

## Hent SSB data

```
suppressPackageStartupMessages({
  library(PxWebApiData)
  library(tidyverse)
  library(lubridate)
})
knitr::opts_chunk$set(echo=FALSE, include = FALSE)
```

```
# vector med relevante kommunenummer
load("knr.Rdata")
```

*Prøv å benytt de samme navnene på objekter og variabler som jeg har benyttet. Det gjør det langt lettere å hjelpe dere hvis dere kjører dere fast.*

Vi starter med å hente data om gjennomsnittlig kvadratmeterpris for eneboliger for perioden 2002 til 2017. Gjennomsnittet er regnet ut for hver kommune. Dataene finnes i tabell 06035. Gi resultatet navnet `pm2_raw`.

```
pm2_raw <- ApiData(
  urlToData = "06035",
  Region = knr,
  ContentsCode = "KvPris",
  Boligtype = "01",
  Tid = c(as.character(2002:2017))
)
```

Vi ønsker å benytte variabelnavnene `knr` for Region, `aar` for Tid og `pm2` for value. SSB gir som svar på vårt api-kall en liste med to element. Stort sett vil vi være interessert i det 2. elementet kalt `dataset`. Det første elementet har et langt navn der tabellnummer og tittelen på statistikken inngår. Vi står fritt til å hente variabler fra dette elementet når det er påkrevd.

- i. Hent ut `dataset` elementet fra `pm2_raw`.
- ii. Sjekk i `pm2_raw` at alle observasjonene er av Boligtype “01”.
- iii. Dropp variablene `Boligtype` og `ContentsCode`.
- iv. Skift navn på variablene Region, Tid og value som angitt ovenfor.

Vi skal også hente ut det offisielle kommunenavn fra det første elementet i `pm2_raw`. For å slippe å hanske med det uhåndterlig lange navnet på dette elementet skifter vi først navn på dette til “desc”.

```
names(pm2_raw)[[1]] <- "desc"
```

Vi legger kommunenavnene til `pm2` som variabelen `knavn`.

```
pm2 <- pm2 %>%
  mutate(
    knavn = pm2_raw$desc$region
  )
```

Vi ønsker å være politisk korrekt og benytte alle de offisielle navnene på kommunene. Etter kommunenavnene er det imidlertid angitt i parentes når de ble dannet og oppløst. Vi finner dette skjemmende og vil derfor fjerne disse parentesene. Samtidig gir det oss en mulighet til å praktisere regexp (kap. 14 i r4ds). Vi må da finne det rette mønsteret før vi så kan bruke dette til å fikse `knavn` vha. `mutate` og `str_replace`.

```
## [1] "Halden (-2019)"
## [2] "Sarpsborg (1992-2019)"
## [3] "Våler (Østfold) (-2019)"
## [4] "Haugesund"
## [5] "Porsanger - Porsángu - Porsanki (1964-2019)"
```

- i. Tenk fra høyre. Vi har linjeslutt, så ingen eller flere mellomrom, så en parentes slutt, så tall eller bindestrek, så parentes start før vi til slutt har en eller flere mellomrom. Husk at vi må benytte to backslash for escape i R.

```
load("test_string_tib.Rdata")
# Legg inn regex mønster
moenster <- '.*'

test_string_tib %>%
  mutate(
    knavn = str_replace(knavn, moenster, "")
  )
```

```
## # A tibble: 5 x 1
##   knavn
##   <chr>
## 1 ""
## 2 ""
## 3 ""
## 4 ""
## 5 ""
```

- i. Når mønsteret gir følgende svar kan du bruke det til å endre knavn i pm2.

```
## # A tibble: 5 x 1
##   knavn
##   <chr>
## 1 Halden
## 2 Sarpsborg
## 3 Våler (Østfold)
## 4 Haugesund
## 5 Porsanger - Porsángu - Porsanki
```

- i. Sjekk hvor mange NA verdier vi har i pm2 (Hint! Kjør funksjonen `is.na()` på alle kolonnene i pm2 vha. `map_df`. Tabuler output vha. `table()` til slutt i pipe.)
- ii. Sjekk hvor mange complete.cases vi har fra 2006 til 2017 (dvs. antall kommuner som har data for alle årene f.o.m. 2006 t.o.m. 2017. Svaret skal bli 197.)
- iii. Sjekk hvor mange complete.cases vi har i 2008 (dvs. antall kommuner som har data for alle årene f.o.m. 2008 t.o.m. 2017. Svaret skal bli 214.)

