

[[Tittel]]

[[Navn // Kand.nr]]

2022-08-24

R markdown

Dette er et R-markdown-dokument (.Rmd). Her kan vi enkelt implementere kode og resultater rundt teksten vi skriver. Dette gjøres via *chunks*:

```
'''{r eks_chunk, echo=TRUE}

klossmajor <- readLines("skjoente_ikke_du_var_borte.txt")

klossmajor[1]
'''
```

```
## [1] "Jeg mener alt i denne sangen"
```

Her har jeg satt `echo=TRUE` i chunk headeren. Det gjør at koden blir repetert i selve dokumentet. Det er sjelden vi ønsker det når vi skal skrive en artikkel/oppgave. Derfor kan det være lurt å sette `knitr::opts_chunk$set(echo = FALSE)` i en oppsett-chunk i starten av dokumentet.

Lage plots

Det er enkelt å visualisere resultat fra data i Rmarkdown:

```
'''{r ploteks, out.width="0.75\\textwidth"}
library(ggplot2)

ggplot(NULL, aes(fill = sample(letters[1:3], 200, replace = TRUE),
                      x = rnorm(200))) +
  geom_density(alpha = .35) +
  theme_bw() +
  labs(x = "Tilfeldig normalfordelte tall",
       y = "Tetthet (density)",
       fill = "Bokstaver")
'''
```

Laste kode fra .R-filer

Man kan også lage separate R-skript som man kan hente inn i .Rmd-fila. Dette kan gjøre at dokumentet blir noe mer oversiktlig:

```
'''{r hente_r_skript, file = "./tokenize_klossmajor.R"}

'''
```

Modellering som tar lang tid

Noen kodebiter fungerer dårlig å kjøre hver gang man skal compilere et .Rmd-dokument. Det mest åpenbare eksempelet er tunge modelleringer eller iterative modelleringer (som emnemodeller). Det beste trikset for å løse dette er å lagre modellen lokalt og laste denne inn i en egen chunk. For eksempel, kan man ha en chunk som ikke kjører, men viser koden:

```
data(cases, package = "stortingscrape")

cases <- cases$root

library(tidytext)
library(dplyr)
library(stm)

cases_dfm <- cases %>% group_by(id) %>%
  unnest_tokens(token, title) %>%
  count(token) %>%
  cast_dfm(., id, token, n)

cases_stm <- stm(cases_dfm, K = 10, init.type = "Spectral")
# save(cases_stm, file = "./cases_stm.rda")

plot(cases_stm, labeltype = "frex")
```

Og en chunk som laster inn modellen, men ikke viser koden:

```
''{r usynlig_modellaster, eval=TRUE, echo=FALSE}
library(tidytext)
library(dplyr)
library(stm)

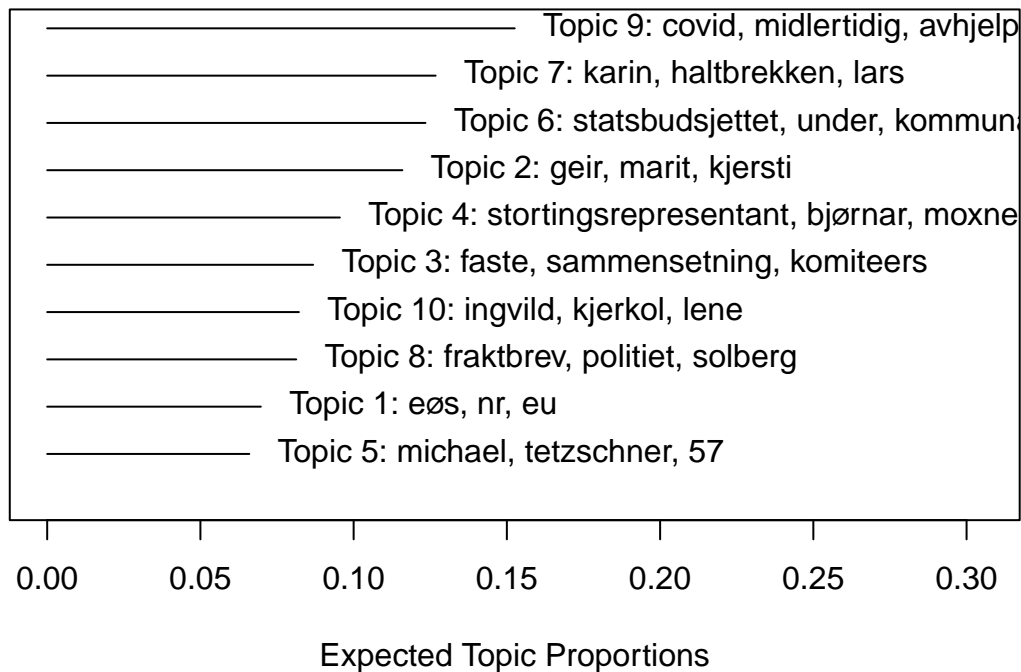
data(cases, package = "stortingscrape")

cases <- cases$root

cases_dfm <- cases %>% group_by(id) %>%
  unnest_tokens(token, title) %>%
  count(token) %>%
  cast_dfm(., id, token, n)

load("./cases_stm.rda")
plot(cases_stm, labeltype = "frex")
''
```

Top Topics



Hurtigtaster

For å kompilere en .Rmd-fil kan man enten trykke på “Knit”-knappen i RStudio eller bruke hurtigtasten `ctrl/cmd + shift + k`. Øvrige hurtigtaster kan man finne i [rmarkdown-cookbook](#).

Referanser og referanselister

Om du vil ha med autogeneratede referanselister basert på kildene du siterer, kan man gjøre dette via .bib-systemet. Spesifiser da helt i toppen av .Rmd-filen:

```
bibliography: filbane/til/referanser.bib
csl: filbane/til/american-political-science-association.csl
```

.bib-filen inneholder et referansebibliotek. Du kan gjerne starte med [biblioteket vi bruker i STV2022](#). Men det kan også være lurt å bruke et program som er egnet til å legge inn og systematisere referansene dine. Selv bruker jeg [JabRef](#), men andre alternativer er feks [Zotero](#), [Paperpile](#), [Mendeley](#) og [EndNote](#).

`american-political-science-association.csl` spesifiserer hvilken stil vi ønsker å referere i. I dette tilfellet [APSA](#) sin.

Du kan føre en referanse med følgende koder:

1. @Grimmer2022 → Grimmer, Roberts, and Stewart (2022)
2. [Grimmer2022] → (Grimmer, Roberts, and Stewart 2022)
3. [Grimmer2022, p.22] → (Grimmer, Roberts, and Stewart 2022, 22)
4. [Grimmer2022, @Silge2017] → Silge and Robinson (2017)

Referanser (autogenerert liste)

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton University Press.

Silge, Julia, and David Robinson. 2017. *Text Mining with R: A Tidy Approach*. O'Reilly Media, Inc.
<https://www.tidytextmining.com/>.