

# model.Rmd

Jonathan

12/30/2021

```
suppressPackageStartupMessages({  
  library(tidyverse)  
  library(lubridate)  
  library(modelr)  
  library(broom)  
  library(lmtest)  
  library(sandwich)  
  library(viridis)  
})
```

Henter PM2 filen:

```
pm2 <- read_csv("data/pm2.csv", show_col_types = FALSE)
```

## Modeller

### Leser inn data

```
# Starter med å lese unn dataene  
pm2 <- read_csv("data/pm2.csv", show_col_types = FALSE)
```

```
pm2 <- pm2 %>%  
  mutate(  
    aar_f = parse_factor(as.character(aar)),  
    aar_d = date(paste0(aar, "-01-01"))  
  )
```

## Modell

Funksjon:

```
mod1 <- 'pm2 ~ aar_f + Total_ya_p + rel_inc_l + rel_inc_h + uni_k_mf + uni_l_mf + nytt_bareal_pp'
```

i) Generer et lm objekt (lm1) utfra mod1 og datasettet pm2

Modell 1:

```
lm1 <- lm(mod1, data = pm2, subset = complete.cases(pm2))
```

ii) Legg residualene fra den lineære modellen til datasettet pm2

```
pm2 %>%  
  add_residuals(lm1)
```

```
## # A tibble: 2,364 x 21
```

```
##      knr      knavn      aar      pm2 Menn_ya_p Kvinner_ya_p Total_ya_p Total_t rel_inc_l
##      <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 0101 Halden 2006 12052 59.8 56.8 58.3 27722 5.7
## 2 0101 Halden 2007 12363 59.7 56.8 58.2 27835 6
## 3 0101 Halden 2008 13427 59.7 56.8 58.3 28092 5.1
## 4 0101 Halden 2009 13095 59.8 57.0 58.4 28389 5.4
## 5 0101 Halden 2010 13832 59.6 57.1 58.3 28776 4.2
## 6 0101 Halden 2011 14915 59.8 57.2 58.5 29220 4.3
## 7 0101 Halden 2012 15473 59.5 57.0 58.2 29543 3.3
## 8 0101 Halden 2013 15461 59.0 56.7 57.9 29880 3.6
## 9 0101 Halden 2014 17164 58.8 56.7 57.7 30132 2.5
## 10 0101 Halden 2015 17427 58.7 56.8 57.8 30328 2.2
## # ... with 2,354 more rows, and 12 more variables: rel_inc_h <dbl>,
## # uni_k_mf <dbl>, uni_k_m <dbl>, uni_k_f <dbl>, uni_l_mf <dbl>,
## # uni_l_m <dbl>, uni_l_f <dbl>, nytt_bareal <dbl>, nytt_bareal_pp <dbl>,
## # aar_f <fct>, aar_d <date>, resid <dbl>
```

Summary:

```
summary(lm1)
```

```
##
## Call:
## lm(formula = mod1, data = pm2, subset = complete.cases(pm2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9288.7 -1488.2   -58.8  1328.1 15273.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -24523.01    2055.68  -11.929 < 2e-16 ***
## aar_f2007      1351.20     249.16   5.423 6.46e-08 ***
## aar_f2008      1422.85     249.66   5.699 1.36e-08 ***
## aar_f2009      1422.70     249.58   5.700 1.35e-08 ***
## aar_f2010      2167.01     250.06   8.666 < 2e-16 ***
## aar_f2011      2929.87     251.30  11.659 < 2e-16 ***
## aar_f2012      3466.12     252.41  13.732 < 2e-16 ***
## aar_f2013      4041.73     254.10  15.906 < 2e-16 ***
## aar_f2014      4026.33     255.60  15.752 < 2e-16 ***
## aar_f2015      4697.04     257.85  18.216 < 2e-16 ***
## aar_f2016      5435.37     260.53  20.863 < 2e-16 ***
## aar_f2017      6417.06     262.34  24.461 < 2e-16 ***
## Total_ya_p      570.84      35.57  16.049 < 2e-16 ***
## rel_inc_l     -178.32      33.41  -5.337 1.04e-07 ***
## rel_inc_h      233.41      18.47  12.639 < 2e-16 ***
## uni_k_mf       -63.92      26.47  -2.415 0.0158 *
## uni_l_mf      1270.59      37.75  33.661 < 2e-16 ***
## nytt_bareal_pp   78.07     216.96   0.360 0.7190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2468 on 2340 degrees of freedom
## (6 observations deleted due to missingness)
## Multiple R-squared:  0.8432, Adjusted R-squared:  0.8421
```

```
## F-statistic: 740.2 on 17 and 2340 DF, p-value: < 2.2e-16
```

## Modell oppgaver

i) Forklar hva vi kan lese ut av verdien på års-koeffisientene.

Årskoeffisientene til modellen forklarer hvor mye  $Y$ , altså  $pm2$  øker i kvadratmeter for hvert år som går. Eksempelvis ser vi at det øker med 1351,2 i 2007, 4026,33 i 2014, helt til 6417,06 i året 2017.

ii) Diskuter om fortegnet er som forventet på de øvrige koeffisientene.

Modellen fremviser at alle fortegnene til koeffisientene er positive utenom de som tilhører  $rel\_inc\_l$  og  $uni\_k\_mf$ , som da innehar negative fortegn. Grunnlaget til hvorfor disse er negative vil være at  $rel\_inc\_l$  er som sagt differensen mellom variablene  $inc\_l$  (lav inntekstsgrense 0-249k) og  $inv\_hh\_l\_m$  (Gjennomsnitt), og  $uni\_k\_mf$  (lav universitetsutdannelse). Individene innenfor disse kategoriene vil komme dårligst ut av økning i pris per kvadratmeter.

### Testing for heteroskedasitet

i) Benytter en Breuch-Pagen test (bptest fra lmtest pakken) der  $H_0$  er at residualene er trukket fra en fordeling med konstant varians.

```
bptest(lm1)
```

```
##
## studentized Breusch-Pagan test
##
## data: lm1
## BP = 407.56, df = 17, p-value < 2.2e-16
```

ii) Har vi problemer med heteroskedastisitet her?

Ja, vi vil ha problemer med heteroskedastisitet med henhold til at p-verdien er større enn 0,05.

iii) I så fall bør vi rapportere robuste standard feil og tilhørende robuste t-verdier (Se `coeftest()` fra `lmtest` pakken. Vi trenger også `vcovHC()` fra `sandwich` pakken for å spesifisere kovariansmatrisen.)

```
# Hvordan man får robuste standard errors i R. HCO er standard i Stata
coeftest(lm1, vcov = vcovHC(lm1, type = "HC3"))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24523.011   2498.848 -9.8137 < 2.2e-16 ***
## aar_f2007    1351.203    203.819  6.6294 4.170e-11 ***
## aar_f2008    1422.850    207.909  6.8436 9.828e-12 ***
## aar_f2009    1422.705    204.908  6.9431 4.949e-12 ***
## aar_f2010    2167.007    204.156 10.6145 < 2.2e-16 ***
## aar_f2011    2929.866    214.854 13.6365 < 2.2e-16 ***
## aar_f2012    3466.120    219.338 15.8026 < 2.2e-16 ***
## aar_f2013    4041.728    225.339 17.9362 < 2.2e-16 ***
## aar_f2014    4026.327    231.187 17.4159 < 2.2e-16 ***
## aar_f2015    4697.035    254.298 18.4706 < 2.2e-16 ***
## aar_f2016    5435.367    278.941 19.4857 < 2.2e-16 ***
## aar_f2017    6417.063    306.831 20.9140 < 2.2e-16 ***
## Total_ya_p    570.843     42.443 13.4496 < 2.2e-16 ***
## rel_inc_l    -178.323     30.979 -5.7562 9.732e-09 ***
```

```
## rel_inc_h      233.411      18.598 12.5505 < 2.2e-16 ***
## uni_k_mf       -63.918      36.640 -1.7445  0.08121 .
## uni_l_mf      1270.590      69.344 18.3230 < 2.2e-16 ***
## nytt_bareal_pp  78.075     206.400  0.3783  0.70527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

iv) Legg residualene fra lm1 til datasettet pm2

```
# Nå har vi complete.cases i pm2
pm2 <- pm2 %>%
  add_residuals(lm1)
```

v) Bruk variabelen aar til å lage en nye variabel aar\_d av typen date. Bruk datoen 1. jan..

