Assignment4h23

Innholdsfortegnelse

```r
# xml skal ha mer detaljert info
# toc_xml <- get_eurostat_toc()
# tekstversjonen har trolig nok info for vårt formål
toc_txt <- get_eurostat_toc(mode = "txt")
```

```r
gdp_tabs <- toc_txt |>
   # Regex AND external to regex
      filter(
        str_detect(
string = title,
# For å matche både små og store bokstaver
pattern = '[Gg][Dd][Pp]'
# AND vha. &
) &
        str_detect(
          string = title,
          # For å matche både små og store bokstaver og
          # space eller ikke før 3
          pattern = '[Nn][Uu][Tt][Ss]\\s*3'
          )
) |>
     select(title, code)
```

```r
gdp_tabs |>
    select(title, code) |>
    flextable()  |>
    width(1, width = 3.5) |>
    width(2, width = 1.5)
```

title	code
Average annual population to calculate regional GDP data (thousand persons) by NUTS 3 regions	nama_10r_3popgdp
Gross domestic product (GDP) at current market prices by NUTS 3 regions	nama_10r_3gdp
European Union trade mark (EUTM) applications per billion GDP by NUTS 3 regions	ipr_ta_gdpr
Community design (CD) applications per billion GDP by NUTS 3 regions	ipr_da_gdpr

```r
# description nama_10r_3gdp
   dsd_gdp <- get_eurostat_dsd("nama_10r_3gdp")
  dsd_gdp |>
  head(n = 15) |>
  flextable()  |>
  width(1, width = 1) |>
  width(2, width = 1) |>
  width(3, width = 3.5)
```

concept	code	name
freq	A	Annual
unit	MIO_EUR	Million euro
unit	EUR_HAB	Euro per inhabitant
unit	EUR_HAB_EU27_2020	Euro per inhabitant in percentage of the EU27 (from 2020) average
unit	MIO_NAC	Million units of national currency
unit	MIO_PPS_EU27_2020	Million purchasing power standards (PPS, EU27 from 2020)

concept	code	name
unit	PPS_EU27_2020_HAB	Purchasing power standard (PPS, EU27 from 2020), per inhabitant
unit	PPS_HAB_EU27_2020	Purchasing power standard (PPS, EU27 from 2020), per inhabitant in percentage of the EU27 (from 2020) average
geo	EU27_2020	European Union - 27 countries (from 2020)
geo	BE	Belgium
geo	BE1	Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
geo	BE10	Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
geo	BE100	Arr. de Bruxelles-Capitale/Arr. Brussel-Hoofdstad
geo	BE2	Vlaams Gewest
geo	BE21	Prov. Antwerpen

```r
# Gross domestic product (GDP) at current market prices by NUTS 3 regions
  # id: nama_10r_3gdp
  nama_10r_3gdp <- get_eurostat_data(
    id = "nama_10r_3gdp",
    filters = list(
      geo = c("AT", "DE", "DK", "FR", "EL", "ES",
              "IT", "NL", "BE", "IE", "PL", "PT", "NO", "SE", "FI", "CH"),
      nuts_level = "3",
      unit = "MIO_PPS_EU27_2020"
    ),
    exact_match = FALSE,
    date_filter = 2000:2020,
    stringsAsFactors = FALSE
    ) |>
    mutate(
      gdp = 1000000 * values
    ) |>
    select(-c(unit, values)) |>
    # Vil bare ha NUTS 3 nivå (5 karakterer). Vil aggregere selv til NUTS2, NUTS1 og NUTSc
     filter(str_length(geo) == 5)
```

Oppgave 1

Vi vil i hovedsak bruke befolkningstabellen som har teksten: «Average annual population to calculate regional GDP data (thousand persons) by NUTS 3 regions», men siden denne synes å ha manglende data for noen regioner vil vi supplere med data fra tabellen med teksten «Population on 1 January by broad age group, sex and NUTS 3 region».

Funnet ved søk:

«Average annual population to calculate regional GDP data (thousand persons) by NUTS 3 regions» = nama_10r_3popgdp

«Population on 1 January by broad age group, sex and NUTS 3 region» = demo_r_pjanaggr3

