Assignment 4

```
suppressPackageStartupMessages({
library(tidyverse)
library(lubridate)
library(modelr)
library(broom)
library(lmtest)
library(sandwich)
library(viridis)
})
# Chunk opsjonene satt nedenfor er for mitt bruk
# Gjør at jeg kan holde noen av kortene skult ;-)
knitr::opts_chunk$set(echo=FALSE, include = FALSE)
```

Modeller

Leser inn data

Lag en ny fylke faktorvariabel. Husk at `parse_factor()` har en del fordeler sammenlignet med klassik-Rs `factor()`. Fylkesnummeret er de to første siffrene i kommunenummeret. Sjekk kapittel 14 i r4ds for å finne funksjon som henter ut deler av en tekststreng. Lag også en faktorvariabel `aar_f` fra årsvariabelen. I tillegg bør variabelen `Trade_pc` skaleres til handel målt i hundretusen NOK. Kall denne `Trade_pc_100K`.

Gjør du det riktig bør dette se slik ut.

Table 1: De 4 første rekkene

knr	fnr	aar_f	Trade_pc_100K
0101	01	2008	0.56266
0101	01	2009	0.56366
0101	01	2010	0.57210
0101	01	2011	0.58010

Modell

La oss starte med følgende modell

```
mod1 <- 'pm2 ~ aar_f + Total_ya_p + inc_k1 + inc_k5 + uni_k_mf + uni_l_mf + Trade_pc_100K'
```

og:

i. generer et `lm` objekt (lm1) utfra `mod1` og datasettet `pm2`.
ii. Legg residualene fra den lineære modellen til datasettet `pm2`.

Har du fulgt meg vil `summary` fra modellen være:

i. Forklar hva vi kan lese ut av verdien på års-koeffisientene.
ii. Diskuter om fortegnet er som forventet på de øvrige koeffisientene.

Vi bør teste for heteroskedastisitet.

i. Benytter en Breuch-Pagen test (`bptest` fra `lmtest` pakken) der H0 er at residualene er trukket fra en fordeling med konstant varians.

ii. Har vi problemer med heteroskedastisitet her?

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.)

iv. Legg residualene fra `lm1` til datasettet `pm2`.

v. Bruk variabelen `aar` til å lage en nye variabel `aar_d` av typen `date`. Bruk datoen 1. jan..

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

vii. Regn ut gjennomsnittlig residual per fylke per år og plot disse som linjer. En linje for hvert fylke og år (vha. `aar_d` generert ovenfor) på x-aksen.

viii. Bruk farge på linjene til å angi fylke.

ix. Plasser `legend` under selve plottet (se `theme` og `legend.position`).

x. Legg inn en horisontal linje for y = 0. (se `geom_hline`).

Dummy fylke og år

i. Innfører en dummy for hvert fylke hvert år. (Husk `*` gir interaksjonsvariabler automatisk i Rs formula). Bruk interaksjon mellom `fnr` og `aar_f` istedenfor `aar_f`. La modell 2 ellers være lik modell 1.

ii. Generer `lm2` fra modell 2 og datasettet `pm2`.

