

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
```

```
library(modelr)
```

```
library(broom)
```

```
##
```

```
## Attaching package: 'broom'
```

```
## The following object is masked from 'package:modelr':
```

```
##
```

```
##     bootstrap
```

Modeller

Leser inn data

```
arblos <- read_csv("data/al9914m.csv")
```

```
##
## -- Column specification -----
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   aar = col_double(),
##   mnd = col_double(),
##   al_Menn = col_double(),
##   al_Kvinner = col_double(),
##   alp_Menn = col_double(),
##   alp_Kvinner = col_double(),
##   alp_15_74 = col_double(),
##   alp_15_29 = col_double(),
##   alp_30_74 = col_double()
## )
```

```
bef <- read_csv("data/bef9914MK.csv")
```

```
##
## -- Column specification -----
## cols(
##   knr = col_character(),
##   knavn = col_character(),
##   aar = col_double(),
##   bef_K_0_14 = col_double(),
##   bef_K_15_29 = col_double(),
##   bef_K_30_74 = col_double(),
##   bef_K_75_105 = col_double(),
##   bef_M_0_14 = col_double(),
##   bef_M_15_29 = col_double(),
##   bef_M_30_74 = col_double(),
##   bef_M_75_105 = col_double(),
##   bef_MK_0_14 = col_double(),
##   bef_MK_15_29 = col_double(),
##   bef_MK_30_74 = col_double(),
##   bef_MK_75_105 = col_double()
## )
```

Modeller med data fra *bef* (befolkning)

```
names(bef)
```

```
## [1] "knr"          "knavn"        "aar"          "bef_K_0_14"
## [5] "bef_K_15_29"  "bef_K_30_74"  "bef_K_75_105" "bef_M_0_14"
## [9] "bef_M_15_29"  "bef_M_30_74"  "bef_M_75_105" "bef_MK_0_14"
## [13] "bef_MK_15_29" "bef_MK_30_74" "bef_MK_75_105"
```

```
names(arblos)
```

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

Arbeidsledighetsprosenten blir beregnet som: $\text{arbl\%} = \text{antall arb. ledige} / \text{arbeidsstyrken}$. Arbeidsstyrken er her dem man anser egnet for arbeid dvs. uføretrygdete etc. er trukket ut. Arbeidsstyrken i en alderskategori er derfor langt mindre enn befolkningen i den tilsvarende alderskategorien.

Dessverre kjenne vi ikke arbeidsstyrken, men vi kan beregne den vha.: $\text{arbeidsstyrken} = \text{antall arb. ledige} / \text{arb. ledighetsprosent}$.

```
arblos <- arblos %>%
  mutate(
    wf_K = (al_Kvinner/alp_Kvinner)*100,
    wf_M = (al_Menn/alp_Menn)*100,
    wf_KM = wf_K + wf_M
  )
```

```
arblos
```

```
## # A tibble: 77,330 x 14
##   knr   knavn   aar   mnd al_Menn al_Kvinner alp_Menn alp_Kvinner alp_15_74
##   <chr> <chr> <dbl> <dbl>   <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
## 3 0101 Hald~ 1999     3    290         230         4         3.8         3.9
## 4 0101 Hald~ 1999     4    244         207        3.4         3.4         3.4
## 5 0101 Hald~ 1999     5    210         179        2.9         3         2.9
## 6 0101 Hald~ 1999     6    227         203        3.2         3.4         3.2
## 7 0101 Hald~ 1999     7    265         273        3.7         4.5         4.1
## 8 0101 Hald~ 1999     8    288         278         4         4.6         4.3
## 9 0101 Hald~ 1999     9    230         201        3.2         3.3         3.3
## 10 0101 Hald~ 1999    10    225         207        3.1         3.4         3.3
## # ... with 77,320 more rows, and 5 more variables: alp_15_29 <dbl>,
## #   alp_30_74 <dbl>, wf_K <dbl>, wf_M <dbl>, wf_KM <dbl>
```

Arbeidsstyrken

Når befolkningen øker vil også arbeidsstyrken øke. Det er derfor mer naturlig å se på arbeidsstyrken relativt til delen av befolkningen som er i yrkesaktiv alder (15-74 år her).

```
names(bef)
```

```
## [1] "knr"          "knavn"        "aar"          "bef_K_0_14"
## [5] "bef_K_15_29"  "bef_K_30_74"  "bef_K_75_105" "bef_M_0_14"
## [9] "bef_M_15_29"  "bef_M_30_74"  "bef_M_75_105" "bef_MK_0_14"
## [13] "bef_MK_15_29" "bef_MK_30_74" "bef_MK_75_105"
```

Vi skal starte med å lage et datasett med arbeidsstyrken (wf) for hele landet samlet, men fordelt på de tre kategorien kvinner, menn og kvinner + menn.

Bruk data for januar hvert år til å beregne wf på landsbasis

```
# årlige data landet samlet
wf <- arblos %>%
  filter(mnd == 1) %>%
```

```
group_by(aar) %>%
  summarise(
    wf_K = sum(wf_K, na.rm = TRUE),
    wf_M = sum(wf_M, na.rm = TRUE),
    wf_KM = wf_K + wf_M
  )
```

'summarise()' ungrouping output (override with '.groups' argument)

```
dim(wf)
```

```
## [1] 16 4
```

```
names(wf)
```

```
## [1] "aar" "wf_K" "wf_M" "wf_KM"
```

Summer de ulike årskategoriene for de to kjønnene og menn+kvinner for å finne total befolkning de ulike årene. Bruk mutate til å lage de nye variablene.

```
bef <- bef %>%
  mutate(
    bef_K = bef_K_0_14 + bef_K_15_29 + bef_K_30_74 + bef_K_75_105,
    bef_M = bef_M_0_14 + bef_M_15_29 + bef_M_30_74 + bef_M_75_105,
    bef_KM = bef_K + bef_M
  )
```

```
dim(bef)
```

```
## [1] 6688 18
```

```
names(bef)
```

```
## [1] "knr" "knavn" "aar" "bef_K_0_14"
## [5] "bef_K_15_29" "bef_K_30_74" "bef_K_75_105" "bef_M_0_14"
## [9] "bef_M_15_29" "bef_M_30_74" "bef_M_75_105" "bef_MK_0_14"
## [13] "bef_MK_15_29" "bef_MK_30_74" "bef_MK_75_105" "bef_K"
## [17] "bef_M" "bef_KM"
```

Legg befolkningsdata variablene bef_K, bef_M, bef_KM til wf. Husk at de må aggregeres for hele landet (group_by() og så summarise() før de «joines»). Bruk tilslutt mutate() make_date() for å lage en ny variabel år som er en date, dvs aar + month=1L + day=1L.

