Ekstra hent SSB data

```
# 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(2006:2017))
)
```

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. Gangen i det følgende er:

1. Vi henter ut `dataset` elementet fra `pm2_raw`.
2. Vi sjekker i pm2_raw at alle observasjonene er av `Boligtype` "01".
3. Vi dropper variablene `Boligtype` og `ContentsCode`.
4. Vi skifter så navn på variablene Region, Tid og value som angitt ovenfor.

Vi bruker funksjonene `select()`, `mutate()` og `rename()` fra tidyverse som er dekket i kap. 5 i r4ds og i tilsvarende slides fra forelesning.

```
pm2 <- pm2_raw$dataset %>%
  select(-Boligtype, -ContentsCode) %>%
  rename(
    knr = Region,
    aar = Tid,
    pm2 = value
)
```

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() er en klassisk R funksjon
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.

i. Forklar hver komponent i følgende regexp.
ii. Hvorfor har vi «\\»?

```
moenster <- '\\s*\\([-\\d\\s]*\\)\\s*$'
```

iii. Bruk dette mønsteret som et argument i funksjonen `str_replace()` for å endre kommunenavnene i `knavn`. Strenger og regexp er behandlet i kap. 14 r4ds og i slides og videre i slides.

Finn kommunenummer for de kommunene som har `complete.cases()` fra 2006

Finner kommunenummeret for de kommunene som ikke mangler data (dvs. de kommunene hvor `complete.cases()` er TRUE) og legger dem i en tibble kalt knr_c.c. Vær oppmerksom på at `complete.cases()` er en klassisk funksjon som krever at du viser hvor dataene skal inn vha. «.», dvs. bruk `complete.cases(.)`. Vi trenger også øvrige funksjoner beskrevet i kap. 5.

```
# Lager tibble med knr for de kommuner som har
# complete.cases i perioden 2006-2017.
knr_c.c <- pm2 %>%
  mutate(
    c.c = complete.cases(.)
  ) %>%
  group_by(knr) %>%
  summarise(sum_c.c = sum(c.c)) %>%
  filter(
    sum_c.c == max(sum_c.c)
  )
# Lager en vector med knr for complete cases
knr_c.c <- knr_c.c$knr
```

Bruk knr_c.c til å redusere pm2 til de kommunene som har «complete cases».

```
pm2 <- pm2 %>%
  filter(
    knr %in% knr_c.c
  )
```

```
dim(pm2)
```

```
## [1] 2364    4
```

Vi har altså 197 kommuner med registrert pm2 for hvert år i perioden 2006-2017. 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

Se dokumentet api.txt i data mappen for forklaring av spørring mot SSBs database vha. api-en. Ovenfor er et eksempel fra tabellen 06035. Nedenfor følger et eksempel som benytter befolknings-statistikk for å finne

andel av befolkning i yrkesaktiv alder. Dette er litt kronglete så koden for å gjøre det er gjengitt nedenfor. Det følgende vil også fungere som et eksempel på bruk av api-en.

«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. Først henter vi antall i yrkesaktiv alder fra tabell 07459 og plasser i variabelen `pop_08_17_ya_raw`. Velger at vi vil ha dataene fordelt på kjønn. Vi kan regne ut total som Menn + Kvinner. Vi lar 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. Vi henter så grunnlag for å beregne befolkning totalt. Det enklest er å benytte den minst detaljerte inndelingen, 0-17 og 18+. Velger også her fordelt på kjønn. Plasser dataene i `pop_08_17_raw` og gjør denne *tidy* (bruk `pivot_wider`). Legger så tidy varianten i `pop_08_17`.Bruker variabelnavnene knr, kjonn, alder, aar , pop. Beregner så Menn_t = Menn_H17 + Menn_H18, Kvinner_t = Kvinner_H17 + Kvinner_H18, Total_t = Menn_t + Kvinner_t (bruker `mutate` for å definere nye variabler).

```
pop_06_17_ya_raw <- ApiData(
    urlToData = "07459",
    Region = knr,
    Alder = list("agg:TredeltGrupperingB2",
                 c("F20-64")),
    Kjonn = TRUE,
    Tid = as.character(2006:2017)
)
```

```
# Change the cumberome name of first list item
names(pop_06_17_ya_raw)[[1]] <- "desc"
```

```
# Pick from both desc and dataset
pop_06_17_ya <- tibble(
  knr = pop_06_17_ya_raw$dataset$Region,
  kjonn = pop_06_17_ya_raw$desc$kjønn,
  aar = pop_06_17_ya_raw$desc$år,
  ya = pop_06_17_ya_raw$dataset$value
)
```

```
# Pivot wider for kjonn (sex) and calculate total
pop_06_17_ya <- pop_06_17_ya %>%
  pivot_wider(
    id_cols = c(knr, aar),
    names_from = kjonn,
    names_prefix = "ya_",
    values_from = ya
  ) %>%
  mutate(
    ya_Total = ya_Menn + ya_Kvinner
  )
```