iii. Legg residualene fra `lm2` til `pm2` og kall dem `res_m2`

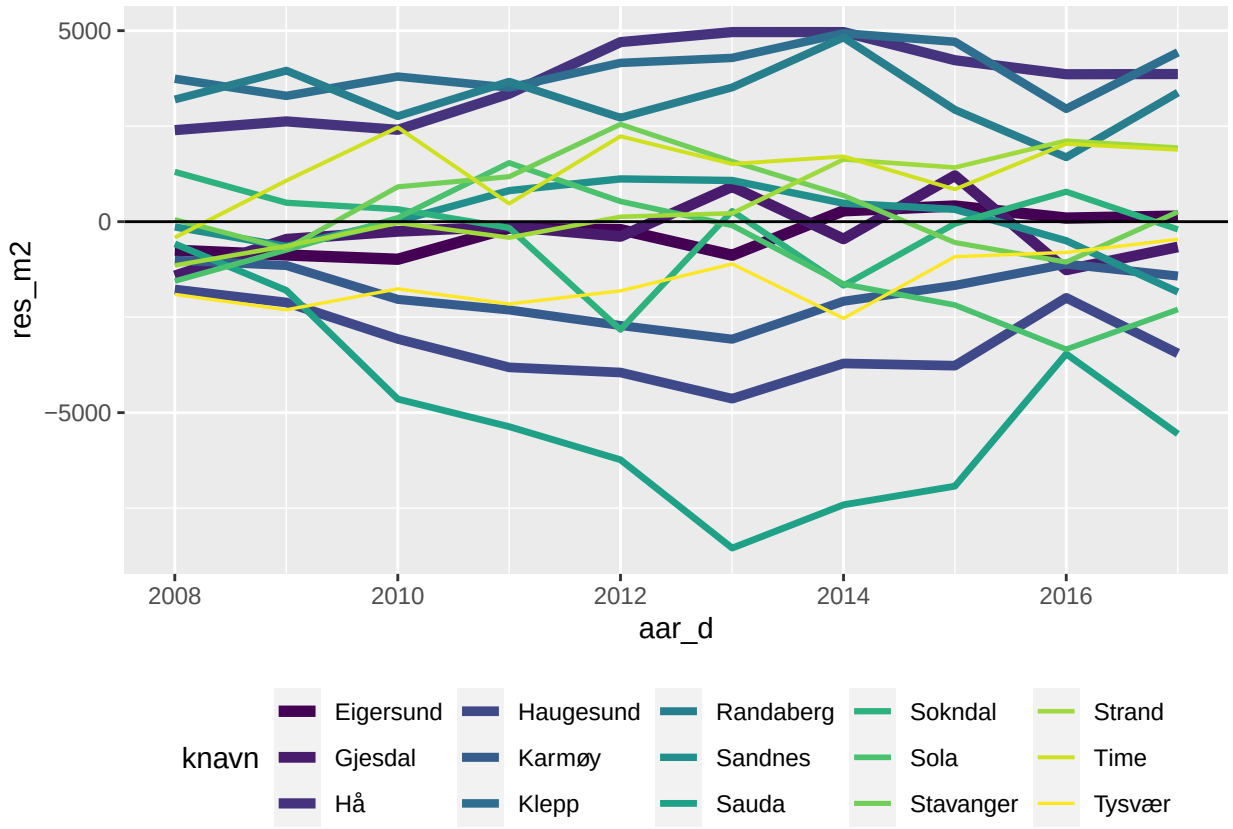
iv. Filtrer ut fylkene "01", "02", "04", "11" og "12" fra pm2 og plot residualene fra `lm2` for hver kommune som linjer. Lag et del-plot (facet) for hvert fylke . La `aar_d` være horisontal akse

Fra figuren vil du se at linjen for noen kommuner ligger over 0-linjen hele perioden, andre ligger under 0-linjen og noen krysser den.

i. Diskuter hva dette sier om kvaliteten på modell 2.
ii. Er det grunn til å mistenke at vi mangler viktige variabler i modell 2?
iii. Filtrer så med hensyn på fylke "11".

Du skal få en figur som ser slik ut.

```
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')
```



i. Gjenta plottet ovenfor men nå bare for kommunene "1119", "1120", "1127", "1121"," 1130", "1135", "1106", "1149"

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

Modell for hvert år

For å se hvor stabile estimatene er over tid skal vi også kjøre modell 2 for hvert enkelt år i perioden 2008 til 2017. For å redusere minnebruk og også gjøre kodingen litt lettere reduserer vi datasettet til de variablene vi bruker i modellen.

Vi ønske å ha ett datasett for hvert år i pm2_n så vi vil benyttes oss av «list-columns», dvs. en variabel som inneholder hele dataframes/tibbles som verdier.

i. Lag en list-column data i `pm2_n` som inneholder et datasett for hvert av årene 2008 til 2017.

