Übungsblatt 2 Lösung

Christian Kahmann

12 10 2020

```
require(XML)
require(RCurl)
require(stringi)
library(tm)
library(slam)
library(proxy)
library(cluster)
library(wordcloud)
library(dplyr)
library(quanteda)
```

Vorbereitung

Viele Funktionen, die R bietet, sind in sogenannte *Pakete* organisiert. Zum Beispiel werden für die Aufgabe 1 die Pakete `RCurl` und `XML` benötigt. Sie können mit folgendem Befehl installiert werden:

```
install.packages(c('RCurl', 'XML'))
```

Hinweis: Für die korrekte Installation von `RCurl` brauchen Sie weitere Dependenzen (z.B. `libcurl4-openssl-dev` auf einem Debian/Ubuntu System). Beachten Sie die Hinweise in der R-Konsole.

Textgewinnung aus RSS-Feeds

Aufgabe 1

Schreiben Sie die Funktion `parse.rss(url)`, die ein RSS-Feed herunterlädt, parst und als `data.frame` darstellt. Benutzen Sie die Funktion, um das RSS-feed der "Tagesschau" herunterzuladen.

Hinweis: Benutzen Sie die Funktion `getURL` aus dem Paket `RCurl`, um das Feed herunterzuladen, sowie `xmlParse` und `xmlToDataFrame` aus `XML`, um die Daten zu extrahieren.

```
parse.rss <- function(url) {
  doc <- xmlParse(getURL(url, .encoding="UTF-8"))
  df <- xmlToDataFrame(doc["//item"], stringsAsFactors=FALSE)
  return(df)
}
```

```
rss <- parse.rss("http://www.tagesschau.de/newsticker.rdf")
rss <- rss[grep("tagesschau.de", rss$link),]
names(rss)
```

```
## [1] "title"       "link"        "description" "guid"        "category"
```

```
head(rss$title, 5)
```

```
## [1] "Was hat der \"Lockdown-light\" gebracht?"
## [2] "Vor Corona-Gipfel mit Merkel: Was die Länder planen"
```

```
## [3] "Liveblog: ++ Bundesweite Daten zur Lage an Schulen geplant ++"
## [4] "Corona-Fälle in Pflegeheimen häufen sich"
## [5] "Impfstoff von AstraZeneca und Uni Oxford wirkt zu 70 Prozent"
```

```
head(rss$link, 5)
```

```
## [1] "https://www.tagesschau.de/faktenfinder/lockdown-light-103.html"
## [2] "https://www.tagesschau.de/inland/corona-plan-bundeslaender-101.html"
## [3] "https://www.tagesschau.de/newsticker/liveblog-coronavirus-montag-157.html"
## [4] "https://www.tagesschau.de/inland/corona-pflegeheime-103.html"
## [5] "https://www.tagesschau.de/ausland/impfstoff-astrazeneca-103.html"
```

Aufgabe 2

Schreiben Sie die Funktion `get.page(url)`, die eine HTML-Seite herunterlädt und den Text aus den `<p>`-Tags extrahiert.

Hinweis: Benutzen Sie die Funktion `htmlParse` aus dem Paket `XML` ähnlich, wie in der vorigen Aufgabe.

```
get.page <- function(url) {
  doc <- htmlParse(getURL(url))
  paragraphs <- lapply(doc["//p"], xmlValue)
  text <- stri_flatten(paragraphs, collapse="\n")
  text <- stri_replace_all_regex(text, "<.+?>", " ")
  return(text)
}
```

Korpus-Erstellung

Aufgabe 3

Wenden Sie die Funktionen von Aufgaben 1-2 an, um aus den heutigen Nachrichtenartikeln ein Korpus zu erstellen:

- extrahieren Sie die URLs der Artikel aus dem RSS-Feed,
- laden Sie die Artikeltexte herunter,
- erstellen Sie aus den heruntergeladenen Texten ein Objekt der Klasse `corpus` aus dem Paket `quanteda`.

```
text <- sapply(rss$link, get.page)
```

```
#corpus <- VCorpus(VectorSource(text))
corpus<-quanteda::corpus(x = text)
```

```
summary(corpus)
```

```
## Corpus consisting of 33 documents:
##
##                                                                                 Text
##                            https://www.tagesschau.de/faktenfinder/lockdown-light-103.html
##                         https://www.tagesschau.de/inland/corona-plan-bundeslaender-101.html
##                    https://www.tagesschau.de/newsticker/liveblog-coronavirus-montag-157.html
##                                https://www.tagesschau.de/inland/corona-pflegeheime-103.html
##                             https://www.tagesschau.de/ausland/impfstoff-astrazeneca-103.html
##                         https://www.tagesschau.de/inland/coronavirus-spahn-impfungen-101.html
##                             https://www.tagesschau.de/ausland/asien-corona-strategie-101.html
```

```
##                                https://www.tagesschau.de/wirtschaft/e-health-corona-101.html
##                       https://www.tagesschau.de/inland/coronavirus-karte-deutschland-101.html
##                                    https://www.tagesschau.de/ausland/coronavirus-karte-101.html
##                           https://www.tagesschau.de/ausland/netanyahu-saudi-arabien-101.html
##                                    https://www.tagesschau.de/ausland/brexit-gespraeche-115.html
##                                      https://www.tagesschau.de/ausland/prozess-sarkozy-101.html
##          https://www.tagesschau.de/ausland/bundeswehr-untersuchung-tuerkei-gestoppt-101.html
##                           https://www.tagesschau.de/wirtschaft/bahn-wasserstoff-zug-101.html
##                                      https://www.tagesschau.de/wirtschaft/airbus-mdax-101.html
##                       https://www.tagesschau.de/ausland/hongkong-joshua-wong-prozess-101.html
##                           https://www.tagesschau.de/ausland/china-internet-konferenz-101.html
##                        https://www.tagesschau.de/ausland/aethiopien-tigray-offensive-101.html
##                                    https://www.tagesschau.de/ausland/wahl-burkina-faso-103.html
##  https://www.tagesschau.de/multimedia/podcasts/malangenommen-bruttonationalglueck-101.html
##                                  https://www.tagesschau.de/inland/corona-impfstoff-faq-101.html
##                                     https://www.tagesschau.de/inland/tsvorzwanzigjahren100.html
##                                                  https://www.tagesschau.de/inland/spahn-465.html
##                       https://www.tagesschau.de/inland/corona-massnahmen-deutschland-115.html
##                                     https://www.tagesschau.de/inland/gruenen-parteitag-141.html
##                                    https://www.tagesschau.de/ausland/usa-biden-blinken-101.html
##                        https://www.tagesschau.de/ausland/corona-suedtirol-massentest-101.html
##         https://www.tagesschau.de/investigativ/ndr-wdr/uiguren-china-menschenrechte-101.html
##                               https://www.tagesschau.de/ausland/afghanistan-konferenz-113.html
##                                https://www.tagesschau.de/wirtschaft/corona-impfstoffe-105.html
##                  https://www.tagesschau.de/sport/sportschau/bundesliga-berlin-koeln-101.html
##                               https://www.tagesschau.de/sport/sportschau/freiburg-mainz-101.html
##  Types Tokens Sentences
##    530   1095        59
##    783   1807        98
##     78    226         5
##    585   1183        60
##    239    371        19
##    648   1855        86
##    473    918        56
##    405    751        39
##    245    479        21
##    296    598        31
##    236    378        21
##    423    806        33
##    423    748        40
##    267    466        26
##    361    636        40
##    350    702        38
##    389    694        35
##    384    732        35
##    326    637        40
##    506   1039        48
##    261    492        22
##   1049   2568       137
##    114    464        24
##    530   1055        51
##    610   1341        64
##    612   1279        64
##    279    464        30
```

```
##    196    303        20
##    495    988        43
##    353    663        31
##    592   1392        81
##    299    501        29
##    344    643        41
##
## Source: /home/christian/Schreibtisch/Text Mining WS2021/Text Mining Übung/Übung 3/* on x86_64 by chri
## Created: Mon Nov 23 14:17:22 2020
## Notes:
```

Aufgabe 4

Wenden Sie die vom `quanteda`-Paket angebotenen Vorverarbeitungsschritte auf das Korpus an: `remove_punct`, `remove_numbers` und `remove_symbols`. Benutzen Sie die Funktion `tokens`, um die Operationen auf das ganze Korpus anzuwenden. Erstellen Sie aus dem Korpus eine Dokument-Term-Matrix.