Befolkning Menn, Kvinner og Totalt

```
# Population, men, women and total
pop_06_17_raw <- ApiData(
    urlToData = "07459",
    Region = knr,
    Alder = list("agg:TodeltGrupperingB",
                 c("H17", "H18")),
    Kjonn = TRUE,
    Tid = as.character(2006:2017)
)
```

```
# Change the cumberome name of first list item
names(pop_06_17_raw)[[1]] <- "desc"
```

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

```
# Pick from both desc and dataset
pop_06_17 <- tibble(
  knr = pop_06_17_raw$dataset$Region,
  kjonn = pop_06_17_raw$desc$kjønn,
  alder = pop_06_17_raw$dataset$Alder,
  aar = pop_06_17_raw$desc$år,
  pop = pop_06_17_raw$dataset$value
)
```

```r
pop_06_17 <- pop_06_17 %>%
  pivot_wider(
    id_cols = c(knr, aar),
    names_from = c(kjonn, alder),
    values_from = pop
  ) %>%
  mutate(
    Menn_t = Menn_H17 + Menn_H18,
    Kvinner_t = Kvinner_H17 + Kvinner_H18,
    Total_t = Menn_t + Kvinner_t
  ) %>%
  select(knr, aar, Menn_t, Kvinner_t, Total_t)
```

Beregner prosentandel i yrkesaktiv alder for kvinner (`Kvinner_ya_p`), menn (`Menn_ya_p`) og totalt (`Total_ya_p`). Velger så variablene `knr`, `aar`, `Menn_ya_p`, `Kvinner_ya_p`, `Total_ya_p`

Her er starten av `pop_06_17_ya_p` for kontroll.

```
head(pop_06_17_ya_p, n = 5)
```

```
## # A tibble: 5 x 6
##   knr   aar   Menn_ya_p Kvinner_ya_p Total_ya_p Total_t
##   <chr> <chr>     <dbl>        <dbl>      <dbl>   <int>
## 1 0101  2006       59.8         56.8       58.3   27722
## 2 0101  2007       59.7         56.8       58.2   27835
## 3 0101  2008       59.7         56.8       58.3   28092
## 4 0101  2009       59.8         57.0       58.4   28389
## 5 0101  2010       59.6         57.1       58.3   28776
```

Legger ya variablene til pm2 tibble-en

Rydder opp og sletter objekter som ikke lenger trengs vha. `rm()`.

Inntekt for husholdningene

i. Hent data fra tabell **07183: Samlet inntekt for husholdninger. Antall og prosent (K) 2006 - 2019**

ii. La summen av de to laveste kategoriene dvs. inntekter fra 0-249999 være variabelen `inc_l` og inntekter over 750000 variabelen `inc_h`. De to variablene vil da angi andelen av husholdninger i en kommune som har en samlet inntekt under 250000 og andelen somhar en inntekt over 750000.