```
# Unødvendig, du har laget aar_d lengre oppe
# pm2 <- pm2 %>%
#   mutate(aar_d = make_date(aar))
```

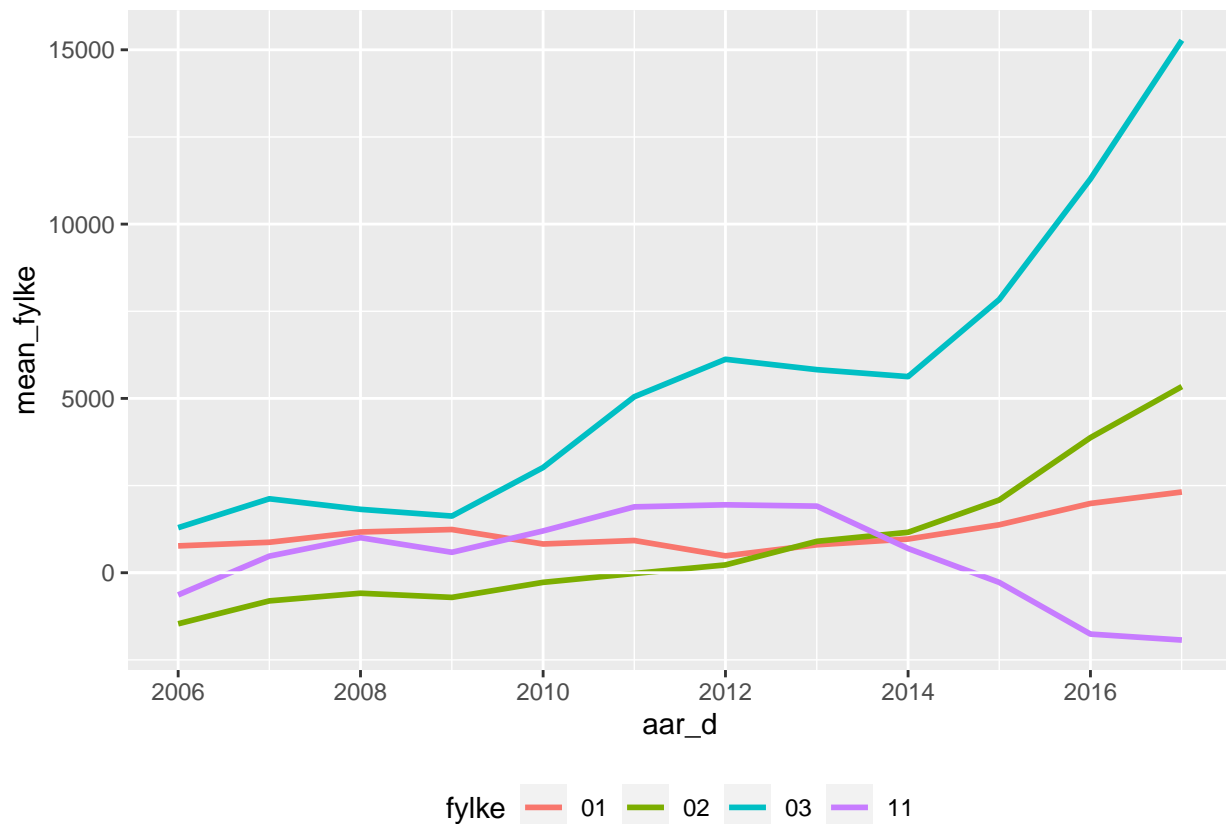
vi) Filtrer ut fylkene Østfold, Akershus, Oslo, Rogaland og Hordaland.

```
pm2 <- pm2 %>%
  mutate(fylke = substr(knr, start = 1, stop = 2))
```

vii) - x

```
# Fin figur ;-)
pm2 %>%
  filter(fylke %in% c("01", "02", "03","11")) %>%
  unnest(c(fylke)) %>%
  group_by(fylke, aar_d) %>%
  summarise(mean_fylke = mean(resid)) %>%
  ggplot(mapping = aes(x= aar_d, y= mean_fylke, colour = fylke)) +
  geom_line(lwd=1) +
  geom_hline(yintercept = 0, colour = "white") +
  theme(legend.position = "bottom")
```

## 'summarise()' has grouped output by 'fylke'. You can override using the '.groups' argument.



#

## Dummy fylke og år

### Del 1

i) Innfører en faktor-variabel for fylke i modellen. La modell 2 ellers være lik modell 1.

```
pm2 <- pm2 %>%
  mutate(
    fnr = parse_factor(as.character(fylke))
  )
```

Funksjon:

```
mod2 <- 'pm2 ~ aar_f + fnr + Total_ya_p + rel_inc_l + rel_inc_h + uni_k_mf + uni_l_mf + nytt_bareal_pp'
```

ii. Generer lm2 fra modell 2 og datasettet pm2.

Modell 2:

```
# du må bruke subset complete.cases også her for å droppe
# observasjonene som ikke er komplette cases. Tror kanskje det greieste er å bare droppe de seks
# fra pm2. complete.cases() gir true hvis vi ikke har noen NA. Får altså en vektor med TRUE og FALSE
# på rekkeplassen i pm2[complete.cases(pm2), ]. Uttrykket tar bare med de rekkene der vi har TRUE
# dropper rekker med manglende observasjoner. Nå har vi 2358 obs.
pm2 <- pm2[complete.cases(pm2), ]
lm2 <- lm(mod2, data = pm2)
```

iii. Legg residualene fra lm2 til pm2 og kall dem res\_m2

