

Oppgaver 4 - løsning

October 6, 2025

```
[ ]: renv::autoload()
```

1 Oppgaver 4

```
[ ]: library(tidyverse)
```

1.1 Uføretrygdede

Du skal finne andelen personer mellom 18 og 67 år som mottar uføretrygd i hvert fylke og sjekke at denne andelen stemmer med den som rapporteres i tabellen for uføretrygdede.

Last inn datasettene `uføretrygdede` og `befolkning` i cellen nedenfor. + `uføretrygdede` - tabell 11695: Uføretrygdede, etter kjønn + `befolkning` - tabell 07459: Alders- og kjønnsfordeling i kommuner, fylker og hele landets befolkning

Lag et nytt objekt `befolkning_18_67_per_fylke` ut fra `befolkning` der du har gjort følgende: + Opprett en ny variabel som heter `Alder_num` som inneholder verdiene fdra `Alder` omgjort til numerisk + Opprett en ny variabel som heter `Region_ny` der følgende verdier fra `Region` har blitt omkodet (øvrige verdier skal være uendret): + + Viken: 31, 32, 33 -> 30 + + Vestfold og Telemark: 39, 40 -> 38 + + Troms og Finnmark: 55, 56 -> 54 + Filtrer rader der antall tegn i `Region_ny` er lik 2, verdiene i `Alder_num` er i intervallet 18-67 og verdien i `value` er høyere enn 0 + Grupper datasettet etter variabelen `Region_ny` og beregn sum etter kolonnen `value`. Kall den nye variabelen med sum for `personer`

Lag et nytt objekt `uføretrygdede_pros` ut fra `uføretrygdede` der du har gjort følgende: + Filtrer rader der antall rader i `Region` er lik 2, `ContentsCode` er lik "UforetygdPers", `Kjonn` er lik 0 + Kobler på `befolkning_18_67_per_fylke` etter køblingsnøkkelen `c(Region = Region_ny)` + Oppretter en ny variabel `andel` som inneholder uføretrygdede i prosent av befolkningen per fylke. Rund av til én desimal + Sjekk at resultatene blir det samme som i objektet `uføretrygdede` (`ContentsCode = "UforetrygdPros"`)

```
[ ]: uforetrygdede <- PxWebApiData::ApiData(11695,
                                           ContentsCode = T,
                                           Region = T,
                                           Kjonn = T,
                                           Tid = "2023")[[2]]

befolkning <- PxWebApiData::ApiData(07459,
```

```

ContentsCode = T,
Region = T,
Kjonn = T,
Alder = T,
Tid = "2024")[[2]]

```

```

[ ]: befolkning_18_67_per_fylke <- befolkning %>%
mutate(Alder_num = as.numeric(gsub("\\+", "", Alder)),
       Region_ny = case_when(Region %in% c("31", "32", "33") ~ "30",
                             Region %in% c("39", "40") ~ "38",
                             Region %in% c("55", "56") ~ "54",
                             TRUE ~ Region)) %>%
filter(nchar(Region_ny)==2,
       Alder_num %in% 18:67,
       value > 0) %>%
group_by(Region_ny) %>%
summarise(personer = sum(value))

uforetrygdede_pros <- uforetrygdede %>%
filter(nchar(Region) == 2,
       ContentsCode == "UforetrygdPers",
       Kjonn == 0,
       value > 0) %>%
left_join(befolkning_18_67_per_fylke, by = c("Region" = "Region_ny")) %>%
mutate(andel = round(value/personer*100, digits = 1))

trygd_pros_statbank <- uforetrygdede %>%
filter(nchar(Region) == 2,
       ContentsCode == "UforetrygdPros",
       Kjonn == 0,
       value != 0) %>%
select(Region, value) %>%
rename(UforetrygdPros = value)

uforetrygdede_pros %>%
left_join(trygd_pros_statbank, by = "Region") %>%
mutate(diff = andel-UforetrygdPros)

```

1.2 Yrkesfordelt månedslønn

Du skal lage et datasett som viser gjennomsnittlig månedslønn på ulike detaljersnivåer i yrkeskoder, slik at du kan sammenligne lønnsnivået for spesifikke yrker med de mer overordnede yrkesgruppene de tilhører.