Generelt har andelen av husholdninger i inc_l sunket pga. inflasjon og velstandsvekst. Tilsvarende har andelen i inc_h økt. Vi ønsker derfor å se hvordan disse størrelsene utvikler seg relativt til gjennomsnittet av de kommunene vi betrakter.

iii. Finn derfor gjennomsnittlig inc_l og inc_h for hvert år. Kall variablene inc_hh_l_m og inc_hh_h_m.

iv. Lag så variablene rel_inc_l som er differansen mellom inc_l og inc_hh_l_m. Lag også tilsvarende variabel rel_inc_h.

v. Sjekk at variablene ser OK. Er det gjort riktig bør starten se slik ut.

```
inc_hh_rel %>%
  head()
```

```
## # A tibble: 6 x 4
##   knr   aar   rel_inc_l rel_inc_h
##   <chr> <chr>     <dbl>     <dbl>
## 1 0101  2006        5.7      -5.3
## 2 0101  2007        6        -6.1
## 3 0101  2008        5.1      -7.2
## 4 0101  2009        5.4      -6.2
## 5 0101  2010        4.2      -6.2
## 6 0101  2011        4.3      -6.7
```

vi. Når OK legg dem så til pm2.

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.

i. Hent dataene. Legg resultatet av spørringen inn i uni_p_raw. Vil være en liste med to komponenter.

Endre navnet på første komponenten til desc.

```
# Change the cumberome name of first list item
names(uni_p_raw)[[1]] <- "desc"
```

1. Legg datene inn i en tibble uni-p med variablene knr, kjonn, nivaa, uni_p og aar. Hent kjønn og nivå fra komponenten desc, de øvrige fra komponenten dataset.

Du skal ende opp med:

```
names(uni_p)
```

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

```
dim(uni_p)
```

```
## [1] 14184     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   2006
## 2 0101  Begge kjønn Universitets- og høgskolenivå, kort  17.4 2007
## 3 0101  Begge kjønn Universitets- og høgskolenivå, kort  17.8 2008
## 4 0101  Begge kjønn Universitets- og høgskolenivå, kort  18.2 2009
## 5 0101  Begge kjønn Universitets- og høgskolenivå, kort  18.6 2010
```

i. 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*.

Funksjonen `fct_recode()` er behandlet i avsnitt 15.5 i r4ds. Tidy datasett er behandlet i kap. 12 i r4ds.

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  2006      17      14.8    19.1      3.6     5.3     2
## 2 0101  2007      17.4    15.1    19.6      3.7     5.3     2.2
## 3 0101  2008      17.8    15.1    20.4      3.9     5.4     2.4
## 4 0101  2009      18.2    15.4    20.9      3.9     5.4     2.5
## 5 0101  2010      18.6    15.6    21.6      4.1     5.5     2.7
## 6 0101  2011      19      15.8    22.2      4.4     5.8     3
## 7 0101  2012      19.6    16.2    22.9      4.6     5.9     3.3
## 8 0101  2013      19.9    16.4    23.3      4.6     5.8     3.4
```