```
wf_hjelp <- bef %>%
  group_by(aar) %>%
  summarise(bef_K = sum(bef_K),
            bef_M = sum(bef_M),
            bef_KM = sum(bef_KM))
```

'summarise()' ungrouping output (override with '.groups' argument)

```
wf <- left_join(wf, wf_hjelp)
```

```
## Joining, by = "aar"
```

```
wf <- wf %>%
  select(aar, bef_K, bef_M, bef_KM, wf_K, wf_M, wf_KM) %>%
  mutate(år = make_date(aar, month = 1L, day = 1L))
rm(wf_hjelp)
```

```
wf
```

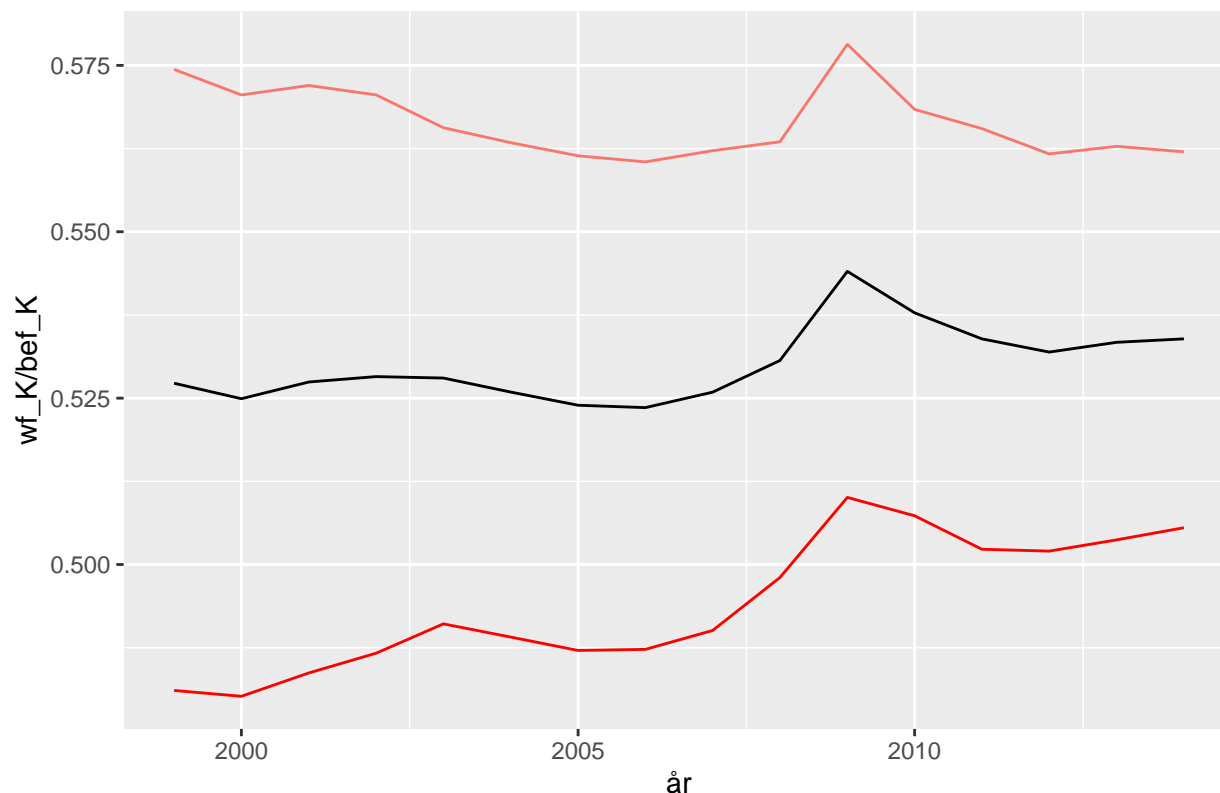
```
## # A tibble: 16 x 8
##   aar    bef_K    bef_M    bef_KM    wf_K    wf_M    wf_KM år
##   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <date>
## 1 1999 2144642 2099157 4243799 1031744. 1205745. 2237489. 1999-01-01
## 2 2000 2159722 2115843 4275565 1037097. 1207206. 2244303. 2000-01-01
## 3 2001 2170177 2129548 4299725 1049731. 1218061. 2267791. 2001-01-01
## 4 2002 2180894 2141321 4322215 1061392. 1221762. 2283154. 2002-01-01
## 5 2003 2195125 2155667 4350792 1077983. 1219325. 2297307. 2003-01-01
## 6 2004 2206714 2167751 4374465 1079308. 1221288. 2300596. 2004-01-01
## 7 2005 2220663 2182844 4403507 1081663. 1225478. 2307142. 2005-01-01
## 8 2006 2236334 2200374 4436708 1089654. 1233306. 2322960. 2006-01-01
## 9 2007 2252272 2222818 4475090 1103816. 1249628. 2353444. 2007-01-01
## 10 2008 2274215 2256203 4530418 1132662. 1271414. 2404076. 2008-01-01
## 11 2009 2299522 2289506 4589028 1172942. 1323707. 2496649. 2009-01-01
## 12 2010 2325447 2319852 4645299 1179755. 1318575. 2498330. 2010-01-01
## 13 2011 2352727 2353491 4706218 1181768. 1330901. 2512669. 2011-01-01
## 14 2012 2380218 2390822 4771040 1194903. 1342914. 2537817. 2012-01-01
## 15 2013 2407722 2426904 4834626 1212788. 1365955. 2578743. 2013-01-01
## 16 2014 2434035 2458438 4892473 1230477. 1381665. 2612141. 2014-01-01
```

Vi vil nå se på arbeidsstyrke relativt til befolkning på landsbasis, dvs. wf_K/bef_K etc.

Plot dataen vha. geom_line() for de tre kategoriene. Bruk år som x-variabel.

