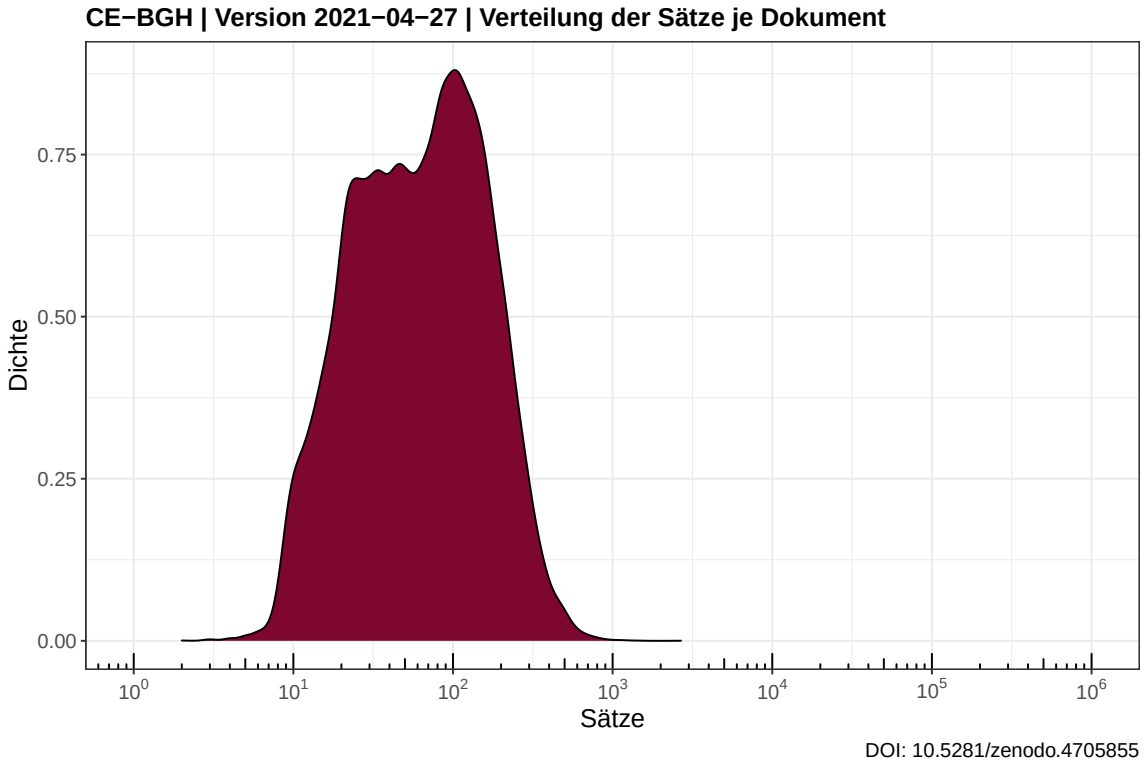
13.4.3 Diagramm: Verteilung Typen

```
ggplot(data = summary.corpus)+
    geom_density(aes(x = typen),
                 fill = "#7e0731")+
    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)
    )
```



13.4.4 Diagramm: Verteilung Sätze

```
ggplot(data = summary.corpus)+
    geom_density(aes(x = saetze),
                 fill = "#7e0731")+
    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)
    )
```



13.5 Anzahl Variablen im Korpus

```
length(txt.bgh)
```

```
## [1] 31
```

13.6 Namen der Variablen im Korpus

```
names(txt.bgh)
```

```
##  [1] "doc_id"            "text"              "gericht"
##  [4] "spruchkoerper_db"  "leitsatz"          "datum"
##  [7] "spruchkoerper_az"  "registerzeichen"   "eingangsnummer"
## [10] "eingangsjahr_az"   "zusatz_az"         "name"
## [13] "kollision"         "entscheidungsjahr" "eingangsjahr_iso"
## [16] "praesi"            "v_praesi"          "verfahrensart"
## [19] "aktenzeichen"      "entscheidung_typ"  "ecli"
## [22] "bemerkung"         "berichtigung"      "doi_concept"
## [25] "doi_version"       "version"           "lizenz"
## [28] "zeichen"           "tokens"            "typen"
## [31] "saetze"
```

