

Oppgaver 3

September 2, 2024

1 Oppgaver 3

1.1 Hent inn pakken `tidyverse` som skal brukes i oppgavene med funksjonen `library()`

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

1.2 Last inn filen “`befolkning_per_kommune.parquet`” som ligger i mappen `data` kursmateriellet. Kall objektet for `befolkning_per_kommune`.

```
[ ]: befolkning_per_kommune <- arrow::read_parquet("../data/befolkning_per_kommune.  
      ↳parquet")
```

1.3 Print de 6 første radene i `befolkning_per_kommune`

```
[ ]: head(befolkning_per_kommune)
```

1.4 Hvilke variabeltyper har kolonnene i filen?

```
[ ]: summary(befolkning_per_kommune)  
     pillar::glimpse(befolkning_per_kommune)
```

1.5 Hva er gjennomsnittet og medianen til kolonnen `value`?

```
[ ]: # glimpse(befolkning_per_kommune)  
     summary(befolkning_per_kommune)
```

1.6 Bruk funksjonen `rename()` til å endre navnet på kolonnen `Region` til `kommunennummer` og `value` til `personer`.

```
[ ]: befolkning_per_kommune <- befolkning_per_kommune %>%  
     rename(kommunennummer = Region,  
            personer = value)
```

1.7 Bruk funksjonen `select()` til å fjerne variabelen `Tid` og endre rekkefølgen på kolonnene slik: `kommunenummer`, `kommunenavn`, `personer`.

```
[ ]: befolkning_per_kommune <- befolkning_per_kommune %>%
      select(kommunenummer, kommunenavn, personer)

befolkning_per_kommune %>%
  head()
```

1.8 Hvor mange personer bor det i Bergen kommune?

```
[ ]: befolkning_per_kommune %>%
      filter(kommunenavn == "Bergen")
```

1.9 Bruk funksjonen `filter()` for å beholde kommuner med befolkning på lavere enn eller lik 1000. Hvor mange kommuner blir det?

```
[ ]: befolkning_per_kommune %>%
      filter(personer <= 1000)
```

1.10 Hvilke to kommuner har flest og færrest antall innbyggere? Hvor mange innbyggere har disse kommunene?

```
[ ]: befolkning_per_kommune %>%
      filter(personer == min(personer) |
             personer == max(personer))
```

1.11 Bruk funksjonen `mutate()` til å opprette en ny variabel som heter `fylkesnummer` som inneholder fylkesnummeret til hver kommune. Hint: fylkesnummeret er de to første sifrene i kommunenummeret.

```
[ ]: befolkning_per_kommune <- befolkning_per_kommune %>%
      mutate(fylkesnummer = substr(kommunenummer, 1, 2))

befolkning_per_kommune %>%
  head()
```

1.12 Bruk funksjonene `group_by()` og `summarise()` for å beregne summen og gjennomsnittet av befolkning per fylke. Lagre resultatene i et nye objekter som heter `befolkning_per_fylke_sum` og `befolkning_per_fylke_gjennomsnitt` og kall de nye variablene for `sum` og `gjennomsnitt`

```
[ ]: befolkning_per_fylke_sum <- befolkning_per_kommune %>%
  group_by(fylkesnummer) %>%
  summarise(sum = sum(personer))

befolkning_per_fylke_gjennomsnitt <- befolkning_per_kommune %>%
  group_by(fylkesnummer) %>%
  summarise(gjennomsnitt = mean(personer))
```

1.13 Koble sammen datasettene `befolkning_per_fylke_sum` og `befolkning_per_fylke_gjennomsnitt` og kall det nye objektet for `befolkning_per_fylke_2`

```
[ ]: befolkning_per_fylke_2 <- befolkning_per_fylke_sum %>%
  left_join(befolkning_per_fylke_gjennomsnitt, by = "fylkesnummer")

head(befolkning_per_fylke_2)
```

1.14 Last inn filen `"../data/fylkesinndeling.csv"` med funksjonen `read.csv2()` og print ut den første raden og kolonnenavnene i filen (filen er lagret med encoding `"latin1"`). Gjør deretter det samme, men legg til argumentet `header = FALSE` i funksjonen `read.csv2()`. Hva blir forskjellen?

```
[ ]: fylkesinndeling <- read.csv2("../data/fylkesinndeling.csv", sep = ";", header =
  ↪FALSE, encoding = "latin1")
fylkesinndeling[1,]
colnames(fylkesinndeling)

fylkesinndeling <- read.csv2("../data/fylkesinndeling.csv", sep = ";", header =
  ↪TRUE, encoding = "latin1")
fylkesinndeling[1,]
colnames(fylkesinndeling)
```

1.15 Last inn filen `"../data/fylkesinndeling.csv"` og kall objektet `fylkesinndeling`. Endre navn på kolonnen `V1` til `fylkesnummer` og legg til ledende null med funksjonen `str_pad()`

```
[ ]: fylkesinndeling <- read.csv("../data/fylkesinndeling.csv", sep = ";", header =
  ↪FALSE) %>%
  rename(fylkesnummer = V1,
         fylkesnavn = V2) %>%
```

```
mutate(fylkesnummer = stringr::str_pad(fylkesnummer, width = 2, "left", pad = "0"))
```

```
fylkesinndeling
```

