Case 1

277

Denne oppgaven er tilpasset fra Case 1, skrevet av Øystein Myrland for kurset SOK-1004, høsten 2021. Eventuelle feil og mangler er mine egne. Rett spørsmål og kommentarer til even.c.hvinden@uit.no.

Instruksjoner

Denne oppgaven skal løses interaktivt i RStudio ved å legge inn egen kode og kommentarer. Det ferdige dokumentet lagres med kandidatnummeret som navn `[kandidatnummer]_SOK1004_C1_H22.qmd` og lastes opp på deres GitHub-side. Hvis du har kandidatnummer 43, så vil filen hete `43_SOK1004_C1_H22.qmd`. Påse at koden kjører og at dere kan eksportere besvarelsen til pdf. Dere leverer lenken til GitHub-repositoriumet i Canvas.

Bakgrunn

Vi skal analysere utviklingen i bruttonasjonalprodukt (BNP) per person i Norge. Vi bruker data Statistisk Sentralbyrå (SSB), tabell "09842: BNP og andre hovedstørrelser (kr per innbygger), etter statistikkvariabel og år". Tabellen inneholder årlige data på BNP per innbygger, fra 1970 til 2021.

I. API, visualisering

SSB gir oss tilgang til sine data via en API (*Application Programming Interface*), programvare som lar to applikasjoner kommunisere med hverandre. SSB tilbyr en API med ferdige datasett. Her er det om lag 250 kontinuerlig oppdaterte datasett med en fast URL over de mest brukte tabellene i Statistikkbanken.

For å få tilgang til tabellen med bruttonasjonalprodukt må vi benytte tjenesten PxWebApi. Her finner du en API konsoll med en søkefunksjon. Prøv å søk på "`bnp`" og merk forslaget: tabell 09842. Søk på denne, og noter URL-en. Den vil vi bruke etterpå.

Til å laste ned dataene skal vi bruke en R-pakke, PxWebApiData, som SSB har laget. I første omgang skal vi bruke funksjonen `ApiData()`. Syntaksen er ikke den samme som i `tidyverse`, og har noen litt uvante egenskaper, herunder lagring i tegnformat og en kombinasjon av norsk og engelsk.

Tips: Det er typisk instruktivt å se på eksempel på bruk. Da har man et intuitivt utgangspunkt for hvordan koden kan brukes.

Jeg vil nå vise dere trinnvis hvordan å laste ned dataene. Formålet er å gi dere en idé på hvordan man kan lære seg å bruke en ny pakke eller funksjon. Vi begynner med å laste inn nødvendige pakker:

```
rm(list=ls())
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(PxWebApiData)
```

NB! Du må installere `PxWebApiData` først. Kjør kommandoen `install.packages("PxWebApiData")` i konsollen. Det må kun gjøres én gang.

Vi bruker funksjonen `ApiData()` til å hente tabell 09842. Som notert ovenfor fant vi URL-en ved hjelp av søkefunksjonen til SSB. Først prøver vi å laste ned dataene direkte, uten ytterligere tilvalg, og tar en titt på hva vi får.

```
lenke <- "http://data.ssb.no/api/v0/no/table/09842"

df <- lenke %>%
  ApiData()

df %>%
  print()
```

```
$`09842: BNP og andre hovedstørrelser (kr per innbygger), etter statistikkvariabel og år`
                                statistikkvariabel   år  value
1                            Bruttonasjonalprodukt 1970  23616
2                            Bruttonasjonalprodukt 2020 633965
3                            Bruttonasjonalprodukt 2021 765836
4 Konsum i husholdninger og ideelle organisasjoner 1970  12283
5 Konsum i husholdninger og ideelle organisasjoner 2020 278844
6 Konsum i husholdninger og ideelle organisasjoner 2021 298804
7   MEMO: Bruttonasjonalprodukt. Faste 2015-priser 1970 214756
8   MEMO: Bruttonasjonalprodukt. Faste 2015-priser 2020 604951
9   MEMO: Bruttonasjonalprodukt. Faste 2015-priser 2021 625077

$dataset
  ContentsCode  Tid  value
1          BNP 1970  23616
2          BNP 2020 633965
3          BNP 2021 765836
4    KonsumHIO 1970  12283
5    KonsumHIO 2020 278844
6    KonsumHIO 2021 298804
7      MEMOBNP 1970 214756
8      MEMOBNP 2020 604951
9      MEMOBNP 2021 625077
```