Toppen av første element `pm2_n$data` bør se slik ut:

```
pm2_n$data[[1]] %>%
  head(n = 5)
```

```
## # A tibble: 5 x 13
##     pm2 fnr   knr     aar aar_f Menn_ya_p Kvinner_ya_p Total_ya_p inc_k1 inc_k5
##   <dbl> <chr> <chr> <dbl> <fct>     <dbl>        <dbl>      <dbl>  <dbl>  <dbl>
## 1 13427 01    0101   2008 2008       59.7         56.8       58.3   24.5   13.6
## 2 18299 01    0104   2008 2008       60.7         58.7       59.7   22.8   16.2
## 3 14981 01    0105   2008 2008       60.9         58.1       59.5   22.2   13.6
## 4 15671 01    0106   2008 2008       59.8         57.8       58.8   21.8   16.2
## 5 18844 01    0111   2008 2008       61.7         61.3       61.5   17.8   19
```

```
## # ... with 3 more variables: uni_k_mf <dbl>, uni_l_mf <dbl>,
## #   Trade_pc_100K <dbl>
```

```
dim(pm2_n)
```

```
## [1] 10  2
```

i. Skriv en funksjon `kom_model` for å kjøre følgende modell for hvert enkelt år:

```
pm2 ~ fnr + Total_ya_p + inc_k1 + inc_k5 + uni_k_mf + uni_l_mf + Trade_pc_100K
```

i. Utfør `kom_model` på hvert element i `pm2_n`.

Modellen for år 2008 bør se slik ut:

```
##
## Call:
## lm(formula = pm2 ~ fnr + Total_ya_p + inc_k1 + inc_k5 + uni_k_mf +
##     uni_l_mf + Trade_pc_100K, data = a_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4643.7 -1014.1   -62.3  1049.1  4422.7
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -21323.12    6210.25  -3.434 0.000732 ***
## fnr02            270.94     646.91   0.419 0.675827
## fnr03           4881.16    1955.07   2.497 0.013392 *
## fnr04          -1918.28     648.11  -2.960 0.003472 **
## fnr05          -2448.43     624.11  -3.923 0.000122 ***
## fnr06          -1689.23     636.36  -2.655 0.008619 **
## fnr07           -386.22     887.87  -0.435 0.664063
## fnr08          -3418.79     721.55  -4.738 4.23e-06 ***
## fnr09          -1056.76     756.64  -1.397 0.164159
## fnr10           -259.64     720.32  -0.360 0.718918
## fnr11            495.00     715.93   0.691 0.490161
## fnr12           -348.05     662.35  -0.525 0.599862
## fnr14          -2658.06     996.48  -2.667 0.008306 **
## fnr15          -3331.71     653.36  -5.099 8.25e-07 ***
## fnr16          -1283.11     634.47  -2.022 0.044550 *
## fnr17          -2437.25     782.79  -3.114 0.002136 **
## fnr18          -2049.05     660.42  -3.103 0.002212 **
## fnr19          -2995.65    1083.85  -2.764 0.006277 **
## fnr20          -2254.93     977.89  -2.306 0.022200 *
## Total_ya_p       464.29      90.03   5.157 6.31e-07 ***
## inc_k1           -50.14      71.27  -0.703 0.482632
## inc_k5           233.05      57.31   4.066 7.00e-05 ***
## uni_k_mf         181.57      74.45   2.439 0.015662 *
## uni_l_mf         554.37     126.50   4.382 1.94e-05 ***
## Trade_pc_100K   1028.58     530.45   1.939 0.053982 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1701 on 189 degrees of freedom
## Multiple R-squared:  0.873,  Adjusted R-squared:  0.8569
## F-statistic: 54.15 on 24 and 189 DF,  p-value: < 2.2e-16
```

i. Bruk funksjonen `glance` fra `broom` pakken til å lage en `mod_summary` variabel og `unnest()` denne. Legg resultatet i `mod_sum`.

Vi plukker ut koeffisientene fra modellen og legger dem i tibble-en `coef_df`. (Hint: `map_df(1)` før resulattet pipes til `tibble()`)

i. Lag en ny variabel av type date i `coef_df` som angir år.

