---
title: "Mappeoppgave 3 1005"
output: html_notebook

Nyregistreringer og konkurser blant norske foretak

Introduksjon

Om oppgaven

[Lenke til oppgavetekst]
(https://uit.instructure.com/courses/33631/files/folder/Mappeoppgaver?
preview=2918081)

Denne oppgaven består i å skrape data angående nyregistreringer og konkurser (der spesifikt konkurs-åpning)fra [Brønnøysundsregistret]
(https://w2.brreg.no/kunngjoring/) mellom 1.1.2017 og dagens dato (vis i r dagens dato når det åpnes). Det skal lages en tabell som inneholder: - Foretaksnavn - Org.nr. (9 siffer)(der det skal fjernes mellomrom og privatpersoner med fødselsdato (6 siffer)) - Registreringsdato (dd.mm.åååå) - Kunngjøringstype (Nyregistrering eller Konkurs/Tvangsavvikling)

Deretter skal det lages en tidsserie som viser endringen i nyåpninger mot konkurser per måned enten per fylke eller hele landet. Og så lages en selvforklarende figur som viser endringen i nyåpninger mot konkurser før, under og etter COVID-19.

Instrukser til oppgaven

- Skal bruke purrr-pakken i r map()-funksjonen til å skrape dataene
- Rensk dataen; fjern duplikater og manglende verdier, og kategoriser dataene etter fylke
- Benytte dato til å aggregere månedsdata for hele landet og per fylke
- Lag en tidsserie for differansen mellom nyregistreringer og konkurser, ved navn "netto"
- Lag en selvforklarende figur som viser "netto" foretak per måned før, under og etter korona fra januar 2017 til februar 2024. Mars 2020 er første "pandemi-måned" og Mai 2023 er første "etter-pandemi-måned". Dermed blir tidsseriene seende slik ut:
- "Før-pandemi": 01.01.2017 - 31.01.2020
- "Under-pandemi":01.02.2020 - 30.04.2023
- "Etter-pandemi":01.05.2023 - 29.02.2024

Gjennomføring av oppgaven

Laste inn data

Siden hvert søk kun gir rom for maks 5000 verdier per søk (og det er langt flere verdier enn det i det tidsrommet jeg skal se på) starter jeg med å lage en liste over datoer som jeg deretter skal putte inn i url'en slik at søkene blir gjort etterhvert som r itererer over listen. Jeg hanter også opp en liste over fylker fra 2024 fra wikipedia som jeg bruker som utgangspunkt til å separere etter fylker. Jeg gjør dette fordi strukturen i nettsiden som skal scrapes gjør at det er mye enklere og mer effektivt å separere etter fylker før jeg henter inn dataene enn å prøve å separere etter de er lastet inn.

Deretter lager jeg en funksjon som tar utgangspunkt i url'en for søk etter nyregistreringer og erstatter "nummer" , "start" og "slutt" med fylkesnummer (hentet fra listen jeg har scrapet fra wikipedia), og datoene fra listen jeg allerede har generert. Dermed får jeg alle url'ene til alle nyregistreringene i

hele landet per måned. Da en del av oppgaven spesifikt er å bruke map() fra purrr-pakken bruker jeg denne her.

(Dette er foreløpig kun nødvendig for nyregistreringer da antall konkurser (av typen konkurs-åpning) for øyeblikket ligger under 5000 i hele tidsperioden, men jeg har også her valgt å separere etter fylke for enkelhets skyld)

Siden en del av oppgaven er å lage en tabell med de nevnte kolonner bruker jeg html-scraping til å hente den relevante informasjonen mens jeg itererer over url'ene. Jeg gjør dette ved først å velge fylkesnummer og så mer fylkesnummer så itererer jeg over datolistene, dette fører til en liste med 17 lister (en for hvert fylke) som igjen har 88 URL'er (en for hver måned per fylke, inkludert Jan Mayen). Her legger jeg også til hvilket fylke foretaket er registrert i siden det er det jeg forstår med å kategorisere dataene per fylke. Jeg lager her en ekstra kolonne som definerer om det er før, under eller etter pandemien etter de kriterier foreskrevet i oppgaven. Jeg fjerner også her alle registreringer med organisasjonsnummer under 9 siffer, for deretter å sjekke etter duplikater.

Nå sitter jeg igjen med to dataframes, en for konkurser og én for nyåpninger, som skal renskes.

Rensking av data

Sjekker etter duplikater og fjerner alle utenom den første av hver type.

Deretter teller jeg nyregistreringer og konkurser per måned for hele landet og per fylke og lagrer dette i en ny dataframe, separerer ut måned og år, og oppretter "netto"-kolonnen ved å trekke fra konkurser fra nyåpninger.

