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_C4_H22.qmd` og lastes opp på deres GitHub-side. Hvis du har kandidatnummer 43, så vil filen hete `43_SOK1004_C4_H22.qmd`. Påse at koden kjører og at dere kan eksportere besvarelsen til pdf. Lever så lenken til GitHub-repositoriumet i Canvas.

Bakgrunn, læringsmål

Innovasjon er en kilde til økonomisk vekst. I denne oppgaven skal vi se undersøke hva som kjennetegner bedriftene som bruker ressurser på forskning og utvikling (FoU). Dere vil undersøke FoU-kostnader i bedriftene fordelt på næring, antall ansatte, og utgiftskategori. Gjennom arbeidet vil dere repetere på innhold fra tidligere oppgaver og øve på å presentere fordelinger av data med flere nivå av kategoriske egenskaper.

Last inn pakker

```
# output | false
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(rjstat)
```

```
Attaching package: 'rjstat'

The following object is masked from 'package:dplyr':

    id
```

```
library(gdata)
```

```
gdata: Unable to locate valid perl interpreter
gdata:
gdata: read.xls() will be unable to read Excel XLS and XLSX files
gdata: unless the 'perl=' argument is used to specify the location of a
```

```
gdata: valid perl intrpreter.
gdata:
gdata: (To avoid display of this message in the future, please ensure
gdata: perl is installed and available on the executable search path.)
gdata: Unable to load perl libaries needed by read.xls()
gdata: to support 'XLX' (Excel 97-2004) files.

gdata: Unable to load perl libaries needed by read.xls()
gdata: to support 'XLSX' (Excel 2007+) files.

gdata: Run the function 'installXLSXsupport()'
gdata: to automatically download and install the perl
gdata: libaries needed to support Excel XLS and XLSX formats.

Attaching package: 'gdata'

The following objects are masked from 'package:dplyr':

    combine, first, last

The following object is masked from 'package:purrr':

    keep

The following object is masked from 'package:stats':

    nobs

The following object is masked from 'package:utils':

    object.size

The following object is masked from 'package:base':

    startsWith
```

```
library(httr)
```

