

Nemt Danmarkskort i R

Aleksander Bang-Larsen

2024-06-29

Indholdsfortegnelse

Baggrund	1
Pakker	1
Data	2
Optimering af data	3
Visualisering	3
Visualisering af udsnit	4
Samlet kode	5

Baggrund

Da jeg skulle lære at lave kort i R, lavede jeg et par hurtige googlesøgninger, hvor jeg kom frem til en god guide lavet af Mikkel Freltoft Krogsholm (2021). Jeg opdagede dog hurtigt, at guiden er out-of-date - og derfor skriver jeg denne guide.

I denne guide vil jeg gennemgå hvordan man kan skabe et kort over alle afstemningssteders områder i Danmark. Vi vil både producere et kort for Aarhus Kommune (Figur 2) og et kort for alle afstemningsområder i hele Danmark (Figur 1).

Pakker

I R-universet findes der flere pakker til at arbejde med kort. Jeg har valgt at bruge `{sf}` som den pakke jeg sætter mig ind i, hvad kan. De fleste kan grundlæggende det samme og er bygget

op omkring **geometry**, der er den kolonne (eller variabel) i en dataframe, der indeholder figurene til kortet.

Først indlæser vi alle pakker der skal bruges.

```
library(tidyverse)      # Her får vi %>%, ggplot2 og andre smarte funktioner.  
library(sf)             # Skal bruges til at arbejde med "simple features" (figurer).
```

Data

For at tegne et præcist danmarkskort kan vi hente **geojson** data fra Styrelsen for Dataforsyning og Infrastruktur (2024). De udstiller en udmærket **API** der kan levere kortdata til os. For at vi kan benytte en **API** i R skal vi først definere et **URL** og dernæst bede om at downloade den fil der hører til på den hjemmeside.

1. Først definerer vi **API**ens **URL**.

```
## Gemmer URL til API-kald  
url <- "https://api.dataforsyningen.dk/afstemningsomraader?format=geojson"
```

2. Dernæst beder vi **R** om at downloade den efterspurgte fil til en midlertidig placering i computerens hukommelse.

Du behøver *ikke* at vide hvad en **API** er for at kunne gennemføre denne guide.

```
# Skaber midlertidig fil  
geofile <- tempfile()  
  
# Henter geojson til tempfile  
download.file(url, geofile)
```

Vi bruger funktionen **download.file()** og giver den **API**ens **URL** og den midlertidige dataplacering, hvor vi ønsker dataene hentet til.

3. Til sidst omformer vi **geoJSON** til et **sf** format som **R** kan forstå.

```
# Læser datafilen ind i R
geodata_st <- st_read(geofile)
afstemningssteder_geodata <- st_as_sf(geodata_st)
```

Nu har vi en dataframe i `afstemningssteder_geodata` der indeholder de figurer vi skal bruge for at lave et danmarkskort!

Her omformes først til `st` og dernæst til `sf`.

Optimering af data

For at vi nemt kan renderes vores plots er det en god ide at gøre figurerne en smule mindre. *Især* når vi er helt nede på afstemningsstedsniveau. Det er nok ikke så relevant, hvis vi arbejder med kommuner eller landsdele.

```
afstemningssteder_geodata <- rmapshaper::ms_simplify(afstemningssteder_geodata,
keep = 0.01, keep_shapes = TRUE)
```

Visualisering

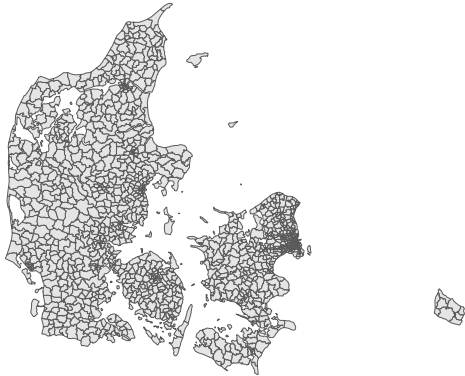
Nu kommer vi til den sjove del - At lave selve kortet! Vi skal bruge `{ggplot2}` fra det velkendte *tidyverse*, præcis som når vi laver grafer i mange andre sammenhænge.

Vi bruger `geom_sf`, der er indbygget i `{ggplot2}` til at tegne de figurer der er opbevaret i vores dataframe. Jeg putter derudover også `ggthemes::theme_map()` på, fordi jeg ikke synes kort behøver akselinjer og en baggrund. `theme_map()` er mit klart foretrukne tema til kort.

Her indlæser vi `rmapshaper::ms_simplify()` med pakkens navn for at slippe for at indlæse hele pakken - Vi skal nemlig ikke bruge det hele! Det samme har vi gjort med `ggthemes::theme_map()` nedenfor, fordi der kun skal bruges det ene *theme*

```
ggplot(afstemningssteder_geodata) +
  geom_sf() +
  ggthemes::theme_map() +
  labs(title = "Afstemningssteder i Danmark",
       caption = "Kilde: DAWA/DAGI") +
  theme(legend.position = "none",
        plot.title = element_text(size = 20),
        plot.caption = element_text(size = 10))
```

Afstemningssteder i Danmark



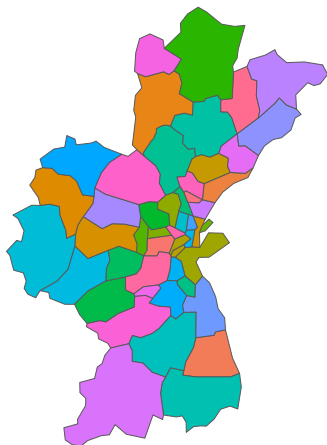
Kilde: DAWA/DAGI

Figur 1: Kort over alle afstemningsområder i hele Danmark. Kilde og titel tilføjet.

Visualisering af udsnit

For at kun vise de afstemningsområder der ligger i Aarhus Kommune kan vi benytte os af `filter()` funktionen fra `{dplyr}` pakken. Med den kan vi filtrere i vores dataframe, så der kun vises afstemningssteder, hvor kommunenavnet er "Aarhus". Derudover har jeg tilføjet et `fill` på afstemningsområdetets navn for at give hver område sin egen farve.

```
afstemningssteder_geodata %>%  
  filter(kommunenavn == "Aarhus") %>%  
  ggplot(aes(fill = navn)) +  
  geom_sf() +  
  ggthemes::theme_map() +  
  theme(legend.position = "none")
```



Figur 2: Kort over alle afstemningsområder i Aarhus Kommune.

Samlet kode

Alt hvad jeg har gennemgået i denne guide kan findes i et samlet r-script [på min github](#). Den kan også findes [her på hjemmesiden](#).

Krogsholm, Mikkel Freltoft. 2021. “Easy Maps of Denmark in R | LinkedIn”. <https://www.linkedin.com/pulse/easy-maps-denmark-r-mikkel-freltoft-krogsholm/>.

Styrelsen for Dataforsyning og Infrastruktur. 2024. “DAWA: Danmarks Adressers Web API Hos Dataforsyningen”. <https://dawadocs.dataforsyningen.dk/dok/guides>.