14 CSV-Dateien erstellen

14.1 CSV mit vollem Datensatz speichern

```
csvname.full <- paste(datasetname,
                      datestamp,
                      "DE_CSV_Datensatz.csv",
                      sep = "_")

fwrite(txt.bgh,
       csvname.full,
       na = "NA")
```

14.2 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")
```

15 Dateigrößen analysieren

15.1 Gesamtgröße

15.1.1 Korpus-Objekt in RAM (MB)

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

```
## 799.2 MB
```

15.1.2 CSV Korpus (MB)

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

```
## [1] 784.0522
```

15.1.3 CSV Metadaten (MB)

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

```
## [1] 30.39824
```

15.1.4 PDF-Dateien (MB)

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

pdf.MB <- file.size(files.pdf) / 10^6
sum(pdf.MB)
```

```
## [1] 6067.462
```

15.1.5 TXT-Dateien (MB)

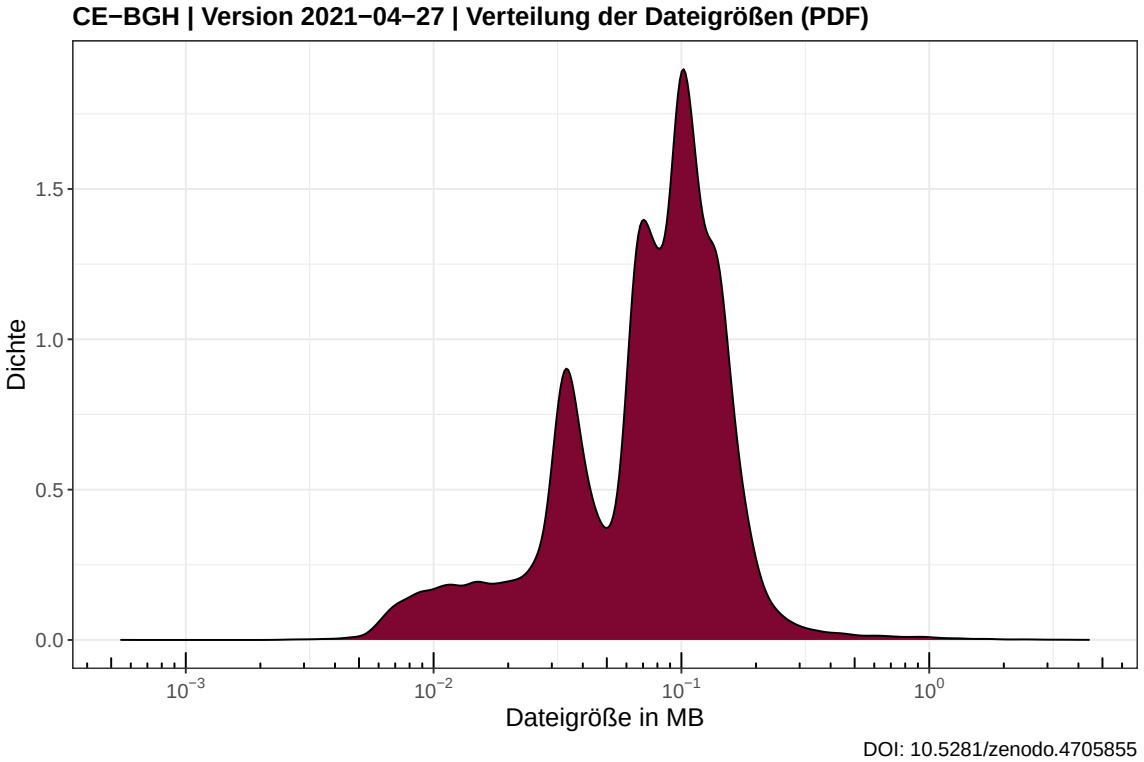
```
files.txt <- list.files(pattern = "\\.txt$",
                        ignore.case = TRUE)

txt.MB <- file.size(files.txt) / 10^6
sum(txt.MB)
```

```
## [1] 766.4664
```

15.2 Diagramm: Verteilung der Dateigrößen (PDF)

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

```
ggplot(data = dt.plot,
       aes(x = pdf.MB)) +
    geom_density(fill = "#7e0731") +
    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)
    )
```