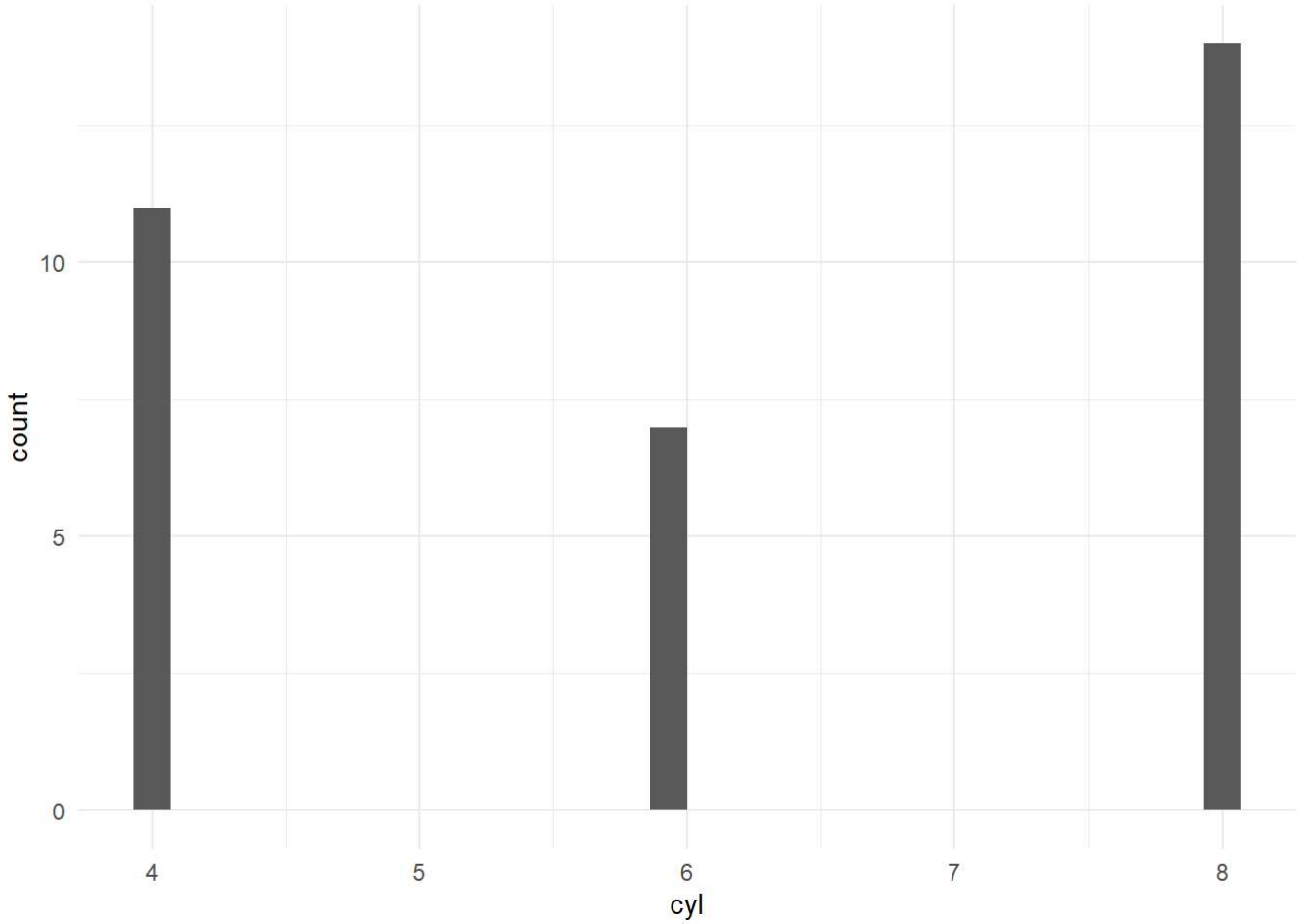
Oppgave I: Introduksjon til histogram

Et histogram eller frekvensfordeling er en figur som viser hvor ofte forskjellige verdier oppstår i et datasett. Frekvensfordelinger spiller en grunnleggende rolle i statistisk teori og modeller. Det er avgjørende å forstå de godt. En kort innføring følger.

La oss se på et eksempel. I datasettet `mtcars` viser variabelen `cyl` antall sylindere i motorene til kjøretøyene i utvalget.

```
data(mtcars)
mtcars %>%
  ggplot(aes(cyl)) +
  geom_histogram() +
  theme_minimal()
```