```
pm2 <- pm2 %>%
  mutate(res_m2 = resid(lm2))
```

```
lm2
```

```
##
## Call:
## lm(formula = mod2, data = pm2)
##
## Coefficients:
##      (Intercept)      aar_f2007      aar_f2008      aar_f2009      aar_f2010
##      -17085.60       1371.32       1433.26       1428.08       2139.00
##      aar_f2011      aar_f2012      aar_f2013      aar_f2014      aar_f2015
##       2883.77       3410.59       3954.71       4017.67       4707.98
##      aar_f2016      aar_f2017      fnr02       fnr03       fnr04
##       5431.32       6385.50       427.35       7059.82      -1891.00
##       fnr05       fnr06       fnr07       fnr08       fnr09
##      -2330.51      -1472.82       114.55      -2825.09      -1483.12
##       fnr10       fnr11       fnr12       fnr14       fnr15
##      -824.84      -446.00      -737.67      -446.88      -3742.47
##       fnr16       fnr17       fnr18       fnr19       fnr20
##      -1586.64      -2426.22      -1556.68      -2791.75      -1997.03
##      Total_ya_p      rel_inc_l      rel_inc_h      uni_k_mf      uni_l_mf
##       430.59       -49.82       257.40       135.12       926.99
## nytt_bareal_pp
##       -173.81
```

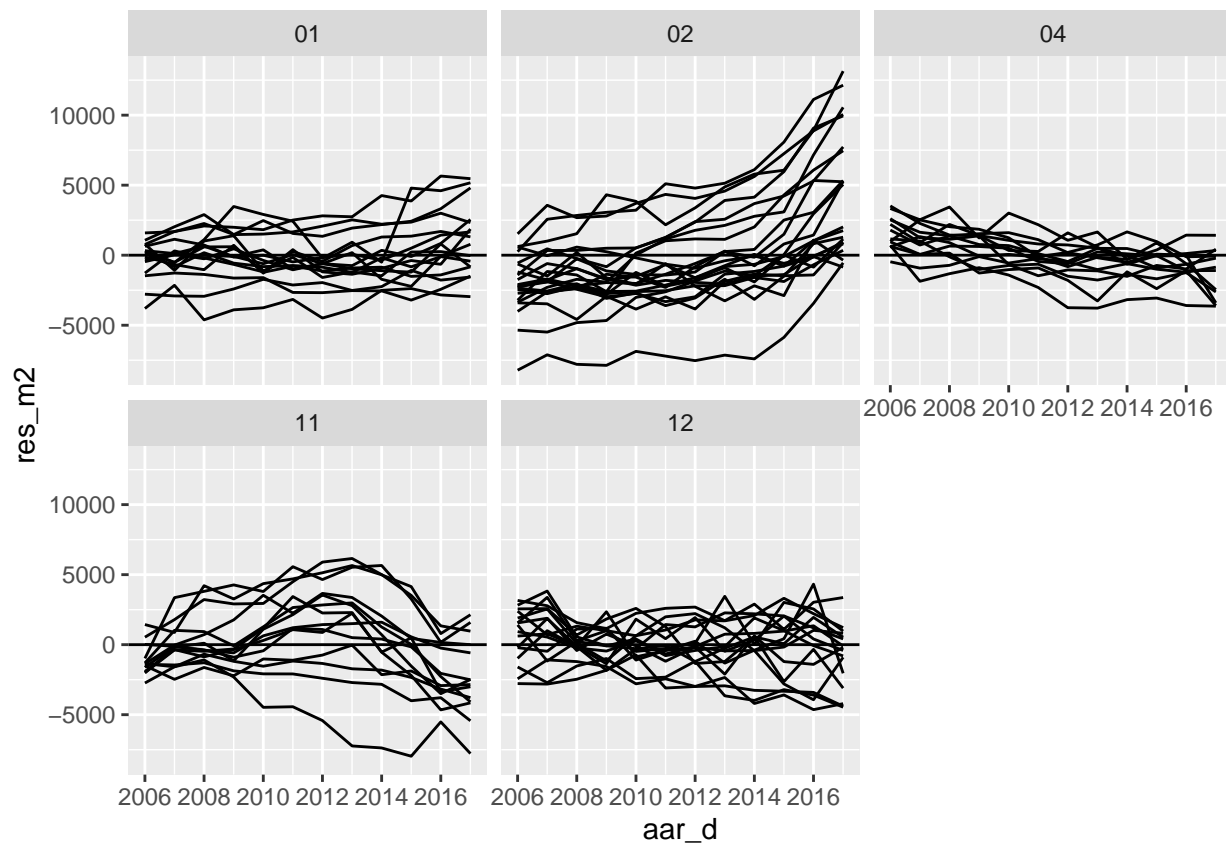
iv) Rapportert summary() for mod2

```
summary(mod2)
```

```
##      Length      Class      Mode
##           1 character character
```

v) Filtrer ut fylkene “01”, “02”, “04”, “11” og “12” fra pm2 og plot residualene fra lm2 for hver kommune som linjer. Lag et del-plot (facet) for hvert fylke . La aar\_d være horisontal akse. Dette skal gi en figur som den følgende.

```
pm2 %>% filter(fnr %in% c("01", "02", "04", "11", "12")) %>%
  ggplot(mapping = aes(x = aar_d, y = res_m2)) +
  geom_line(aes(group = knavn)) +
  scale_size_manual(values = c(seq(2.0, 0.5, by = -0.1))) +
  geom_hline(yintercept = 0) +
  theme(legend.position = 'bottom') +
  facet_wrap(~fylke)
```



## Del 2

### i) Diskuter hva dette sier om kvaliteten på modell 2.

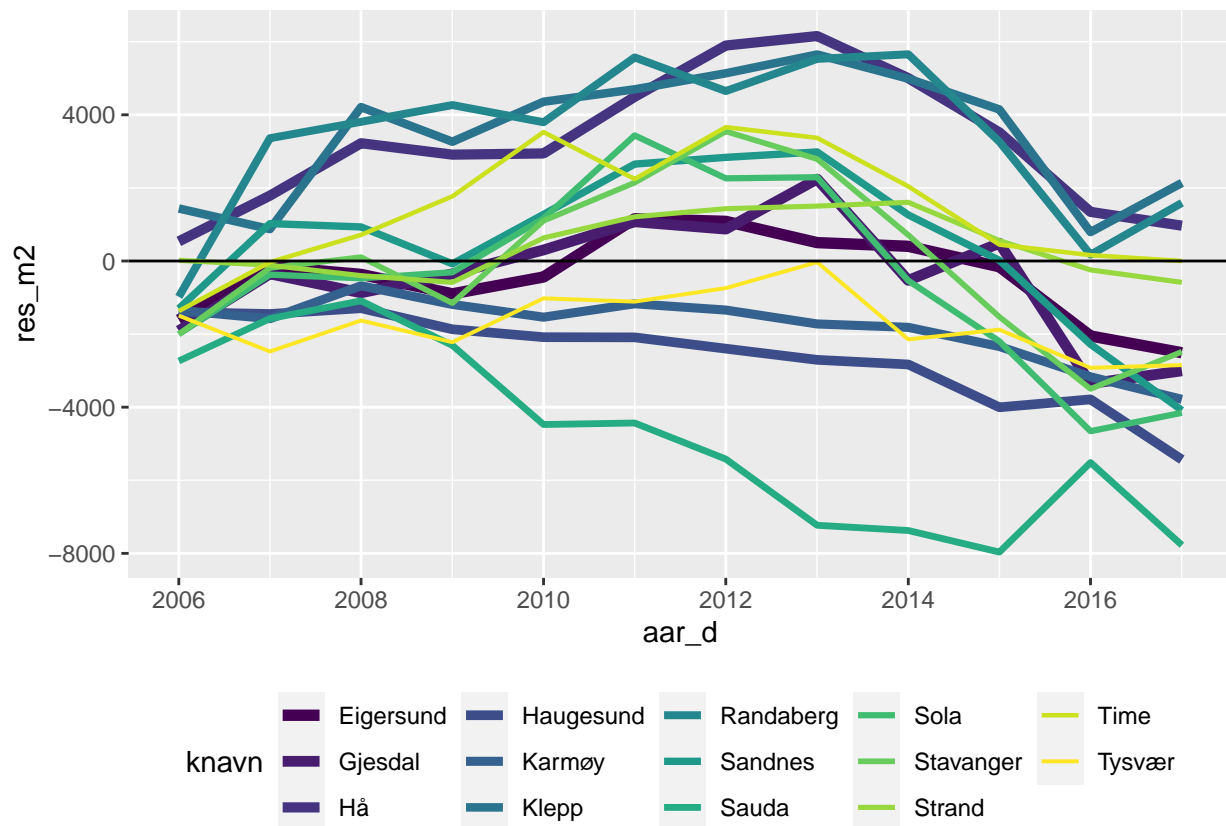
Med henhold til at modellene fremviser at noen kommuner ligger over 0 linjen og andre under, kan vi konkludere at modell 2 har svært høye variasjoner som ikke er gunstig.

### ii) Er det grunn til å mistenke at vi mangler viktige variabler i modell 2?

Variasjonene kan være preget av at vi mangler viktige variabler som ikke er inkludert i modellen.

### iii) Filtrer så med hensyn på fylke "11".

