Oppgave 1: Forslag til løsning

Bjørn Høyland

1/23/2019

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

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

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(readr)
library(modelr)
library(broom)
```

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

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

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

V-dem: Demokratisk utvikling

I forskningen på demokratisk utvikling er det for tiden en rivende utvikling blant annet på grunn av nye og rike datasett som dekker både mange aspekter ved demokrati og dekker en lang tidsperiode. Det rikeste datasettet er muligens v-dem, https://www.v-dem.net/en/. Flere forskere ved Institutt for Statsvitenskap

ved UiO har bedratt til å bygge opp dette datasettet, særlig historiske data. Det være nyttig å ta en liten titt i kodeboka for å et inntrykk av dette datasettet og for å se hva variablene vi bruker fange opp.

Jeg har ekskludert land som kun fantes før 1952, trukket ut et lite utvalg av variabler og tilfeldig valgt ut 10000 rader og lagret dette som en csv-fil. Filen heter **oppgave_1_data.csv** og ligger i samme mappe som denne fila. Last inn dette datasettet med `read_csv`. Kall objektet **vdem_train**. Bruk navnet på objektet til å se på datasettet.

Last inn data og gi noen variabler nye navn

Etterpå bruker du `rename` til å gi variablene navn som er lettere å huske. Kall **v2x_polyarchy** demokrati, **v2pepwrses** likhet, **v2pepwrgen** likestilling, **v2psoppaut** opposisjon og **v2x_corr** korrupsjon. Ikke skift navn på de andre variablene. Sørg også for at variablene ikke endrer rekkefølge.

Vi ønsker å beskrive utviklingen i demokrati over tid og mellom grupper av land og bygge modeller som kan predikere nivå på demokrati utenfor trenings-datasettet vårt.

For at koden skal kjøre må du endre `eval=FALSE` til `eval=TRUE` i topplinjen på r-kode biten.

Mange av dere har ikke endret `eval=FALSE` til `eval=TRUE`. Det har resultert i 0 poeng, evt ingen poeng på de kode-bitene hvor `eval=FALSE`. Ettersom det er noen av dere som aktivt kun har endret de kode-bitene som virker, er det ikke en aktuell løsning for meg å endre det for alle.