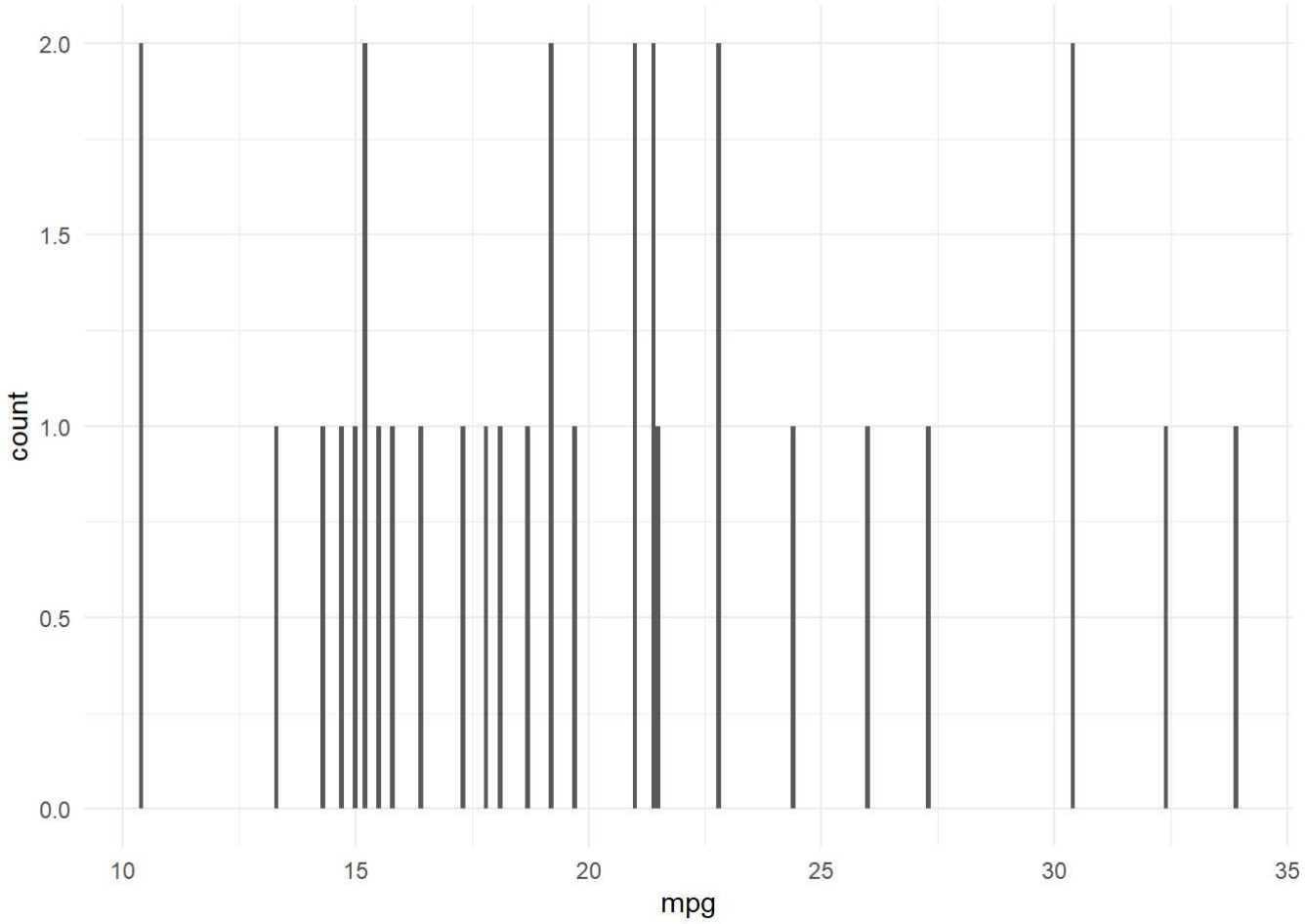
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Verdiene av variabelen er er gitt ved den horisontale aksen, antall observasjoner på den vertikale aksen. Vi ser at det er 11, 7, og 14 biler med henholdsvis 4, 6, og 8 sylindere.

La oss betrakte et eksempel til. Variabelen `mpg` i `mtcars` måler gjennomsnittlig drivstoffforbruk i uanstendige engelske enheter. Variabelen er målt med ett desimal i presisjon.

```
data(mtcars)
mtcars %>%
  ggplot(aes(mpg)) +
  geom_histogram(binwidth=0.1) +
  theme_minimal()
```