Oppgave 2

```
# description nama_10r_3popgdp
 dsd_gdp <- get_eurostat_dsd("nama_10r_3popgdp")
dsd_gdp |>
head(n = 15) |>
flextable()  |>
width(1, width = 1) |>
width(2, width = 1) |>
width(3, width = 3.5)
```

concept	code	name
freq	A	Annual
unit	THS	Thousand
geo	EU27_2020	European Union - 27 countries (from 2020)
geo	BE	Belgium
geo	BE1	Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
geo	BE10	Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
geo	BE100	Arr. de Bruxelles-Capitale/Arr. Brussel-Hoofdstad
geo	BE2	Vlaams Gewest
geo	BE21	Prov. Antwerpen

concept	code	name
geo	BE211	Arr. Antwerpen
geo	BE212	Arr. Mechelen
geo	BE213	Arr. Turnhout
geo	BE22	Prov. Limburg (BE)
geo	BE223	Arr. Tongeren
geo	BE224	Arr. Hasselt

```
# id: nama_10r_3popgdp
nama_10r_3popgdp <- get_eurostat_data(
  id = "nama_10r_3popgdp",
  filters = list(
    geo = c("AT", "DE", "DK", "FR", "EL", "ES",
            "IT", "NL", "BE", "IE", "PL", "PT", "NO", "SE", "FI", "CH"),
    nuts_level = "3",
    unit = "THS"
  ),
  exact_match = FALSE,
  date_filter = 2000:2020,
  stringsAsFactors = FALSE
  ) |>
  mutate(
    pop.x = 1000 * values
  ) |>
  select(-c(unit, values)) |>
  # Vil bare ha NUTS 3 nivå (5 karakterer). Vil aggregere selv til NUTS2, NUTS1 og NUTSc
   filter(str_length(geo) == 5)
```

Oppgave 3

```
# id: demo_r_pjanaggr3
demo_r_pjanaggr3 <- get_eurostat_data(
  id = "demo_r_pjanaggr3",
  filters = list(
    geo = c("AT", "DE", "DK", "FR", "EL", "ES",
            "IT", "NL", "BE", "IE", "PL", "PT", "NO", "SE", "FI", "CH"),
```

```r
      nuts_level = "3",
      unit = "NR", sex = "T", age = "TOTAL"
    ),
    exact_match = FALSE,
    date_filter = 2000:2020,
    stringsAsFactors = FALSE
    ) |>
    select(-c(unit, age, sex)) |> select(geo, time, pop.y = values) |>
    # Vil bare ha NUTS 3 nivå (5 karakterer). Vil aggregere selv til NUTS2, NUTS1 og NUTSc
     filter(str_length(geo) == 5)
```

Oppgave 4

```r
nuts3_missing_in_demo_r_pjanaggr3 <- setdiff(
  nama_10r_3popgdp$geo,
  demo_r_pjanaggr3$geo
)

# Vis resultatet
# ag_comment: setter width så fåre en finere output
print(nuts3_missing_in_demo_r_pjanaggr3, width = 78)
```

```
 [1] "DKZZZ" "ESZZZ" "ITG2D" "ITG2E" "ITG2F" "ITG2G" "ITG2H" "ITZZZ" "NLZZZ"
[10] "NO020" "NO074" "NO081" "NO082" "NO091" "NO092" "NO0A1" "NO0A2" "NO0A3"
[19] "NO0B2" "NOZZZ"
```

Oppgave 5

```r
# Identifiser NUTS3-soner som mangler i nama_10r_3gdp
nuts3_missing_in_nama_10r_3gdp <- setdiff(
  demo_r_pjanaggr3$geo,
  nama_10r_3popgdp$geo
)

# Vis resultatet
# ag_comment: setter width så fåre en finere output
print(nuts3_missing_in_nama_10r_3gdp, width = 78)
```

```
 [1] "BE221" "BE222" "BE321" "BE322" "BE324" "BE325" "BE326" "BE327" "FRXXX"
[10] "ITG25" "ITG26" "ITG27" "ITG28" "ITG29" "ITG2A" "ITG2B" "ITG2C" "NO011"
[19] "NO012" "NO021" "NO022" "NO031" "NO032" "NO033" "NO034" "NO041" "NO042"
[28] "NO043" "NO051" "NO052" "NO053" "NO061" "NO062" "NO072" "NO073"
```

Oppgave 6

```r
# hsh pipe over for å få console output
# Utfører en full_join
full_pop_nuts3 <- full_join(demo_r_pjanaggr3, nama_10r_3popgdp, by = c("geo", "time"))

# Vis de første radene av det kombinerte datasettet
head(full_pop_nuts3)
```

```
     geo time pop.y pop.x
1: AT111 2001 37732 38050
2: AT111 2002 37778 37730
3: AT111 2003 37703 37650
4: AT111 2004 37640 37580
5: AT111 2005 37522 37450
6: AT111 2006 37413 37450
```

