

Assignment 4

```
library(PxWebApiData)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2      v purrr 0.3.4
## v tibble 3.0.4       v dplyr 1.0.2
## v tidyr 1.1.2        v stringr 1.4.0
## v readr 1.4.0        v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

knitr::opts_chunk$set(eval = TRUE)
```

Assignment 4; ssb data

Data hentet

Data sist hentet 30. november 2020. Dette dokumentet skal bare kjøres hvis nye data må hentes inn. `eval = FALSE` over i setup chunk må endres til `eval = TRUE`, så må setup chunk-en kjøres før en til slutt velger `Restart R and run all chunks` fra Run menyen.

Vektoren med de 418 kommunene vi vil bruke

I perioden 1999 t.o.m 2014 er det relativt få kommunesammenslåinger. De som er har vi ekskludert fra vårt utvalg. Vi ender opp med en liste på 418 kommuner. Kommunenummer for disse er listet under.

```
komp_vec <- c("0101", "0104", "0105", "0106", "0111", "0118", "0119", "0121", "0122", "0123", "0124", "0125")
```

```
length(komp_vec)
```

```
## [1] 418
```

1999 t.o.m. 2014 månedsbasis

Vi bruker perioden januar 1999 t.o.m. mai 2014. Perioden har relativt få kommunejusteringer og god tilgang på data. Lag en vektor `tid_vec` på formen “1999M01”, “1999M02”, “1999M03”, ... “2014M04”, “2014M05”. Bruk `c()`, `paste()`, `rep()` og `str_pad()`. Husk at `rep()` har et `each` argument. Et tips er å lage opp til 2014M12 og så fjerne de siste 7 elementene.

```
# combine med c(), rep() se argument each, str_pad()
# til å lage 01, 02...11, 12
tid_vec <- paste(rep(1999:2014, each = 12), "M", str_pad(1:12, width = 2, pad = "0"), sep = "")
tid_vec <- tid_vec[1:185]
```

```
length(tid_vec)
```

```
## [1] 185
```

```
c(head(tid_vec, n = 14), "...", tail(tid_vec, n = 14))
```

```
## [1] "1999M01" "1999M02" "1999M03" "1999M04" "1999M05" "1999M06" "1999M07"
## [8] "1999M08" "1999M09" "1999M10" "1999M11" "1999M12" "2000M01" "2000M02"
## [15] "...      "2013M04" "2013M05" "2013M06" "2013M07" "2013M08" "2013M09"
## [22] "2013M10" "2013M11" "2013M12" "2014M01" "2014M02" "2014M03" "2014M04"
## [29] "2014M05"
```

Kjønn og aldersgruppe

Vi trenger også disse som jeg har definert for dere.

```
alder_vec <- c("15-74", "15-29", "30-74")
kjonn_vec <- c("Menn", "Kvinner")
```

Tabell 10540, arbeidsløse i %, per kommune, fordelt på alderskategorier, månedlig 1991M01-2014M05

Arbeidsløshet i % for alderskategoriene “15-74 år” “15-29 år” og “30-74 år”.

Tabellen er på 418*185*3=231990 rekker som er under grensen på 800000 rekker. Kan derfor hente alt på en gang.

ApiData() returnerer en liste med to elementer, hhv. det som kalles variabel-beskrivelse og verdier. For oss er det greit å hente `region` som er kommunenavn, alder og kjønn (“Kvinner”, “Menn”) fra den første og resten fra den andre listen. Husk at du kan hente innholdet i de to (dataframe) frem med hhv. `[[1]]` og `[[2]]`.

```
arbl_p_alder <- ApiData(
  10540,
  Region = kom_vec,
  Tid = tid_vec,
  Alder = alder_vec
)
df_arbl_p_alder <- data.frame(
  knr = arbl_p_alder[[2]]$Region,
  knavn = arbl_p_alder[[1]]$region,
  alder = arbl_p_alder[[1]]$alder,
  tid = arbl_p_alder[[2]]$Tid,
  # arbeidsløse i prosent
  alp = arbl_p_alder[[2]]$value
)
dim(df_arbl_p_alder)
```

```
## [1] 231990      5
```

Tabell 04471, arbeidsløse i %, per kommune, etter kjønn, månedlig 1991M01-2014M05

Her fordelt på kjønn. Vi har $418 \cdot 185 \cdot 2 = 154660$ rekker som er under grensen på 800k. Kan derfor hente alt på en gang. Det er nå ikke lenger Alder som er statistikk variabel så den må byttes ut med Kjønn =

```
dim(df_arbl_p_kjonn)
```

```
## [1] 154660      5
```

```
names(df_arbl_p_kjonn)
```

```
## [1] "knr" "knavn" "kjonn" "tid" "alp_k"
```

Tabell 10594, Antall arbeidsløse per kommune, etter kjønn, månedlig 1991M01-2014M05

Antall rekker vil være $418 \cdot 185 \cdot 2 = 154660$.