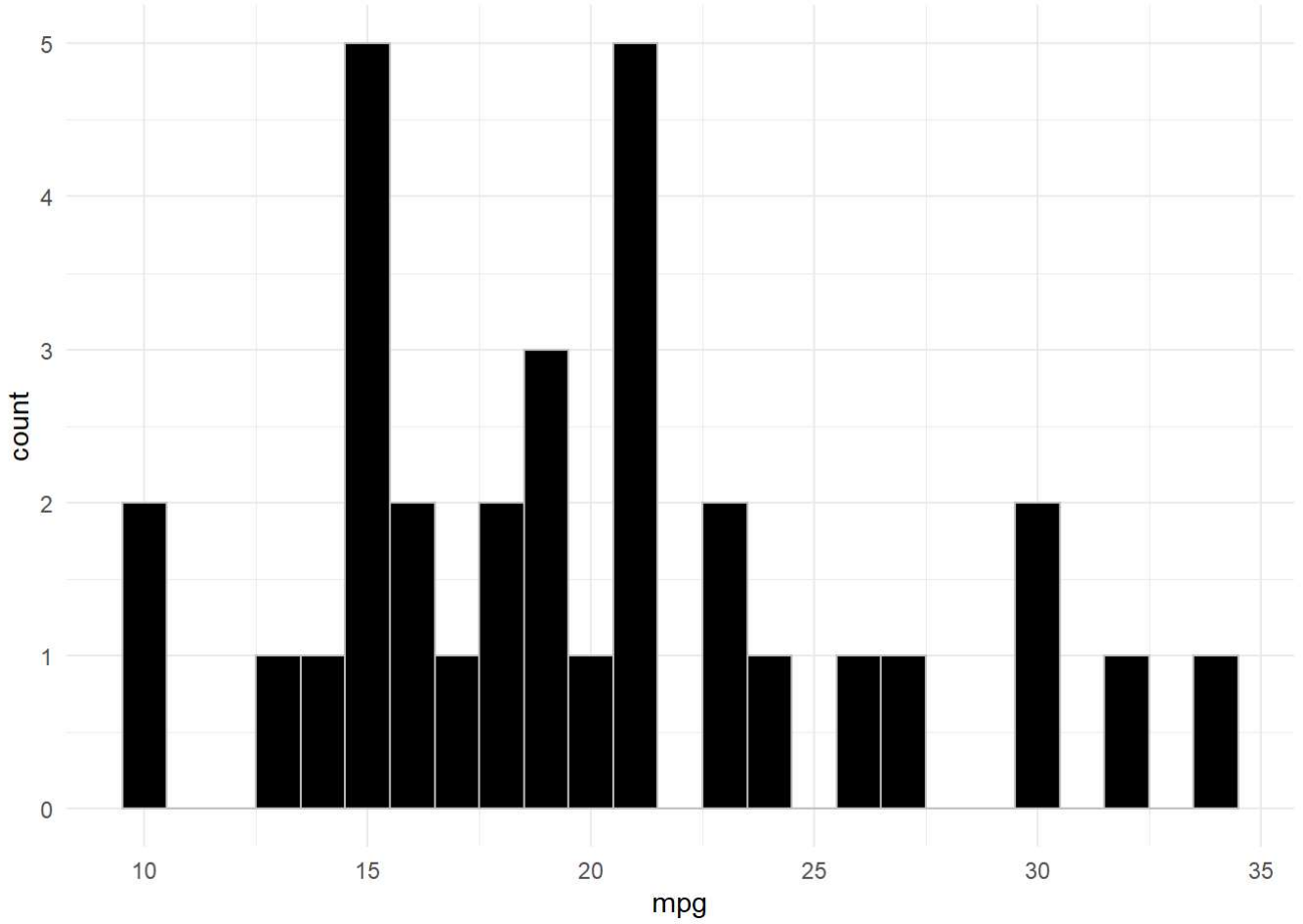
Datasettet inneholder mange unike verdier, hvilket gir utslag i et flatt histogram, noe som er lite informativt. Løsningen da er å gruppere verdier som ligger i nærheten av hverandre. Kommandoen Kommandoen `binwidth` i `geom_histogram()` bestemmer bredden av intervallene som blir slått sammen. Kan du forklare hvorfor alle unike verdier blir telt ved å å bruke `binwidth = 0.1`?

Svar:

Binwidth = 0.1 inkluderer alle de unike verdiene fordi verdiene har en desimal. Derfor vil for eksempel 0.1 og 0.2 bli telt individuelt.

Eksperimenter med forskjellige verdier for `binwidth` og forklar hva som kjennetegner en god verdi.

```
# løs oppgave I her
data(mtcars)
mtcars %>%
  ggplot(aes(mpg)) +
  geom_histogram(binwidth=1, fill="black", col="grey") +
  theme_minimal()
```



Svar:

En god verdi må en god mengde data, men også ikke være så lav at man får for få grupper.

Etter noe ekspremintering har jeg kommet frem til at 1 er en god verdi.

Jeg har også lagt inn litt farger som fungerer for å skille søylene fra hverandre, slik at det blir lettere å se de forskjellige gruppene.

Oppgave II: Last ned og rydd i data

Vi skal nå undersøkene dataene i Tabell 07967: Kostnader til egenutført FoU-aktivitet i næringslivet, etter næring (SN2007) og sysselsettingsgruppe (mill. kr) 2007 - 2020 SSB. Dere skal laster de ned ved hjelp av API. Se brukerveiledningen her.