Oppgave 7

```r
# Identifiser NUTS3-soner som mangler i GDP-tabellen
nuts3_missing_in_gdp <- setdiff(
  full_pop_nuts3$geo,
  nama_10r_3popgdp$geo
)

# Vis resultatet
print(nuts3_missing_in_gdp, width = 78)
```


```
 [1] "BE221" "BE222" "BE321" "BE322" "BE324" "BE325" "BE326" "BE327" "FRXXX"
[10] "ITG25" "ITG26" "ITG27" "ITG28" "ITG29" "ITG2A" "ITG2B" "ITG2C" "NO011"
[19] "NO012" "NO021" "NO022" "NO031" "NO032" "NO033" "NO034" "NO041" "NO042"
[28] "NO043" "NO051" "NO052" "NO053" "NO061" "NO062" "NO072" "NO073"
```

Oppgave 8

```r
# Identifiserer NUTS3-soner som mangler i full_pop_nuts3
nuts3_missing_in_full_pop_nuts3 <- setdiff(
  nama_10r_3gdp$geo,
  full_pop_nuts3$geo
)

# Vis resultatet
print(nuts3_missing_in_full_pop_nuts3, width = 78)
```

```
[1] "ATZZZ" "BEZZZ" "FIZZZ" "FRZZZ" "PTZZZ" "SEZZZ"
```

Oppgave 9

```r
full_pop_nuts3 |>
     filter(geo %in% c("NO053", "NO060", "NO061")) |>
     filter(time %in% 2014:2020) |>
     arrange(time, geo)
```

```
      geo time  pop.y  pop.x
 1: NO053 2014 261458     NA
 2: NO060 2014 441193 443090
 3: NO061 2014 306067     NA
 4: NO053 2015 263736     NA
 5: NO060 2015     NA 447910
 6: NO061 2015 310093     NA
 7: NO053 2016 265151     NA
 8: NO060 2016 449457 452090
 9: NO061 2016 313105     NA
10: NO053 2017 266274     NA
11: NO060 2017 454596 457000
12: NO061 2017 317363     NA
13: NO053 2018 266858     NA
14: NO060 2018 458742 460170
15: NO061 2018 320884     NA
16: NO053 2019 267420     NA
17: NO060 2019 462032 465910
18: NO053 2020 267642     NA
19: NO060 2020 465136 469910
```

```r
full_pop_nuts3 <- full_pop_nuts3 %>%
  filter(!str_detect(str_sub(geo, start = 3, end = 5), "ZZZ$"))
```

Oppgave 10

```r
 full_pop_nuts3 <- full_pop_nuts3 |>
     mutate(
       pop = ifelse(
         test = is.na(pop.x) == TRUE,
         yes = pop.y,
         no = pop.x
         )
     ) |>
     select(-pop.x, -pop.y)
```

Oppgave 11

```r
full_pop_nuts3 <- full_pop_nuts3 %>%
    mutate(pop = ifelse(pop == 0, NA, pop))
```

Oppgave 12

```r
eu_data <- left_join(nama_10r_3gdp, full_pop_nuts3, by = c("geo", "time"))
```

```r
eu_data <- eu_data %>%
  filter(!str_detect(str_sub(geo, start = 3, end = 5), "ZZZ$"))
```

```r
dim(eu_data)
```

```
[1] 21062     4
```

Dette er feil tall! Skal bli slik:

21159 4

```r
eu_data <- eu_data |>
    mutate(
      country = str_sub(geo, start = 1L, end = 2L)
    )
```

```r
eu_data |>
    distinct(geo, .keep_all = TRUE) |>
    group_by(country) |>
    summarise(Antall = n(), .groups = "drop")
```

```
# A tibble: 16 x 2
   country Antall
   <chr>    <int>
 1 AT          35
 2 BE          44
 3 CH          26
 4 DE         401
 5 DK          11
 6 EL          52
 7 ES          59
 8 FI          19
 9 FR         101
10 IE           8
11 IT         107
12 NL          40
13 NO          12
14 PL          73
15 PT          25
16 SE          21
```

```r
eu_data |>
   summary()
```

```
     geo                 time                gdp                  pop
 Length:21062       Length:21062       Min.   :8.512e+07   Min.   :   8400
 Class :character   Class :character   1st Qu.:2.957e+09   1st Qu.: 132240
 Mode  :character   Mode  :character   Median :5.342e+09   Median : 241050
                                       Mean   :1.004e+10   Mean   : 373869
                                       3rd Qu.:1.037e+10   3rd Qu.: 440690
                                       Max.   :2.606e+11   Max.   :6757000
```

```
                                                          NA's   :13
  country
Length:21062
Class :character
Mode  :character
```

```
eu_data <- eu_data |>
    select(country, NUTS3 = geo, year = time, gdp, pop)
  # Rydder opp
  # Sletter alle objekt utenom eu_data
  # don't use if you don't mean it
  rm(list = setdiff(ls(), "eu_data"))
```

