Übung 3

Christian Kahmann

13 10 2020

```
library(quanteda)
library(slam)
library(plotly)
```

Vorverarbeitung

In diesem Übungsblatt benutzen wir wieder das Tagesschau-Korpus. Mit folgendem Code lesen wir das Korpus und die Metadaten ein und erstellen eine Dokument-Term-Matrix:

```
texts <- readtext::readtext(file = "tagesschau_corpus/text/")
corpus <- quanteda::corpus(x = texts)
corpus <- corpus %>%
  tokens(remove_punct = TRUE, remove_numbers = TRUE, remove_symbols = TRUE) %>%
  tokens_select(pattern = stopwords("de"),selection = "remove")

DTM<-dfm(corpus)
```

```
articles <- read.csv('tagesschau_corpus/links.csv',
                     header=F, sep='\t', stringsAsFactors=F)
colnames(articles) <- c('url', 'date', 'category', 'filename')
rownames(articles) <- articles$filename
articles$date <- as.Date(articles$date, "%d.%m.%Y")
```

Aufgabe 1

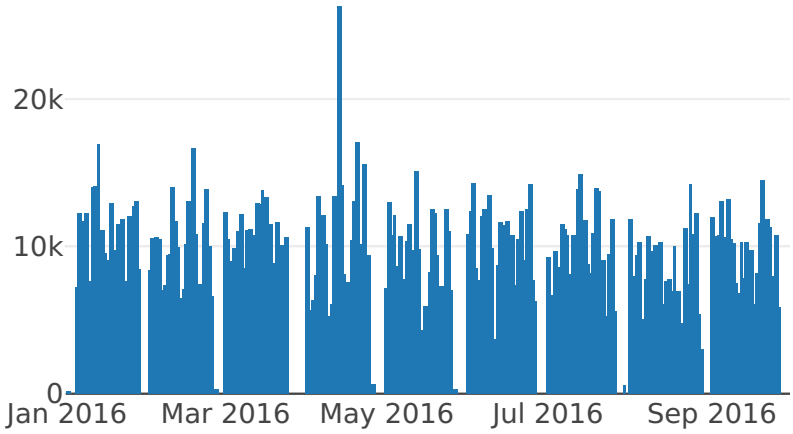
Erzeugen Sie einen Vektor `sizes`, der für jeden Tag die Textgröße (Anzahl Tokens) enthält. Visualisieren Sie diesen Vektor als Barplot.

```
dates <- seq(as.Date("2016-01-01"), as.Date("2016-09-30"), "days")
sizes <- sapply(dates, function(x) sum(DTM[articles[articles$date==x,"filename"],]))
names(sizes) <- dates
```

```
sizes[1:10]
```

```
## 2016-01-01 2016-01-02 2016-01-03 2016-01-04 2016-01-05 2016-01-06
##        108          0          0          0       7244      12254
## 2016-01-07 2016-01-08 2016-01-09 2016-01-10
##      11728      12270       7647       6799
```

```
plot_ly(x=dates, y=sizes, type="bar")
```



Aufgabe 2

Schreiben Sie die Funktion `daily.freqs`, die für ausgewählte Terme tägliche Häufigkeiten liefert. Das Ergebnis soll eine Martix sein mit einer Zeile für jeden Term und einer Spalte für jeden Tag.

```
daily.freqs <- function(terms) {
  dates <- seq(as.Date("2016-01-01"), as.Date("2016-09-30"), "days")
  freqs <- sapply(dates, function(x) colSums(DTM[articles[articles$date==x,"filename"],terms]))
  if (length(terms) == 1) {
    freqs <- t(as.matrix(freqs))
  }
  colnames(freqs) <- as.character(dates)
  rownames(freqs) <- terms
  freqs
}
```