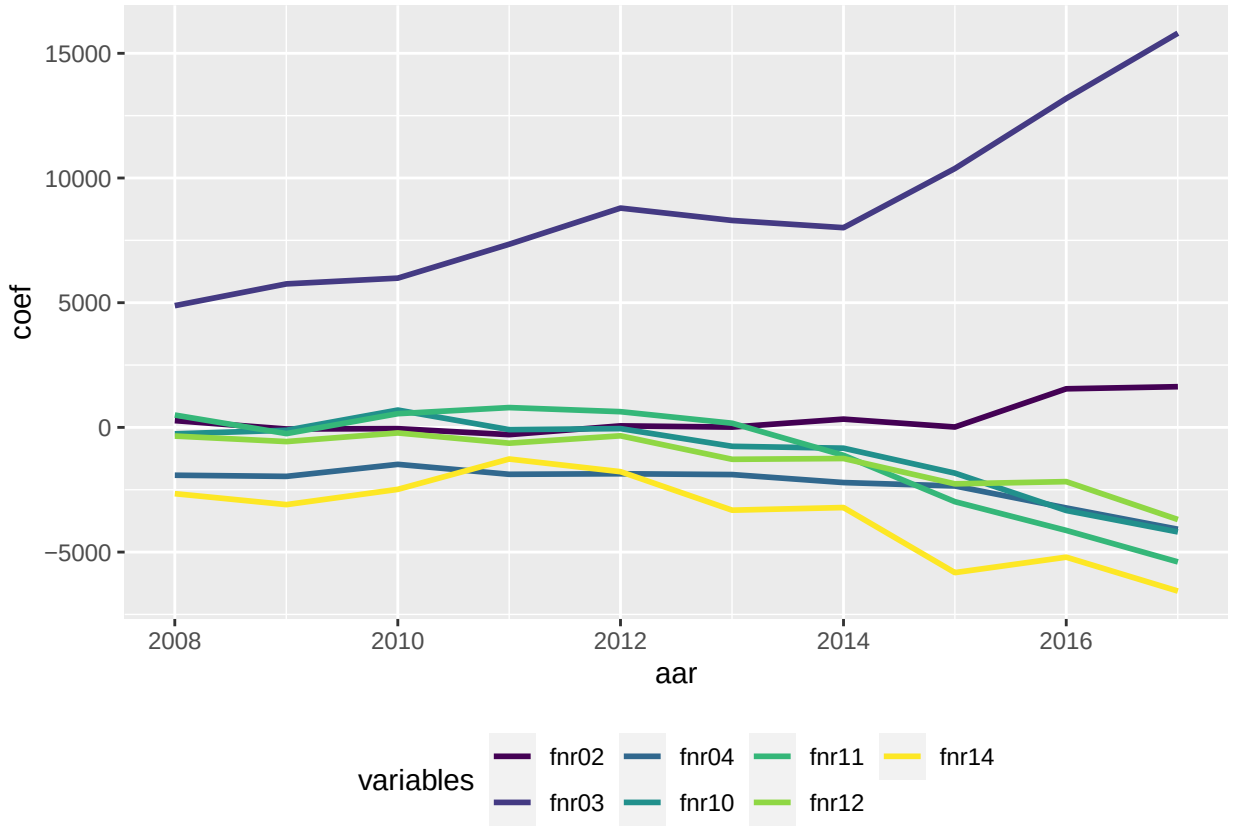
ii. Pivot_longer `coef_df` til `coef_df_long`.

iii. Lag så et plot av fylke-faktorvariablenes koeffisienter for fylkene "fnr02", "fnr03", "fnr04", "fnr10", "fnr11", "fnr12", "fnr14" fra år 2008 til 2017.

iv. Hva sier plot-et oss om prisutviklingen i disse fylkene?