Oppgave 13

```
# Beregner GDP per capita
eu_data <- eu_data %>%
  mutate(gdp_per_capita = round(gdp / pop, 2))
```

```
  eu_data |>
    select(gdp_per_capita) |>
    summary()
```

```
gdp_per_capita
Min.   :  3359
1st Qu.: 18324
Median : 23270
Mean   : 25308
3rd Qu.: 29377
Max.   :177427
NA's   :13
```

Oppgave 14

```r
# Legger til variabelen country_name basert på verdier i geo
eu_data <- eu_data %>%
  mutate(
    country_name = case_when(
      country == "AT" ~ "Østerrike",
      country == "DE" ~ "Tyskland",
      country == "DK" ~ "Danmark",
      country == "FR" ~ "Frankrike",
      country == "EL" ~ "Hellas",
      country == "ES" ~ "Spania",
      country == "IT" ~ "Italia",
      country == "NL" ~ "Nederland",
      country == "BE" ~ "Belgia",
      country == "IE" ~ "Irland",
      country == "PL" ~ "Polen",
      country == "PT" ~ "Portugal",
      country == "NO" ~ "Norge",
      country == "SE" ~ "Sverige",
      country == "FI" ~ "Finland",
      country == "CH" ~ "Sveits",
      TRUE ~ as.character(country)  # Default: Behold country som country_name hvis ingen
    )
  )
```

Oppgave 15

```r
# Lager NUTS2, NUTS1, og NUTSc
eu_data <- eu_data %>%
  mutate(
    NUTS2 = str_sub(NUTS3, start = 1, end = 4),
    NUTS1 = str_sub(NUTS3, start = 1, end = 3),
    NUTSc = str_sub(NUTS3, start = 1, end = 2)
  ) %>%
  # Velger ønsket rekkefølge av variabler
  select(country_name, country, year, NUTS3, NUTS2, NUTS1, NUTSc, gdp, pop, gdp_per_capita
```

Oppgave 16

```
gini_NUTS2 <- eu_data %>%
  group_by(NUTS2, country_name, country, year) %>%
  summarise(
    gini_nuts2 = Gini(
      x = gdp_per_capita,
      weights = pop,
      na.rm = TRUE
    ),
    pop = sum(pop, na.rm = TRUE),
    gdp = sum(gdp, na.rm = TRUE),
    gdp_per_capita = gdp / pop,
    num_nuts3 = n(),
    .groups = "drop"
  ) %>%
  select(country_name, country, NUTS2, year, pop, gdp, gdp_per_capita, num_nuts3, gini_nut
```

```
gini_NUTS2 |>
     summary() |>
     print(width = 80)
```

```
country_name         country              NUTS2                year
Length:4193        Length:4193        Length:4193        Length:4193
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



     pop                 gdp            gdp_per_capita    num_nuts3
Min.   :       0   Min.   :8.512e+07   Min.   : 3359   Min.   : 1.000
1st Qu.:  714880   1st Qu.:1.628e+10   1st Qu.:19425   1st Qu.: 2.000
Median : 1451900   Median :3.416e+10   Median :24498   Median : 4.000
Mean   : 1876835   Mean   :5.042e+10   Mean   :  Inf   Mean   : 5.023
3rd Qu.: 2374900   3rd Qu.:6.267e+10   3rd Qu.:30870   3rd Qu.: 7.000
Max.   :12363480   Max.   :6.996e+11   Max.   :  Inf   Max.   :23.000

  gini_nuts2
Min.   :0.0001
1st Qu.:0.0591
```

```
 Median :0.1014
 Mean   :0.1196
 3rd Qu.:0.1603
 Max.   :0.4547
 NA's   :703
```

```
gini_NUTS2 |>
    select(-country_name) |>
    filter(gini_nuts2 < 0.001)
```

```
# A tibble: 4 x 8
  country NUTS2 year      pop         gdp gdp_per_capita num_nuts3 gini_nuts2
  <chr>   <chr> <chr>   <dbl>       <dbl>          <dbl>     <int>      <dbl>
1 ES      ES43  2010  1100400 18879360000         17157.         2   0.000405
2 IT      ITF5  2006   588300 11135870000         18929.         2   0.000545
3 NO      NO07  2010   467100 13738470000         29412.         2   0.000479
4 PL      PL43  2020  1010100 18762060000         18574.         2   0.000148
```

