Case 1

19

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   1.0.10
v tidyr   1.2.1      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2
-- 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

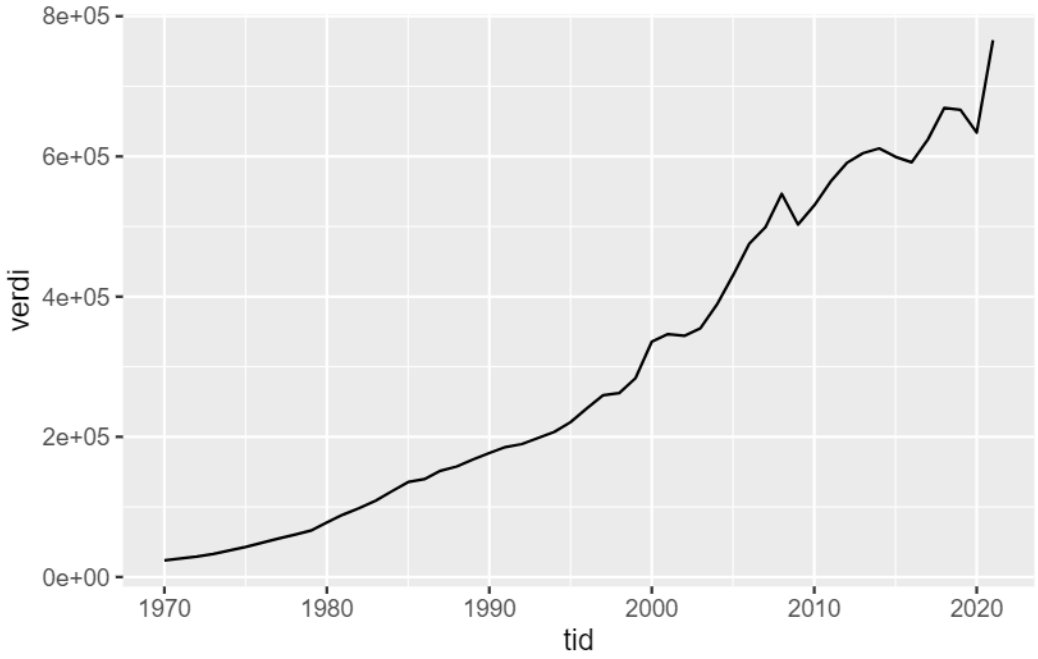
Skriv kode som lagrer dataene som én tibble med anstendige variabelnavn og årstall som heltall. Fremover bruker jeg "var", "tid", og "verdi" for "statistikkvariabel", "Tid", og "value".

```r
# Oppgave Ia løses her
df <- df[2] $dataset
df$Tid <- as.integer(df$Tid)
df$value <- as.numeric(df$value)
df <- tibble(df)
df <- rename(df, var = ContentsCode)
df <- rename(df, tid = Tid)
df <- rename(df, verdi = value)
```

Oppgave Ib: Lag en figur

Følgende kode skaper en enkel figur.