```
wf %>%
  ggplot(wf, mapping = aes(x = år)) +
  # geom_line(mapping = aes(y = wf_K/bef_K, color = "...")) +
  # geom_line(mapping = aes(y = wf_M/bef_M, color = "darkblue")) +
  # geom_line(mapping = aes(y = wf_KM/bef_KM)) +
  geom_line(mapping = aes(y = wf_K/bef_K, color = "red")) +
  geom_line(mapping = aes(y = wf_M/bef_M, color = "darkblue")) +
  geom_line(mapping = aes(y = wf_KM/bef_KM, color = "black")) +
  ggtitle("Arbeidsstyrke relativt til befolkning. Menn blå, kvinner rød og totalt svart.") +
  theme(legend.position = "none")
```

Arbeidsstyrke relativt til befolkning. Menn blå, kvinner rød og totalt svart.



Hvordan kan pukkelen rett før 2010 forklares? Dette kan skyldes finanskrisen.

```
names(arblos)
```

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

Vi vil nå generere dat for arbeidsstyrken på fylkesbasis. Husk at de to første sifferene i knr angir fylket en kommune ligger i. Bruk dataene i arblos til å finne arbeidsstyrken på fylkesbasis (wf_f). Start med å bruke mutate() til å lage en ny variabel fylke. Grupper så og finn wf_K_f, wf_M_f og wf_KM_f vha. summarise(). Lag til slutt en ny variabel år som en date. Velg til slutt de relevante variablen vha. select().

```
# årlige data per fylke
wf_f <- arblos %>%
  mutate(
    fylke = substr(knr, start = 1, stop = 2)
  ) %>%
  group_by(aar, mnd, fylke) %>%
  summarise(
    wf_K_f = sum(wf_K, na.rm = TRUE),
    wf_M_f = sum(wf_M, na.rm = TRUE),
    wf_KM_f = wf_K_f + wf_M_f
  ) %>%
  mutate(år = make_date(aar, mnd, day=1L)) %>%
  select(aar, mnd, år, fylke, wf_K_f, wf_M_f, wf_KM_f)
```

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

```
print(wf_f, n = 5)
```

```
## # A tibble: 3,515 x 7
## # Groups:   aar, mnd [185]
##   aar    mnd år      fylke wf_K_f wf_M_f wf_KM_f
##   <dbl> <dbl> <date>   <chr>   <dbl>   <dbl>   <dbl>
## 1  1999     1 1999-01-01 01     57671.  67408. 125079.
## 2  1999     1 1999-01-01 02     120670. 133018. 253688.
## 3  1999     1 1999-01-01 03     133500  147097. 280597.
## 4  1999     1 1999-01-01 04      42237.  49356.  91593.
## 5  1999     1 1999-01-01 05      41178.  47990.  89168.
## # ... with 3,510 more rows
```

Lag også en ny tibble bef_f fra bef som inneholder befolkningen i hvert fylke hvert år.

```
bef_f <- bef %>%
  mutate(
    fylke = substr(knr, start = 1, stop = 2)
  ) %>%
  group_by(aar, fylke) %>%
  summarise(
    bef_K_f = sum(bef_K, na.rm = TRUE),
    bef_M_f = sum(bef_M, na.rm = TRUE),
    bef_KM_f = bef_K_f + bef_M_f
  ) %>%
  select(aar, fylke, bef_K_f, bef_M_f, bef_KM_f)
```

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

```
bef_f
```

```
## # A tibble: 304 x 5
## # Groups:   aar [16]
##   aar fylke bef_K_f bef_M_f bef_KM_f
##   <dbl> <chr>   <dbl>   <dbl>   <dbl>
## 1  1999 01     123893  119292  243185
## 2  1999 02     229550  224826  454376
## 3  1999 03     257046  238442  495488
## 4  1999 04      93317   91073  184390
## 5  1999 05      87879   86384  174263
## 6  1999 06     117734  114475  232209
## 7  1999 07      72957   70085  143042
## 8  1999 08      82644   80016  162660
## 9  1999 09      50362   49878  100240
## 10 1999 10      76790   75175  151965
## # ... with 294 more rows
```

Lag til slutt tibble-en wf_f_bef som innholde arbeidsstyrke (wf-f) og befolkning (bef-f) på fylkesnivå for hvert år.

```
wf_f_bef <- wf_f %>%
  right_join(bef_f) %>%
  arrange(fylke, aar, mnd)
```

```
## Joining, by = c("aar", "fylke")
```

Da skal wf_f_bef se slik ut:

```
print(arrange(wf_f_bef, fylke, aar, mnd))
```

```
## # A tibble: 3,515 x 10
## # Groups:   aar, mnd [185]
##   aar   mnd år      fylke wf_K_f wf_M_f wf_KM_f bef_K_f bef_M_f bef_KM_f
##   <dbl> <dbl> <date>   <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1999     1 1999-01-01 01    57671.  67408.  125079.  123893  119292  243185
## 2 1999     2 1999-02-01 01    57693.  67526.  125220.  123893  119292  243185
## 3 1999     3 1999-03-01 01    57108.  67551.  124659.  123893  119292  243185
## 4 1999     4 1999-04-01 01    57526.  67355.  124881.  123893  119292  243185
## 5 1999     5 1999-05-01 01    57285.  67189.  124474.  123893  119292  243185
## 6 1999     6 1999-06-01 01    57529.  66792.  124321.  123893  119292  243185
## 7 1999     7 1999-07-01 01    57393.  67690.  125083.  123893  119292  243185
## 8 1999     8 1999-08-01 01    57531.  66998.  124529.  123893  119292  243185
## 9 1999     9 1999-09-01 01    57264.  67609.  124873.  123893  119292  243185
## 10 1999    10 1999-10-01 01    57702.  66907.  124610.  123893  119292  243185
## # ... with 3,505 more rows
```

Plot nå arbeidsstyrke relativt til befolkning vha. geom_line()

```
wf_f_bef %>%
  ggplot() +
  geom_line(wf_f_bef, mapping = aes(x = år,
                                     y = wf_KM_f/bef_KM_f,
                                     color = fylke)) +
  theme(legend.position = "bottom")
```




Lage regioner

Alle fylkene blir litt rotete så vi definerer istedet seks regioner vha. `case_when()` og lager en ny tibble `wf_r` fra `wf_f_bef`.