Oppgave 17

```
gini_NUTS1 <- eu_data %>%
  group_by(NUTS1, country_name, country, year) %>%
  summarise(
    gini_nuts1 = Gini(
      x = gdp_per_capita,
      weights = pop,
      na.rm = TRUE
    ),
    pop = sum(pop, na.rm = TRUE),
    gdp = sum(gdp, na.rm = TRUE),
    gdp_per_capita = gdp / pop,
    num_nuts2 = n_distinct(NUTS2),
    .groups = "drop"
  ) %>%
  select(country_name, country, NUTS1, year, pop, gdp, gdp_per_capita, num_nuts2, gini_nut
```

```
gini_NUTS1 |>
  summary() |>
    print(width = 80)
```

```
 country_name          country             NUTS1               year          
 Length:1545        Length:1545        Length:1545        Length:1545       
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  
```

```
      pop                gdp            gdp_per_capita    num_nuts2    
 Min.   :   25740   Min.   :6.815e+08   Min.   : 6423   Min.   :1.000  
 1st Qu.: 2544800   1st Qu.:5.422e+10   1st Qu.:19819   1st Qu.:1.000  
 Median : 4032210   Median :9.979e+10   Median :24765   Median :3.000  
 Mean   : 5093573   Mean   :1.368e+11   Mean   :26180   Mean   :2.714  
 3rd Qu.: 6076380   3rd Qu.:1.649e+11   3rd Qu.:31275   3rd Qu.:4.000  
 Max.   :17939970   Max.   :6.996e+11   Max.   :63383   Max.   :7.000  
                                                                       
   gini_nuts1     
 Min.   :0.01983  
 1st Qu.:0.08361  
 Median :0.12644  
 Mean   :0.13387  
 3rd Qu.:0.16753  
 Max.   :0.39082  
 NA's   :144      
```

Oppgave 18

```r
gini_NUTSc <- eu_data %>%
  group_by(NUTSc, country_name, country, year) %>%
  summarise(
    gini_nutsc = Gini(
      x = gdp_per_capita,
      weights = pop,
      na.rm = TRUE
    ),
    pop = sum(pop, na.rm = TRUE),
    gdp = sum(gdp, na.rm = TRUE),
    gdp_per_capita = gdp / pop,
    num_nuts1 = n_distinct(NUTS1),
    .groups = "drop"
  ) %>%
```

```r
  select(country_name, country, NUTSc, year, pop, gdp, gdp_per_capita, num_nuts1, gini_nut
```

```r
gini_NUTSc |>
    summary() |>
    print(width = 80)
```

```
 country_name          country             NUTSc               year          
 Length:312         Length:312         Length:312         Length:312        
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  
                                                                            
                                                                            
                                                                            
      pop                 gdp            gdp_per_capita    num_nuts1     
 Min.   : 3543470   Min.   :9.547e+10   Min.   : 8865   Min.   : 1.000  
 1st Qu.: 7997358   1st Qu.:2.037e+11   1st Qu.:23421   1st Qu.: 2.000  
 Median :10557885   Median :3.121e+11   Median :28361   Median : 3.000  
 Mean   :25222983   Mean   :6.776e+11   Mean   :28676   Mean   : 4.952  
 3rd Qu.:43837275   3rd Qu.:1.010e+12   3rd Qu.:34222   3rd Qu.: 7.000  
 Max.   :83161210   Max.   :3.147e+12   Max.   :61599   Max.   :16.000  
   gini_nutsc    
 Min.   :0.1110  
 1st Qu.:0.1430  
 Median :0.1691  
 Mean   :0.1755  
 3rd Qu.:0.2004  
 Max.   :0.3826  
```

Oppgave 19

```r
gini_NUTS2_nest <- gini_NUTS2 |>
     group_by(country_name, country) |>
     nest(.key = "NUTS2_data") |>
     ungroup()
```

Oppgave 20

```
gini_NUTS1_nest <- gini_NUTS1 %>%
  group_by(country_name, country) %>%
  nest(.key = "NUTS1_data") %>%
  ungroup()
```

Oppgave 21

```
gini_NUTSc_nest <- gini_NUTSc %>%
  group_by(country_name, country) %>%
  nest(.key = "NUTSc_data") %>%
  ungroup()
```