Bruk en JSON-spørring til å laste ned alle statistikkvariable for alle år, næringer, og sysselsettingsgrupper med 10-19, 20-49, 50-99, 100-199, 200 - 499, og 500 eller flere ansatte. Lagre FoU-kostnader i milliarder kroner. Sørg for at alle variabler har riktig format, og gi de gjerne enklere navn og verdier der det passer.

Hint. Bruk lenken til SSB for å hente riktig JSON-spørring og tilpass koden fra case 3.

```
# besvar oppgave II her

#JSON spørring fra SSB

url <- "https://data.ssb.no/api/v0/no/table/07967"

query <- '{
  "query": [
    {
      "code": "NACE2007",
      "selection": {
        "filter": "item",
        "values": [
          "A-N",
          "C",
          "G-N",
          "A-B_D-F"
        ]
      }
    },
    {
      "code": "SyssGrp",
      "selection": {
        "filter": "item",
        "values": [
          "10-19",
          "20-49",
          "50-99",
          "100-199",
          "200-499",
          "500+"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "FoUKostnader",
          "FoUDriftskostnader",
          "Lonnskostnader",
          "AndreDriftsKost",
          "FoUInvesteringer",
          "KostInnleidPers"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
```

```
          "2007",
          "2008",
          "2009",
          "2010",
          "2011",
          "2012",
          "2013",
          "2014",
          "2015",
          "2016",
          "2017",
          "2018",
          "2019",
          "2020",
          "2021"
        ]
      }
    }
  ],
  "response": {
    "format": "json-stat2"
  }
}'

# Kode rappet fra Case 3 for å lage df

hent_indeks.tmp <- url %>%
  POST(body = query, encode = "json")

df <-  hent_indeks.tmp %>%
  content("text") %>%
  fromJSONstat() %>%
  as_tibble()

# Del på 1000 for å gjøre millioner til milliarder, og endre navn.
df <- df %>%
  mutate(value = value/1000) %>%
  rename(næring = "næring (SN2007)", gruppe = sysselsettingsgruppe, variabel = statistikkvaria
```

Oppgave III: Undersøk fordelingen

Vi begrenser analysen til bedrifter med minst 20 ansatte og tall fra 2015 - 2020. Lag en figur som illustrerer fordelingen av totale FoU-kostnader fordelt på type næring (industri, tjenesteyting, andre) og antall ansatte i bedriften (20-49, 50-99, 100-199, 200-499, 500 og over). Tidsdimensjonen er ikke vesentlig, så bruk gjerne histogram.