```
# region
wf_r <- wf_f_bef %>%
  mutate(
    dato = ymd(paste(aar, mnd, "01", sep = "-")),
    region = case_when(
      as.numeric(fylke) == 3 ~ "Oslo",
      as.numeric(fylke) %in% c(1:2, 4:8) ~ "Østlandet",
      as.numeric(fylke) %in% c(9, 10) ~ "Sørlandet",
      as.numeric(fylke) %in% c(11, 12, 14, 15) ~ "Vestlandet",
      as.numeric(fylke) %in% c(16, 17) ~ "Trøndelag",
      as.numeric(fylke) %in% c(18, 19, 20) ~ "Nord-Norge"
    )
  ) %>%
  group_by(dato, region) %>%
  summarise(
    wf_K_r = sum(wf_K_f, na.rm = TRUE),
    wf_M_r = sum(wf_M_f, na.rm = TRUE),
    wf_KM_r = wf_K_r + wf_M_r,
    bef_K_r = sum(bef_K_f, na.rm = TRUE),
    bef_M_r = sum(bef_M_f, na.rm = TRUE),
```

```

    bef_KM_r = bef_K_r + bef_M_r
  ) %>%
  select(dato, region, wf_K_r, wf_M_r, wf_KM_r, bef_K_r, bef_M_r, bef_KM_r)

```

'summarise()' regrouping output by 'dato' (override with '.groups' argument)

Plot nå for regionene wf_KM_r/bef_KM_r, både vha. `geom_line()` og `geom_smooth()`. La farge vise regionene. Sett denne i `ggplot()` slik at det gjelder for både `geom_line()` og `geom_smooth()`. Sett i tillegg `alpha = 0.5` for `geom_line()` og `se = FALSE` for `geom_smooth()`. Legg til `theme(legend.position = "bottom")` til slutt for å få legend under plottet.

```

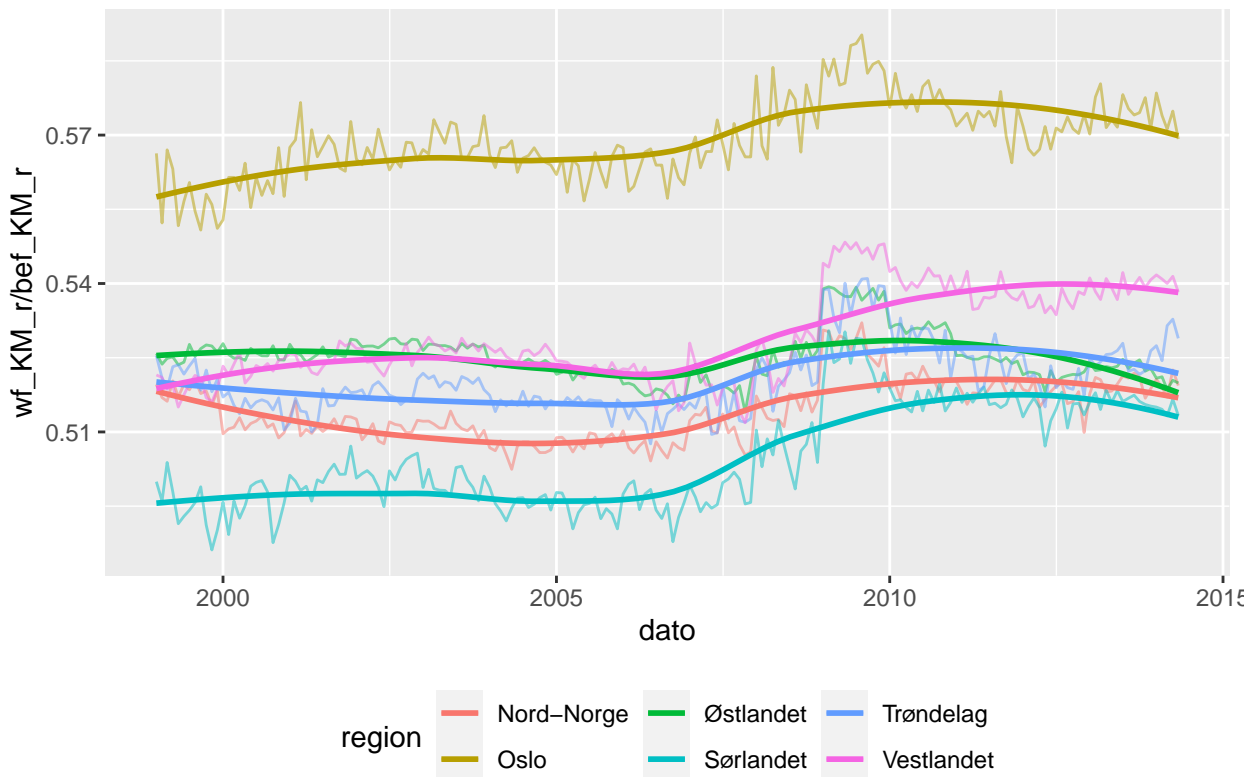
wf_r %>%
  ggplot(wf_r, mapping = aes(x = dato,
                             y = wf_KM_r/bef_KM_r,
                             color = region)) +

