Assignment 4

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Assignment 4;del 2; Tidy data

I denne filen skal vi lese inn rådataene vi hentet fra SSB i `get-ssb-data.Rmd`. Vi skal gjøre dataene «tidy» og kombinere arbeidsløshetsdataene som er månedlige til en stor «tidy» tibble. Videre skal vi kombinere befolkningsdataene som er årlige til en stor «tidy» tibble. Disse to tibble-ene skal så lagres som `.csv` filer i mappen data. Disse to vil så senere bli lest inn i filen `model.Rmd` og være utgangspunktet for vårt modelleringsarbeide.

Leser inn filene

```
df_arbl_p_alder <- read_csv(file = "ssb_df_arbl_p_alder.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   alder = col_character(),
##   tid = col_character(),
##   alp = col_double()
## )
```

```
df_arbl_p_kjonn <- read_csv(file = "ssb_df_arbl_p_kjonn.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   kjonn = col_character(),
##   tid = col_character(),
##   alp_k = col_double()
## )
```

```
df_arbl <- read_csv(file = "ssb_df_arbl.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   tid = col_character(),
##   kjonn = col_character(),
##   al = col_double()
## )
```

```
df_bef_K <- read_csv(file = "ssb_df_bef_K.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   tid = col_double(),
##   kjonn = col_character(),
##   alder = col_character(),
##   bef = col_double()
## )
```

```
df_bef_M <- read_csv(file = "ssb_df_bef_M.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   tid = col_double(),
##   kjonn = col_character(),
##   alder = col_character(),
##   bef = col_double()
## )
```

Problem med NA

Vi har et problem med NA for små kommuner i `df_arbl` . Årsaken er såkalt «prikking» utfra personvernhensyn. Er det færre enn 3 (dvs. 0, 1, 2) personer i en kategori så mener SSB at det blir mulig å identifisere enkeltpersoner. Derfor «prikker» de, dvs. lager NA, verdier for observasjonene. Vi lar dette ligge nå, men et problem som det er greit å være oppmerksom på. Dere kommer ganske sikkert til å støte på det senere i studiet.

```
#Utsira
#View(df_arbl[df_arbl$knr == "1151",])
#View(df_arbl_p_alder[df_arbl_p_alder$knr == "1151",])
```

Gjøre arbeidsløshet «tidy»

Gjøre arbeidsløse i % aldersgrupper tidy; (`df_arbl_p_alder`)

```
head(df_arbl_p_alder, n = 2)
```

```
## # A tibble: 2 x 5
##   knr   knavn          alder    tid       alp
##   <chr> <chr>          <chr>    <chr>   <dbl>
## 1 0101  Halden (-2019) 15-74 år 1999M01     4
## 2 0101  Halden (-2019) 15-74 år 1999M02     4
```

Endre kategorier i variabelen alder

Å ha kategoriene "15-74 år" etc. er å be om problemer senere. Vi endrer til "15_74" etc..

```
df_arbl_p_alder <- df_arbl_p_alder %>%
    mutate(
        alder = case_when(
            alder == "15-74 år" ~ "15_74",
            alder == "15-29 år" ~ "15_29",
            alder == "30-74 år" ~ "30_74"
        )
    )
```

Bruk separate for splitte tid til aar og mnd

Fikse knavn

Vi skal fjerne (-2019) etc som skjemmer navnene. Her får vi bruk for «regular expressions» fra kapittel 14 *Strings*. Jeg vil anbefale funksjonen `str_remove()`. Når det gjelder `pattern` så husk at det vi vil fjerne starter med *mellomrom*, før vi får en parentes, så kommer tall og en bindestrek (bruk character class) før det hele avsluttes med en parentes slutt. Husk at vi må benytte \\ for «escape».

```
# For å finne riktig pattern, fjern når ferdig
# Fjern når du har funnet et pettern som virker
test <- tibble(test_navn = c("Halden (-2019)", "Moss (-2019)", "Sarpsborg (1992-2019)", "Fredrikstad (19
str_remove(
    string = test$test_navn,
    pattern = "H"
    )
```

```
## [1] "alden (-2019)"           "Moss (-2019)"
## [3] "Sarpsborg (1992-2019)"   "Fredrikstad (1994-2019)"
## [5] "valer (-2019)"
```