```
pm2 %>% filter(fnr %in% c("11")) %>%
  ggplot(mapping = aes(x = aar_d, y = res_m2)) +
  scale_color_viridis(discrete = TRUE, option = "D") +
  geom_line(aes(group = knavn, colour = knavn, size = knavn)) +
  scale_size_manual(values = c(seq(2.0, 0.5, by = -0.1))) +
  geom_hline(yintercept = 0) +
  theme(legend.position = 'bottom')
```

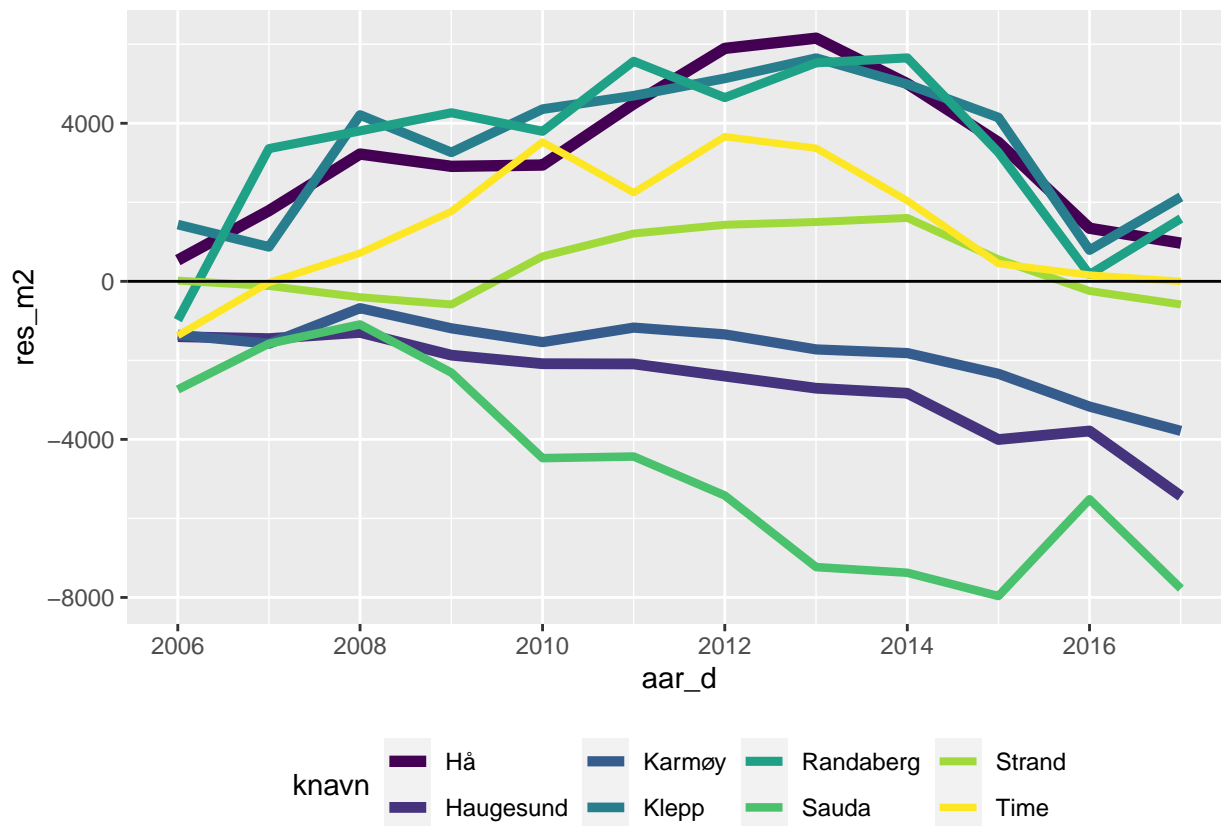


### Del 3

i) Gjenta plottet ovenfor men nå bare for kommunene “1119”, “1120”, “1127”, “1121”, “1130”, “1135”, “1106”, “1149”

```
pm2 %>% filter(knr %in% c("1119", "1120", "1127", "1121", "1130", "1135", "1106", "1149")) %>%
  ggplot(mapping = aes(x = aar_d, y = res_m2)) +
  scale_color_viridis(discrete = TRUE, option = "D") +
  geom_line(aes(group = knavn, colour = knavn, size = knavn)) +
  scale_size_manual(values = c(seq(2.0, 0.5, by = -0.1))) +
  geom_hline(yintercept = 0) +
  theme(legend.position = 'bottom')
```





ii) Hva kjennetegner de kommune i Rogaland hvor vår enkle modell hhv. overvurderer og undervurderer pris per kvadratmeter?

Kommunene i nærheten Haugesund undervurderes og de i nærheten av Stavanger overvurderes.

OBS! Motsatt blir det! Husk at vi her betrakter residualene. Systematisk positive residualer betyr at virkelig verdi er høyere enn den modellen predikerer (kommunene rundt Stavanger), mens systematisk negative residualer betyr at modellen predikerer høyere verdier enn de virkelige. Vår enkle modell undervurderer altså prisene rundt stavanger, mens den overvurderer prisene rundt Haugesund. Tyder på at modellen mangler en viktig variabel, og da vet vi fra økonometrien at våre koeffisient estimater for de eksisterende variablene antakelig er biased.

## Modell for hvert fylke

```
pm2_k <- pm2 %>%
  group_by(knr, knavn, fnr) %>%
  nest()

dim(pm2_k)

## [1] 197 4

pm2_k %>%
  head()

## # A tibble: 6 x 4
## # Groups:   knr, knavn, fnr [6]
##   knr   knavn   fnr data
##   <chr> <chr>   <fct> <list>
```

```
## 1 0101 Halden      01    <tibble [12 x 21]>
## 2 0104 Moss       01    <tibble [12 x 21]>
## 3 0105 Sarpsborg  01    <tibble [12 x 21]>
## 4 0106 Fredrikstad 01    <tibble [12 x 21]>
## 5 0111 Hvaler     01    <tibble [12 x 21]>
## 6 0119 Marker     01    <tibble [12 x 21]>
```

i) Skriv en funksjon `k_model` for å kjøre følgende modell for hvert enkelt fylke

```
k_model <- function(a_df) {
  lm(pm2 ~ aar + Total_ya_p + rel_inc_l + rel_inc_h + uni_k_mf + uni_l_mf + nytt_bareal_pp,
    data = a_df)
}
```

ii) Bruk funksjonen på data elementet i `pm2_k` og generer et nytt model element som inneholder en lineær modell for hver kommune.

```
pm2_k <- pm2_k %>%
  mutate(model = map(data, .f = k_model))
```

iii) Dropp kommunene “0811” og “0821” som ikke har komplette data.

Skal bare droppe kommunene Bø og Siljan som ikke hadde komplette da. Bare bruke en pipe på `pm2` og ta med alle kommuner som IKKE er “0811” eller “0821”. Husk at ! betyr ikke og `%in%` er svært hensiktsmessig.

```
# Lager et redusert datasett der vi har tatt ut kommuner
# som ikke har obs. for alle årene
pm2_k <- pm2_k %>%
  filter(!knr %in% c("0811", "0821"))
```

iv) Legg til residualene som en variabel i `pm2_k` (jmf. slide 24 i kap.25).