Merk følgende: `df` inneholder to datasett i formatet `data.frame`. Datasettene heter `"09842: BNP og andre hovedstørrelser (kr per innbygger), etter statistikkvariabel og år"` og `dataset`. Datasettene inneholder 9 verdier av 3 variabler. Variabelen `value` er identisk. Variablene `år` og `Tid` inneholder de identiske verdiene `"1970"`, `"2020"` og `"2020"`. Merk at disse er i tegnformat `<chr>` (derav anførselstegnene) og ikke en numerisk verdi, for eksempel `<dbl>`. Variabelen `statistikkvariabel` og `ContentsCode` inneholder henholdsvis verdiene `BNP`, `KonsumHIO MEMOBNP` og `Bruttonasjonalprodukt, Konsum i husholdninger og ideelle organisasjoner` og `MEMO: Bruttonasjonalprodukt. Faste 2015-priser`.

Vi har altså ikke fått hele tabell 09842, men verdiene for tre statistikkvariabler over tre tidsperioder, lagret med forskjellige variabelnavn og verdier.

Det vi trenger er **metadata**: Informasjon som beskriver innholdet i dataene, slik at vi kan filtrere API-spørringen. Kjør følgende kode.

```r
metadata <- lenke %>%
  ApiData(returnMetaData = TRUE)
```

Åpner vi listen `metadata` fra minnet så kan vi se nærmere på den i øvre venstre vindu i Rstudio. Her ser vi to lister kalt `[[1]]` og `[[2]]`. Listene beskriver variablene vi kan filtrere på. Liste `[[1]]` har fire variable: `code`, `text`, `values`, og `valueTexts`. Alle variablene er `<chr>`. Liste `[[2]]` har de samme foregående fire variablene samt en variabel `time`.

- `code` viser navnene på variablene vi bruker i funksjonen `ApiData()` for å filtrere. Den tar verdiene `ContentsCode` og `Tid`. Legg merke til at utviklerne i SSB her blander norsk og engelsk.
- `text` er en unik tekstverdi tilknyttet verdien på `code` som forklarer hva vi ser på. Den tar verdien `statistikkvariabel` og `år`. Vi kan altså filtrere på statistikkvariabel og år.
- `values` viser hvilke verdier av statistikkvariabel og år vi kan velge, med henholdsvis 6 og 52 forskjellige verdier. Du vil kjenne igjen tre av hver fra den første spørringen ovenfor.
- `valueTexts` gir en unik tekstverdi tilknyttet verdien på `values` som forklarer oss hva vi ser på. For `Tid` og `år` er de identiske, men for `ContentsCode` og `statistikkvariabel` får vi en mer fullstendig forklaring.
- `time` er en logisk variabel, og tar derfor to verdier: `TRUE` og `FALSE`. I dette tilfellet indikerer den at variabelen `Tid` måler tid, hvilket gjør at funksjonene i pakken vil behandle `Tid` på en annen måte enn en statistikkvariabel.

Vi har nå informasjonen vi trenger til å laste ned BNP-tall mellom 1970 og 2021. Jeg velger å ta BNP med både løpende og faste priser.

```
df <- lenke %>%
  ApiData(Tid = paste(1970:2021), ContentsCode = c("BNP","MEMOBNP"))
```

På venstre side av likhetstegnet bruker vi `code` fra metadata. På høyre side velger vi verdier fra `values`. Merk at jeg bruker funksjonen `paste()` for å konvertere numeriske verdier, for eksempel `<dbl>` til tegn `<chr>`.

La oss rydde i data. Det er tre ting å ta tak i:

1. `df` lagrer informasjonen i to tabeller med samme informasjon, som vist over. Det er unødvendig.
2. Årstallene er lagret som tegn, `<chr>`. Disse skulle heller være heltall, `<int>`.
3. Formatet `data.frame` er underlegent `tibble`.