```
vdem_train <- read_csv("oppgave_1_data.csv")
vdem_train <- vdem_train %>%
  rename("demokrati" = v2x_polyarchy,
         "likhet" = v2pepwrses,
         "likestilling" = v2pepwrgen,
         "opposisjon" = v2psoppaut,
         "korrupsjon" = v2x_corr)
```

Lag noen figurer

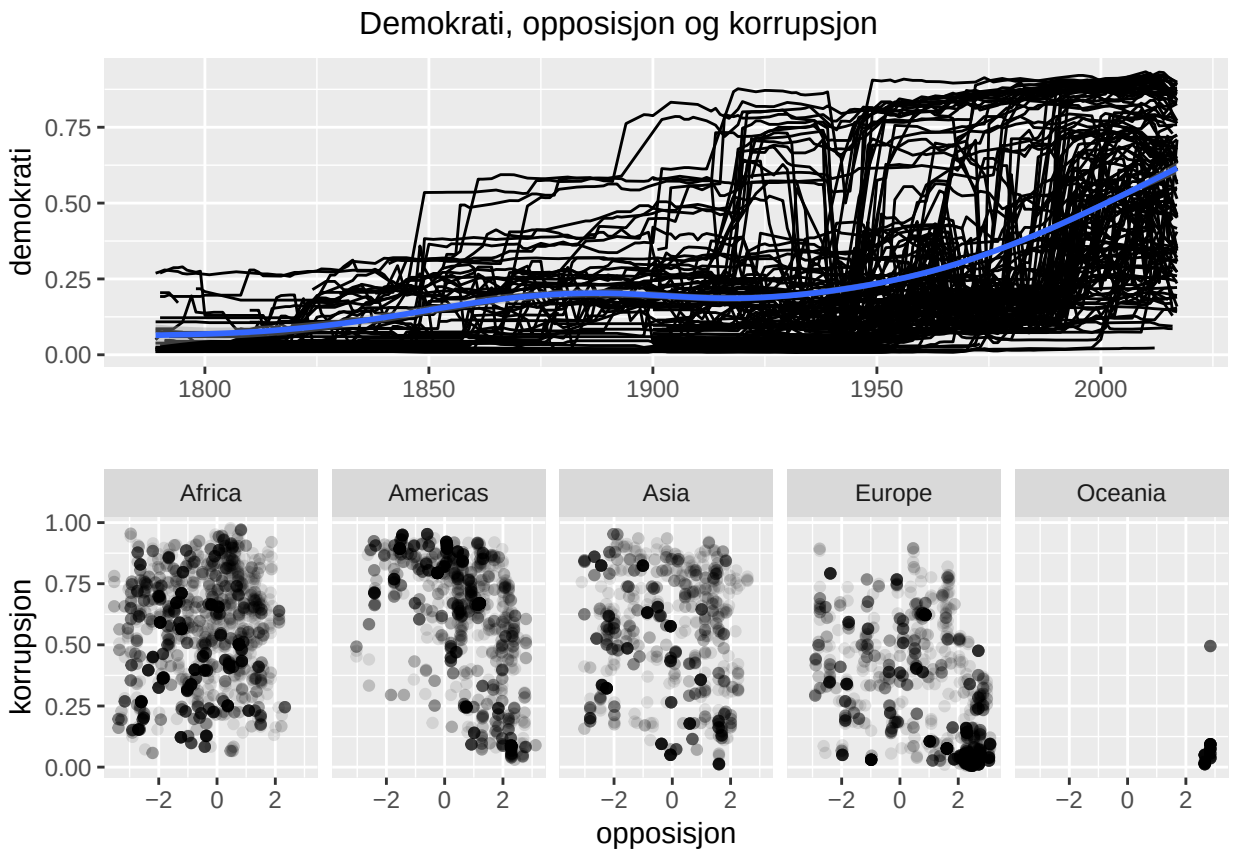
Nå vil jeg at du skal lage to figurer. Den første figuren skal vise utvikling i demokrati over tid innad i hvert enkelt land. Det skal ikke stå **year** på x-aksen, men det skal stå **demokrati** på y-aksen. Gi figuren til objektet **demo_tid**. Legg på semi-parametrisk regresjonslinje som viser sammenhengen på tvers av land.

I den andre figuren skal du vise samvariasjon mellom **korrupsjon** (y-akse) og **opposisjon** (x-akse). Hver observasjon skal være et punkt. Gjør punktene semi-gjennomsiktige slik at vi kan se mønster i data ved å sette alpha = .1 i punkt-argumentet. Bruk **facet_wrap** for å lage en dele opp etter verdensdel. Gi figuren til objektet **korr_opp**. Pass på at alle verdensdelene står ved siden av hverandre (Hint! bruk ncol).

```
demo_tid <- vdem_train %>%
  ggplot(aes(year,demokrati, group = country_name)) +
    geom_line() +
  geom_smooth(group = 1) +
  scale_x_continuous("")
# Det er ikke nødvendig å legge inn scale_y_continuous
# demokrati er allerede på y aksen

korr_opp <- vdem_train %>%
  ggplot(aes(opposisjon,korrupsjon)) +
  geom_point(alpha = .1) +
  facet_wrap(~ continent, ncol =5)
```