Oppgave 22

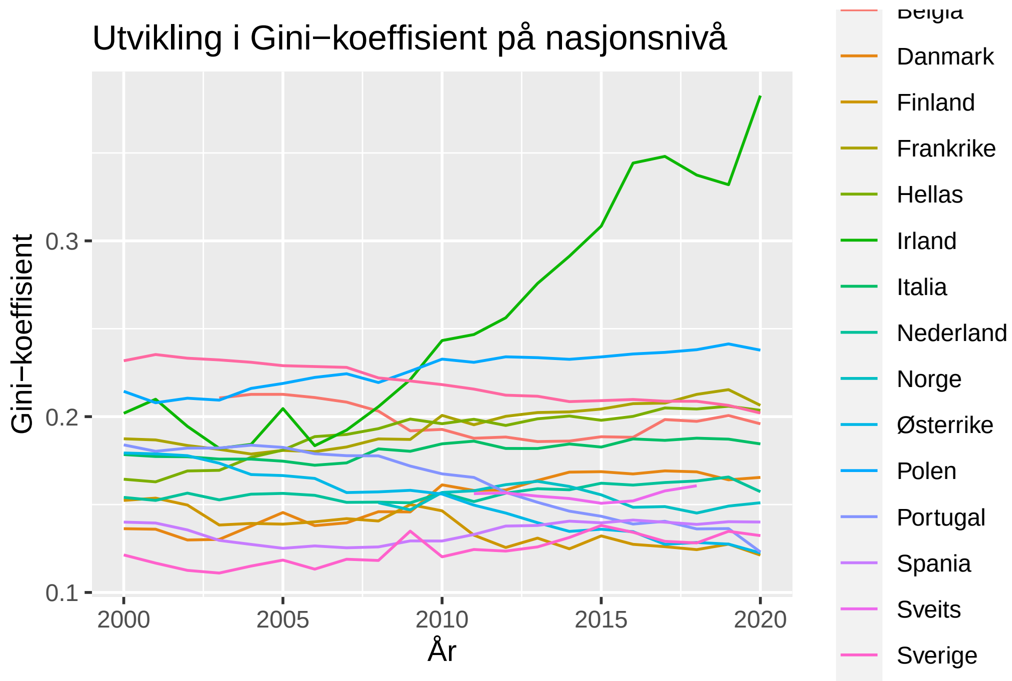
```
# Anta at eu_data er ditt opprinnelige datasett

# Grupper etter land og neste dataene på NUTS3-nivå
eu_data_nested <- eu_data %>%
  group_by(country_name, country) %>%
  nest() %>%
  rename(NUTS3_data = data) %>%
  ungroup()

# Sørg for at de andre gini-datasettene har én rad per land
# Du må kanskje utføre en aggregering her hvis det er nødvendig
gini_NUTS2_nest_unique <- gini_NUTS2_nest %>% distinct(country_name, country, .keep_all =
gini_NUTS1_nest_unique <- gini_NUTS1_nest %>% distinct(country_name, country, .keep_all =
gini_NUTSc_nest_unique <- gini_NUTSc_nest %>% distinct(country_name, country, .keep_all =

# Left join med de unike gini-datasettene
eu_data_nested <- eu_data_nested %>%
  left_join(gini_NUTS2_nest_unique, by = c("country_name", "country")) %>%
  left_join(gini_NUTS1_nest_unique, by = c("country_name", "country")) %>%
  left_join(gini_NUTSc_nest_unique, by = c("country_name", "country"))

# Fjerner gruppestrukturen og eventuelle duplikater
```

```
eu_data_nested <- eu_data_nested %>%
  ungroup() %>%
  distinct(country_name, country, .keep_all = TRUE)

# Sjekk strukturen til den endelige dataframen
print(eu_data_nested)
```

```
# A tibble: 16 x 6
   country_name country NUTS3_data           NUTS2_data NUTS1_data NUTSc_data
   <chr>        <chr>   <list>               <list>     <list>     <list>
 1 Østerrike    AT      <tibble [735 x 8]>   <tibble>   <tibble>   <tibble>
 2 Belgia       BE      <tibble [712 x 8]>   <tibble>   <tibble>   <tibble>
 3 Sveits       CH      <tibble [208 x 8]>   <tibble>   <tibble>   <tibble>
 4 Tyskland     DE      <tibble [8,421 x 8]> <tibble>   <tibble>   <tibble>
 5 Danmark      DK      <tibble [231 x 8]>   <tibble>   <tibble>   <tibble>
 6 Hellas       EL      <tibble [1,092 x 8]> <tibble>   <tibble>   <tibble>
 7 Spania       ES      <tibble [1,239 x 8]> <tibble>   <tibble>   <tibble>
 8 Finland      FI      <tibble [399 x 8]>   <tibble>   <tibble>   <tibble>
 9 Frankrike    FR      <tibble [2,121 x 8]> <tibble>   <tibble>   <tibble>
10 Irland       IE      <tibble [162 x 8]>   <tibble>   <tibble>   <tibble>
11 Italia       IT      <tibble [2,247 x 8]> <tibble>   <tibble>   <tibble>
12 Nederland    NL      <tibble [840 x 8]>   <tibble>   <tibble>   <tibble>
13 Norge        NO      <tibble [156 x 8]>   <tibble>   <tibble>   <tibble>
14 Polen        PL      <tibble [1,533 x 8]> <tibble>   <tibble>   <tibble>
15 Portugal     PT      <tibble [525 x 8]>   <tibble>   <tibble>   <tibble>
16 Sverige      SE      <tibble [441 x 8]>   <tibble>   <tibble>   <tibble>
```