1.16 Legg til navn på fylke ved å koble sammen `befolkning_per_fylke_2` og `fylkesinndeling` med funksjonen `full_join()`. Kall det nye objektet `befolkning_per_fylke_3`

```
[ ]: befolkning_per_fylke_3 <- befolkning_per_fylke_2 %>%
      full_join(fylkesinndeling, by = "fylkesnummer")

befolkning_per_fylke_3
```

1.17 Fjern rader med missing (NA) på variabelen `sum`.

```
[ ]: befolkning_per_fylke_3 <- befolkning_per_fylke_3 %>%
      filter(!is.na(sum))
```

1.18 Restrukturer datasettet `befolkning_per_fylke_3` fra “bredt” til “langt” format. Hint: kolonnene `sum` og `gjennomsnitt` skal slås sammen til én kolonne.

```
[ ]: befolkning_per_fylke_3_long <- befolkning_per_fylke_3 %>%
      pivot_longer(cols = c("sum", "gjennomsnitt"),
                   names_to = "variabel",
                   values_to = "personer")

nrow(befolkning_per_fylke_3)
nrow(befolkning_per_fylke_3_long)
```

1.19 Last inn filen `befolkning` fra statistikkbanken ved å kjøre kodesnutten nedenfor og beregn befolkning (totalt) for hele landet, per kommune og per fylke i tre separate objekter. Behold kun aktive fylker og kommuner (dvs. som har innbyggere i 2024)

Hint + Region: 0 = hele landet, to siffer = fylke, fire siffer = kommune + Kjønn: 1 = menn, 2 = kvinner

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

```
[ ]: befolkning_hele_landet <- befolkning %>%
  filter(Region == 0) %>%
  group_by(Region) %>%
  summarise(value = sum(value))

befolkning_fylke <- befolkning %>%
  filter(nchar(Region) == 2) %>%
  group_by(Region) %>%
  summarise(value = sum(value)) %>%
  filter(value != 0)

befolkning_kommune <- befolkning %>%
  filter(nchar(Region) == 4) %>%
  group_by(Region) %>%
  summarise(value = sum(value)) %>%
  filter(value != 0)
```

1.20 Bruk befolkning fra forrige oppgave og opprett en ny variabel som heter aldersgruppe der alder er gruppert etter følgende inndeling: “0-15”, “16-24”, “25-34”, “35-44”, “45-54”, “55-64”, “65-74”, “75+”

Beregn deretter befolkning per aldersgruppe for hele landet, per kommune og per fylke i tre separate objekter.

```
[ ]: befolkning_2 <- befolkning %>%
  mutate(Alder = gsub("\\+", "", Alder),
         Alder_num = as.numeric(Alder),
         aldersgruppe = case_when(Alder_num %in% 0:15 ~ "0-15",
                                   Alder_num %in% 16:24 ~ "16-24",
                                   Alder_num %in% 25:34 ~ "25-34",
                                   Alder_num %in% 35:44 ~ "35-44",
                                   Alder_num %in% 45:54 ~ "45-54",
                                   Alder_num %in% 55:64 ~ "55-64",
                                   Alder_num %in% 65:74 ~ "65-74",
                                   Alder_num >= 75 ~ "75 og eldre",
                                   TRUE ~ ""))

befolkning_2_hele_landet <- befolkning_2 %>%
  filter(Region == 0) %>%
  group_by(Region, aldersgruppe) %>%
  summarise(value = sum(value))

befolkning_2_fylke <- befolkning_2 %>%
  filter(nchar(Region) == 2) %>%
  group_by(Region, aldersgruppe) %>%
  summarise(value = sum(value)) %>%
  filter(value != 0)
```

```
befolkning_2_kommune <- befolkning_2 %>%
  filter(nchar(Region) == 4) %>%
  group_by(Region, aldersgruppe) %>%
  summarise(value = sum(value)) %>%
  filter(value != 0)
```

1.21 Bruk funksjonen `bind_rows()` for å legge sammen de tre objektene du opprettet i forrige oppgave til ett nytt objekt som heter `befolkning_per_aldersgrupper`.

```
[ ]: befolkning_per_aldersgrupper <- bind_rows(befolkning_2_hele_landet,
  ↪ befolkning_2_fylke, befolkning_2_kommune)
```

1.22 Lagre objektet `befolkning_per_aldersgrupper` i mappen `../data/` i `.csv`-format og les deretter inn filen for å kontrollere at filen ble lagret riktig. Lagre også filen i `.xlsx` og `.parquet`-format (og les inn filene etterpå)

```
[ ]: write.csv2(befolkning_per_aldersgrupper, "../data/befolkning_per_aldersgrupper.
  ↪ csv")

openxlsx::write.xlsx(befolkning_per_aldersgrupper, file = "../data/
  ↪ befolkning_per_aldersgrupper.xlsx",
  rowNames = FALSE,
  showNA = FALSE,
  overwrite=T)

arrow::write_parquet(befolkning_per_aldersgrupper, "../data/
  ↪ befolkning_per_aldersgrupper.parquet")
```

1.23 Åpen oppgave: Les inn et datasett du jobber med og bruk funksjonene `rename()`, `select()`, `filter()`, `group_by()`, `summarise()` osv. til å gjøre gjørene oppgaver du vanligvis må løse.

```
[ ]:
```

```
[ ]:
```