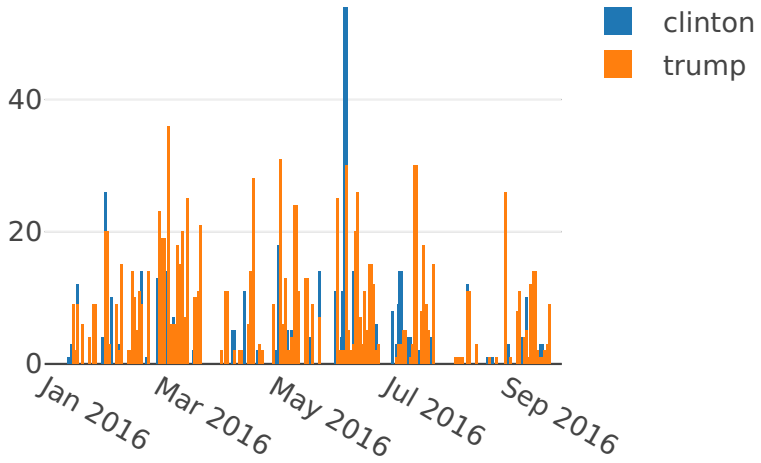
```
daily.freqs(c("clinton", "trump"))[,100:110]
```

```
##         2016-04-09 2016-04-10 2016-04-11 2016-04-12 2016-04-13 2016-04-14
## clinton          0          5          0          0          0          0
## trump            0          2          0          0          2          0
##         2016-04-15 2016-04-16 2016-04-17 2016-04-18 2016-04-19
## clinton         11          0          0          0          8
## trump            0          0          6          0         14
```

Aufgabe 3

Visualisieren Sie als Barplot die Tageshäufigkeiten von `trump` (rot) und `clinton` (blau).

```
freqs <- daily.freqs(c("clinton", "trump"))
plot_ly(x=dates, y=freqs[1,], name="clinton", type="bar") %>%
  add_trace(y=freqs[2,],name="trump")
```

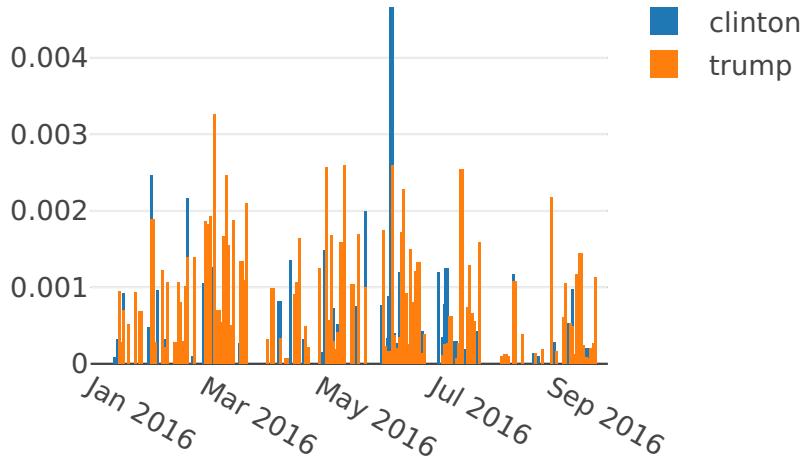


Aufgabe 4

Wiederholen Sie die vorige Aufgabe mit relativen Häufigkeiten.

Hinweis: benutzen Sie den Vektor `sizes` von der Aufgabe 1.

```
freqs <- daily.freqs(c("clinton", "trump"))
relfreqs <- t(t(freqs)/sizes)
relfreqs[is.nan(relfreqs)] <- 0
relfreqs[is.infinite(relfreqs)] <- 0
plot_ly(x=dates, y=relfreqs[1,], name="clinton", type="bar") %>%
  add_trace(y=relfreqs[2,],name="trump")
```



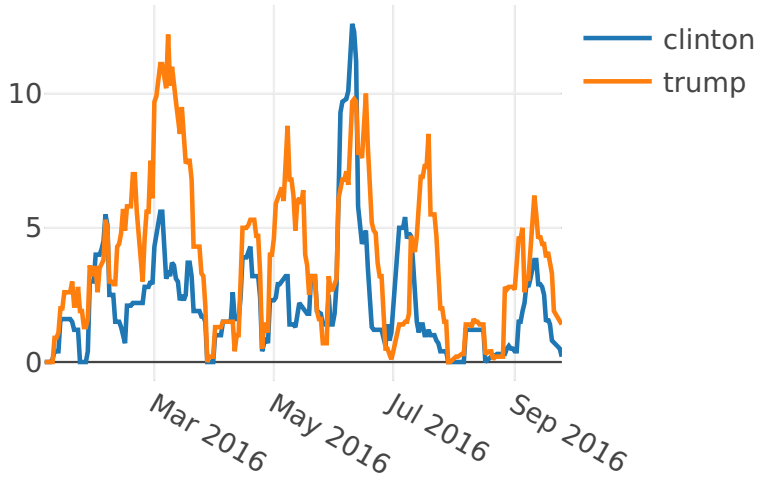
Aufgabe 5

Um längerfristige Trends in Zeitverläufen zu beobachten, benutzen wir den *gleitenden Durchschnitt*, d.h. den Durchschnitt der Werte in der Umgebung eines Punktes.

