hent-ssb-data.Rmd

Jonathan

12/30/2021

```
library(PxWebApiData)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
load("knr.Rdata")
```

Hente data om gjennomsnittlig kvadratmeterpris for eneboliger

```
pm2_raw <- ApiData(
  urlToData = "06035",
  Region = knr,
  ContentsCode = "KvPris",
  Boligtype = "01",
  Tid = c(as.character(2006:2017)),
  # Neste linje gjør at din nyere versjon av ApiData() oppfører
  # likt som den tidligere versjonen som jeg har brukt
  makeNAstatus = FALSE
)
```

Droppe variabler og endre navn

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

Endre navn på elementet

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

Legge til kommunenavnene til pm2 som variablen *knavn*

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

Fjerne paranteser i knavn

```
# se help str_replace.str_replace_all(string, pattern, replacement). Vi skal altså
# sende inn en tekststreng, vår eksisterende knavn variabel, bruke en regex for å matche de
# parentesene vi vil bli kvitt. Det er denne rgex-en vi har i moenster. Til slutt må vi angi
# det vi vil erstatte teksen med. I vårt tilfelle vil vi bare slette så replacement er ""
# som angir ingenting. Legg ut i et objekt tmp først for å teste at riktig, så endre til
# pm2 <- pm2 %>% .... For så å oppdatere pm2 med nye navn
# Fyll inn argumentene til str_replace under. Sjekk at tmp ser ut slik du ønsker. Bytt så ut
# tmp <- pm2 %>% ...  med pm2 <- pm2 %>% ...
pm2 <- pm2 %>%
  # oppdaterer knavn så bruker mutate
  mutate(
  knavn = str_replace(
    string = knavn,
    pattern = '\\s*\\([-\\d\\s]*\\)\\s*$' ,
    replacement = ""
  )
  )
```

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

```
# 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
```

Bruke knr_cc til å redusere pm2 til kommunene som har "complete cases".

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

```
dim(pm2)
```

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

Andel yrkesaktive

```
pop_06_17_ya_raw <- ApiData(
    urlToData = "07459",
    #oppdatert fra knr til knr_c.c
    Region = knr_c.c,
    Alder = list("agg:TredeltGrupperingB2",
                 c("F20-64")),
    Kjonn = TRUE,
    Tid = as.character(2006:2017),
    # Neste linje gjør at din nyere versjon av ApiData() oppfører
    # likt som den tidligere versjonen som jeg har brukt
    makeNAstatus = FALSE
)
```

```
# 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_c.c,
    Alder = list("agg:TodeltGrupperingB",
                 c("H17", "H18")),
    Kjonn = TRUE,
    Tid = as.character(2006:2017),
    # Neste linje gjør at din nyere versjon av ApiData() oppfører
    # likt som den tidligere versjonen som jeg har brukt
    makeNAstatus = FALSE
)
```

```
# 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
)
```

```
    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`

```
# join ya og total pop (pop_08_17)
pop_06_17_ya_p <- pop_06_17_ya %>%
  left_join(pop_06_17, by = c("knr", "aar"))
```

```
# calculate ya percentage and select relevant variables
pop_06_17_ya_p <- pop_06_17_ya_p %>%
  mutate(
    Menn_ya_p = (ya_Menn/Menn_t)*100,
    Kvinner_ya_p = (ya_Kvinner/Kvinner_t)*100,
    Total_ya_p = (ya_Total/Total_t)*100
  ) %>%
  # Tar med Total_t, for å kunne regne ut nye m2 per person
  select(knr, aar, Menn_ya_p, Kvinner_ya_p, Total_ya_p, Total_t)
```

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

```
# update pm2
pm2 <- pm2 %>%
  left_join(pop_06_17_ya_p, by = c("knr", "aar")) %>%
  select(knr, knavn, aar, everything())
```

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

```
# clean up
rm(pop_06_17, pop_06_17_raw, pop_06_17_ya, pop_06_17_ya_raw, pop_06_17_ya_p)
```

Inntekt