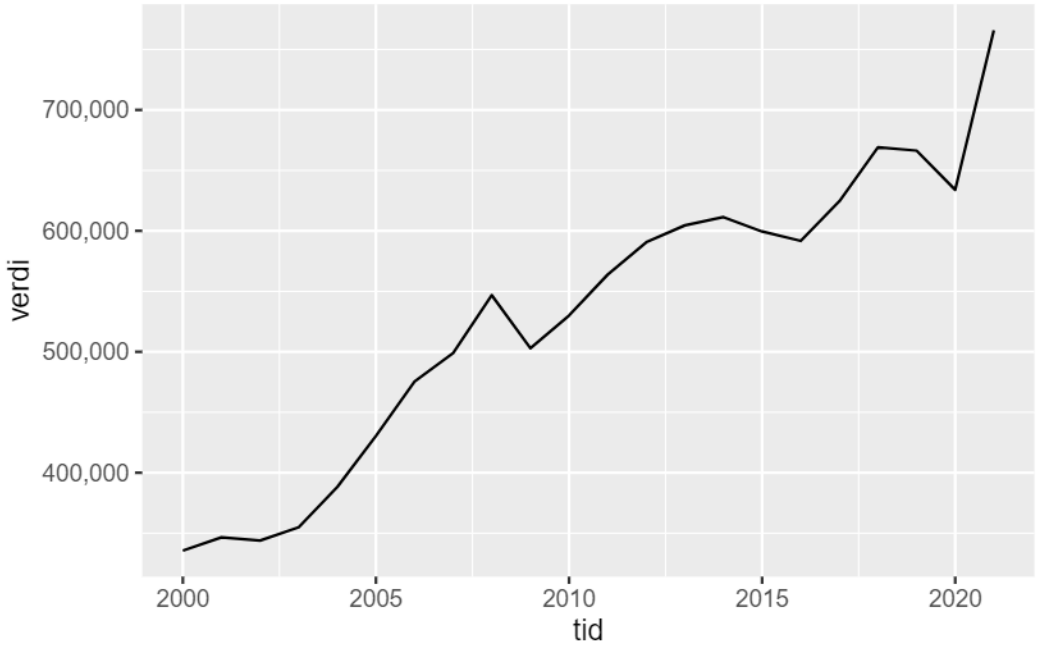
```r
df %>%
  filter(var == "BNP") %>%
  ggplot(aes(x=tid,y=verdi)) +
  geom_line()
```



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?

```
# Oppgave Ib løses her
df %>%
  filter(var == "BNP") %>%
  filter(tid %in% c(2000:2021)) %>%
  ggplot(aes(x=tid,y=verdi)) +
  scale_y_continuous(labels=scales::comma) +
  geom_line()
```



Svar på spørsmålet

Grafen viser utviklingen av BNP over en periode på 21 år. I løpet av denne tidsperioden ser vi en klar økning. BNP i løpende priser er de faktiske prisene uten hensyn til inflasjon og endret pengeverdi. Faste priser tar hensyn til disse og gir et mye klarere bilde på landets reelle verdiskapning. Derfor er også faste priser også faste priser mest relevante når du ønsker å se landets reelle verdiøkning.

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

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

Svar på spørsmålet

pivot_wider() legger til flere kolonner og dermed minsker antallet rader. I dette tilfellet lagde den egne kolonner for både BNP og MEMOBNP.

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:

df_wide <- df_wide %>%

relocate("LBNP", .before = "MEMOBNP")

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

# legger variablene i rekkefølge

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

df_wide
```

```
# A tibble: 52 x 5
     tid   BNP  LBNP MEMOBNP LMEMOBNP
   <int> <dbl> <dbl>   <dbl>    <dbl>
 1  1970 23616    NA  214756       NA
 2  1971 26363 23616  225352   214756
 3  1972 29078 26363  235557   225352
 4  1973 32805 29078  244518   235557
 5  1974 37734 32805  252539   244518
 6  1975 42884 37734  263586   252539
 7  1976 48711 42884  277636   263586
 8  1977 54652 48711  287968   277636
 9  1978 60091 54652  297971   287968
10  1979 66069 60091  309942   297971
# ... with 42 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`.

```r
# Besvar oppgave IIb her
df_wide <- df_wide %>%
  mutate(DBNP = BNP-LBNP) %>%
  mutate(DMEMOBNP = MEMOBNP-LMEMOBNP)

#flytter den til riktig plass
df_wide <- df_wide %>%
  relocate("DBNP", .before = "MEMOBNP")
```

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
df_long <- df_wide %>%
  pivot_longer(cols = everything(), names_to = "var", values_to = "verdi") %>%
  arrange(var)
df_long
```

```
# A tibble: 364 x 2
   var   verdi
   <chr> <dbl>
 1 BNP   23616
 2 BNP   26363
 3 BNP   29078
 4 BNP   32805
 5 BNP   37734
 6 BNP   42884
 7 BNP   48711
 8 BNP   54652
 9 BNP   60091
10 BNP   66069
# ... with 354 more rows
```