Slik skal de 10 første navnene nå se ut:

```
head(unique(arblos_aldersgrp$knavn), n = 10)
```

```
##  [1] "Halden"      "Moss"        "Sarpsborg"   "Fredrikstad" "Hvaler"
##  [6] "Aremark"     "Marker"      "Rømskog"     "Trøgstad"    "Spydeberg"
```

Pivot for å gjøre tidy

arblos_aldersgrp er ikke tidy. Hver observasjon er spredd over tre rekke, hhv. 15_29, 30_74 og 15_74. Pivot dataframen `arblos_aldersgrp` slik at vi får en observasjon per rekke. Bruk opsjonen `names_prefix` og sett denne lik "alp_" slik at de tre nye variabelnavnene starter med dette. Bruk samtidig mutate til å endre variablene `aar`og `mnd` til integer.

Slik skal `arblos_aldersgrp` nå se ut.

```
print(arblos_aldersgrp, n = 2)
```

```
## # A tibble: 77,330 x 7
##   knr   knavn    aar   mnd alp_15_74 alp_15_29 alp_30_74
##   <chr> <chr>  <int> <int>     <dbl>     <dbl>     <dbl>
## 1 0101  Halden  1999     1         4       6.3       3.2
## 2 0101  Halden  1999     2         4       6.1       3.3
## # ... with 77,328 more rows
```

Da skal `arblos_aldersgrp` være «tidy».

Gjøre arbeidløse (prosent) fordelt på kjønn tidy; (`df_arbl_p_kjonn`)

`df_arbl_p_kjonn` er *ikke* tidy.

```
print(arrange(df_arbl_p_kjonn, knr, knavn, tid, kjonn), n = 2)
```

```
## # A tibble: 154,660 x 5
##   knr   knavn          kjonn   tid     alp_k
##   <chr> <chr>          <chr>   <chr>   <dbl>
## 1 0101  Halden (-2019) Kvinner 1999M01   4.1
## 2 0101  Halden (-2019) Menn    1999M01   3.9
## # ... with 154,658 more rows
```

Hva som må fikses i`df_arbl_p_kjonn`:

1. Fiks knavn vha. `str_remove()`
2. Splitte tid i `aar` og `mnd`, gjør til integer i samme slengen
3. En observasjon per rekke, alp_Kvinner, alp_Menn som variabelnavn

Fikser knavn

```
print(arrange(df_arbl_p_kjonn, knr, knavn, tid, kjonn), n = 2)
```

```
## # A tibble: 154,660 x 5
##   knr   knavn  kjonn   tid     alp_k
##   <chr> <chr>  <chr>   <chr>   <dbl>
## 1 0101  Halden Kvinner 1999M01   4.1
## 2 0101  Halden Menn    1999M01   3.9
## # ... with 154,658 more rows
```

Splitte tid

```
print(arrange(df_arbl_p_kjonn, knr, knavn, aar, mnd, kjonn), n = 2)
```

```
## # A tibble: 154,660 x 6
##   knr   knavn  kjonn   aar   mnd   alp_k
##   <chr> <chr>  <chr>   <chr> <chr> <dbl>
## 1 0101  Halden Kvinner 1999  01      4.1
## 2 0101  Halden Menn    1999  01      3.9
## # ... with 154,658 more rows
```

En observasjon per rekke

```
print(arrange(df_arbl_p_kjonn, knr, knavn, aar, mnd), n = 2)
```

```
## # A tibble: 77,330 x 6
##   knr   knavn    aar   mnd alp_Menn alp_Kvinner
##   <chr> <chr>  <int> <int>    <dbl>       <dbl>
## 1 0101  Halden  1999     1      3.9         4.1
## 2 0101  Halden  1999     2      4           3.9
## # ... with 77,328 more rows
```

Da er `df_arbl_p_kjonn` «tidy».