Oppgave Ia: Rydd i data

Skriv kode som lagrer dataene som én tibble med anstendige variabelnavn og årstall som heltall. Fremover bruker jeg "var", "tid", og "verdi" for "statistikkvariabel", "Tid", og "value".

```r
# Oppgave Ia løses her.

# Ønsker å starte med "blanke ark" så sletter minnet og importerer pakker.
rm(list=ls())
library(tidyverse)
library(PxWebApiData)

# Henter inn data og fester det til "data".
data <- "http://data.ssb.no/api/v0/no/table/09842"

# Henter inn data for BNP, angir periode og hvilken variabler jeg vil ha med "BNP" og "MEM
df1 = data %>%
  ApiData(Tid = paste(2000:2021), ContentsCode = c("BNP", "MEMOBNP"))

# Har nå fått ett datasett som inneholder 2 lister, trenger kun den en av disse listene så
df2 = df1[[2]]

# Konverterer (mutate) tid fra <chr> til <int> (parse_integer) og endrer navn på kolonnene
df3 <- df2 %>%
  mutate(Tid=parse_integer(Tid)) %>%
  rename(var=ContentsCode, tid=Tid, verdi=value)

# Konverterer dataframen til en tibble og lager en ny gruppe som jeg kaller for df_tibble.
df1_tibble <- as_tibble(df3)

# Sjekker om konverteringen fungerer.
is_tibble(df3)
```

```
[1] FALSE
```

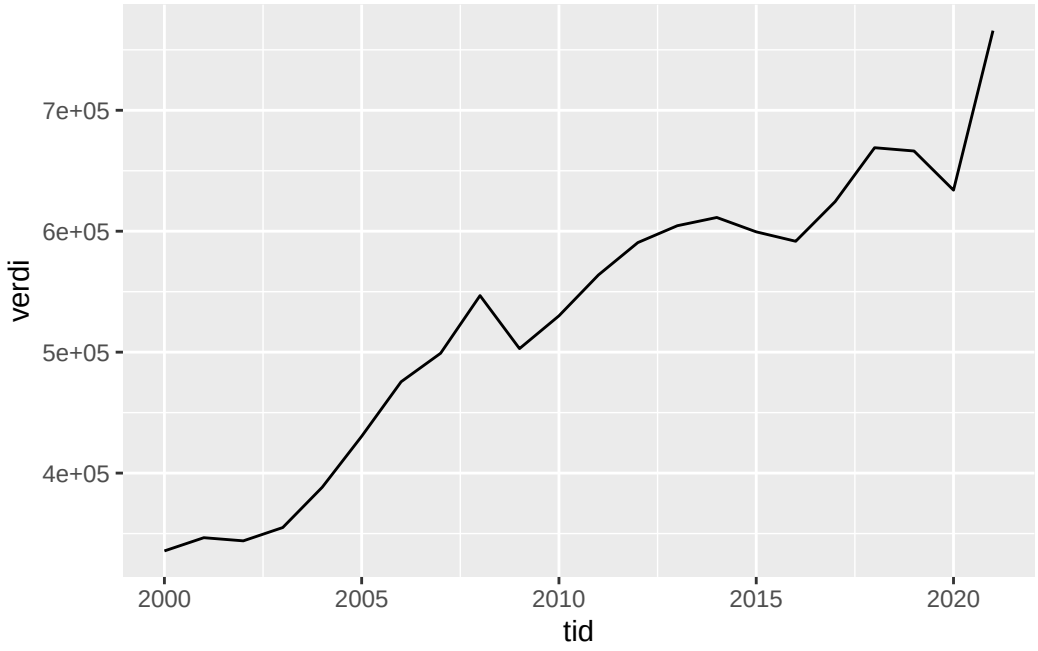
```r
is_tibble(df1_tibble)
```

```
[1] TRUE
```

Oppgave Ib: Lag en figur

Følgende kode skaper en enkel figur.

```r
df1_tibble %>%
  filter(var == "BNP") %>%
  ggplot(aes(x=tid,y=verdi)) +
  geom_line()
```