Berechnen Sie den gleitenden Durchschnitt für die beiden Terme aus der vorigen Aufgabe und visualisieren Sie diesen als Liniendiagramm. Probieren Sie verschiedene Filter aus, z.B. `rep(1/10, 10)`.

Hinweis: benutzen Sie die Funktion `filter` in Kombination mit `apply`, um den gleitenden Durchschnitt zu berechnen.

```
filter <- rep(1/10, 10)
#filter <- c(seq(1, 10)/sum(seq(1,10))/2, seq(10, 1)/sum(seq(10,1))/2)
freqs_fil <- t(apply(freqs, 1, function(x) stats::filter(x, filter=filter)))
plot_ly(x = dates, y = freqs_fil["clinton",],type="scatter",mode="lines",name="clinton") %>%
  add_trace(y = freqs_fil["trump",],name="trump")
```

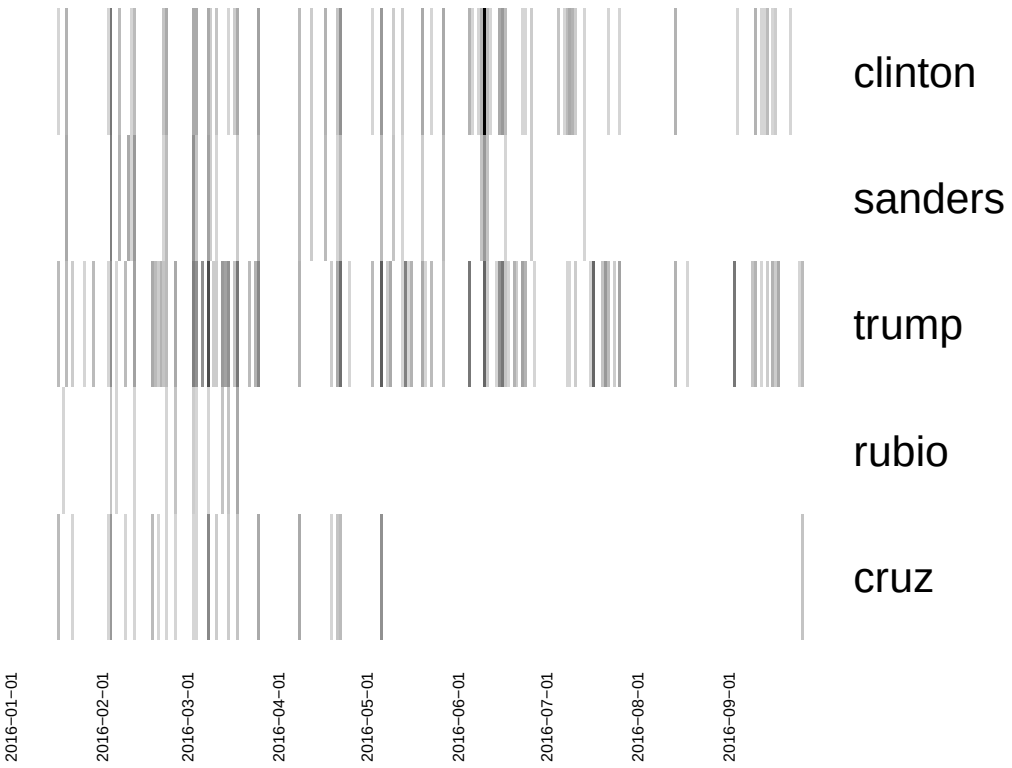


Aufgabe 6

Visualisieren Sie die Tageshäufigkeit von ausgewählten Termen im Korpus als Heatmap (siehe Grafik).

Hinweis: benutzen Sie die Funktion `heatmap.2` aus dem Paket `gplots`.

```
freqs <- daily.freqs(c("clinton", "sanders", "trump", "rubio", "cruz"))
labels <- dates
labels[(dates %in% seq(as.Date("2016-01-01"), as.Date("2016-09-30"), "months")) == FALSE] <- NA
gplots::heatmap.2(freqs, Colv=F, Rowv=F, key=F, trace='none', dendrogram='none', col=gray(c(30, 25:0)/3
```


clinton
sanders
trump
rubio
cruz
2016−01−01
2016−02−01
2016−03−01
2016−04−01
2016−05−01
2016−06−01
2016−07−01
2016−08−01
2016−09−01