Vi ser at ved å velge perioden 2008-2017 istedenfor 2006-2017 får vi 17 ekstra complete.cases. Velger derfor å studere perioden 2008-2017.

Vi har altså 214 kommuner med registrert pm2 for hvert år i perioden 2008-2017. Det er disse kommunene vi skal bruke i den følgende analysen

```
# Skifter navn til knr for listen av kommune nummer vi skal bruke
# i analysen
knr <- knr_c.c
# Time to clean up
rm(knr_c.c, pm2_raw)
```

# Befolkning

## «Yrkesaktiv alder»

Vi ønsker å finne prosentandelen av befolkningen i hver kommune som er i yrkesaktiv alder. Dessverre finnes ikke variabelen direkte tilgjengelig, men fra tabell 07459 kan vi finne antall personer i alderen 20-64 år. Fra samme tabell kan vi også finne total befolkning i hver kommune (krever litt jobb med rådataene). Vi kan da finne prosent i yrkesaktiv alder ved å dividere de to størrelsene og multiplisere med 100.

Vi henter data i to omganger:

- i. Hent antall i yrkesaktiv alder fra tabell 07459 og plasser i variabelen `pop_08_17_ya_raw`. Velg at du vil ha dataene fordelt på kjønn. Vi kan regne ut total som Menn + Kvinner. La variabelen `kjonn` i `pop_08_17_ya_raw` angi kjønn og `ya` antall i yrkesaktiv alder. La `ya_Total = ya_Menn + ya_Kvinner` i `pop_08_17_ya_raw`.
- ii. Hent grunnlag for å beregne befolkning totalt. Det enklest å benytte den minst detaljerte inndelingen, 0-17 og 18+. Velg også her fordelt på kjønn. Plasser dataene i `pop_08_17_raw` og gjør denne *tidy* (bruk `pivot_wider`). Legg tidy varianten i `pop_08_17`. Bruk variabelnavnene `knr`, `kjonn`, `alder`, `aar`, `pop`. Beregn `Menn_t = Menn_H17 + Menn_H18`, `Kvinner_t = Kvinner_H17 + Kvinner_H18`, `Total_t = Menn_t + Kvinner_t` (bruk `mutate` for å definere nye variabler).

## Befolkning Menn, Kvinner og Totalt

Få med både kjønn fra element 1 og Alder fra element 2.

Under ser du hva du bør ha kommet frem til.

```
dim(pop_08_17_ya)
```

```
## [1] 2140    5
```

```
names(pop_08_17_ya)
```

```
## [1] "knr"      "aar"      "ya_Menn"  "ya_Kvinner" "ya_Total"
```

```
dim(pop_08_17)
```

```
## [1] 2140    5
```

```
names(pop_08_17)
```

```
## [1] "knr"      "aar"      "Menn_t"   "Kvinner_t" "Total_t"
```

- i. Slå sammen `pop_08_17_ya` og `pop_08_17` og legg resultatet i `pop_08_17_ya_p`.
- ii. Beregn prosentandel i yrkesaktiv alder for kvinner (`Kvinner_ya_p`), menn (`Menn_ya_p`) og totalt (`Total_ya_p`).
- iii. Velg variablene `knr`, `aar`, `Menn_ya_p`, `Kvinner_ya_p`, `Total_ya_p`