Pipe

%>% nennt sich Forward-Pipe und erhöt die Lesbarkeit von R Code. Statt f(x, ...) schreibt man nun x %>% f(...).

```
# Beispiel
x <- c(0.109, 0.359, 0.63, 0.996, 0.515, 0.142, 0.017, 0.829, 0.907)
y1<-round(exp(diff(log(x))), 1)

y2<- x %>% log() %>%
    diff() %>%
    exp() %>%
    round(1)
y1==y2
```

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

```
corpus_tokens <- corpus %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE, remove_symbols = TRUE)

# Alternative ohne Pipe:
#corpus_tokens<-tokens(x = corpus,remove_punct=T, remove_numbers=T,remove_symbols=T)
```

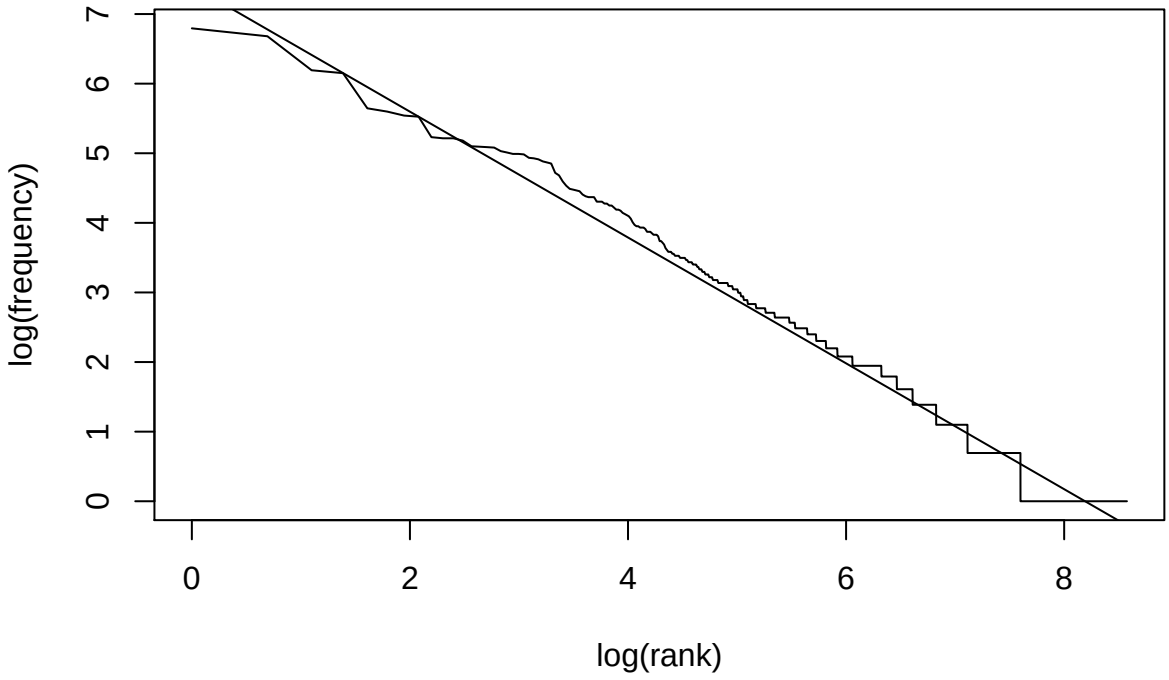
Worthäufigkeiten

Aufgabe 5

Visualisieren Sie das Zipfsche Gesetz für das Korpus.

Hinweis: Benutzen Sie die Funktion `Zipf_plot` aus dem Paket `tm`.

```
dfm<-dfm(corpus_tokens)
Zipf_plot(as.matrix(dfm))
```




```
## (Intercept)           x
##   7.4057273  -0.9040865
```

Aufgabe 6

Visualisieren Sie eins von den Dokumenten als eine Wortwolke.

Hinweis: Benutzen Sie die Funktion `textplot_wordcloud` aus dem Paket `quanteda`.

```
textplot_wordcloud(
  x=dfm[1,],
  max_words = 200,
  min_size = 1,
  random_color = T,
  color = randomcoloR::randomColor(5, luminosity="dark")
)
```


```
# Ohne Stopworte
stopwords_de<-tm::stopwords(kind="de")
dfm<-dfm[,-which(colnames(dfm)%in%stopwords_de)]
textplot_wordcloud(
  x=dfm[1,],
  max_words = 200,
  min_size = 2,
  random_color = T,
  color = randomcoloR::randomColor(5, luminosity="dark")
)
```


angesichts
weitere
umstritten
anfang
wurde
inzidenz
länder
kaum
r−wert
mitte
maßnahmen
seit
zahl
dass
ende
weiterhin
mehr
oktober
bereits
merkel
schule
schulen
neuinfektionen
viele
mittlerweile
corona−pandemie
daten
liegt
fälle
zudem
fast
infektionen
hoch