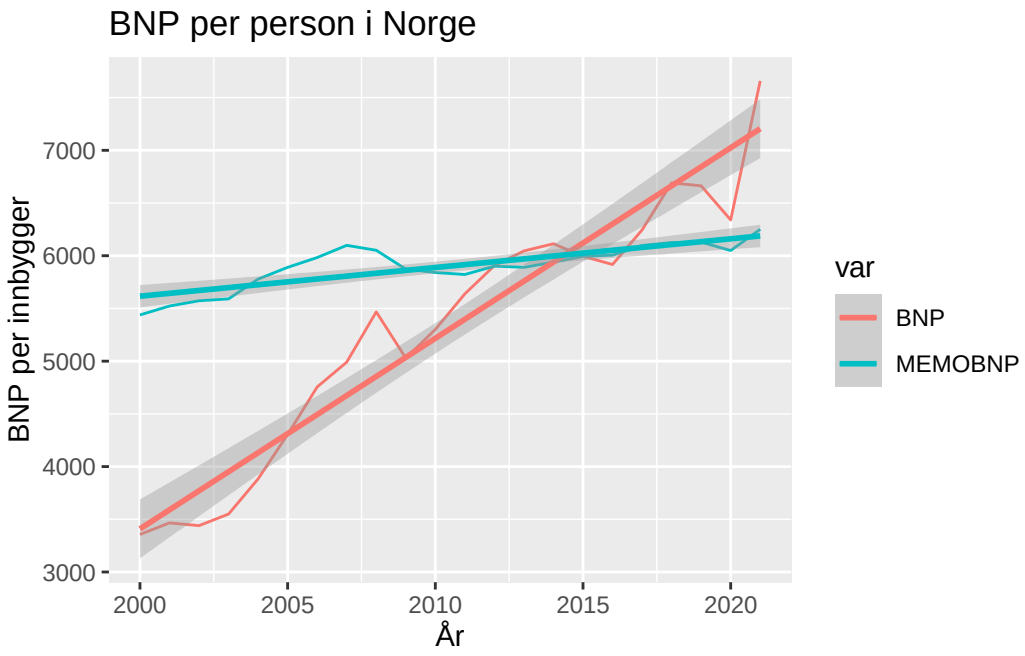



```r
# Fordandret fra "df" -> "df_tibble" slik at den passet inn i mitt dokument.
```

Lag en pen figur som viser BNP i tusener av kroner per person, i både løpende og faste priser, mellom 2000 og 2021. Skriv en tydelig forklaring og tolkning av figuren. Hvordan har inntektene utviklet seg? Forklar forskjellen mellom BNP i løpende og faste priser. Til hvilke formål er de mest relevante?

```r
# Lager figuren. Konverterer (mutate) verdiene i "df1_tibble" til å vise BNP i tusner med
df1_tibble %>%
  mutate(verdi=verdi/100) %>%
  ggplot(aes(x=tid, y=verdi, col=var)) +
  geom_line() +
  geom_smooth(method = lm) +
  labs(title = "BNP per person i Norge",
       x = "År",
       y = "BNP per innbygger")
```

```
`geom_smooth()` using formula 'y ~ x'
```



Figuren over viser veksten i BNP per inbygger fra år 2000 til år 2020. På den røde linjen kan vi se BNP i Norge og den blå linjen kan vi se BNP i Norge med samme priser som i 2015 (faste priser). Visst vi skal se på reel utvikling i BNP bør vi se på den blå grafen ettersom den viser hvor mye mer Norge produserer visst prisene er kontast det samme kontra den røde grafen som har flytende prise (kjøpekraft).

II. Transformasjon, visualisering

Våre data er en tidsserie, hvilket betyr at rekkefølgen i observasjonene er ordnet etter tid. Vi skal nå regne prosentvis, årlig endring. La x_t være BNP i år t. For eksempel vil x_{1970} være 23616.

Den årlige endringen i BNP fra år $t-1$ til t er gitt ved $x_t - x_{t-1}$. I samfunnsøkonomi er det vanlig å betegne dette som $\Delta x_t := x_t - x_{t-1}$. Tegnet Δ er den greske bokstaven delta og betegner differanse. For eksempel vil $\Delta x_{1971} = 26363 - 23616 = 2747$ kroner.

I mange tilfeller er vi interesserte i relativ vekst: Hvor mye økte BNP, relativt til hva den var i utgangspunkt? Den mest brukte enheten er hundredeler eller prosentvis endring, gitt ved $100 \times \Delta x_t / x_{t-1}$. For eksempel var den prosentvise endringen i BNP i 1971 $100 \times \Delta x_{1971}/x_{1970} = 100 \times (2747/23616) \approx 11.6$, hvor $\approx$ betegner "omtrent lik" da jeg viser svaret med kun én

desimal. Tilsvarende kan man skrive at $\Delta x_{1971}/x_{1970} = 2747/23616 \approx 0.116 = 11.6\%$, hvor tegnet % betegner at beløpet oppgis i hundredeler eller prosent.