Oppgave 23

```
# Konverter year til numerisk format
gini_NUTSc$year <- as.numeric(gini_NUTSc$year)

# Plotting med femårsintervaller på x-aksen
ggplot(gini_NUTSc, aes(x = year, y = gini_nutsc, group = country_name, color = country_nam
  geom_line() +
  labs(title = "Utvikling i Gini-koeffisient på nasjonsnivå",
       x = "År",
       y = "Gini-koeffisient") +
  scale_x_continuous(breaks = seq(min(gini_NUTSc$year), max(gini_NUTSc$year), by = 5))
```




```r
eu_data_nested |>
  unnest(NUTSc_data) |>
  filter(year == 2020) |>
  select(country_name, gini_nutsc) |>
  arrange(desc(gini_nutsc)) |>
  flextable() |>
  width(1, width = 1.5) |>
  width(2, width = 1.5)
```

country_name	gini_nutsc
Irland	0.3826165
Polen	0.2378284
Frankrike	0.2064403
Hellas	0.2036007
Tyskland	0.2020493
Belgia	0.1959298
Italia	0.1845053

country_name	gini_nutsc
Danmark	0.1654528
Nederland	0.1573150
Norge	0.1510297
Spania	0.1400519
Sverige	0.1323442
Portugal	0.1230546
Østerrike	0.1224705
Finland	0.1212452

Oppgave 24

```
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "IE") %>%
  filter(year == 2000:2020) %>%
  select(year, NUTS2, gdp_per_capita, gini_nuts2) %>%
  arrange(desc(gini_nuts2)) %>%
  flextable() %>%
  width(1, width = 1.5) %>%
  width(2, width = 1.5)
```

year	NUTS2	gdp_per_capita	gini_nuts2
2020	IE06	64,839.24	0.43423978
2019	IE06	61,274.46	0.39684316
2014	IE06	43,838.62	0.39008370
2013	IE06	40,634.37	0.38360258
2020	IE05	76,983.92	0.38215439
2018	IE06	59,937.07	0.37901152
2017	IE06	57,092.90	0.36839547
2012	IE06	38,688.55	0.35534175

year	NUTS2	gdp_per_capita	gini_nuts2
2016	IE06	53,774.71	0.35346251
2011	IE06	37,708.15	0.34423906
2015	IE06	49,405.75	0.33548581
2018	IE05	71,209.31	0.33340668
2010	IE06	38,186.56	0.32317347
2020	IE04	23,696.39	0.32178852
2019	IE05	74,539.22	0.27480470
2001	IE06	30,485.86	0.27125525
2009	IE06	36,569.62	0.26782869
2008	IE06	39,852.99	0.26258757
2005	IE06	38,579.77	0.25673014
2000	IE06	29,192.95	0.24578629
2012	IE05	33,372.58	0.24476866
2011	IE05	32,826.67	0.24429044
2007	IE06	42,350.73	0.23501259
2002	IE05	29,920.73	0.23133080
2003	IE06	33,002.40	0.22778539
2004	IE06	36,195.35	0.22757404
2006	IE06	40,448.89	0.22348125
2014	IE04	22,524.99	0.21387000
2014	IE05	33,924.39	0.21325659
2002	IE06	31,805.13	0.21019030
2016	IE04	22,942.09	0.20662891
2015	IE04	24,124.46	0.20607425
2010	IE05	30,680.67	0.19852713
2009	IE05	29,667.47	0.19583205
2012	IE04	24,150.40	0.19386944

year	NUTS2	gdp_per_capita	gini_nuts2
2019	IE04	25,184.98	0.17456392
2013	IE04	21,574.19	0.17280122
2011	IE04	24,164.51	0.17174129
2001	IE05	26,998.52	0.17157286
2018	IE04	25,146.39	0.16895623
2010	IE04	21,949.62	0.16571218
2013	IE05	32,815.19	0.16406741
2003	IE05	30,664.21	0.15978361
2005	IE05	31,326.17	0.15701296
2007	IE05	35,714.88	0.15286466
2006	IE05	33,455.42	0.15216101
2017	IE04	25,035.40	0.15140781
2000	IE05	23,835.52	0.14272675
2004	IE05	30,657.57	0.14181869
2008	IE05	32,355.49	0.13684589
2001	IE04	17,661.33	0.12956190
2009	IE04	20,296.05	0.12750682
2000	IE04	16,692.46	0.12312561
2008	IE04	22,675.35	0.09187490
2004	IE04	21,620.15	0.08109587
2007	IE04	24,585.47	0.07999625
2005	IE04	22,089.61	0.07961662
2006	IE04	24,331.68	0.06434471
2002	IE04	19,128.84	0.06318599
2003	IE04	19,637.24	0.03032886
2015	IE05	30,172.28	
2016	IE05	29,699.72	