Her er starten av `pop_08_17_ya_p` for kontroll.

```
head(pop_08_17_ya_p, n = 5)
```

```
## # A tibble: 5 x 5
##   knr    aar  Menn_ya_p Kvinner_ya_p Total_ya_p
##   <chr> <chr>    <dbl>      <dbl>      <dbl>
## 1 0101  2008     59.7       56.8       58.3
## 2 0101  2009     59.8       57.0       58.4
## 3 0101  2010     59.6       57.1       58.3
## 4 0101  2011     59.8       57.2       58.5
## 5 0101  2012     59.5       57.0       58.2
```

## Legg ya variablene til pm2 tibble-en

- Rydd opp! Slett objekter som ikke lenger trengs vha. `rm()`.

## Inntektsdesiler

En desil vil være 1/10 av variabelen, der variabelen er ordnet etter størrelse. Grensen mellom øverste og nest-øverste desil for inntekt vil altså være den inntekten som gjør at du er blant de 10% med høyest inntekt på landsbasis. Hvor mange prosent av husholdningene som er over denne grensen vil selvsagt variere fra kommune til kommune. Vi er interessert i kvintiler (1/5), nærmere best hvor stor prosentandel av husholdningene i hver kommune som er i nedre og øvre kvartil. For å beregne disse to variablene må vi hente ned opplysninger om desilene 1, 2, 9 og 10. Vi summerer desil 1 og 2 til kvartil 1 og desil 9 og 10 til kvartil 5.

Dataene finnes i tabell 12558. Finn aktuelle parameter vha. SSBs web-side for tabell 12558. Opplysningene kan en finne hvis en velger API-spørring nederst på siden. Se filen `api.txt` (i denne mappen) for hvordan vi kommer frem til parameterverdiene vi skal gi funksjonen `ApiData`.

Her er noen opplysninger om hva du skal ende opp med:

```
names(inc_k_08_17)

## [1] "knr"      "aar"      "inc_k1" "inc_k5"

dim(inc_k_08_17)

## [1] 2140      4
```

- Når du har et resultat som ovenfor. Legg de to nye variablene til `pm2`.
- Rydd opp, dvs. slett objekter du ikke lenger har bruk for (vha. `rm()`).

## Prosent av befolkning med universitets/høgskole utdanning

Dataene finner vi i tabell 09429. Vi er interessert i hvor mange prosent av befolkningen som har hhv. kort og lang universitets/høgskole-utdanning.

- Hent dataene. Legg datene inn i en tibble `uni_p` med variablene `knr`, `kjonn`, `nivaa`, `uni_p` og `aar`.

Du skal ende opp med:

```
names(uni_p)

## [1] "knr"      "kjonn" "nivaa" "uni_p" "aar"

dim(uni_p)

## [1] 12840      5

head(uni_p, n = 5)

## # A tibble: 5 x 5
##   knr   kjonn   nivaa                uni_p aar
##   <chr> <chr>      <chr>              <dbl> <chr>
## 1 0101 Begge kjønn Universitets- og høgskolenivå, kort 17.8 2008
## 2 0101 Begge kjønn Universitets- og høgskolenivå, kort 18.2 2009
## 3 0101 Begge kjønn Universitets- og høgskolenivå, kort 18.6 2010
## 4 0101 Begge kjønn Universitets- og høgskolenivå, kort 19   2011
## 5 0101 Begge kjønn Universitets- og høgskolenivå, kort 19.6 2012
```

- Benytt `fct_recode` til å rekode nivåene for variabelen `nivaa` til `uni_k` of `uni_l`.

- ii. Benytt `fct_recode` til å rekode nivåene for variabelen `kjonn` til `mf`, `f` og `m`.
- iii. Gjør dataene *tidy*.