Last inn objektene `yrkesfordelt_manedslonn` og `yrkesklassifisering_klass` i cellen nedenfor. Variabelen `Yrke` inneholder ulike yrkeskoder med 1-4 siffer. + `yrkesfordelt_manedslonn`: - tabell 11418: Yrkesfordelt månedslønn, etter sektor, kjønn og arbeidstid + `yrkesklassifisering_klass` - klassifikasjon 7: Standard for yrkesklassifisering

Lag et nytt objekt som heter `yrkesfordelt_manedslonn_4` som kun inneholder rader med yrkeskoder som består av fire siffer. + Endre navn på variablene `Yrke` og `value` til henholdsvis `Yrke_4` og `Manedslonn_4`. Gjør deretter det samme med yrker med 3, 2 og 1 siffer slik at du endre opp med totalt fire datasett med forskjellige nivåer av yrkeskoder (`yrkesfordelt_manedslonn_4`, `yrkesfordelt_manedslonn_3`, `yrkesfordelt_manedslonn_2` og `yrkesfordelt_manedslonn_1`). + Opprett tre nye variabler (`Yrke_3`, `Yrke_2` og `Yrke_1`) i datasettet `yrkesfordelt_manedslonn_4` som inneholder de 3 første, 2 første og det første sifferet i variabelen `Yrke_4`. + Koble deretter sammen alle objektene slik at datasettet inneholder alle disse kolonnene: `Yrke_4`, `Manedslonn_4`, `Yrke_3`, `Yrke_2`, `Yrke_1`, `Manedslonn_3`, `Manedslonn_2`, `Manedslonn_1` + Koble på slutt navn på yrkesvariablene fra kodelisten `yrkesklassifisering_klass` + Sjekk månedslønnen til utvalgte yrker på 4-sifternivå (f.eks. Legespesialister: 2212 og Sykepleiere: 2223) og se hvordan lønnen skiller seg fra yrkesgruppene disse tilhører.

```
[ ]: yrkesfordelt_manedslonn <- PxWebApiData::ApiData(11418,
                                                    ContentsCode = "Manedslonn",
                                                    MaaleMetode = "02", # 02: gjennomsnitt, 01:
                                                    ↪ median

                                                    Yrke = T,
                                                    Sektor = "ALLE",
                                                    Kjonn = "0",
                                                    AvtaltVanlig = "0",
                                                    Tid = "2023")[[2]] %>%

filter(Yrke != "0-9")

yrkesklassifisering_klass <- klassR::GetKlass(7)
```

```
[ ]: yrkesfordelt_manedslonn_4 <- yrkesfordelt_manedslonn %>%
filter(nchar(Yrke) == 4) %>%
select(Yrke, value) %>%
rename(Yrke_4 = Yrke,
       Manedslonn_4 = value) %>%
mutate(Yrke_3 = substr(Yrke_4, 1, 3),
       Yrke_2 = substr(Yrke_4, 1, 2),
       Yrke_1 = substr(Yrke_4, 1, 1))

yrkesfordelt_manedslonn_3 <- yrkesfordelt_manedslonn %>%
filter(nchar(Yrke) == 3) %>%
select(Yrke, value) %>%
rename(Yrke_3 = Yrke,
       Manedslonn_3 = value)

yrkesfordelt_manedslonn_2 <- yrkesfordelt_manedslonn %>%
filter(nchar(Yrke) == 2) %>%
select(Yrke, value) %>%
rename(Yrke_2 = Yrke,
       Manedslonn_2 = value)
```

```
yrkesfordelt_manedslonn_1 <- yrkesfordelt_manedslonn %>%
  filter(nchar(Yrke) == 1) %>%
  select(Yrke, value) %>%
  rename(Yrke_1 = Yrke,
         Manedslonn_1 = value)

yrkesfordelt_manedslonn_alle <- yrkesfordelt_manedslonn_4 %>%
  left_join(yrkesfordelt_manedslonn_3, by = "Yrke_3") %>%
  left_join(yrkesfordelt_manedslonn_2, by = "Yrke_2") %>%
  left_join(yrkesfordelt_manedslonn_1, by = "Yrke_1") %>%
  left_join(yrkesklassifisering_klass, by = c("Yrke_4" = "code"))

yrke <- c("2223",
         "2212")

yrkesfordelt_manedslonn_alle %>%
  filter(Yrke_4 %in% yrke)
```