  geom_line(alpha = 0.5) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  ggtitle("Arbeidsstyrkens andel av befolkningen for regionene.")

```

'geom_smooth()' using method = 'loess' and formula 'y ~ x'

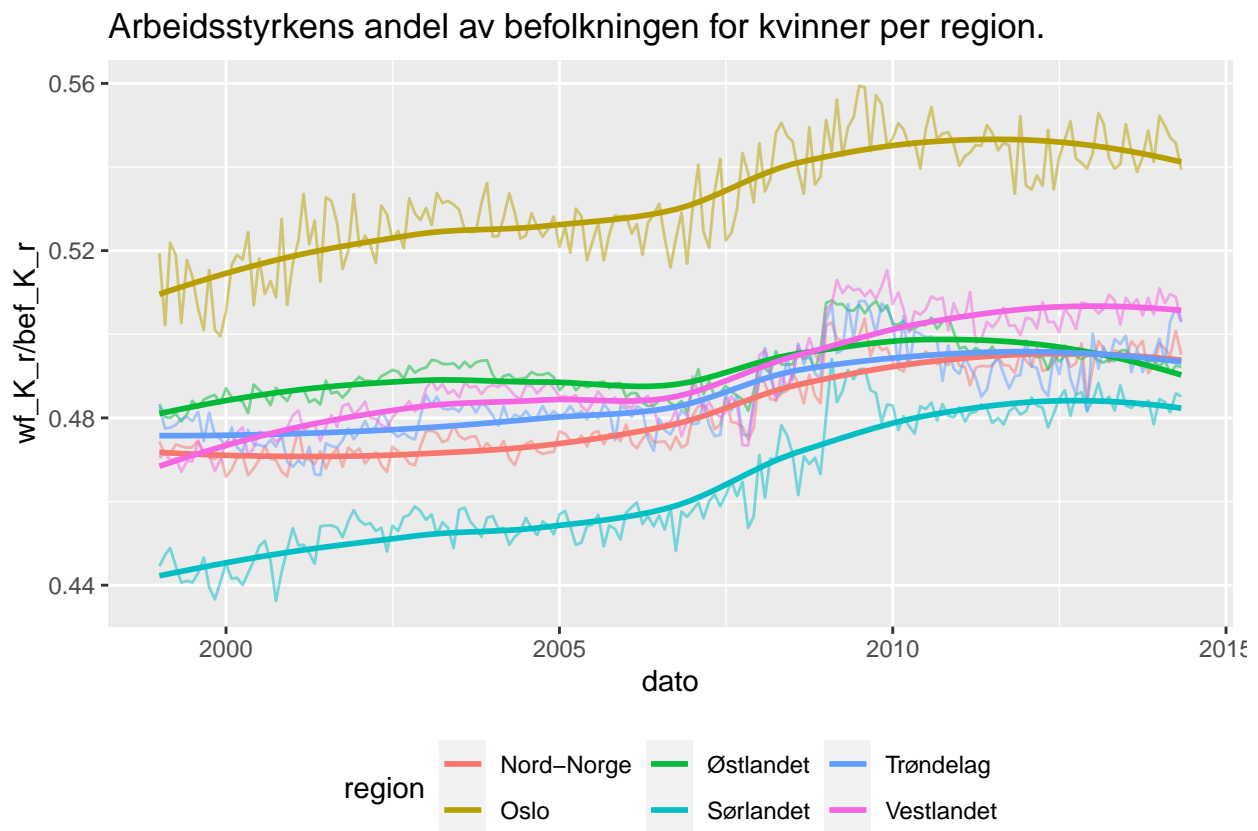
Arbeidsstyrkens andel av befolkningen for regionene.



Plot for kvinner.

```
wf_r %>%
  ggplot(wf_r, mapping = aes(x = dato,
                             y = wf_K_r/bef_K_r,
                             color = region)) +
  geom_line(alpha = 0.5) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  ggtitle("Arbeidsstyrkens andel av befolkningen for kvinner per region.")
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

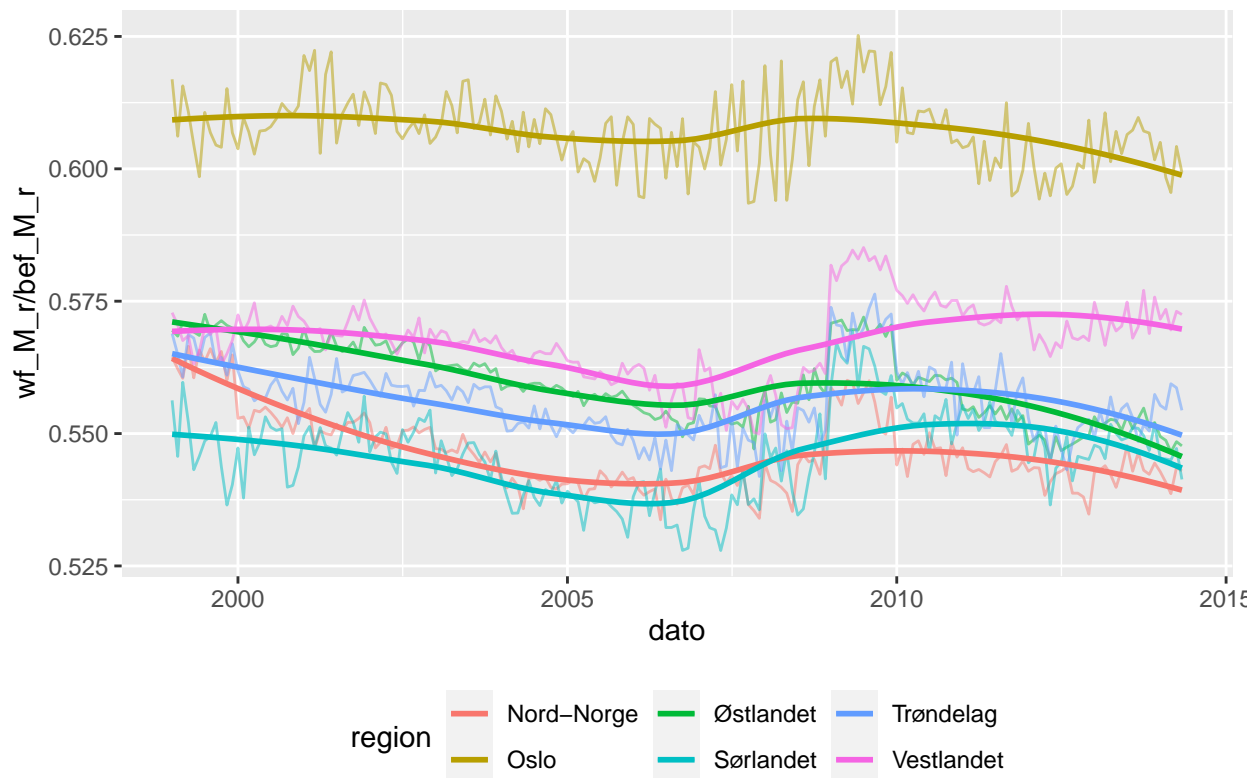


Plot for menn.

```
wf_r %>%
  ggplot(wf_r, mapping = aes(x = dato,
                             y = wf_M_r/bef_M_r,
                             color = region)) +
  geom_line(alpha = 0.5) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  ggtitle("Arbeidsstyrkens andel av befolkningen for menn per region.")
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

Arbeidsstyrkens andel av befolkningen for menn per region.



Forlar kort den generelle utviklingen i arbeidsstyrken. Hva er det som «redder oss»? menn har en nedgang i arbeidsstyrkens andel av befolkningen fra 2000-2007, men det er en liten stigning før det er nedgang mot 2015.

nest() arblos

Da skal vi jobbe direkte med arbeidsløshet og lage lineære modeller for hver av de 418 kommunene. Modellen vi skal lage er på ingen måte perfekt. Vi er interessert i selve teknikken med å organisere dataene og kjøre modeller på mange subsett av dataene (her for hver kommune).

Vi vil se på en modell der vi forklarer arbeidsstyrken i en kommune vha. ungdomsledighet og ledighet blant litt eldre arbeidstakere (30-74 år). En hypotese er at vi vil se en negativ sammenheng mellom ungdomsledighet og arbeidsstyrken. De unge flytter hvis det ikke er jobb.

Vi starter med å gruppere på kommune og nest-e dataene.