Oppgave IIa: Omorganisere datasett med `pivot_wider()`

Vi skal lage to variable `dBNP` og `dMEMOBNP` som viser relativ endring i `BNP` og `MEMOBNP`. Til dette formålet skal vi bruke kommandoene `pivot_wide()` og `pivot_long()` til å omorganisere dataene. Jeg anbefaler dere først å lese kapittel 12.3 i pensum. Betrakt følgende kode.

```
df_wide <- df1_tibble %>%
  pivot_wider(names_from = var, values_from = verdi)

# Fordandret fra df -> df1_tibble så det passer inn i mitt dokument.
```

Beskriv konkret hva koden gjorde. Sammenlign `df` og `df_wide`.

Koden bytter om hvordan informasjonen vises. Den gjør listen kortere med å legge "MEMOBNP" på siden og fjerner verdi kategorien og legger verdiene under "TID", "BNP" og "MEMOBNP". Den gjør det enklere å lese både "BNP" og "MEMOBNP" samtidig.

Oppgave IIb: Beregn vekst

Til å beregne endring er funksjonen `lag()` meget nyttig. I denne konteksten er begrepet *lag* et engelsk verb som beskriver foregående observasjon. Bruker vi funksjoenen `lag()` på en variabel (kolonne) så returnerer den en ny kolonne hvor verdien er lik foregående observasjon. Betrakt følgende kode:

```
df1_wide <- df_wide %>%
  mutate(LBNP = lag(BNP,n=1L)) %>%
  mutate(LMEMOBNP = lag(MEMOBNP,n=1L))

# legger variablene i rekkefølge

df2_wide <- df1_wide %>%
  relocate("LBNP", .before = "MEMOBNP")

df2_wide
```

```
# A tibble: 22 x 5
     tid    BNP   LBNP MEMOBNP LMEMOBNP
   <int>  <int>  <int>   <int>    <int>
```

```
 1  2000 335626     NA  543829       NA
 2  2001 346565 335626  552311   543829
 3  2002 343978 346565  557285   552311
 4  2003 354965 343978  559068   557285
 5  2004 388296 354965  577835   559068
 6  2005 430427 388296  588981   577835
 7  2006 475535 430427  598277   588981
 8  2007 499065 475535  609848   598277
 9  2008 546765 499065  605164   609848
10  2009 502924 546765  587259   605164
# ... with 12 more rows
# i Use `print(n = ...)` to see more rows
```

Hvis vi bruker den matematiske notasjonen diskutert tidligere så har vi nå kolonner med x_t (`BNP`, `MEMOBNP`) og x_{t-1} (`LBNP`, `LMEMOBNP`).

Bruk funksjonen `mutate()` til å lage en ny variabel med relativ endring i `BNP` og `MEMOBNP` i `df_wide` og lagre de som `DBNP` og `DMEMOBNP`.

```
# Besvar oppgave IIb her

# Legger til "DNBP" og "DMEMOBNP" og fester dette til df3_wide, og definerer deres verdier
df3_wide <- df2_wide %>%
  mutate(DBNP = BNP - lag(BNP,n=1L)) %>%
  mutate(DMEMOBNP = MEMOBNP - lag(MEMOBNP,n=1L))


# Kaller på funksjonen for å få opp tabellen.
df3_wide
```

```
# A tibble: 22 x 7
     tid    BNP   LBNP MEMOBNP LMEMOBNP   DBNP DMEMOBNP
   <int>  <int>  <int>   <int>    <int>  <int>    <int>
 1  2000 335626     NA  543829       NA     NA       NA
 2  2001 346565 335626  552311   543829  10939     8482
 3  2002 343978 346565  557285   552311  -2587     4974
 4  2003 354965 343978  559068   557285  10987     1783
 5  2004 388296 354965  577835   559068  33331    18767
 6  2005 430427 388296  588981   577835  42131    11146
 7  2006 475535 430427  598277   588981  45108     9296
 8  2007 499065 475535  609848   598277  23530    11571
 9  2008 546765 499065  605164   609848  47700    -4684
```

```
10  2009 502924 546765  587259   605164 -43841   -17905
# ... with 12 more rows
# i Use `print(n = ...)` to see more rows
```

Oppgave IIc: Omorganisere datasett med `pivot_longer()`

Bruk nå funksjonen `pivot_longer()` til å transformere `df_wide` til det opprinnelige formatet, altså med variablene `var` og `verdi`. Kall den transformerte tabellen for `df_long`.