```
grid.arrange(demo_tid,korr_opp, nrow = 2, top = "Demokrati, opposisjon og korrupsjon")  # dette skal IK
```



Tell opp andel missing for hvert år

Nå skal du lage din egen funksjon!

Oppgaven er å finne ut hvordan andel missing varierer over tid på de ulike variablene i datasettet. For å gjøre det trenger vi først en funksjon som gi oss andelen missing for en variabel. Denne lager vi med funksjonen `function`. Kall funksjonen **na_prop**. Husk at i logiske variabler er FALSE = 0 og TRUE = 1.

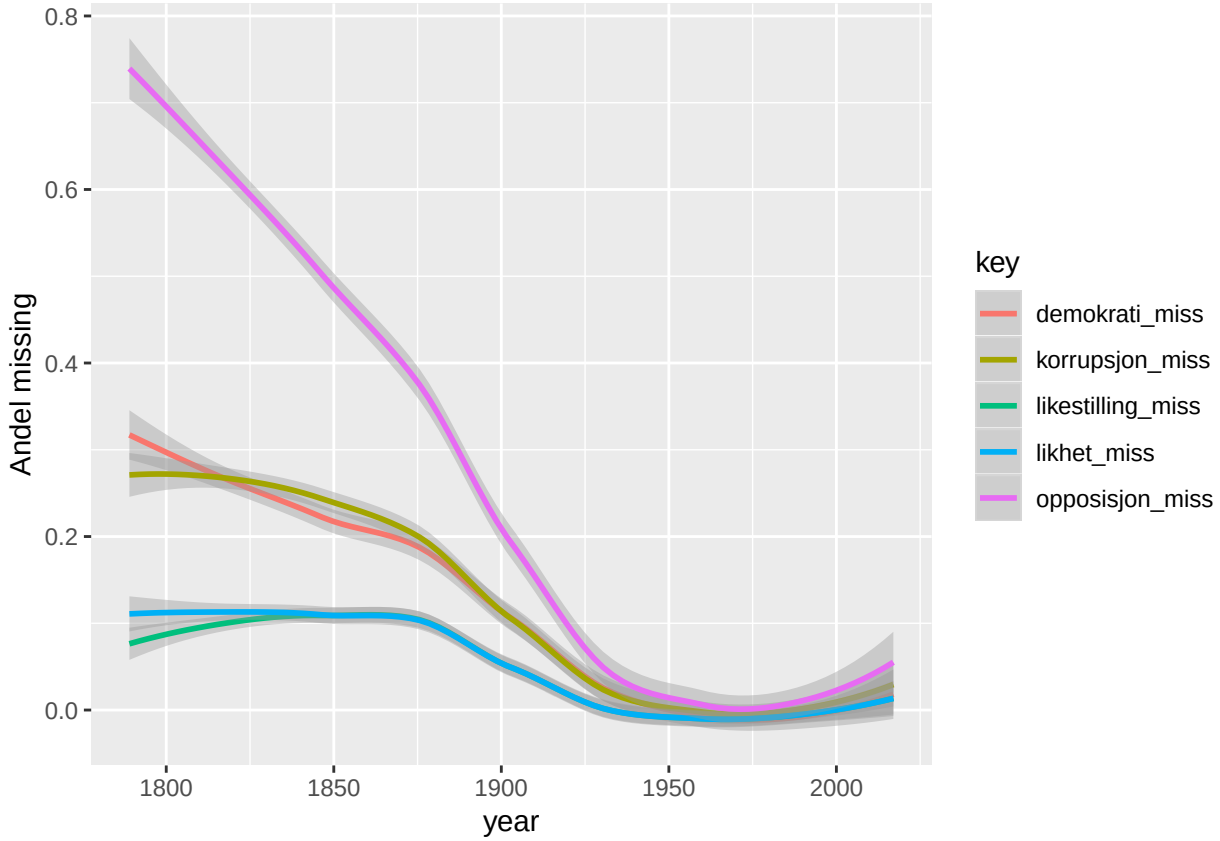
```
na_prop <- function(x) {
 mean(is.na(x))
}
```

Nå vil jeg at du skal bruke **na_prop** til å regne ut andel missing for hvert år på demokrati, likhet, likestilling, opposisjon og korrupsjon sette disse sammen til et datasett (tibble) og sortet etter årstall. Du kan bruke `gather` for å gfjøre dette. Kall første argument i `gather` for **key**. Det er denne som blir gruppe-variabelen din. Legg **_miss** til det ordinære variable-navnet. Legg til snitt-verdi for den ordinære variabelen for det året. Kall dette **year_miss**. Med **year_miss** lag en figur som viser en glattet utvikling i missing over tid for hver av missing variablene. Bruk forskjellige farger for hver variable. Disse må korrespondere med Kall y-aksen for "Andel missing". Kall figuren **fig_year_miss**.