```
arblos_by_knr <- arblos %>%
  group_by(knr, knavn) %>%
  nest()
```

```
print(arblos_by_knr, n = 4)
```

```
## # A tibble: 418 x 3
## # Groups:   knr, knavn [418]
##   knr   knavn   data
##   <chr> <chr>   <list>
```

```
## 1 0101 Halden      <tibble [185 x 12]>
## 2 0104 Moss       <tibble [185 x 12]>
## 3 0105 Sarpsborg  <tibble [185 x 12]>
## 4 0106 Fredrikstad <tibble [185 x 12]>
## # ... with 414 more rows
```

```
arblos_by_knr$data[[1]]
```

```
## # A tibble: 185 x 12
##   aar   mnd al_Menn al_Kvinner alp_Menn alp_Kvinner alp_15_74 alp_15_29
##   <dbl> <dbl>   <dbl>       <dbl>   <dbl>       <dbl>   <dbl>   <dbl>
## 1 1999     1    283         248     3.9         4.1     4       6.3
## 2 1999     2    291         236     4           3.9     4       6.1
## 3 1999     3    290         230     4           3.8     3.9     5.9
## 4 1999     4    244         207     3.4         3.4     3.4     4.9
## 5 1999     5    210         179     2.9         3       2.9     3.8
## 6 1999     6    227         203     3.2         3.4     3.2     4.2
## 7 1999     7    265         273     3.7         4.5     4.1     5.2
## 8 1999     8    288         278     4           4.6     4.3     6.6
## 9 1999     9    230         201     3.2         3.3     3.3     4.8
## 10 1999    10    225         207     3.1         3.4     3.3     4.8
## # ... with 175 more rows, and 4 more variables: alp_30_74 <dbl>, wf_K <dbl>,
## #   wf_M <dbl>, wf_KM <dbl>
```

Vi har nå en tibble med data for hver kommune inne i tibble-en `arblos_by_knr`.

Skriv en funksjon som kjører den lineære modellen $wf_KM \sim alp_15_29 + alp_30_74$ på en input dataframe `a_df`. Kall funksjonen `mod1` (i magel på noe bedre navn).

Kjøre så modellen vha. funksjonen `mod1` på data i `arblos_by_knr`, og lag en list-column i `arblos_by_knr` som inneholder modellen. Kjør også `tidy` og `glance` fra `broom` på modellene for å få hhv. koeffisienter og ulike summary av modellen. Lagre resultatene i hhv. `mod1_arblos`, `mod1_arblos_coef` og `mod1_arblos_sum` (i siste er sum forkortelse for summary).

```
arblos_by_knr <- arblos_by_knr %>%
  mutate(mod1_arblos = map(data, .f = mod1)) %>%
  mutate(mod1_arblos_coef = map(.x = mod1_arblos, .f = tidy)) %>%
  mutate(mod1_arblos_sum = map(.x = mod1_arblos, .f = glance))
```

```
arblos_by_knr
```

```
## # A tibble: 418 x 6
## # Groups:   knr, knavn [418]
##   knr   knavn      data      mod1_arblos mod1_arblos_coef mod1_arblos_sum
##   <chr> <chr>    <list>      <list>      <list>      <list>
## 1 0101 Halden  <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 2 0104 Moss   <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 3 0105 Sarpsborg <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 4 0106 Fredrikst~ <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 5 0111 Hvaler  <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 6 0118 Aremark  <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 7 0119 Marker   <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
## 8 0121 Rømskog  <tibble [185 ~ <lm>      <tibble [3 x 5]> <tibble [1 x 12~
```

```
## 9 0122 Trøgstad <tibble [185 ~ <lm> <tibble [3 x 5]> <tibble [1 x 12~
## 10 0123 Spydeberg <tibble [185 ~ <lm> <tibble [3 x 5]> <tibble [1 x 12~
## # ... with 408 more rows
```

Kommunestørrelse

Vi lager oss så en ny kategori variabel for kommunestørrelse.

```
kom_str <- bef %>%
  filter(aar == 2014) %>%
  mutate(
    k_str = case_when(
      bef_KM <= 2500 ~ "Svært liten",
      bef_KM > 2500 & bef_KM <= 6000 ~ "Liten",
      bef_KM > 6000 & bef_KM <= 20000 ~ "Middels",
      bef_KM > 20000 & bef_KM <= 60000 ~ "Stor",
      bef_KM > 60000 & bef_KM <= 100000 ~ "Svært stor"
    )
  ) %>%
  select(knr, knavn, bef_KM, k_str)
head(kom_str)
```

```
## # A tibble: 6 x 4
##   knr   knavn   bef_KM k_str
##   <chr> <chr>   <dbl> <chr>
## 1 0101 Halden    29827 Stor
## 2 0104 Moss      30987 Stor
## 3 0105 Sarpsborg 53480 Stor
## 4 0106 Fredrikstad 76810 Svært stor
## 5 0111 Hvaler     4354 Liten
## 6 0118 Aremark    1396 Svært liten
```

Pakker ut og henter model karakteristika

```
mod_arbl_re <- arblos_by_knr %>%
  unnest(c("mod1_arblos_coef", "mod1_arblos_sum"), names_repair = "universal") %>%
  select(knr, knavn, term, estimate, std.error, p.value...9, adj.r.squared)
```

```
## New names:
## * statistic -> statistic...8
## * p.value -> p.value...9
## * statistic -> statistic...13
## * p.value -> p.value...14
```

```
print(mod_arbl_re, n = 10)
```

```
## # A tibble: 1,254 x 7
## # Groups:   knr, knavn [418]
##   knr   knavn   term      estimate std.error p.value...9 adj.r.squared
##   <chr> <chr>   <chr>      <dbl>      <dbl>      <dbl>      <dbl>
```