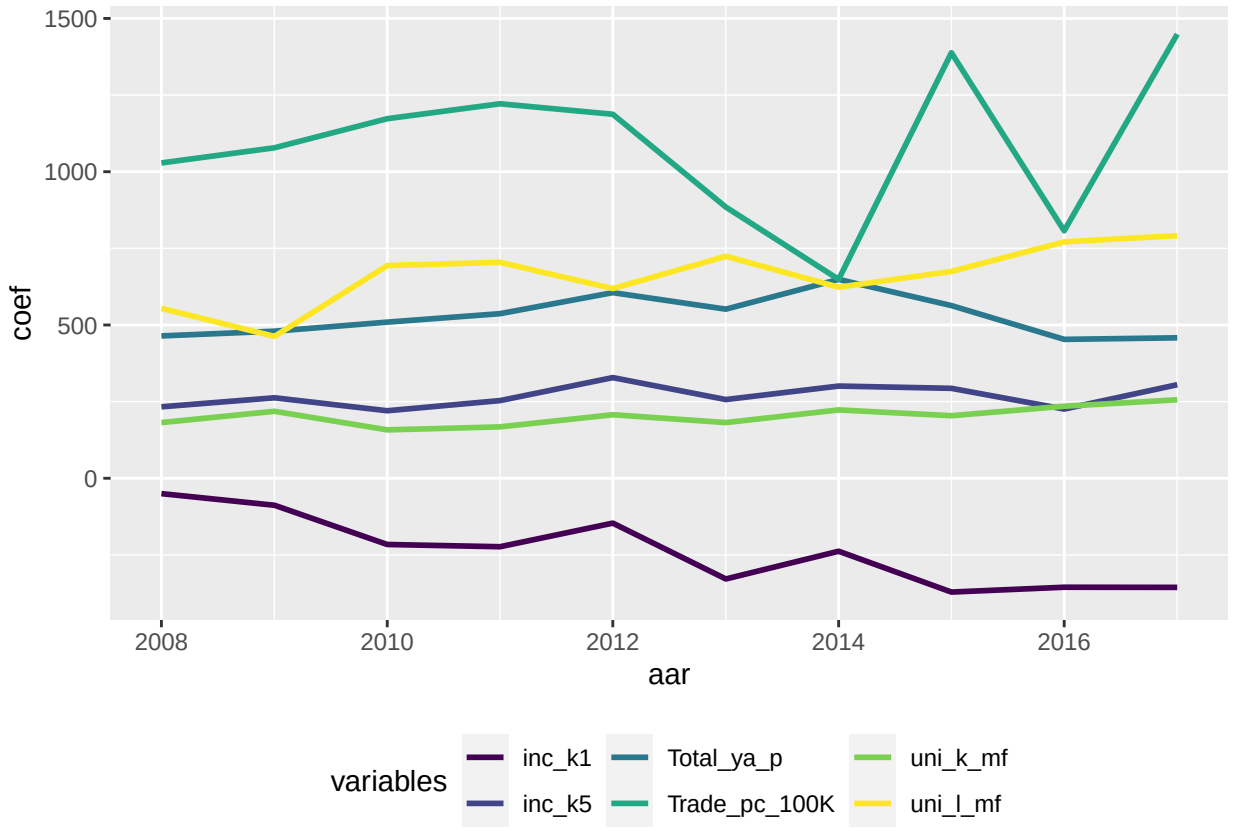
v. Hva skjedde i 2014?

```
coef_df_long %>%
  select(aar, variables, coef) %>%
  filter(
    variables %in% c("fnr02", "fnr03", "fnr04", "fnr10", "fnr11", "fnr12", "fnr14")
    ) %>%
  ggplot(mapping = aes(x = aar, y = coef, colour = variables)) +
  scale_color_viridis(discrete = TRUE, option = "D") +
  geom_line(aes(group = variables), lwd = 1) +
  theme(legend.position = 'bottom')
```



i. Lag et plot tilsvarende det ovenfor for fnr, men nå for variablene Total_ya_p, inc_k1, inc_k5, uni_k_mf, uni_l_mf og Trade_pc_100K. (Plottet er gjengitt nedenfor, dere skal gjenskape det vha ggplot)

ii. Diskuter om koeffisientene ser ut til å være stabile over tid.


1500
1000
500
0
coef
2008
2010
2012
2014
2016
aar
variables
inc_k1
inc_k5
Total_ya_p
Trade_pc_100K
uni_k_mf
uni_l_mf