```
year_miss <- vdem_train %>%
  group_by(year) %>%
  summarise(demokrati_miss = na_prop(demokrati),
         likhet_miss = na_prop(likhet),
         likestilling_miss = na_prop(likestilling),
```

```
        opposisjon_miss = na_prop(opposisjon),
        korrupsjon_miss = na_prop(korrupsjon))
fig_year_miss <- year_miss %>%
  gather(key, value, demokrati_miss,likhet_miss,
         likestilling_miss,opposisjon_miss,
         korrupsjon_miss) %>%
  ggplot(aes(year,value, color = key)) +
    geom_smooth() +
  scale_y_continuous("Andel missing")

  # gi y aksen som viser andel missing navnet missing
fig_year_miss
```



Regresjon

Vi har nå kommet frem til regresjonsmodeller. Vi fjerner først missing data med `na.omit` og kaller det nye datasettet **vdem_complete**. Nå skal du lage en minimalistisk prediksjonsmodell for demokratisk nivå. Med 2 variabler, lag en lm modell hvor $r^2 > .7$. Du kan ikke ha "country_name" i modellen. Ved kategoriske variabler teller en variable med 2 nivå som 1 variabler, mens en med 3 nivå som 2 variabler, osv. Tilsvarende, en interaksjon teller som tre variabler. Dette er fordi vi må estimere en β for hver av dem. Husk at du må angi datasettet med argumentet **data** etter "," .

```
vdem_complete <- na.omit(vdem_train)
lm_mod <- lm(demokrati ~ likestilling + opposisjon, data = vdem_complete)
summary(lm_mod) # vis resultatene, litt mer info enn du egentlig trenger
```

```
##
## Call:
## lm(formula = demokrati ~ likestilling + opposisjon, data = vdem_complete)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45481 -0.08392  0.00403  0.09515  0.36271
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3363107  0.0017425  193.00   <2e-16 ***
## likestilling 0.0831589  0.0012124   68.59   <2e-16 ***
## opposisjon   0.0919776  0.0009927   92.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1375 on 7971 degrees of freedom
## Multiple R-squared:  0.7324, Adjusted R-squared:  0.7323
## F-statistic: 1.091e+04 on 2 and 7971 DF,  p-value: < 2.2e-16
```

Nå skal du lage en god modell for et enkelt-land som en `function`, og anvende denne på tvers av alle land med `map`. Kall denne **reg_fun**. Du må gruppere data som du skal kjøre modellen på på land. Kall dette objektet **by_country**.

Vurder justert r^2, hvor godt passer modellen på tvers av treningsdata? Bruk `summarise` til å lage et oppsummerende datasett I **reg_mods** rapporter følgene oppsummerende variabler, `min`, `mean`, `median`, `max` og `sd` for hver verdensdel. Kall disse variablene henholdsvis **adj_r2_sq_min**, **adj_r2_sq_mean**, **adj_r2_sq_median**, **adj_r2_sq_max**, **adj_r2_sq_sd**. For at modellen skal være god, må **adj_r2_sq_mean** være minst .75 for alle verdensdeler.