```
#Alternativ besvarelse på oppgave IIc

df_long <- df_wide %>%
  pivot_longer(!tid, names_to = "var", values_to = "verdi") %>%
  arrange(var)
df_long
```

```
# A tibble: 312 x 3
     tid var   verdi
   <int> <chr> <dbl>
 1  1970 BNP   23616
 2  1971 BNP   26363
```

```
 3  1972 BNP   29078
 4  1973 BNP   32805
 5  1974 BNP   37734
 6  1975 BNP   42884
 7  1976 BNP   48711
 8  1977 BNP   54652
 9  1978 BNP   60091
10  1979 BNP   66069
# ... with 302 more rows
```

Oppgave IId: Figur med vekst

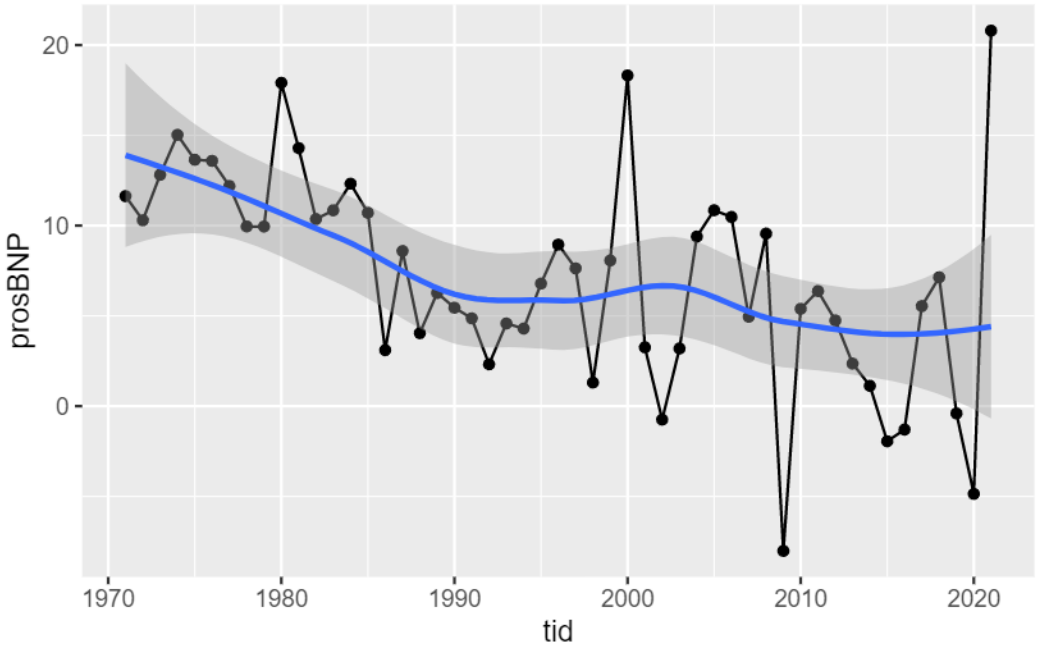
Lag en pen figur med prosentvis vekst i nominelt og reelt BNP per person fra 1970 til 2021. Finnes det observasjoner med negativ vekst i reell BNP? Hva skyldes dette?

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

df_wide %>%
mutate(prosBNP = 100*(BNP - lag(BNP))/lag(BNP)) %>%
filter(tid >=1971) %>%
ggplot(aes(x=tid, y=prosBNP)) +
geom_line() +
geom_point() +
geom_smooth()
```

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



Svar på spørsmålet

Vi finner 4 observbvasjoner med negativ vekst i BNP.Grunner til disse er som ofte knyttet til store globale kriser. Som eksempel, regresjonen i 2008 skyldes den globale finanskrisen og den i 2020 grunnet koronapandemien.