```
## 1 0101 Halden (Intercept) 14288. 153. 1.23e-155 0.110
## 2 0101 Halden alp_15_29 228. 58.0 1.19e- 4 0.110
## 3 0101 Halden alp_30_74 -516. 105. 1.97e- 6 0.110
## 4 0104 Moss (Intercept) 14030. 252. 3.65e-116 0.0102
## 5 0104 Moss alp_15_29 47.0 97.2 6.30e- 1 0.0102
## 6 0104 Moss alp_30_74 66.3 206. 7.48e- 1 0.0102
## 7 0105 Sarpsborg (Intercept) 25250. 435. 2.02e-119 0.0138
## 8 0105 Sarpsborg alp_15_29 273. 164. 9.72e- 2 0.0138
## 9 0105 Sarpsborg alp_30_74 -416. 381. 2.77e- 1 0.0138
## 10 0106 Fredrikstad (Intercept) 36302. 394. 1.15e-154 0.550
## # ... with 1,244 more rows
```

Slå sammen kom_str og mod_arbl_re vha. left_join().

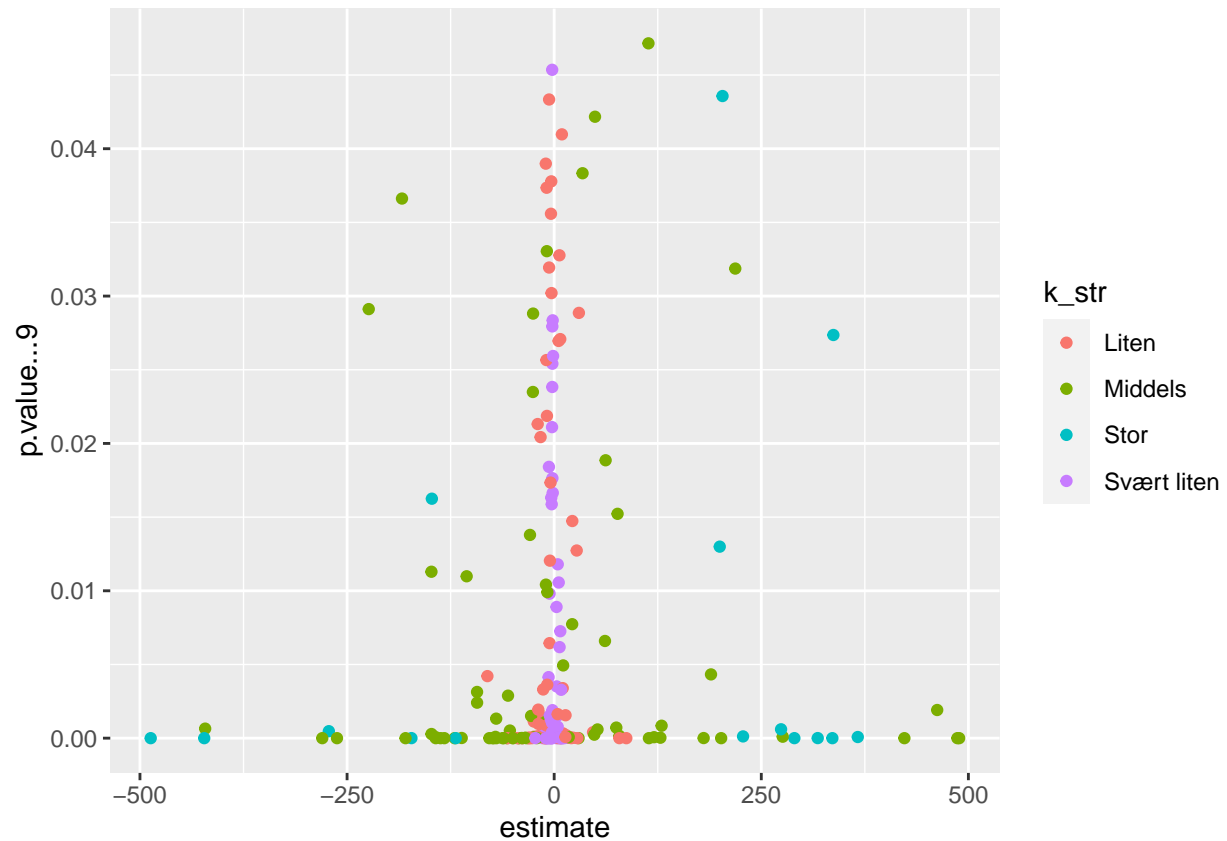
```
kom_str_mod <- mod_arbl_re %>%
  left_join(kom_str, by = c("knr", "knavn"))
```