Igjen er det Kjønn og ikke Alder som er statistikk-variabel.

```
dim(df_arbl)
```

```
## [1] 154660      5
```

```
names(df_arbl)
```

```
## [1] "knr" "knavn" "tid" "kjonn" "al"
```

Tabell 07459, Befolkning årsklasser 000-105+ for årene 1999-2014

Antall rekker blir $418 \cdot 16 \cdot 106 \cdot 2 = 1417856$ som er mer enn maks grensen på 800k. Splitter derfor på kjønn og får 708928 rekker for hvert kjønn.

Bruk `c()` og `str_pad()` til å lage en `alder_klasse_vec` på formen “000”, “001”, “002” ... “104”, “105+”. Bruk denne som verdi for Alder i `ApiData()` kallet, dvs. `Alder = alder_klasse_vec`. Bruk hhv. `Kjønn = “Kvinner”` og `Kjønn = “Menn”` for å hente de respektive datasettene.

```
alder_klasse_vec <- c(str_pad(0:104, width = 3, pad = "0"), "105+")
```

```
length(alder_klasse_vec)
```

```
## [1] 106
```

```
c(head(alder_klasse_vec, n = 5), "...", tail(alder_klasse_vec, n = 5))
```

```
## [1] "000" "001" "002" "003" "004" "... " "101" "102" "103" "104"
## [11] "105+"
```

```
# Må splitte på kjønn pga maks grense på 800000 rekker
bef_K <- ApiData(
  07459,
  Region = kom_vec,
  Tid = as.character(1999:2014),
  Alder = alder_klasse_vec,
  Kjønn = "Kvinner"
)

df_bef_K <- data.frame(
  knr = bef_K[[2]]$Region,
  knavn = bef_K[[1]]$region,
  tid = bef_K[[2]]$Tid,
  # Vil ha Kvinner, Menn ikke 2, 1 i bef_K[[2]]$Kjønn
  kjønn = bef_K[[1]]$kjønn,
  # velger alder som 000 001 etc
  alder = bef_K[[2]]$Alder,
  bef = bef_K[[2]]$value
)
```

```
dim(df_bef_K)
```

Kvinner

```
## [1] 708928      6
```

Menn Hent ned tilsvarende befolkningsdata for menn.

```
dim(df_bef_M)
```

```
## [1] 708928      6
```

Write out the dataframes

For kontroll.

```
names(df_arbl_p_alder)
```

```
## [1] "knr" "knavn" "alder" "tid" "alp"
```

```
names(df_arbl_p_kjønn)
```

```
## [1] "knr" "knavn" "kjønn" "tid" "alp_k"
```

```

names(df_arbl)

## [1] "knr" "knavn" "tid" "kjonnn" "al"

names(df_bef_K)

## [1] "knr" "knavn" "tid" "kjonnn" "alder" "bef"

names(df_bef_M)

## [1] "knr" "knavn" "tid" "kjonnn" "alder" "bef"

dim(df_arbl_p_alder)

## [1] 231990 5

dim(df_arbl_p_kjonnn)

## [1] 154660 5

dim(df_arbl)

## [1] 154660 5

dim(df_bef_K)

## [1] 708928 6

dim(df_bef_M)

## [1] 708928 6

write_csv(df_arbl_p_alder, path = "./ssb_df_arbl_p_alder.csv")

## Warning: The 'path' argument of 'write_csv()' is deprecated as of readr 1.4.0.
## Please use the 'file' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_warnings()' to see where this warning was generated.

write_csv(df_arbl_p_kjonnn, path = "./ssb_df_arbl_p_kjonnn.csv")
write_csv(df_arbl, path = "./ssb_df_arbl.csv")
write_csv(df_bef_K, path = "./ssb_df_bef_K.csv")
write_csv(df_bef_M, path = "./ssb_df_bef_M.csv")

```

Opprydding

Vi har lagret det vi har hentet og kan nå rydde opp.

```
# Fjern # fra neste hvis en vil slette alle objekter  
rm(list = ls())
```