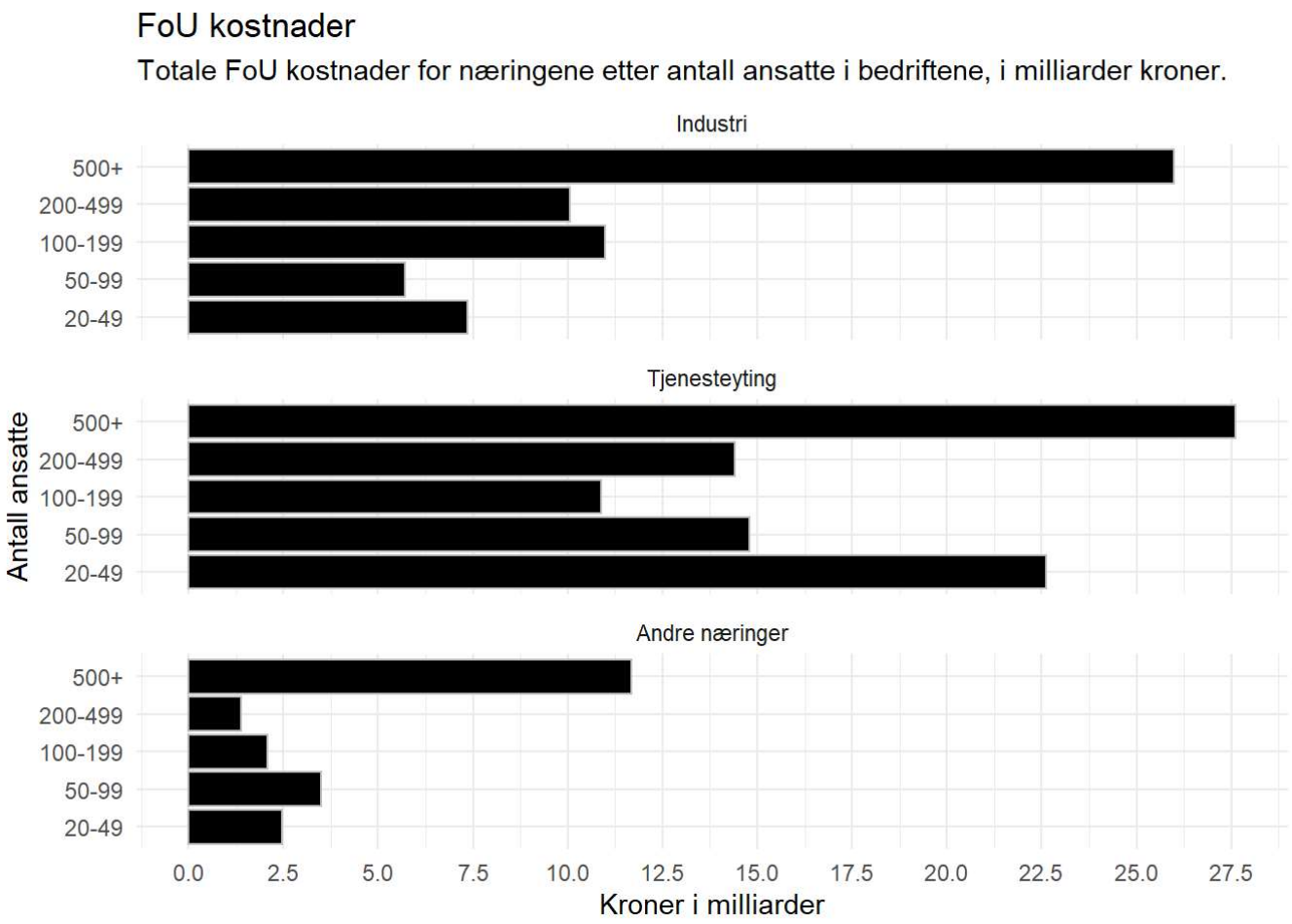
Merknad. Utfordringen med denne oppgaven er at fordelingene er betinget på verdien av to variable. Kommandoen `facet_grid()` kan være nyttig til å slå sammen flere figurer på en ryddig måte.

```
df_Tot <- df %>%
```

```
  filter(variabel == "FoU-kostnader i alt")

#Endre navn på gruppene
df_Tot["gruppe"][df_Tot["gruppe"] == "10-19 sysselsatte"] <- "10-19"
df_Tot["gruppe"][df_Tot["gruppe"] == "20-49 sysselsatte"] <- "20-49"
df_Tot["gruppe"][df_Tot["gruppe"] == "50-99 sysselsatte"] <- "50-99"
df_Tot["gruppe"][df_Tot["gruppe"] == "100-199 sysselsatte"] <- "100-199"
df_Tot["gruppe"][df_Tot["gruppe"] == "200-499 sysselsatte"] <- "200-499"
df_Tot["gruppe"][df_Tot["gruppe"] == "500 sysselsatte og over"] <- "500+"



#Fjern "Alle næringer", bedrifter med 10-19 ansatte og uønskede år
df_Tot<-df_Tot[!(df_Tot$næring=="Alle næringer" | df_Tot$gruppe=="10-19" | df_Tot$år=="2007" |


df_tot <- df_Tot %>%
  group_by(næring, gruppe) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring'. You can override using the
`.groups` argument.
```

```
# besvar oppgave III he



df_tot %>%

#Manuell sortering så bedriftstørrelse kommer i stugende rekkefølge
  mutate(gruppe = fct_relevel(gruppe, "20-49", "50-99", "100-199", "200-499", "500+")) %>%

  ggplot(aes(tot, gruppe)) +
  geom_col(fill="black", col="grey") +
  scale_x_continuous(breaks=c(0, 2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20, 22.5, 25, 27.5))+

  labs(y = "Antall ansatte", x = "Kroner i milliarder", title = "FoU kostnader",
subtitle = "Totale FoU kostnader for næringene etter antall ansatte i bedriftene, i milliarder



  facet_wrap(~ factor (næring, (levels=c("Industri", "Tjenesteyting", "Andre næringer"))), nro
             #  lag tre paneler
             #  titler
                      labeller = as_labeller(
                        c("Industri" = "Industri",
                          "Tjenesteyting" = "Tjenesteyting",
                          "Andre næringer" = "Andre næringer")))+
  theme_minimal()
```



Oppgave IV: Undersøk fordelingen igjen

Kan du modifisere koden fra oppgave II til å i tillegg illustrere fordelingen av FoU-bruken på lønn, innleie av personale, investering, og andre kostnader?