```
# Her må du bruke map2() siden du trenger to argumenter nemlig modellen og funksjonen (add_residual())
# som du skal bruke på modellen
```

```
pm2_k <- pm2_k %>%
  mutate(
    resid = map2(.x = data, .y = model, .f = add_residuals)
  )
```

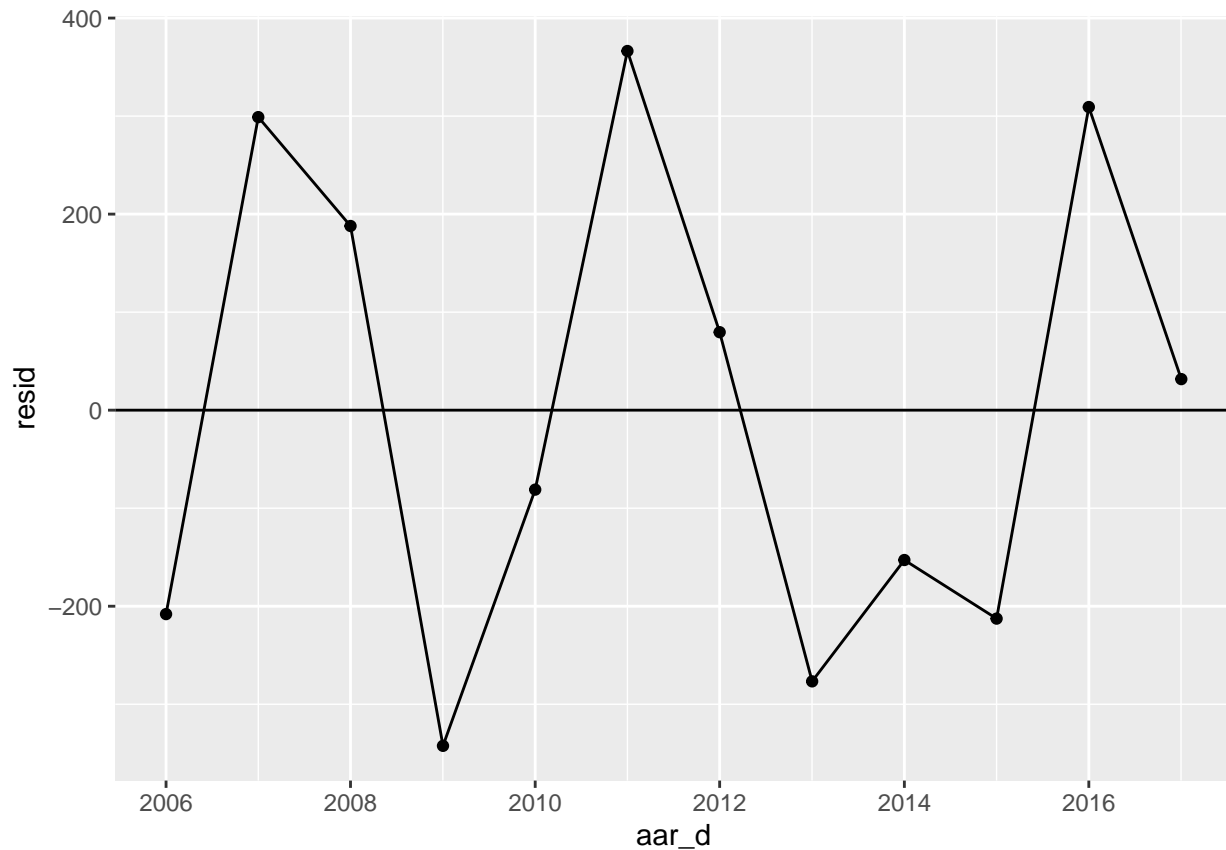
v) List ut residualene for Haugesund.

```
# Filtrer ut Haugesund, før du så unnest() residualene og velger variablene aar og resid
```

```
Hauge <- pm2_k %>%
  filter(knavn == "Haugesund") %>%
  unnest(resids)
```

vi) Lag et plot av residualene for Haugesund (jmf. slide 28 kap. 25).

```
Hauge %>%
  filter(knavn == "Haugesund") %>%
  ggplot(mapping = aes(x = aar_d, y = resid)) +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 0)
```



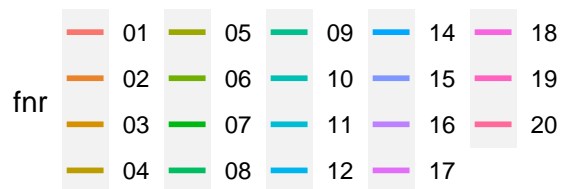
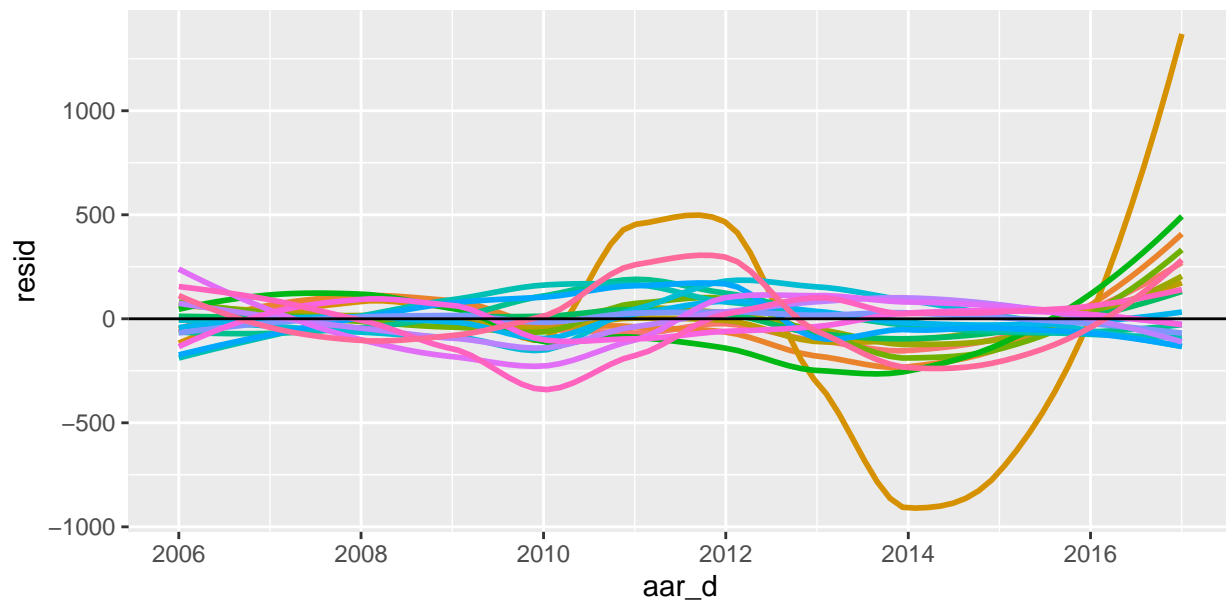
Dataene fra ovenfor kan du bare pipe inn i ggplot, legge på en `geom_point()` og `geom_line()` før du til slutt legger på `geom_hline(yintercept = 0)`

**vii. Bruk `geom_smooth()` til å vise utviklingen i gjennomsnittlige residualer for hvert fylke.**

Bruk `geom_smooth(aes(colour = fnr, group = fnr), se = FALSE)` for å få en farge for hvert fylke