```
reg_fun <- function(df) {
    lm(demokrati ~ likestilling + opposisjon + year, data = df)
}
by_country <-  vdem_complete %>%
    group_by(country_name, continent) %>%
    nest() %>%
  mutate(model = map(data,reg_fun))
reg_mods <- by_country  %>%
  mutate(glance = map(model,glance)) %>%
  unnest(glance, .drop = TRUE) %>%
  select(adj.r.squared, continent) %>%
  group_by(continent)
mods_sum <- reg_mods %>%
  summarise(adj_r2_sq_min = min(adj.r.squared),
            adj_r2_sq_mean = mean(adj.r.squared),
            adj_r2_sq_median = median(adj.r.squared),
            adj_r2_sq_max = max(adj.r.squared),
            adj_r2_sq_sd = sd(adj.r.squared))

mods_sum %>%
  summarise(min(adj_r2_sq_mean)) > .75
```

```
##      min(adj_r2_sq_mean)
## [1,]                TRUE
```

KNN-regresjon

Til slutt prøver vi oss på en ikke-parametrisk modell på tvers av alle land, vi bruker funksjonen `knnreg` i **library(caret)**. Vi evaluerer prediksjonskraft som gjennomsnittet av kvadratroten av den absolutte forskjellen mellom predikert y og faktisk y. Lag funksjonen for dette. Kall den funksjonen `mean_squared_abs_error`.

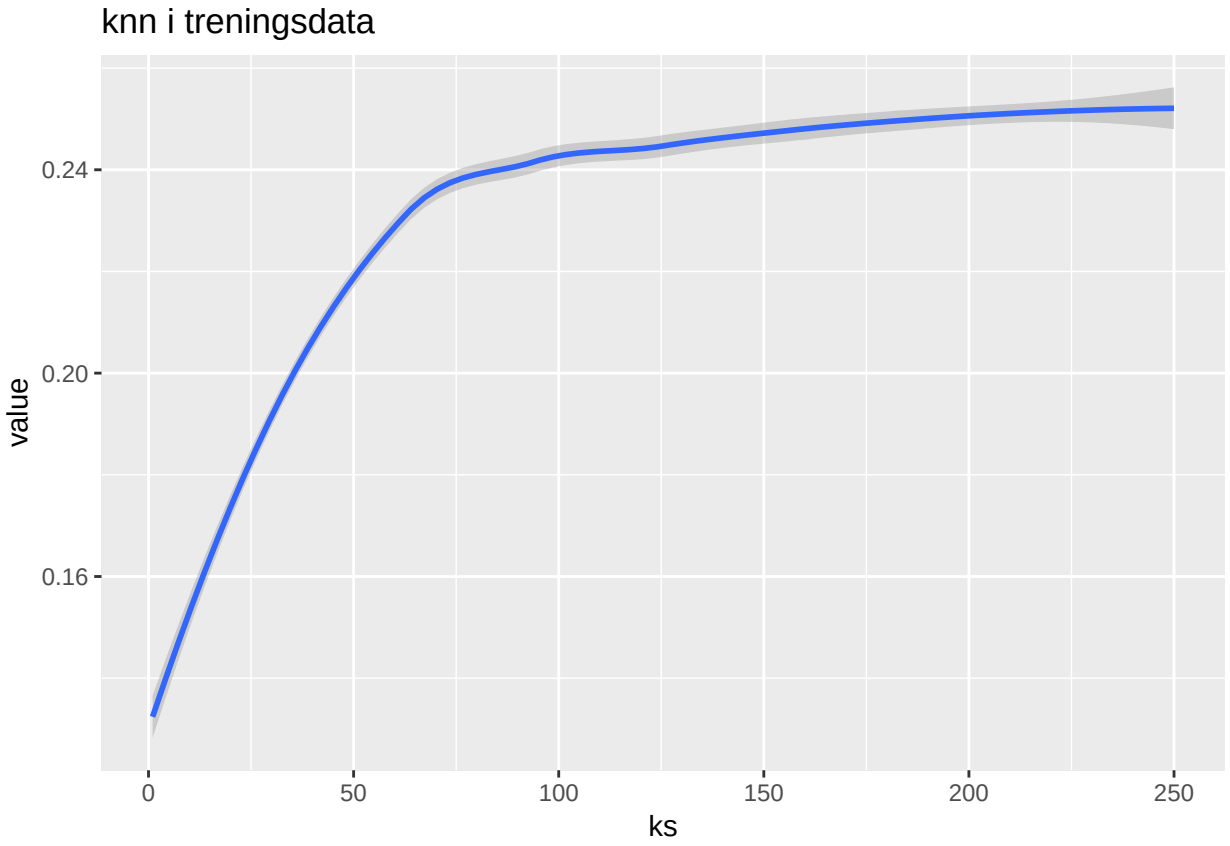
Vi bruker alle variablene utenom "country_name","kontinent" og "year" i modellen. Modellen kaller vi **knn_mod**. Her har vi gitt **k** verdien i.

Varier i fra 1 - 250 med **ks** og prediker y i treningsdata for hver k. Kall dette objektet **knn_mods**.

Kalkuler gjennomsnittlig prediksjonskraft med `mean_squared_abs_error` for hver verdi av **ks** i **knn_mods**. Dette objektet skal være en `tibble` og ha to kolonner, *value* er verdien av, *ks* er verdien på k. Kall dette objected for **knn_k**.

Med **knn_k**, lag en figur i som viser hvordan modellen predikerer i testdata. Legg på en ikke-parametrisk trendlinje. La y-aksen vise verdien fra `mean_squared_abs_error` og x-aksen vise **k**. Du skal ikke endre navn på hverken x eller y aksen. Legg på tittelen "knn i treningsdata". Kall figure **fig_knn**.