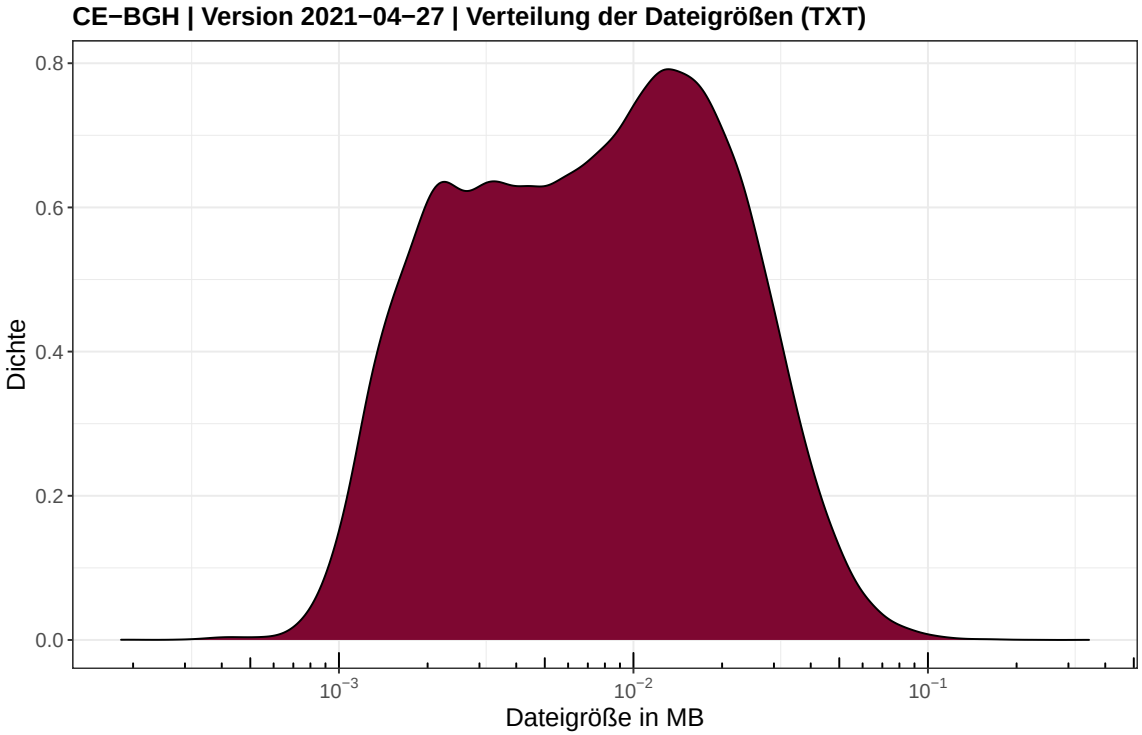


DOI: 10.5281/zenodo.4705855

15.3 Diagramm: Verteilung der Dateigrößen (TXT)

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

```
ggplot(data = dt.plot,
       aes(x = txt.MB)) +
    geom_density(fill = "#7e0731") +
    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 (TXT)"),
        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.4705855

16 Erstellen der ZIP-Archive

16.1 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)


unlink(csvname.meta)
```

16.2 Verpacken der PDF-Dateien

16.2.1 Nur Leitsatz-Entscheidungen

```
files.leitsatz <- gsub("\\.txt",
                       "\\.pdf",
                       txt.bgh[leitsatz == "LE"]$doc_id)
```

```
zip(paste(datasetname,
          datestamp,
          "DE_PDF_Leitsatz-Entscheidungen.zip",
          sep = "_"),
    files.leitsatz)
```

16.2.2 Nur Benannte Entscheidungen

```
files.benannt <- gsub("\\.txt",
                      "\\.pdf",
                      txt.bgh[is.na(name) == FALSE]$doc_id)
```

```
zip(paste(datasetname,
          datestamp,
          "DE_PDF_Entscheidungen-mit-Namen.zip",
          sep = "_"),
    files.benannt)
```

16.2.3 Alle Entscheidungen

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

unlink(files.pdf)
```

16.3 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)
```

16.4 Verpacken der Analyse-Dateien

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

16.5 Verpacken der Source-Dateien

```
files.source <- c(list.files(pattern = "Source"),
                  "buttons")
```

```
files.source <- grep("spin",
                     files.source,
                     value = TRUE,
                     ignore.case = TRUE,
                     invert = TRUE)

zip(paste(datasetname,
          datestamp,
          "Source_Files.zip",
          sep = "_"),
    files.source)
```

17 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.