Grafisk presentasjon

Jeg lager så to tidsserier, en for hele landet og en for per fylke.

Ut fra dataframen lager jeg så en "netto"-tidsserie for hele landet over bedriftene, og så en ny figur som viser "netto" foretak per måned før, under og etter korona.

Gjennomføring av oppgaven

Laste inn data

```{r}
rm = ls()
install.packages(c("janitor", "quantmod")) # Laster inn og installerer pakker
library(purrr)
library(tidyverse)
library(rvest)
library(janitor)
library(lubridate)
library(quantmod)
library(vctrs)
```

```{r}
# Laster inn url'ene til de to ulike søkene
konkurs_url <- "https://w2.brreg.no/kunngjoring/kombisok.jsp?
datoFra=01.01.2017&datoTil=28.03.2024&id_region=000&id_fylke=nummer&id_kommune=-
+-+-&id_niva1=51&id_niva2=56&id_bransje1=0"
nyregistreringer_url <- "https://w2.brreg.no/kunngjoring/kombisok.jsp?
datoFra=start&datoTil=end&id_region=000&id_fylke=nummer&id_kommune=-+-+-
```

```
&id_niva1=2&id_bransje1=0"
```

```{r}
# Liste over datopar
list1 <- format(seq(as.Date("2017-01-01"), as.Date("2024-04-01"), by="months"),
format="%d.%m.%Y") #Startdato
list2 <- format(seq(as.Date("2017-02-01"), as.Date("2024-05-01"), by="months"),
format="%d.%m.%Y") #Enddato
df <- do.call(rbind, Map(data.frame, start=list1, end=list2))#Legger listene
sammen
```

```{r}
# Brukt KI for å streamline koden
# For å kunne iterere over fylkesnumre, henter jeg en oversikt over fylker fra
snl
snl_url <- "https://snl.no/fylkesnummer"

# Skraper data fra tabellen
fylker_table <- read_html(snl_url) %>%
  html_table() %>%
  .[[1]]

# Siden strukturen på tabellen er ganske enkel kan jeg bare hente data og legge
den i rows
fylker <- fylker_table %>%
  setNames(nm = c("NR", "navn")) %>%
  mutate(navn = str_replace(navn, "\\*", "")) # Svalbard og Jan Mayen hadde en
ekstra liten asterisk, så jeg fjerner den for syns skyld

fylker <- as_tibble(fylker) # Gjør om til tibble

fylker
```

Nyregistreringer

```{r}
# Her er det brukt KI for å streamline koden
# Funksjon som generer alle URL'ene jeg behøver, fordelt på datoer og
fylkeskoder
generate_urls_for_fylke <- function(fylke_code, date_pairs) {
  map2(date_pairs$start,
       date_pairs$end, ~ paste0("https://w2.brreg.no/kunngjoring/kombisok.jsp?
datoFra=",.x, "&datoTil=", .y,"&id_region=000&id_fylke=", fylke_code,
"&id_kommune=-+-+-&id_niva1=2&id_bransje1=0"))
}

# Lager en liste med lister over alle URL'ene fordelt på fylker
nyåpning_liste_url_comp <- map(fylker[[1]], ~ generate_urls_for_fylke(.x, df))

```

```{r}
nyåpning_liste_all <- list()
i = 0

# Lager en nested df som lagrer de ulike fylkene og månedene inni hverandre
```

```
# For hvert fylke så laster jeg ned HTML-info
for (i in seq(1, length(fylker[[1]]), by = 1)){
html_content <- lapply(nyåpning_liste_url_comp[[i]], read_html)

# Vasker og henter fram det jeg er interessert i
nyåpninger <- lapply(html_content, function(content) {
  konk_test_p <- html_elements(content, css = ".normal-br-link p")
  df2 <- html_text(konk_test_p)
  df2 <- df2[12:length(df2)]

  # På grunn av hvordan jeg får ut dataen så må jeg konstruere en df der jeg
separerer listen inn og beholder de delene jeg er interessert i
  extracted_info <- map_df(seq(1, length(df2), by = 7), function(j) {
    tibble(
      bedrnavn = df2[j],
      bedrnr = gsub(" ", "", df2[j + 1]),  # Fjerner alle bedrnr under 9
      opprdato = as.Date(df2[j + 2], format = "%d. %m. %Y"),  # Konverter til
dato
      type = df2[j + 4],
      fylke = rep(fylker[[2]][i], length.out = 1) #Legger også til fylket
    )
  })


  # Legger inn ny kolonne som sier om det skjedde før, under, eller etter COVID
  extracted_info <- extracted_info %>%
    filter(nchar(as.character(bedrnr)) >= 9) %>%
    mutate(COVID = case_when(
      opprdato > as.Date("2023-05-01") ~ "Etter",
      opprdato > as.Date("2020-02-01") & opprdato <= as.Date("2023-04-30") ~
"Under",
      TRUE ~ "Før"
    ))
})
#Lagrer listene i en samlet liste
nyåpning_liste_all[[i]] <- nyåpninger
}

# Kombinerer alt i ett
nyåpninger_hele_land <- bind_rows(nyåpning_liste_all)
```