Gjøre antall arbeidsløse fordelt på kjønn tidy; (`df_arbl`)

```
print(df_arbl, n = 2)
```

```
## # A tibble: 154,660 x 5
##   knr   knavn          tid     kjonn    al
##   <chr> <chr>          <chr>   <chr> <dbl>
## 1 0101  Halden (-2019) 1999M01 Menn    283
## 2 0101  Halden (-2019) 1999M02 Menn    291
## # ... with 154,658 more rows
```

Hva som må fikses i `df_arbl`:

1. Fiks knavn vha. `str_remove()`
2. Splitte tid i `aar` og `mnd`, gjør til integer i samme slengen
3. En observasjon per rekke, al_Kvinner, al_Menn som variabelnavn

Gjør som ovenfor.

Har du gjort det riktig skal `df_arbl` nå se slik ut.

```
print(df_arbl, n = 2)
```

```
## # A tibble: 77,330 x 6
##   knr   knavn    aar   mnd al_Menn al_Kvinner
##   <chr> <chr>  <int> <int>   <dbl>      <dbl>
## 1 0101  Halden  1999     1     283        248
## 2 0101  Halden  1999     2     291        236
## # ... with 77,328 more rows
```

Samle månedlige arbeisløshets-data i en tibble

Før vi slår i sammen de tre tibble-ene vi har gjort «tidy» er det lurt å sjekke en siste gang at de er på den formen vi ønsker.

Slår sammen tre tibble til en

Tenk nøye gjennom hvilke variabler man skal «joine» på.

```
## Joining, by = c("knr", "knavn", "aar", "mnd")
## Joining, by = c("knr", "knavn", "aar", "mnd")
```

```
print(al9914m, n = 2)
```

```
## # A tibble: 77,330 x 11
##   knr   knavn   aar   mnd al_Menn al_Kvinner alp_Menn alp_Kvinner alp_15_74
##   <chr> <chr> <int> <int>   <dbl>      <dbl>    <dbl>       <dbl>     <dbl>
## 1 0101  Hald~  1999     1     283        248      3.9         4.1         4
## 2 0101  Hald~  1999     2     291        236      4           3.9         4
## # ... with 77,328 more rows, and 2 more variables: alp_15_29 <dbl>,
## #   alp_30_74 <dbl>
```

```
names(al9914m)
```

```
##  [1] "knr"         "knavn"       "aar"         "mnd"         "al_Menn"
##  [6] "al_Kvinner"  "alp_Menn"    "alp_Kvinner" "alp_15_74"   "alp_15_29"
## [11] "alp_30_74"
```

```
dim(al9914m)
```

```
## [1] 77330    11
```

Gjøre befolkning «tidy»

Vi måtte dele befolkning i Menn og Kvinner da vi hentet dataene. Nå vil vi gjøre dem «tidy» og slå dem sammen.

```
dim(df_bef_M)
```

```
## [1] 708928      6
```

```
dim(df_bef_K)
```

```
## [1] 708928      6
```

```
print(arrange(df_bef_M, knr, knavn, tid, alder), n = 2)
```

```
## # A tibble: 708,928 x 6
##   knr   knavn            tid kjonn alder   bef
##   <chr> <chr>          <dbl> <chr> <chr> <dbl>
## 1 0101  Halden (-2019)  1999 Menn  000     147
## 2 0101  Halden (-2019)  1999 Menn  001     178
## # ... with 708,926 more rows
```

Hver observasjon er spredt over 106 rekker!

Vi vil bruke følgende strategi for å gjøre befolkning «tidy»:

1. Endre variabel-navn bef til hhv. df_bef_K og df_bef_M. Slette variabelen kjonn i både df_bef_M og df_bef_K.
2. Først slå sammen Menn og Kvinner så vi slipper å gjøre alt to ganger.
3. Fikse navn
4. Skifte navn fra tid til aar
5. Gjøre om alder fra chr til integer
6. Lage årsklasser for befolkningen. Vi trenger ikke 106
7. Gjøre pivot_ så vi blir «tidy»

Endre variabel navn, fjerne kjonn

```
df_bef_K <- df_bef_K %>%
    rename(bef_K = bef) %>%
    select(-kjonn)
df_bef_M <- df_bef_M %>%
    rename(bef_M = bef) %>%
    select(-kjonn)
```

Slå sammen df_bef_K og df_bef_M

```
## Joining, by = c("knr", "knavn", "tid", "alder")
```

```
dim(bef9914MK)
```

```
## [1] 708928      6
```

```
names(bef9914MK)
```

```
## [1] "knr"   "knavn" "tid"   "alder" "bef_K" "bef_M"
```

Fikser knavn

Fiks kommunenavnene.