17.1 Liste der ZIP-Archive erstellen

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

17.2 Funktion anzeigen

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

function(x, threads = detectCores()){

```
print(paste("Parallel processing using", threads, "threads."))

begin <- Sys.time()

cl <- makeForkCluster(threads)
registerDoParallel(cl)

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)
                        }
stopCluster(cl)

end <- Sys.time()
duration <- end - begin

print(paste0("Processed ",
             length(x),
             " files. Runtime was ",
             round(duration,
                   digits = 2),
             " ",
             attributes(duration)$units,
             "."))

return(multihashes)
```

}

17.3 Hashes berechnen

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

```
## [1] "Parallel processing using 16 threads."
## [1] "Processed 8 files. Runtime was 24.36 secs."
```

17.4 In Data Table umwandeln

```
setDT(multihashes)
```

17.5 Index hinzufügen

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

17.6 In Datei schreiben

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

17.7 Leerzeichen hinzufügen um Zeilenumbruch zu ermöglichen

Hierbei handelt es sich lediglich um eine optische Notwendigkeit. Die normale 128 Zeichen lange Zeichenfolge wird ansonsten nicht umgebrochen und verschwindet über die Seitengrenze. Das Leerzeichen erlaubt den automatischen Zeilenumbruch und damit einen für Menschen sinnvoll lesbaren Abdruck im Codebook. Diese Variante wird nur zur Anzeige verwendet und danach verworfen.

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

17.8 In Bericht anzeigen

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

index	filename
1	CE-BGH_2021-04-27_DE_ANALYSE.zip
2	CE-BGH_2021-04-27_DE_CSV_Datensatz.zip
3	CE-BGH_2021-04-27_DE_CSV_Metadaten.zip
4	CE-BGH_2021-04-27_DE_PDF_Datensatz.zip
5	CE-BGH_2021-04-27_DE_PDF_Entscheidungen-mit-Namen.zip
6	CE-BGH_2021-04-27_DE_PDF_Leitsatz-Entscheidungen.zip
7	CE-BGH_2021-04-27_DE_TXT_Datensatz.zip
8	CE-BGH_2021-04-27_Source_Files.zip

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

index	sha2.256
1	a540aaec91c760ce93ae76e6a84cb898ca25b0cd380e3d3cd389bd2c436f4e97
2	82b866042cc6def97d64cc19f75cdd30538d87aee5b2be1855b25cad38d20c65
3	d9ff34386e7b63fd38bb450717a16073d690a27d46ca5949a918029412fc4611
4	20cca0295a23a236193f5241c6efde1827cd1c45c99e2947fa9d97b3852c5a74
5	a1b2ff30044f543086d6dd25195dbd6b70f9478923bb03bf972ea31c665e0c39
6	3cdb12f50f246ae3335e1409698783c8458e8359ccf56ba27a1d3c7d1bf4b1b4
7	d07ffff8b906d3b084c461acaf456f3aacbb4eb4df23dec8791f5366ad23dec6
8	fb25f7e64f06eb8126b85a354f11897679bb0029553ee085d978fc4be0118897

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

index	sha3.512
1	5320df555c2e93635bc96a1bcb8620c8c72c778d8c1accb8324b20185072d320 7a28dc718b2e6aeefe89d9dc93d6e7f5f35f1ab03861e150ab3f6f07d8e95a02
2	5523ba5607ff712680cbd99a4edac503500891df2b986ef682befcf37b9834e3 118cd16dc2dd27519604e95e590774327888db9414540b2c78746e3f6e06ad82
3	01001a4da176612c70018daa1375160181e99259ad74eca35a193d0acf524920 342a74f6f7f5239940d0290a8c1d7bd0d9f11c090f7640f12dc3fbc80a700ef7
4	51a00c7a1a96fd997600e445412af551db0a8771e3ae8cb2700b95c0909f0ffd 95f406f15c9b981b7420adeda308a3046d3dd62bd6d130bb81c4707e52721d93
5	46edf055de71b319241647a719ace76b705f8d86c2ebf634def216d5a0d9ef32 bd93830a4a7481de12b8a99ea06a760b46e8e371dff9c309fa543d69c2bb4d08
6	f5622fe70303a2d71e77eb0a9a5bb3a59913ddc27423030fbd0ab5fdae62d331 5e8427f812948bce292cd4ea9a843d481fe7ed2904aac5aae0074410b120582e