```
# Du trenger bare disse residualene i disse plotene, så du
# kan like godt bare pakke dem ut i pipen og så sende videre
# til ggplot. Så kan du bruke pm2_k også for neste plot
pm2_k %>%
  unnest(resids) %>%
  ggplot(mapping = aes(x = aar_d, y = resid)) +
  geom_smooth(aes(colour = fnr, group = fnr), se = FALSE) +
  geom_hline(yintercept = 0) +
  theme(legend.position = "bottom")
```

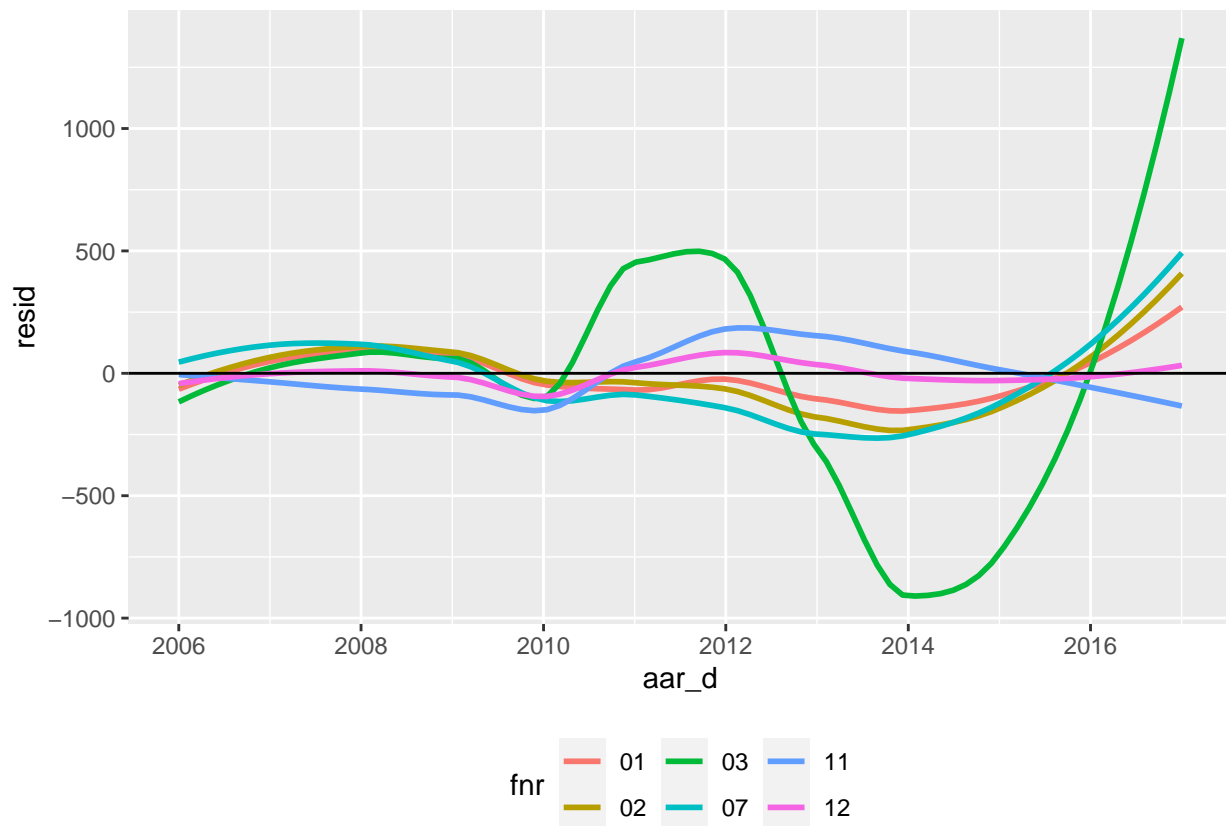
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



viii. La samme plot, men nå bare for fylkene 01, 02, 03, 07, 11 og 12.

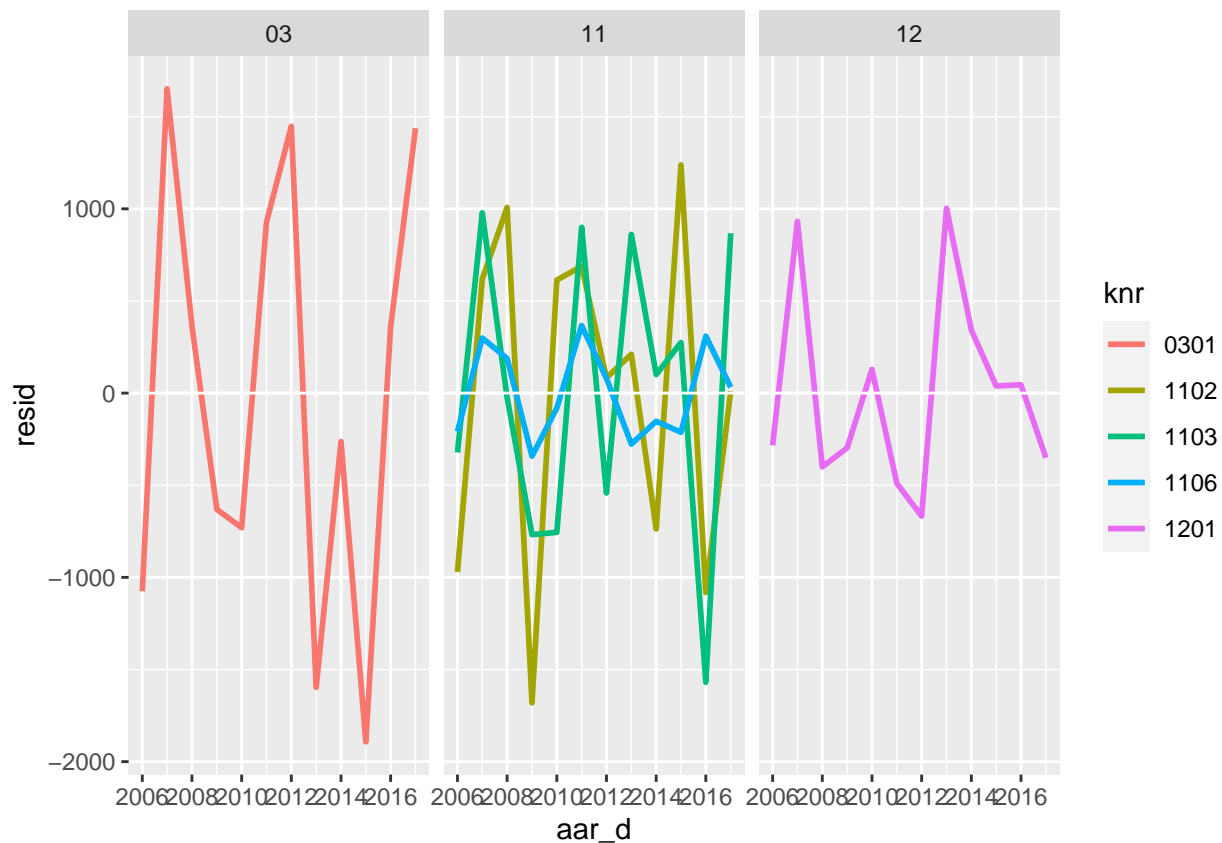
```
pm2_k %>% filter(fnr %in% c("01", "02", "03", "07", "11", "12")) %>%
  # unnest pm2_k igjen for å få tilgang til variablene
  unnest(resids) %>%
  ggplot(mapping = aes(x = aar_d, y = resid)) +
  geom_smooth(aes(colour = fnr, group = fnr), se = FALSE) +
  geom_hline(yintercept = 0) +
  theme(legend.position = "bottom")
```

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



ix) Plukk ut kommunene “0301”, “1102”, “1103”, “1106” og “1201” vis utviklingen i residualer vha. `geom_line()`. Lag facet-plot med `fnr` som variabel.

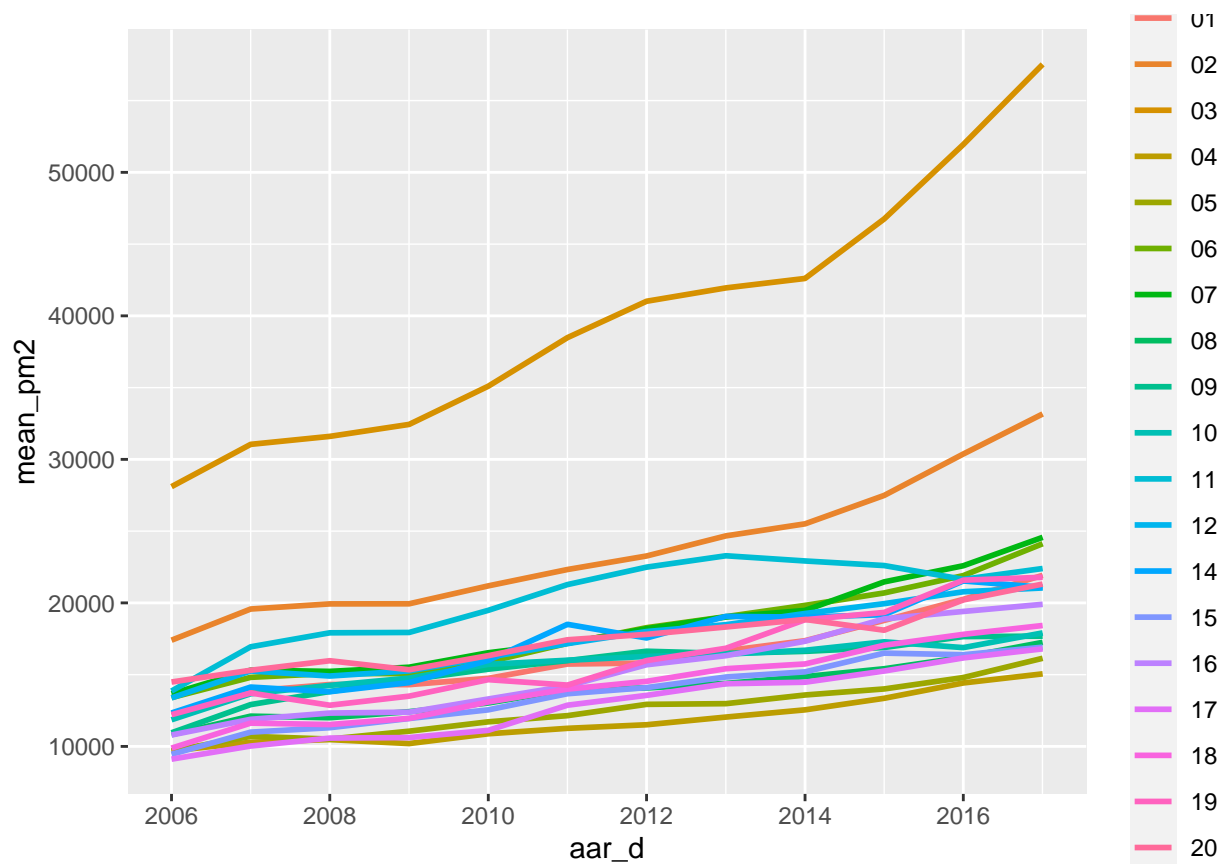
```
pm2_k %>% filter(knr %in% c("0301", "1102", "1103", "1106", "1201")) %>%
  # unnest pm2_k igjen for å få tilgang til variablene
  unnest(resids) %>%
  ggplot(mapping = aes(x = aar_d, y = resid)) +
  # Her kommuner så vi må gruppere mht. knr. Setter
  # også colour så får vi en farge for hver kommune
  # Tar vekk alpha så blir det lettere å se fargene
  geom_line(aes(group = knr, colour = knr), lwd = 1) +
  geom_hline(yintercept = 0, colour = "white") +
  # Tar facet mht. fnr , så samler vi kommuner fra samme fylke
  facet_wrap(~fnr)
```



x) Plot av utviklingen i gjennomsnittlig pm2 for hvert fylke (jmf.slide 34 kap. 25).