```
# hh forkortelse for husholdninger
inc_hh_raw <- ApiData(
  urlToData = "07183",
  # Min feil. Hadde brukt knr istedenfor knr_c.c
  Region = knr_c.c,
  ContentsCode = c("Inntekt1" , "Inntekt2","Inntekt7"),
  Tid = c(as.character(2006:2017)),
  makeNAstatus = FALSE
)
```

Send så inc_hh_raw$dataset inn i en pipe. Bruk rename til å endre variabelnavn til de vi bruker dvs. knr for Region, aar for tid og inc for value. Så må du pivot_wider, dvs få en kolonne for variabelen Inntekt1, en for Inntekt2 og en for Inntekt7. For pivot_wider trenger du tre argumenter første er id_cols som skal være det som identifiserer er observasjon (kommune et gitt år), names_from som er kolonnen hvor de nye variabelnavnene (Inntekt1, etc) skal hentes fra og til slutt values_from som er navnet på kolonnen som har selve verdiene. Kan være greit å avslutte pipen med en select() der du velger de variablene som du trenger å ha med videre.

Når det er gjort kan du lage en ny variabel inc_hh_l som er summen av Inntekt1 og Inntekt2 og inc_hh_h som er Inntekt7.

Så må du beregne gjennomsnittet av inc_hh_l og inc_hh_h for hvert år. Kall dem for inc_hh_l_m og inc_hh_h_m. Gjøres vha. group_by() og summarise(). Inntekt relativt til gjennomsnittet blir da inc_hh_l/inc_hh_l_m og tilsvarende for inc_hh_h

```
inc_hh <- inc_hh_raw$dataset %>%
  rename(
    knr = Region,
    aar = Tid,
    inc = value
  ) %>%
  pivot_wider(
    id_cols = c(knr, aar),
    names_from = ContentsCode,
    values_from = inc
  ) %>%
  mutate(
    # Inntekt1 og Inntekt2 har nå blitt variabler
    inc_l = Inntekt1 + Inntekt2,
    inc_h = Inntekt7
  ) %>%
```

```
  select(knr, aar, inc_l, inc_h)
```

Da har vi husholdningenes inntekt i inc_hh. Må så beregne gjennomsnittlig inc_hh_l og inc_hh_h for hvert år. Kaller disse for inc_hh_l_m og inc_hh_h_m og legger dem i inc_hh_y_mean

```
inc_hh_y_mean <- inc_hh %>%
  # gjennomsnitt per år så grupperer mht. aar
  group_by(aar) %>%
  summarise(
    # Regner gjennomsnitt av dem vi har data for. Avrunder til 1 desimal
    inc_hh_l_m = round(mean(inc_l, na.rm = TRUE), 1),
    inc_hh_h_m = round(mean(inc_h, na.rm = TRUE), 1)
  )
```

Da må vi slå sammen inc_hh med inc_hh_y_mean vha. left_join(). Sett argumentet by til aar. Da vil du få f.eks 24.4 for inc_hh_l_m for alle kommuner i år 2006, 21.9 for inc_hh_l_m for alle kommuner i år 2007 osv. Da er det enkelt å bruke mutate() for å regne ut rel_inc_l og rel_inc_h

```
inc_hh <- left_join(
  inc_hh,
  inc_hh_y_mean,
  by = "aar",
)
```

Mutering av variabler:

```
inc_hh_rel <- inc_hh %>%
  mutate(
    rel_inc_l = inc_l - inc_hh_l_m,
    rel_inc_h = inc_h - inc_hh_h_m
  )
```

Fjerne unødvendige variabler:

```
inc_hh_rel <- inc_hh_rel %>%
  select(-inc_l, -inc_h, -inc_hh_l_m, -inc_hh_h_m)
```

```
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
```

```
# La til by så slipper vi melding i Console
pm2 <- left_join(pm2, inc_hh_rel, by = c("knr", "aar"))
```

Utdanning

Henter datasett