```
print(kom_str_mod, n = 5)
```

```
## # A tibble: 1,254 x 9
## # Groups:   knr, knavn [418]
##   knr   knavn term      estimate std.error p.value...9 adj.r.squared bef_KM k_str
##   <chr> <chr> <chr>      <dbl>      <dbl>      <dbl>      <dbl> <dbl> <chr>
## 1 0101 Halden (Inter~ 14288.      153.      1.23e-155      0.110 29827 Stor
## 2 0101 Halden alp_15~ 228.        58.0      1.19e- 4      0.110 29827 Stor
## 3 0101 Halden alp_30~ -516.       105.      1.97e- 6      0.110 29827 Stor
## 4 0104 Moss (Inter~ 14030.      252.      3.65e-116      0.0102 30987 Stor
## 5 0104 Moss alp_15~ 47.0        97.2      6.30e- 1      0.0102 30987 Stor
## # ... with 1,249 more rows
```

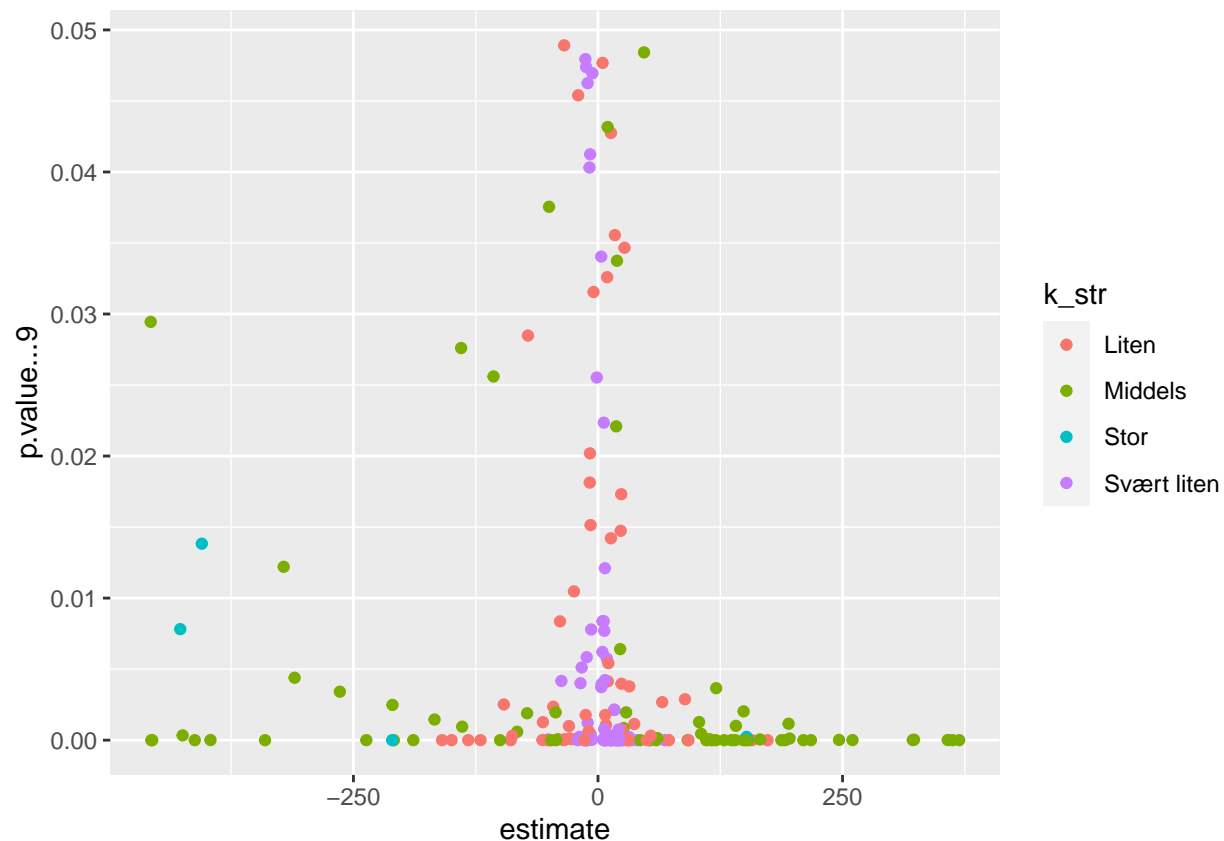
Vi plotter koeffisientene som er signifikante og lar farge vise kommune størrelse. Tar vekk ekstreme estimat.

```
kom_str_mod %>%
  filter(term == "alp_15_29") %>%
  filter(estimate > -500 & estimate < 500 ) %>%
  filter(p.value...9 < 0.05) %>%
  ggplot(mapping = aes(
    x = estimate,
    y = p.value...9,
    colour = k_str
  )) +
  geom_point()
```



Lag tilsvarende plot for alp_30_74.

```
kom_str_mod %>%
  filter(term == "alp_30_74") %>%
  filter(estimate > -500 & estimate < 500 ) %>%
  filter(p.value...9 < 0.05) %>%
  ggplot(mapping = aes(
    x = estimate,
    y = p.value...9,
    colour = k_str
  )) +
  geom_point()
```

Hvor mange har vi?

```
kom_str_mod %>%
  filter(term == "alp_15_29") %>%
  group_by(k_str) %>%
  summarise(n = n())
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 5 x 2
##   k_str      n
##   <chr>    <int>
## 1 Liten    126
## 2 Middels  115
## 3 Stor     39
## 4 Svært liten 128
## 5 Svært stor  10
```

```
kom_str_mod %>%
  filter(term == "alp_15_29") %>%
  filter(estimate > - 500 & estimate < 500 ) %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(n = n())
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 4 x 2
##   k_str      n
##   <chr>    <int>
## 1 Liten      64
## 2 Middels    75
## 3 Stor       15
## 4 Svært liten 58
```

```
kom_str_mod %>%
  filter(term == "alp_30_74") %>%
  filter(estimate > - 500 & estimate < 500 ) %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(n = n())
```

'summarise()' ungrouping output (override with '.groups' argument)

```
## # A tibble: 4 x 2
##   k_str      n
##   <chr>    <int>
## 1 Liten      68
## 2 Middels    67
## 3 Stor        4
## 4 Svært liten 71
```

```
kom_str_mod %>%
  filter(term == "alp_15_29") %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(mean_15_29 = mean(estimate))
```

'summarise()' ungrouping output (override with '.groups' argument)

```
## # A tibble: 5 x 2
##   k_str      mean_15_29
##   <chr>    <dbl>
## 1 Liten      -2.93
## 2 Middels    0.0802
## 3 Stor      -40.5
## 4 Svært liten -1.64
## 5 Svært stor -9901.
```

```
kom_str_mod %>%
  filter(term == "alp_30_74") %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(mean_30_74 = mean(estimate))
```

'summarise()' ungrouping output (override with '.groups' argument)

```
## # A tibble: 5 x 2
##   k_str      mean_30_74
##   <chr>      <dbl>
## 1 Liten      -1.47
## 2 Middels   -2.70
## 3 Stor       6.15
## 4 Svært liten  4.75
## 5 Svært stor 11076.
```

```
kom_str_mod %>%
  filter(term == "alp_15_29") %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(n = n())
```

'summarise()' ungrouping output (override with '.groups' argument)

```
## # A tibble: 5 x 2
##   k_str      n
##   <chr>    <int>
## 1 Liten    64
## 2 Middels  75
## 3 Stor    23
## 4 Svært liten  58
## 5 Svært stor   7
```

```
kom_str_mod %>%
  filter(term == "alp_30_74") %>%
  filter(p.value...9 < 0.05) %>%
  group_by(k_str) %>%
  summarise(n = n())
```

'summarise()' ungrouping output (override with '.groups' argument)

```
## # A tibble: 5 x 2
##   k_str      n
##   <chr>    <int>
## 1 Liten    68
## 2 Middels  70
## 3 Stor    25
## 4 Svært liten  71
## 5 Svært stor   7
```

I litt over 50% av kommunene, hovedsaklig de små, ser modellen ut til virke. Kanskje noe å bygge videre på.

#siste