```
dim(uni_p)
```

```
## [1] 2364    8
```

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

Økt tilbud av boliger

Vi beregner ny tilgang av kvadratmeter i enebolig + enebolig med hybel per innbygger. Vi finner først antall kvadratmeter som er bygd et år og dividerer så med antall innbyggere dette året. Antall innbygger har vi alt i variabelen Total_t fra befolkningsstatistikken.

i. Hent dataene fra tabell 05940.

ii. Legg dataene fra komponenten dataset i variabelen nye_m2_raw.

iii. Gjør nødvendige endringer av variabelnavn, pivot_wider s.a. nytt_bareal = b111 + b112 kan beregnes og velg til slutt variablene knr, aar og nytt_bareal.

iv. Sjekk at nye_m2 har riktig dimensjon og ser ok ut. Bruk så left_join for å legge variabelen nytt_bareal til pm2.

v. Regner ut antall nye m2 per person og velger endelig sett variabler.

vi. Lag en ny fylkesnummer-variabel som faktor. Husk at fylkesnummer er de to første sifferne i kommunenummeret. Fuksjonen str_sub() er behandlet i r4ds kap. 14.2.3, mens funksjonen parse_factor() er behandlet i r4ds kap. 15.2.

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] 2358   18
```

```
names(pm2)
```

```
##  [1] "knr"            "knavn"          "aar"            "pm2"
##  [5] "Menn_ya_p"      "Kvinner_ya_p"   "Total_ya_p"     "Total_t"
##  [9] "rel_inc_l"      "rel_inc_h"      "uni_k_mf"       "uni_k_m"
## [13] "uni_k_f"        "uni_l_mf"       "uni_l_m"        "uni_l_f"
## [17] "nytt_bareal_pp" "fnr"
```

```
pm2 %>%
  select(knr:rel_inc_h) %>%
  head(n=8)
```

```
##    knr  knavn  aar   pm2 Menn_ya_p Kvinner_ya_p Total_ya_p Total_t rel_inc_l
## 1 0101 Halden 2006 12052  59.75007     56.79180   58.26059   27722       5.7
## 2 0101 Halden 2007 12363  59.71609     56.80068   58.24681   27835       6.0
## 3 0101 Halden 2008 13427  59.74892     56.79763   58.26214   28092       5.1
## 4 0101 Halden 2009 13095  59.77860     57.04693   58.40290   28389       5.4
## 5 0101 Halden 2010 13832  59.64298     57.06300   58.34376   28776       4.2
## 6 0101 Halden 2011 14915  59.84630     57.22382   58.53183   29220       4.3
## 7 0101 Halden 2012 15473  59.45122     57.00467   58.22699   29543       3.3
## 8 0101 Halden 2013 15461  58.97797     56.73872   57.85475   29880       3.6
##   rel_inc_h
## 1      -5.3
## 2      -6.1
## 3      -7.2
## 4      -6.2
## 5      -6.2
## 6      -6.7
## 7      -6.8
## 8      -6.9
```

```
pm2 %>%
  select(Menn_ya_p:uni_k_f) %>%
  head(n=8)
```

```
##   Menn_ya_p Kvinner_ya_p Total_ya_p Total_t rel_inc_l rel_inc_h uni_k_mf
## 1  59.75007     56.79180   58.26059   27722       5.7      -5.3     17.0
## 2  59.71609     56.80068   58.24681   27835       6.0      -6.1     17.4
## 3  59.74892     56.79763   58.26214   28092       5.1      -7.2     17.8
## 4  59.77860     57.04693   58.40290   28389       5.4      -6.2     18.2
## 5  59.64298     57.06300   58.34376   28776       4.2      -6.2     18.6
## 6  59.84630     57.22382   58.53183   29220       4.3      -6.7     19.0
## 7  59.45122     57.00467   58.22699   29543       3.3      -6.8     19.6
```

```
## 8  58.97797     56.73872   57.85475   29880       3.6      -6.9     19.9
##   uni_k_m uni_k_f
## 1    14.8    19.1
## 2    15.1    19.6
## 3    15.1    20.4
## 4    15.4    20.9
## 5    15.6    21.6
## 6    15.8    22.2
## 7    16.2    22.9
## 8    16.4    23.3
```

```
pm2 %>%
  select(uni_l_mf:fnr) %>%
  head(n=8)
```

```
##   uni_l_mf uni_l_m uni_l_f nytt_bareal_pp    fnr
## 1      3.6     5.3     2.0      0.3395498 fnr_01
## 2      3.7     5.3     2.2      0.5134543 fnr_01
## 3      3.9     5.4     2.4      0.5771750 fnr_01
## 4      3.9     5.4     2.5      0.2109268 fnr_01
## 5      4.1     5.5     2.7      0.2006881 fnr_01
## 6      4.4     5.8     3.0      0.3308008 fnr_01
## 7      4.6     5.9     3.3      0.3694953 fnr_01
## 8      4.6     5.8     3.4      0.5078313 fnr_01
```

i. Ser ting bra ut kan pm2 lagres til filen `pm2.csv` (bruk tidyverse funksjonen write_csv()).