```
mean_squared_abs_error <- function(preds,y){
  sqrt(abs(preds - y))
}
knn_mod <- function(i,dt){
  knnreg(demokrati ~ likhet + likestilling + opposisjon + korrupsjon,
                  data = dt, k = i)
}
ks <- 1:250
knn_mods <- map(ks,function(i) {
    knn_mod(i,vdem_complete) %>%
    predict(vdem_complete)})
knn_k <- map(knn_mods,mean_squared_abs_error, vdem_complete$demokrati) %>%
  map(mean) %>%
  unlist() %>%
  cbind(value = ., ks) %>%
  as_tibble()
knn_fig <-  ggplot(knn_k, aes(ks, value)) +
  geom_smooth() +
    ggtitle("knn i treningsdata")
knn_fig
```



Til slutt, last inn **oppgave_1_testdata.csv**, gi navn og fjern missing. Bruk datasettet til å predikere missing. Kalkuler snitt av `mean_squared_abs_error` for alle verdier i **ks**. Kall denne vektoren av verdier **test** og lag en ny veriable i **knn_k** som er treningsverdien minus testverdien. Kall denne variablen **diff**. Bruk det oppdaterte *knn_k **til å lage en figur som vise forskjell mellom trening og testdata som en linje hvor** ks **er på x-aksen of** diff **på y-aksen. Gi figuren overskriften "KNN: Forskjell i feil-prediksjon: Trening - Test". Kall figuren** fig_diff**. Kall x-aksen for "k" og y-aksen for "trening - test".