Skifte navn fra tid til aar

Gjøre om alder fra chr til integer

Bruk først `str_replace()` til å endre "105+" til "105". Husk at "+" er et av tegnene som har spesiell betydning og må «escapes» med \\. Gjør så om fra chr til integer.

```
bef9914MK <- bef9914MK %>%
    mutate(
        alder = str_replace(alder, "105\\+", "105"),
        alder = as.integer(alder)
    )
```

Lage årsklasser for befolkningen

Vi vil benytte årsklassene "0_14", "15_29", "30_74" og "75_105" siden disse samsvarer med dem brukt for arbeidsløshet. Bruk `mutate()`, `case_when()` og `alder %in% c(0:14) ~ "0_14"` osv. for å definere årsklassene. Gi variabelen med årsklassene navnet `ald_int`. Bruk så `group_by()` på knr, knavn, aar og ald_int og summer opp vha. `sum()`. La siste linjen i `summarise()` være `bef_MK = bef_K + bef_M` så får vi også med oss total befolkning for de ulike årsklassene.

```
## `summarise()` regrouping output by 'knr', 'knavn', 'aar' (override with `.groups` argument)
```

```
print(bef9914MK, m = 2)
```

```
## # A tibble: 26,752 x 7
## # Groups:   knr, knavn, aar [6,688]
##    knr   knavn  aar   ald_int bef_K bef_M bef_MK
##    <chr> <chr>  <chr> <chr>   <dbl> <dbl>  <dbl>
##  1 0101  Halden 1999  0_14     2351  2534   4885
##  2 0101  Halden 1999  15_29    2445  2530   4975
##  3 0101  Halden 1999  30_74    7110  7067  14177
##  4 0101  Halden 1999  75_105   1597   889   2486
##  5 0101  Halden 2000  0_14     2388  2592   4980
##  6 0101  Halden 2000  15_29    2416  2514   4930
##  7 0101  Halden 2000  30_74    7163  7135  14298
##  8 0101  Halden 2000  75_105   1612   913   2525
##  9 0101  Halden 2001  0_14     2429  2674   5103
## 10 0101  Halden 2001  15_29    2416  2526   4942
## # ... with 26,742 more rows
```

Gjør `pivot_` så vi blir «tidy».

Gjør du ting rett skal `bef9914MK` nå se slik ut:

```
print(bef9914MK, m = 2)
```

```
## # A tibble: 6,688 x 15
## # Groups:   knr, knavn, aar [6,688]
##    knr   knavn aar   bef_K_0_14 bef_K_15_29 bef_K_30_74 bef_K_75_105 bef_M_0_14
##    <chr> <chr> <chr>      <dbl>       <dbl>       <dbl>        <dbl>      <dbl>
##  1 0101  Hald~ 1999        2351        2445        7110         1597       2534
##  2 0101  Hald~ 2000        2388        2416        7163         1612       2592
##  3 0101  Hald~ 2001        2429        2416        7263         1632       2674
##  4 0101  Hald~ 2002        2438        2409        7297         1630       2707
##  5 0101  Hald~ 2003        2468        2381        7370         1627       2729
##  6 0101  Hald~ 2004        2466        2350        7408         1640       2707
##  7 0101  Hald~ 2005        2464        2339        7494         1610       2713
##  8 0101  Hald~ 2006        2424        2377        7583         1574       2678
##  9 0101  Hald~ 2007        2410        2413        7601         1604       2679
## 10 0101  Hald~ 2008        2458        2421        7690         1583       2655
## # ... with 6,678 more rows, and 7 more variables: bef_M_15_29 <dbl>,
## #   bef_M_30_74 <dbl>, bef_M_75_105 <dbl>, bef_MK_0_14 <dbl>,
## #   bef_MK_15_29 <dbl>, bef_MK_30_74 <dbl>, bef_MK_75_105 <dbl>
```

og ha dimensjonene:

```
dim(bef9914MK)
```

```
## [1] 6688   15
```

Skrive «tidy» data til fil

```r
write_csv(al9914m, "al9914m.csv")
write_csv(bef9914MK, "bef9914MK.csv")
```

```r
rm(list = ls())
```