Merknad. Kommandoen `fill = [statistikkvariabel]` kan brukes i et histogram.

```
df_4 <- df

#  filter(variabel == "FoU-kostnader i alt")

#Endre navn på gruppene
df_4["gruppe"][df_4["gruppe"] == "10-19 sysselsatte"] <- "10-19"
df_4["gruppe"][df_4["gruppe"] == "20-49 sysselsatte"] <- "20-49"
df_4["gruppe"][df_4["gruppe"] == "50-99 sysselsatte"] <- "50-99"
df_4["gruppe"][df_4["gruppe"] == "100-199 sysselsatte"] <- "100-199"
df_4["gruppe"][df_4["gruppe"] == "200-499 sysselsatte"] <- "200-499"
df_4["gruppe"][df_4["gruppe"] == "500 sysselsatte og over"] <- "500+"



#Fjern "Alle næringer", bedrifter med 10-19 ansatte og uønskede år
df_4<-df_4[!(df_4$næring=="Alle næringer" | df_4$variabel=="FoU-driftskostnader i alt" | df_4$


#Synn litt med satsesttet for å få de tallene jeg vil ha
```

```
#Surr ttt med satasettet for a ja de tallene jeg vil ha
df_20 <- df_4 %>%

  filter(gruppe == "20-49")

df_50 <- df_4 %>%

  filter(gruppe == "50-99")

df_100 <- df_4 %>%

  filter(gruppe == "100-199")

df_200 <- df_4 %>%

  filter(gruppe == "200-499")

df_500 <- df_4 %>%

  filter(gruppe == "500+")

#Summering
df_20_ <- df_20 %>%
  group_by(næring, gruppe, variabel) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring', 'gruppe'. You can override using
the `.groups` argument.
```

```
df_50_ <- df_50 %>%
  group_by(næring, gruppe, variabel) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring', 'gruppe'. You can override using
the `.groups` argument.
```

```
df_100_ <- df_100 %>%
  group_by(næring, gruppe, variabel) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring', 'gruppe'. You can override using
the `.groups` argument.
```

```
df_200_ <- df_200 %>%
  group_by(næring, gruppe, variabel) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring', 'gruppe'. You can override using
the `.groups` argument.
```

```
df_500_ <- df_500 %>%
```

```
  group_by(næring, gruppe, variabel) %>%
  summarise(tot = sum(value, na.rm=TRUE))
```

```
`summarise()` has grouped output by 'næring', 'gruppe'. You can override using
the `.groups` argument.
```

```
#Legg summene inn i  ett datasett
df_foutot <- rbind(df_20_, df_50_, df_100_, df_200_, df_500_)
```

```
# besvar oppgave III her

df_foutot %>%

#Manuell sortering så bedriftstørrelse kommer i stugende rekkefølge
  mutate(variabel = fct_relevel(variabel, "FoU-investeringer (kapitalkostnader)", "Kostnader t

  ggplot(aes(tot, variabel)) +
  geom_col(fill="black", col="yellow") +
  scale_x_continuous(breaks=c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100))+

  labs(y = "", x = "Kroner i milliarder", title = "FoU kostnader samlet",
subtitle = "Totale FoU kostnader for næringene samlet, i milliarder kroner.") +



  facet_wrap(~ factor (næring, (levels=c("Industri", "Tjenesteyting", "Andre næringer"))), nro
             #  lag tre paneler
             #  titler
                      labeller = as_labeller(
                        c("Industri" = "Industri",
                          "Tjenesteyting" = "Tjenesteyting",
                          "Andre næringer" = "Andre næringer")))+
  theme_minimal()
```

FoU kostnader samlet

Totale FoU kostnader for næringene samlet, i milliarder kroner.