```
uni_p_raw <- ApiData(
  urlToData = "09429",
  # oppdatert til knr_c.c eller får vi 423 kommuner. 197 er nok å holde styr på
  Region = knr_c.c,
  Nivaa = c("03a", "04a"),
  Kjonn = TRUE,
  ContentsCode = "PersonerProsent",
  Tid = c(as.character(2006:2017)),
  makeNAstatus = FALSE
)
```

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

```
# Plukker variabler fra uni_p_raw for å lage uni_p
uni_p <- tibble(
  knr = uni_p_raw$dataset$Region,
  aar = uni_p_raw$dataset$Tid,
  Kjonn = uni_p_raw$desc$kjønn,
  nivaa = uni_p_raw$desc$nivå,
  uni_p = uni_p_raw$dataset$value
)
```

```
dim(uni_p)
```

```
## [1] 14184     5
```

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

```
## # A tibble: 5 x 5
##   knr   aar   Kjonn       nivaa                                uni_p
##   <chr> <chr> <chr>       <chr>                                <dbl>
## 1 0101  2006  Begge kjønn Universitets- og høgskolenivå, kort  17
## 2 0101  2007  Begge kjønn Universitets- og høgskolenivå, kort  17.4
## 3 0101  2008  Begge kjønn Universitets- og høgskolenivå, kort  17.8
## 4 0101  2009  Begge kjønn Universitets- og høgskolenivå, kort  18.2
## 5 0101  2010  Begge kjønn Universitets- og høgskolenivå, kort  18.6
```

i) Benytt fct_recorde til å rekode nivåene for variablen nivaa til uni_k of uni_l.

```
uni_p <- uni_p %>%
  mutate(nivaa = fct_recode(nivaa,
                            "uni_k" = "Universitets- og høgskolenivå, kort",
                            "uni_l" = "Universitets- og høgskolenivå, lang"))
```

ii) Benytt fct_recode til å rekode nivåene for variablen kjonn til mf, f og m.

```
uni_p <- uni_p %>%
  mutate(
    Kjonn = fct_recode(Kjonn,
                       "mf" = "Begge kjønn",
                       "f" = "Kvinner",
                       "m" = "Menn")
```

```
  )
```

iii) Gjør dataene tidy

```
uni_p <- uni_p %>%
  pivot_wider(
    id_cols = c(knr, aar),
    names_from = c(nivaa, Kjonn),
    values_from = uni_p
  )
```

```
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
```

```
pm2 <- left_join(
  pm2,
  uni_p,
  by = c("knr", "aar")
)
```

Økt tilbud av boliger

Henter datasett

```
nye_m2_raw <- ApiData(
  urlToData = "05940",
  Region = knr_c.c,
  # Tar bare med eneboliger og eneboliger med hybel
  Byggeareal = c("111", "112"),
  # Tar bare med de som er fullført i det relevante året
  ContentsCode = "BruksarealFullfort",
  Tid = c(as.character(2006:2017))
)
```

```
# Her kan vi ta dataene rett fra dataset
nye_m2_raw <- nye_m2_raw$dataset
```

```
# Må pivot_wider s.a. vi får en variabel for 111 og en for 112
# Vi kan da lage en ny som er lik b111 + b112
nye_m2 <- nye_m2_raw %>%
  rename(
```

```
knr = Region,
aar = Tid ) %>%
  pivot_wider(
    id_cols = c(knr, aar),
    names_from = Byggeareal,
    names_prefix = "b",
    values_from = value
) %>% mutate(
    nytt_bareal = b111 + b112
  ) %>%
  select(knr, aar, nytt_bareal)
```

```
 pm2 <- pm2 %>%
  left_join(nye_m2, by  = c("knr", "aar"))
```

```
# Nytt boligareal per person i kommunen
pm2 <- pm2 %>%
  mutate(
    nytt_bareal_pp = nytt_bareal/Total_t
  )
```

Skriv datasett til csv-filen pm2.csv

```
# Bruk write_csv() (fra tidyverse) når du skriver til fil
# bedre enn write.csv() som er klassik varianten
```

```
write_csv(pm2, "pm2.csv")
```