NB! Husk å bruk anførselstegn (`"[variabelnavn]"`) når du definerer nye variable i `pivot_longer()`.

```
# Besvar oppgave IIc

# Legger inn dataen fra df3_wide og forandrer fra "wide" til "long" slik at "BNP, ... ,DME
df1_long <- df3_wide %>%
  pivot_longer(c("BNP", "LBNP", "DBNP", "MEMOBNP", "LMEMOBNP", "DMEMOBNP"), names_to = "va

# Kaller på funksjonen for å få opp grafen.
df1_long
```

```
# A tibble: 132 x 3
     tid var       verdi
   <int> <chr>     <int>
 1  2000 BNP      335626
 2  2000 LBNP         NA
 3  2000 DBNP         NA
 4  2000 MEMOBNP  543829
 5  2000 LMEMOBNP     NA
 6  2000 DMEMOBNP     NA
 7  2001 BNP      346565
 8  2001 LBNP     335626
 9  2001 DBNP      10939
10  2001 MEMOBNP  552311
# ... with 122 more rows
# i Use `print(n = ...)` to see more rows
```

Oppgave IId: Figur med vekst

Lag en pen figur med prosentvis vekst i nominelt og reelt BNP per person fra 1970 til 2021. Finnes det observasjoner med negativ vekst i reell BNP? Hva skyldes dette?

Merknad: Det er en del støy i data. Prøv å kombinere `geom_point()` og `geom_smooth()` for å få et bedre inntrykk av den langsiktige utviklingen.

```
# Besvar oppgave IId her

#Her kombinerer jeg egentlig alt jeg har gjort tidligere i denne oppgaven til en ny graf s

#Lager en ny df med tall fra 1970 - 2021 ettersom den gammle var fra 2000 - 2021.
df_1970 = data %>%
  ApiData(Tid = paste(1970:2021), ContentsCode = c("BNP", "MEMOBNP"))

# Har nå fått ett datasett som inneholder 2 lister, trenger kun den en av disse listene så
df_1970 = df_1970[[2]]

# Konverterer (mutate) tid fra <chr> til <int> (parse_integer) og endrer navn på kolonnene
df_1970 <- df_1970 %>%
  mutate(Tid=parse_integer(Tid)) %>%
  rename(var=ContentsCode, tid=Tid, verdi=value)

# Konverterer dataframen til en tibble og lager en ny gruppe som jeg kaller for df_1970_ti
df_1970_tibble <- as_tibble(df_1970)

# Legger til "DNBP" og "DMEMOBNP" og fester dette til df3_wide, og definerer deres verdier
df_1970_tibble <- df_1970_tibble %>%
  pivot_wider(names_from = var, values_from =verdi)

#Legger til verdiene til variabelene året før i nye kolonner.
df_1970_tibble <- df_1970_tibble %>%
  mutate(LBNP = lag(BNP,n=1L)) %>%
  mutate(LMEMOBNP = lag(MEMOBNP, n=1L)) %>%
  relocate(LBNP, .before = MEMOBNP)

# Legger til endringer i BNP og MEMOBNP i prosent, kallt PROSENT_BNP og PROSENT_MEMOBNP
df_1970_tibble <- df_1970_tibble %>%
  mutate(PROSENT_BNP = (100*(BNP - lag(BNP, n=1L))/(lag(MEMOBNP, n=1L)))) %>%
  mutate(PROSENT_MEMOBNP = (100*(MEMOBNP - lag(MEMOBNP, n=1L))/(lag(MEMOBNP, n=1L)))) %>%
  relocate(PROSENT_BNP, .before = PROSENT_MEMOBNP)

# Flyyter alle variablene til en kategori.
df_1970_tibble <- df_1970_tibble %>%
  pivot_longer(c("BNP", "LBNP", "PROSENT_BNP", "MEMOBNP", "LMEMOBNP", "PROSENT_MEMOBNP"),
```

```r
# Velger kun endringer i prosent.
df_1970_tibble <- df_1970_tibble %>%
  filter(var == c("PROSENT_BNP", "PROSENT_MEMOBNP"))

# Lager grafenn.
df_1970_tibble %>%
  mutate(verdi=verdi/100) %>%
  ggplot(aes(x=tid, y=verdi, col=var)) +
  geom_line() +
  geom_smooth() +
  labs(title = "BNP per person i Norge fra 1970 til 2021",
       x = "År",
       y = "BNP per innbygger med prosentvis endring")
```

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

```
Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
Warning: Removed 2 row(s) containing missing values (geom_path).
```