```
# Må sende pm2_k (ikke pm2) inn i pipen siden det er den
# som har data som list-column
pm2_k %>%
  unnest(c(data)) %>%
  # kan ikke gruppere mht. datasettet (pm2). Grupperer mht. fnr (fylke)
  group_by(fnr, aar_d) %>%
  summarise(mean_pm2 = mean(pm2)) %>%
  ggplot(aes(x = aar_d, y = mean_pm2, colour = fnr)) +
  geom_line(lwd = 1)
```

## 'summarise()' has grouped output by 'fnr'. You can override using the '.groups' argument.

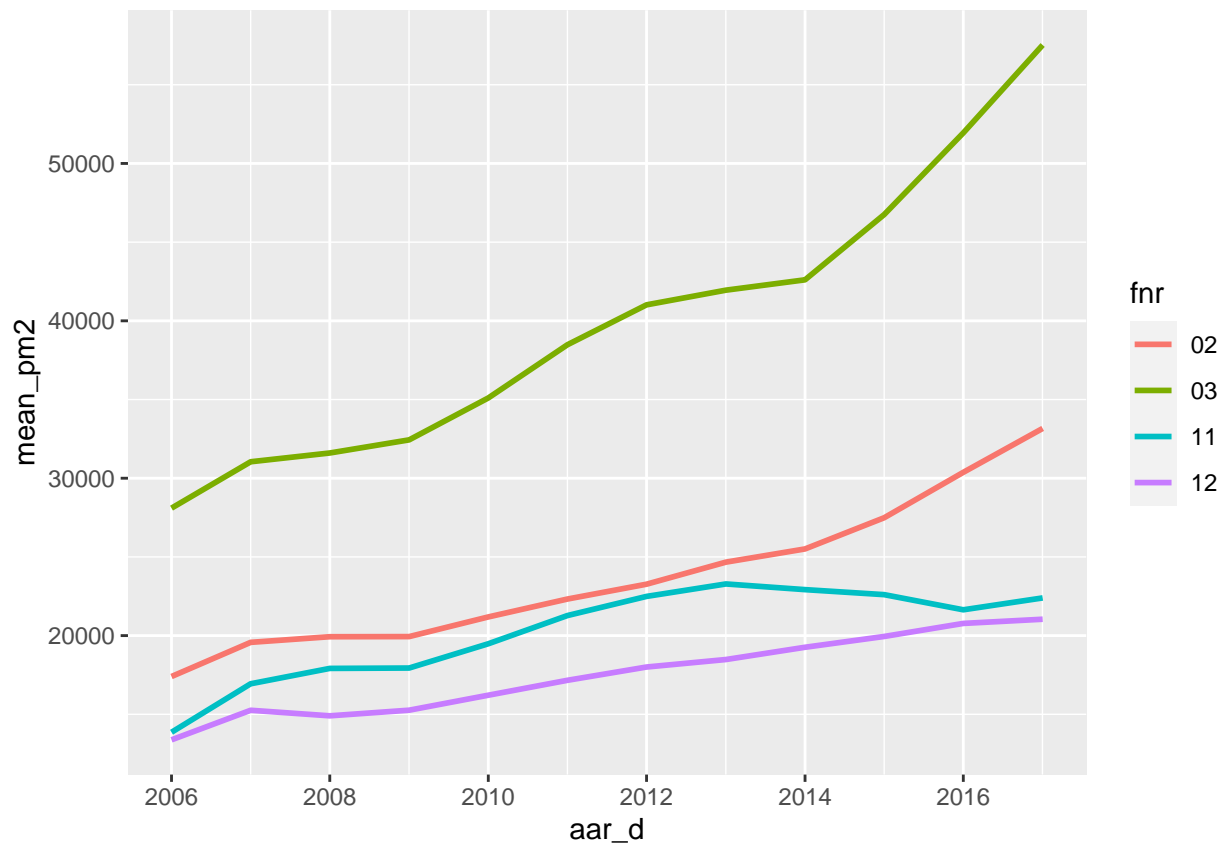


xi. Gjør det samme, men nå bare for fylkene “fnr\_02”, “fnr\_03”, “fnr\_11” og “fnr\_12”.

Bruk koden ovenfor, men kjør et filter på de ønskete fylkene i starten