Du skal ende opp med:

```
head(uni_p, n = 8)
```

```
## # A tibble: 8 x 8
##   knr    aar  uni_k_mf uni_k_m uni_k_f uni_l_mf uni_l_m uni_l_f
##   <chr> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 0101  2008    17.8    15.1    20.4     3.9     5.4     2.4
## 2 0101  2009    18.2    15.4    20.9     3.9     5.4     2.5
## 3 0101  2010    18.6    15.6    21.6     4.1     5.5     2.7
## 4 0101  2011    19      15.8    22.2     4.4     5.8     3
## 5 0101  2012    19.6    16.2    22.9     4.6     5.9     3.3
## 6 0101  2013    19.9    16.4    23.3     4.6     5.8     3.4
## 7 0101  2014    20.6    17      24       4.9     6.1     3.8
## 8 0101  2015    21      17.2    24.8     5.2     6.4     4.1
```

```
dim(uni_p)
```

```
## [1] 2140     8
```

- i. Er `uni_p` i orden kan den nå flettes sammen med `pm2`.
- ii. Slett objekter som ikke lenger trengs.

## Handelsomsetning per innbygger

Dataene finnes i tabellen 04776.

- i. Hent dataene vha. api-en. Legg dem i `trade_08_17`. Kall variabelen for `Trade_pc`.
- ii. Sjekk at dimensjoner etc. stemmer før tabellen flettes sammen med `pm2`.
- iii. Slett objekter som ikke lenger trengs.

## Skriv datasett til csv-filen `pm2.csv`

Har du gjort alt riktig skal du nå ha et datasett `pm2` med følgende karakteristika:

```
dim(pm2)
```

```
## [1] 2140    16
```

```
names(pm2)
```

```
## [1] "knr"      "knavn"    "aar"      "pm2"      "Menn_ya_p"
## [6] "Kvinner_ya_p" "Total_ya_p" "inc_k1"   "inc_k5"   "uni_k_mf"
## [11] "uni_k_m"    "uni_k_f"   "uni_l_mf" "uni_l_m"  "uni_l_f"
## [16] "Trade_pc"
```

```
pm2 %>%
```

```
  select(knr:inc_k5) %>%
```

```
  head(n=8)
```

```
##   knr  knavn  aar  pm2  Menn_ya_p  Kvinner_ya_p  Total_ya_p  inc_k1  inc_k5
## 1 0101 Halden 2008 13427  59.74892    56.79763    58.26214    24.5    13.6
## 2 0101 Halden 2009 13095  59.77860    57.04693    58.40290    24.4    14.1
## 3 0101 Halden 2010 13832  59.64298    57.06300    58.34376    23.9    13.7
## 4 0101 Halden 2011 14915  59.84630    57.22382    58.53183    24.0    14.0
```

```
## 5 0101 Halden 2012 15473 59.45122 57.00467 58.22699 23.9 14.0
## 6 0101 Halden 2013 15461 58.97797 56.73872 57.85475 24.1 13.4
## 7 0101 Halden 2014 17164 58.76014 56.72937 57.74260 23.9 13.5
## 8 0101 Halden 2015 17427 58.71457 56.84787 57.78159 24.0 13.7
```

```
pm2 %>%
  select(uni_k_mf:Trade_pc) %>%
  head(n=8)
```

```
##   uni_k_mf uni_k_m uni_k_f uni_l_mf uni_l_m uni_l_f Trade_pc
## 1    17.8    15.1    20.4     3.9     5.4     2.4    56266
## 2    18.2    15.4    20.9     3.9     5.4     2.5    56366
## 3    18.6    15.6    21.6     4.1     5.5     2.7    57210
## 4    19.0    15.8    22.2     4.4     5.8     3.0    58010
## 5    19.6    16.2    22.9     4.6     5.9     3.3    58787
## 6    19.9    16.4    23.3     4.6     5.8     3.4    59453
## 7    20.6    17.0    24.0     4.9     6.1     3.8    63033
## 8    21.0    17.2    24.8     5.2     6.4     4.1    63747
```

Ser ting bra ut kan pm2 lagres til filen pm2.csv.