Konkurser

```{r}
# Her er det brukt KI for å streamline koden
# Funksjon som generer alle URL'ene jeg behøver, fordelt på datoer og
fylkeskoder
generate_urls_for_fylke <- function(fylke_code, date_pairs) {
  map2(date_pairs$start,
       date_pairs$end, ~ paste0("https://w2.brreg.no/kunngjoring/kombisok.jsp?
datoFra=",.x, "&datoTil=", .y,"&id_region=000&id_fylke=", fylke_code,
"&id_kommune=-+-+-&id_niva1=51&id_niva2=-+-+-&id_bransje1=0"))
}

# Lager en liste med lister over alle URL'ene fordelt på fylker
konkurs_liste <- map(fylker[[1]], ~ generate_urls_for_fylke(.x, df))

```

```{r}
```

```
konkurser <- list()
i = 0

# Lager en nested df som lagrer de ulike fylkene og månedene inni hverandre


# For hvert fylke så laster jeg ned HTML-info
for (i in seq(1, length(fylker[[1]]), by = 1)){
html_content <- lapply(konkurs_liste[[i]], read_html)

# Vasker og henter fram det jeg er interessert i
konkurs_OSL <- lapply(html_content, function(content) {
  konk_test_p <- html_elements(content, css = ".normal-br-link p")
  df2 <- html_text(konk_test_p)
  df2 <- df2[12:length(df2)]

  # På grunn av hvordan jeg får ut dataen så må jeg konstruere en df der jeg
separerer listen inn og beholder de delene jeg er interessert i
  extracted_info <- map_df(seq(1, length(df2), by = 7), function(j) {
    tibble(
      bedrnavn = df2[j],
      bedrnr = gsub(" ", "", df2[j + 1]),  # Fjerner alle bedrnr under 9
      konkurs_OSL <- add_column(konkurs_OSL, x = "konkurs"),
      opprdato = as.Date(df2[j + 2], format = "%d. %m. %Y"),  # Konverter til
dato
      type = df2[j + 4],
      fylke = rep(fylker[[2]][i]) #Legger også til fylket
    )
  })


  # Legger inn ny kolonne som sier om det skjedde før, under, eller etter COVID
  extracted_info %>%
    filter(nchar(as.character(bedrnr)) >= 9) %>%
    mutate(COVID = case_when(
      opprdato > as.Date("2023-05-01") ~ "Etter",
      opprdato > as.Date("2020-02-01") & opprdato <= as.Date("2023-04-30") ~
"Under",
      TRUE ~ "Før"
    ))
})
#Lagrer listene i en samlet liste
konkurser_liste_all[[i]] <- konkurser_OSL
}

# Kombinerer alt i ett
konkurser_hele_land <- bind_rows(konkurser_liste_all)
```

```{r}
konkurser_hele_land
```

Rensking av data

- Fjerne orgnr under 9 siffer (allerede fikset når jeg lastet inn dataen)
- Sjekk duplikatverdier og fjern
- Lag ny liste som teller antall nyregistreringer og konkurser

Konkurser

```{r}
# Finner duplikater og fjerner
konkurser_hele_land %>% group_by(bedrnr) %>% filter(n()>1) #Finner duplikater

konkurser_hele_land <- konkurser_hele_land %>%
  arrange(bedrnavn, opprdato) %>%  # Make sure data is sorted by company_name
and then by date
  group_by(bedrnr) %>%
  slice_head(n = 1) %>%  # Keep the first row of each group, which is the
earliest due to the arrange() call
  ungroup()

konkurser_hele_land %>% group_by(bedrnr) %>% filter(n()>1) #tester
```