```
vdem_test <- read_csv("oppgave_1_testdata.csv")
```

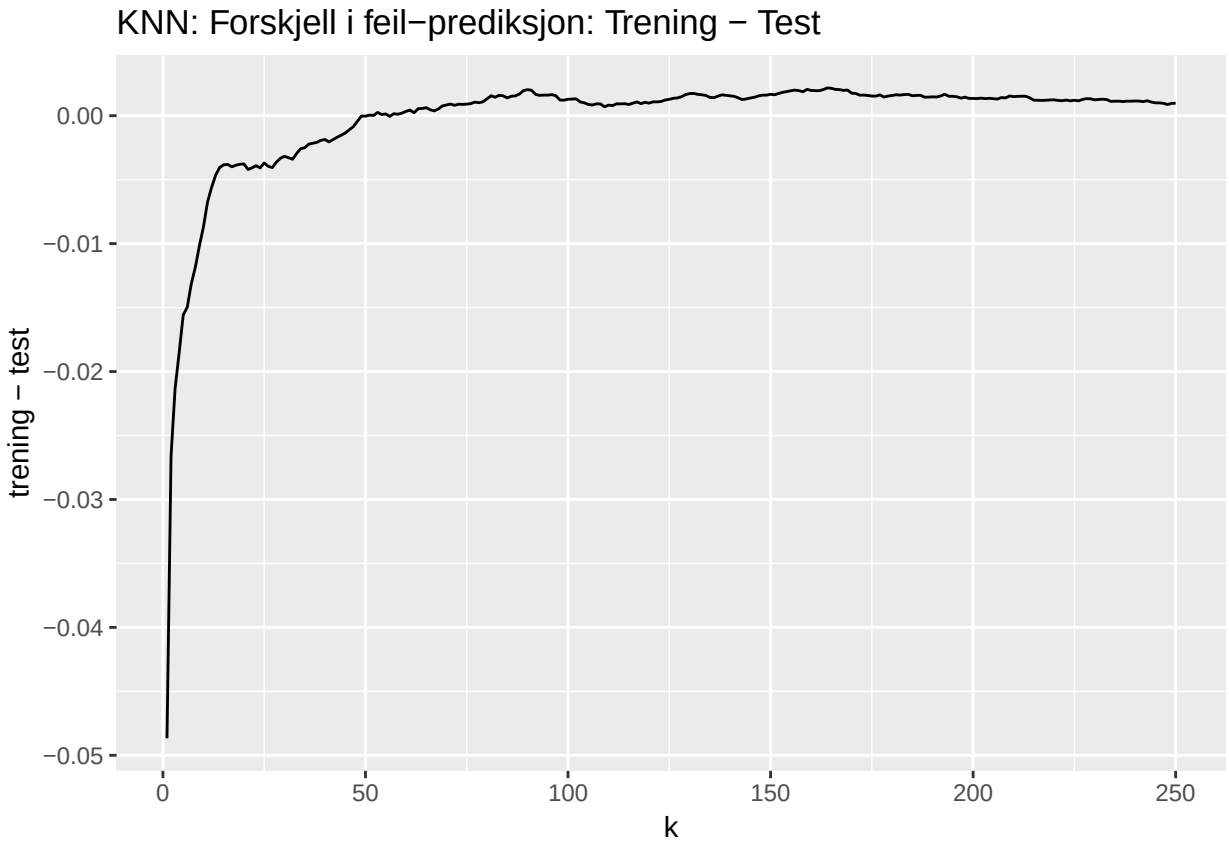
```
## Parsed with column specification:
## cols(
##   country_name = col_character(),
##   year = col_double(),
##   v2x_polyarchy = col_double(),
##   v2pepwrses = col_double(),
##   v2pepwrgen = col_double(),
##   v2psoppaut = col_double(),
##   v2x_corr = col_double(),
##   continent = col_character()
## )
```

```
vdem_test <- vdem_test %>%
  rename("demokrati" = v2x_polyarchy,
         "likhet" = v2pepwrses,
         "likestilling" = v2pepwrgen,
         "opposisjon" = v2psoppaut,
         "korrupsjon" = v2x_corr)
```

```r
vdem_test <- na.omit(vdem_test)
test <- map(ks,function(i) {
    knn_mod(i,vdem_complete) %>%
      predict(vdem_test)})

test<- map(test,mean_squared_abs_error, vdem_test$demokrati) %>%
    map(mean) %>%
    unlist()
fig_diff <- knn_k %>%
  mutate(diff = value - test) %>%
  ggplot(aes(ks, diff)) +
  geom_line() +
  ggtitle("KNN: Forskjell i feil-prediksjon: Trening - Test") +
  scale_x_continuous("k") +
  scale_y_continuous("trening - test")

fig_diff
```



Trykk på "Knit" og last opp oppgave_1.Rmd.