7	f694ef39229758cdc4ca2d52a2ed3a40a2d7f4b2eb2c61f1bc54bcf0e2bf7bca 1b2f90bbbb04e1e89e821501ef443b7bb3399a19ecb896ea55a8ae8e94e6ae0a
8	555c93a182366d8adc48ec88fefc68bab584b33d1c1522e414de0918b80f34eb dd934aaf74b7404db1bf4e94c9439de77250adf7fd7d27ebf877df26edbb1648

18 Abschluss

18.1 Datumsstempel

```
print(datestamp)
```

```
## [1] "2021-04-27"
```

18.2 Datum und Uhrzeit (Anfang)

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

```
## [1] "2021-04-27 03:49:07 CEST"
```

18.3 Datum und Uhrzeit (Ende)

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

```
## [1] "2021-04-27 12:20:23 CEST"
```

18.4 Laufzeit des gesamten Skriptes

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

```
## Time difference of 8.520989 hours
```

18.5 Warnungen

```
warnings()
```

19 Parameter für strenge Replikationen

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

```
## [1] "OpenSSL 1.1.1k  FIPS 25 Mar 2021"
```

```
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 33 (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     data.table_1.14.0 scales_1.1.1
##  [5] ggplot2_3.3.3     doParallel_1.0.16 iterators_1.0.13  foreach_1.5.1
##  [9] pdftools_2.3.1    kableExtra_1.3.4  knitr_1.31        rvest_1.0.0
## [13] httr_1.4.2        mgsub_1.7.2       fs_1.5.0
##
## loaded via a namespace (and not attached):
##  [1] qpdf_1.1           tidyselect_1.1.0   xfun_0.22          purrr_0.3.4
##  [5] lattice_0.20-41    colorspace_2.0-0   vctrs_0.3.6        generics_0.1.0
##  [9] htmltools_0.5.1.1  viridisLite_0.3.0  yaml_2.2.1         utf8_1.2.1
## [13] rlang_0.4.10       pillar_1.5.1       glue_1.4.2         withr_2.4.1
## [17] selectr_0.4-2      lifecycle_1.0.0    stringr_1.4.0      munsell_0.5.0
## [21] gtable_0.3.0       codetools_0.2-18   evaluate_0.14      labeling_0.4.2
## [25] curl_4.3           fansi_0.4.2        highr_0.8          Rcpp_1.0.6
## [29] magick_2.7.1       RcppParallel_5.0.3 webshot_0.5.2      farver_2.1.0
## [33] systemfonts_1.0.1  fastmatch_1.1-0    stopwords_2.2      askpass_1.1
## [37] digest_0.6.27      stringi_1.5.3      dplyr_1.0.5        grid_4.0.4
## [41] tools_4.0.4        magrittr_2.0.1     tibble_3.1.0       crayon_1.4.1
## [45] pkgconfig_2.0.3    Matrix_1.3-2       ellipsis_0.3.1     xml2_1.3.2
## [49] rmarkdown_2.7      svglite_2.0.0      rstudioapi_0.13    R6_2.5.0
## [53] compiler_4.0.4
```

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. 2021. *Data.table: Extension of 'Data.frame'*. https://CRAN.R-project.org/package=data.table.

Ewing, Mark. 2020. *Mgsub: Safe, Multiple, Simultaneous String Substitution*. https://CRAN.R-project.org/package=mgsub.

Hester, Jim, and Hadley Wickham. 2020. *Fs: Cross-Platform File System Operations Based on Libuv*. https://CRAN.R-project.org/package=fs.

Ooms, Jeroen. 2020. *Pdftools: Text Extraction, Rendering and Converting of Pdf Documents*. https://CRAN.R-project.org/package=pdftools.

R Core Team. 2021. *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.

———. 2020. *Httr: Tools for Working with Urls and Http*. https://CRAN.R-project.org/package=httr.

———. 2021. *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, 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/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

Zhu, Hao. 2021. *KableExtra: Construct Complex Table with Kable and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.