Nyåpninger

```{r}
# Finner duplikater og fjerner
nyåpninger_hele_land %>% group_by(bedrnr) %>% filter(n()>1) #Finner duplikater

nyåpninger_hele_land <- nyåpninger_hele_land %>%
  arrange(bedrnavn, opprdato) %>%  # Make sure data is sorted by company_name
and then by date
  group_by(bedrnr) %>%
  slice_head(n = 1) %>%  # Keep the first row of each group, which is the
earliest due to the arrange() call
  ungroup()

nyåpninger_hele_land %>% group_by(bedrnr) %>% filter(n()>1) #tester
```

```{r}
konkurser_hele_land
```

```{r}
# siden jeg skal legge sammen konkurser og nyåpninger lager jeg en kolonne på
hver som denoterer hvilke dataframe de er en del av
konkurser_hele_land <- add_column(konkurser_hele_land, x = "Konkurs")
```

```{r}
nyåpninger_hele_land <- add_column(nyåpninger_hele_land, x = "Nyåpning")
```

```{r}
df <- bind_rows(nyåpninger_hele_land, konkurser_hele_land)
```

```{r}
df
```

Setter sammen dataframene og endrer dato til bare måned og år

```{r}
# Change the date to year and month
df <- df %>%
  mutate(år = year(dato),
         month = month(dato))
```

```

```{r}
netto_df_per_fylke <- df %>%
  group_by(fylke, COVID, år, month, x) %>%
  summarise(n = n(), .groups = "drop")
netto_df_per_fylke <- netto_df_per_fylke %>%
  pivot_wider(names_from = x, values_from = n)
netto_df_per_fylke <- netto_df_per_fylke %>%
   mutate(across(where(is.numeric), ~replace_na(., 0)))
netto_df_per_fylke <- netto_df_per_fylke %>% mutate(netto = nyåpning - konkurs)
```

```{r}
netto_df_hele_land <- df %>%
  group_by(COVID, år, month, x) %>%
  summarise(n = n(), .groups = "drop")
netto_df_hele_land <- netto_df_hele_land %>%
    pivot_wider(names_from = x, values_from = n)
netto_df_hele_land <- netto_df_hele_land %>% mutate(netto = nyåpning - konkurs)
```

Grafisk presentasjon

```{r}
# Tidsserie land

netto_df_hele_land1 <- netto_df_hele_land %>%
  mutate(date = make_date(år, month, 1))
```

```{r}
# tidsserie land del 2
ggplot(netto_df_hele_land1, aes(x = date, y = netto)) +
  geom_line() +  # Draw lines connecting the points
  geom_point() +  # Draw points at each data point
  theme_minimal() +  # Use a minimal theme for a cleaner look
  labs(
    title = "Time Series of Differences",
    x = "Date",
    y = "Difference"
  ) +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 year")+
  facet_wrap(~COVID)
```

```{r}
# tidsserie land del 2
ggplot(netto_df_hele_land1, aes(x = date, y = netto)) +
  geom_line() +  # Draw lines connecting the points
  geom_point() +  # Draw points at each data point
  theme_minimal() +  # Use a minimal theme for a cleaner look
  labs(
    title = "Time Series of Differences",
    x = "Date",
    y = "Difference"
  ) +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 year")
```

```{r}

```r
# Tidsserie fylke
netto_df_per_fylke <- netto_df_per_fylke %>%
  mutate(date = make_date(år, month, 1))
ggplot(netto_df_per_fylke, aes(x = date, y = netto)) +
  geom_line(aes(color = fylke)) +  # Draw lines connecting the points
  geom_point() +  # Draw points at each data point
  theme_minimal() +  # Use a minimal theme for a cleaner look
  labs(
    title = "Tidsserie av netto nyåpninger i Norge 2017-2023",
    x = "Dato",
    y = "Netto nyåpninger (trukket fra konkurser)",
  ) +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year")+
  facet_wrap(~fylke, ncol = 1)+ theme(legend.position = "none")
```

```{r}
ggplot(netto_df_hele_land1, aes(x = date, y = netto, group = COVID, color =
COVID)) +
  geom_line() +
  geom_point() +  # Adds points to each data entry for clarity
  scale_color_manual(values = c("Før" = "blue", "Under" = "red", "Etter" =
"green")) +  # Customize colors
  geom_smooth(method = "lm", se = FALSE, aes(group = COVID)) +
  theme_minimal() +
  labs(
    title = "Net Change Over Time by COVID Period",
    x = "Date",
    y = "Net Change",
    color = "COVID Period"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5),  # Center the title
    legend.position = "bottom"  # Place legend at the bottom
  ) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")  # Customize date
breaks and labels
```

Bruk av KI

Jeg har tatt i bruk KI for å effektivt omgjøre nestede for-loops til map()-funksjoner suksessfullt.

Prøvde også å få hjelp til å effektivisere koden som scrapet data, med mindre hell.