1.3 Barnevern

Du skal beregne hvor mange barnevernsmeldinger det er per 1 000 innbyggere i alderen 0–24 år i hvert fylke, og kontrollere at resultatet stemmer med den ferdigberegnete indikatoren i barnevernstabellen.

Lag et nytt objekt `befolkning_0_24_per_fylke` ut fra `befolkning` der du har gjort følgende: + Opprett en ny variabel som heter `Alder_num` som inneholder verdiene fra `Alder` omgjort til numerisk + Opprett en ny variabel som heter `Region_ny` der følgende verdier fra `Region` har blitt omkodet (øvrige verdier skal være uendret): + Viken: 31, 32, 33 -> 30 + Vestfold og Telemark: 39, 40 -> 38 + Troms og Finnmark: 55, 56 -> 54 + Filtrer rader der antall tegn i `Region_ny` er lik 2, verdiene i `Alder_num` er i intervallet 0-24 og verdien i `value` ikke er missing + Grupper datasettet etter variabelen `Region_ny` og beregn sum etter kolonnen `value`. Kall den nye variabelen med sum for `personer`

Lag et nytt objekt `barnevern_per_1000` ut fra `barnevern` der du har gjort følgende: + Filtrer rader der `ContentsCode` er lik "Melding", `Region` inneholder mønsteret `^[0-9]{2}$` og `value` ikke inneholder missing-verdier. + Kobler på `befolkning_0_24_per_fylke` etter køblingsnøkkelen `c(Region = Region_ny)` + Oppretter en ny variabel `melding_per_1000` som inneholder meldinger per 1000 barn per fylke. Rund av til én desimal + Sjekk at resultatene blir det samme som i objektet `barnevern` (`ContentsCode = "MeldingPer1000"`)

```
[ ]: barnevern <- PxWebApiData::ApiData(10673,
                                       ContentsCode = T,
                                       Region = T,
                                       Kjonn = "Total",
                                       Alder = "999",
                                       Konklusjon = "Om",
                                       Tid = "2023")[[2]]
```

```
befolkning <- PxWebApiData::ApiData(07459,
                                     ContentsCode = T,
                                     Region = T,
                                     Kjonn = T,
                                     Alder = T,
                                     Tid = "2024")[[2]]
```

```
[ ]: befolkning_0_24_per_fylke <- befolkning %>%
mutate(Alder_num = as.numeric(gsub("\\+", "", Alder)),
       Region_ny = case_when(Region %in% c("31", "32", "33") ~ "30",
                             Region %in% c("39", "40") ~ "38",
                             Region %in% c("55", "56") ~ "54",
                             TRUE ~ Region)) %>%
filter(nchar(Region_ny)==2,
       Alder_num %in% 0:24,
       value > 0) %>%
group_by(Region_ny) %>%
summarise(personer = sum(value))

barnevern_per_1000 <- barnevern %>%
filter(ContentsCode == "Melding",
       stringr::str_detect(Region, "[0-9]{2}$"),
       !is.na(value)) %>%
left_join(befolkning_0_24_per_fylke, by = c("Region" = "Region_ny")) %>%
mutate(melding_per_1000 = round(value/(personer/1000), digits = 1))
```