```
pm2_k %>% filter(fnr %in% c("02", "03", "11", "12")) %>%
  unnest(c(data)) %>%
  # kan ikke gruppere mht. datasettet (pm2). Grupperer mht. fnr (fylke)
  group_by(fnr, aar_d) %>%
  summarise(mean_pm2 = mean(pm2)) %>%
  ggplot(aes(x = aar_d, y = mean_pm2, colour = fnr)) +
  geom_line(lwd = 1)
```

## ‘summarise()’ has grouped output by ‘fnr’. You can override using the ‘.groups’ argument.



xii) Bruk funksjonen glance fra broom pakken til å lage en mod\_summary variabel og unnest() denne. Legg resultatet i mod\_sum (jmf. slide 43 kap. 25).

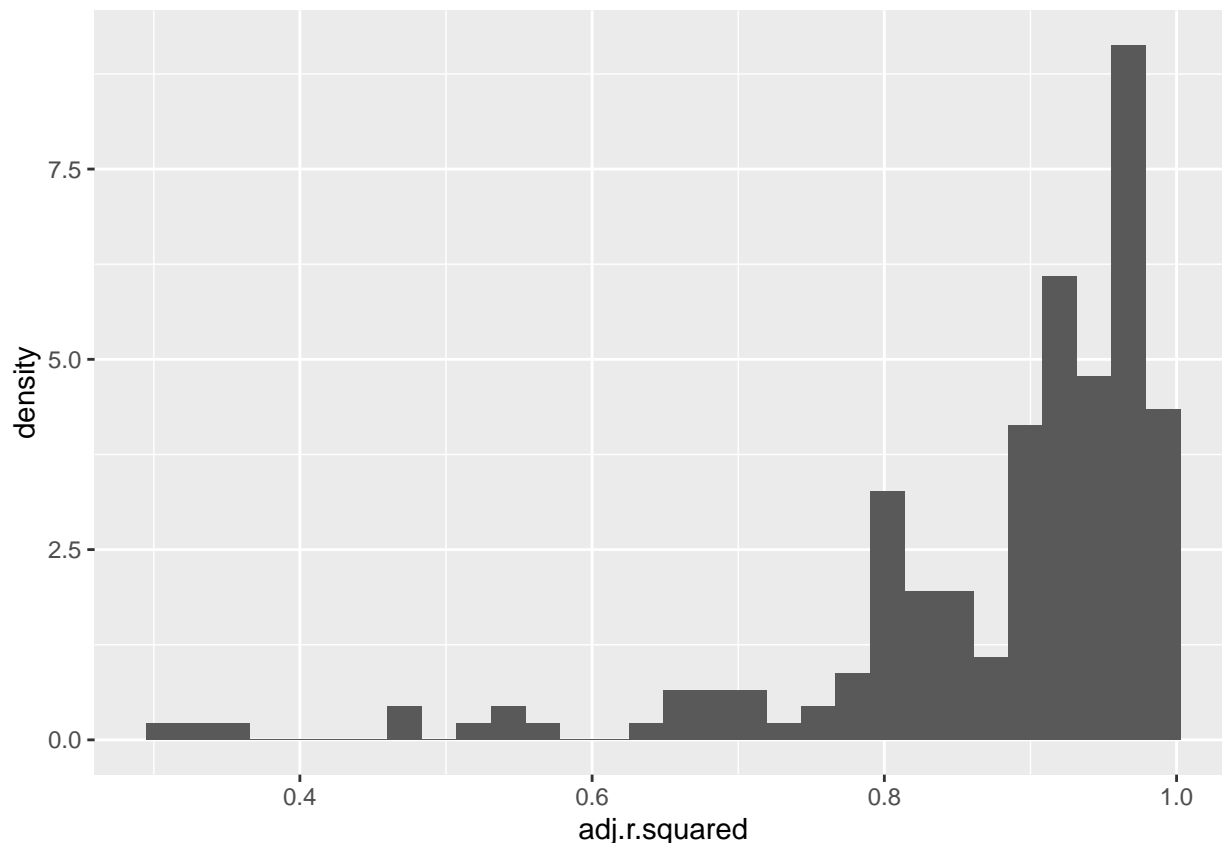
```
mod_sum <- pm2_k %>%
  mutate(mod_summary = map (.x = model, .f = glance)) %>%
  unnest(mod_summary)
```

xiii) Bruk mod\_sum til å lage et histogram av adj.r.squared for modellene våre.

```
mod_sum %>%
  ggplot(aes(x = adj.r.squared, y = ..density..)) +
  geom_histogram()
```

## 'stat\_bin()' using 'bins = 30'. Pick better value with 'binwidth'.





**xiv) Finn de 10 kommunene der vi har lavest adj.r.squared.**

Start pipe med `mod_sum`. Bruk `arrange()` til å sortere etter `adj.r.squared` Velg de variablene du trenger og avslutt med `head(10)`

```
mod_sum %>%
  arrange(adj.r.squared) %>%
  select(knr, knavn, fnr, adj.r.squared) %>%
  head(10)
```

```
## # A tibble: 10 x 4
## # Groups:   knr, knavn, fnr [10]
##   knr   knavn      fnr adj.r.squared
##   <chr> <chr>    <fct>      <dbl>
## 1 1135 Sauda      11      0.314
## 2 0428 Trysil      04      0.327
## 3 0426 Våler (Hedmark) 04      0.345
## 4 0901 Risør       09      0.461
## 5 1238 Kvam        12      0.475
## 6 1517 Hareid       15      0.522
## 7 0520 Ringebu      05      0.541
## 8 0423 Grue         04      0.545
## 9 1535 Vestnes      15      0.562
## 10 1251 Vaksdal     12      0.629
```

**xv) Finn de 10 kommunene der vi har høyest adj.r.squared.**

Tilsvarende, men de 10 høyeste først (altså sortert i avtakende rekkefølge)

```
mod_sum %>%
  arrange(desc(adj.r.squared)) %>%
  select(knr, knavn, fnr, adj.r.squared) %>%
  head(10)
```

```
## # A tibble: 10 x 4
## # Groups:   knr, knavn, fnr [10]
##   knr   knavn   fnr   adj.r.squared
##   <chr> <chr>   <fct>         <dbl>
## 1 0235 Ullensaker 02           0.998
## 2 1149 Karmøy    11           0.997
## 3 1253 Osterøy 12           0.994
## 4 0220 Asker    02           0.993
## 5 0229 Enebakk   02           0.992
## 6 1719 Levanger 17           0.992
## 7 0427 Elverum   04           0.991
## 8 0619 Ål       06           0.991
## 9 1519 Volda    15           0.990
## 10 0216 Nesodden 02           0.988
```