year	NUTS2	gdp_per_capita	gini_nuts2
2017	IE05	29,789.05	

Oppgave 25

```
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "ES") %>%
  filter(year == 2000:2020) %>%
  select(year, NUTS2, gdp_per_capita, gini_nuts2) %>%
  arrange(desc(gini_nuts2)) %>%
  flextable() %>%
  width(1, width = 1.5) %>%
  width(2, width = 1.5)
```

year	NUTS2	gdp_per_capita	gini_nuts2
2020	ES53	22,719.90	0.115920138
2008	ES53	27,875.00	0.108881586
2009	ES53	25,804.49	0.108818282
2007	ES53	27,821.29	0.106439817
2018	ES52	24,235.24	0.104417291
2019	ES52	24,889.05	0.104085705
2017	ES52	23,806.66	0.103034425
2015	ES52	21,970.60	0.102652061
2010	ES53	25,398.78	0.101492167
2013	ES52	20,327.91	0.101380463
2020	ES52	21,907.27	0.099274486
2011	ES53	25,143.55	0.098587046
2014	ES52	21,091.42	0.098430270
2011	ES52	20,796.90	0.097942630
2018	ES41	26,067.97	0.097751373

year	NUTS2	gdp_per_capita	gini_nuts2
2016	ES52	22,639.29	0.097724713
2006	ES53	27,011.18	0.097296825
2012	ES52	20,225.29	0.097239182
2013	ES53	24,957.04	0.095867129
2017	ES41	25,186.94	0.095184432
2012	ES53	24,994.51	0.094246479
2005	ES53	25,403.18	0.093102965
2000	ES41	16,250.51	0.091380632
2019	ES41	26,789.82	0.090390406
2014	ES53	25,888.20	0.089507007
2015	ES53	27,063.03	0.089219337
2019	ES53	30,487.29	0.088023170
2005	ES41	20,797.35	0.088004807
2016	ES53	28,198.96	0.087954646
2017	ES53	29,509.58	0.086849943
2016	ES41	24,217.50	0.086572339
2018	ES53	29,887.61	0.086216597
2004	ES53	24,381.72	0.085631441
2000	ES53	22,559.25	0.085616105
2001	ES41	17,121.63	0.084742840
2002	ES41	18,130.14	0.084396737
2004	ES41	19,626.48	0.083798439
2003	ES41	18,773.22	0.083736872
2008	ES41	23,775.23	0.083446431
2020	ES41	24,169.50	0.083313033
2010	ES52	21,117.14	0.082822520
2001	ES53	23,639.10	0.081719918

year	NUTS2	gdp_per_capita	gini_nuts2
2006	ES41	22,376.87	0.081719578
2007	ES41	23,647.67	0.081498554
2013	ES41	21,996.09	0.080403118
2014	ES41	22,458.95	0.080123176
2015	ES41	23,377.23	0.080008094
2011	ES41	22,405.11	0.079933228
2007	ES21	32,211.55	0.079673609
2010	ES41	22,482.01	0.079638465
2012	ES41	22,226.32	0.078929990
2002	ES53	24,064.83	0.075884284
2009	ES52	21,381.36	0.075363636
2003	ES53	23,816.78	0.075089097
2006	ES21	30,479.83	0.074844020
2009	ES41	22,596.40	0.074426769
2002	ES70	19,311.26	0.073644213
2007	ES52	23,135.58	0.072389466
2008	ES52	23,170.73	0.072362262
2004	ES70	19,986.86	0.071144685
2008	ES21	32,941.89	0.070562382
2006	ES52	22,303.91	0.070075700
2001	ES21	23,175.63	0.068425695
2001	ES70	18,601.86	0.068405079
2003	ES70	19,683.57	0.068395091
2002	ES21	24,347.41	0.068153554
2000	ES52	17,098.41	0.065797668
2005	ES70	20,775.31	0.065761079
2017	ES21	34,988.77	0.065053635

year	NUTS2	gdp_per_capita	gini_nuts2
2002	ES52	19,023.21	0.064524767
2000	ES70	17,537.95	0.064035579
2003	ES52	19,281.20	0.063556157
2005	ES52	20,740.02	0.062739991
2004	ES21	26,175.85	0.061338694
2018	ES21	35,533.49	0.060987851
2001	ES52	18,271.03	0.060504413
2013	ES21	30,381.67	0.059706810
2005	ES21	27,928.65	0.059291473
2003	ES21	25,047.67	0.059151201
2020	ES21	31,983.82	0.058407924
2000	ES61	13,310.46	0.058347514
2002	ES61	14,987.01	0.058024662
2004	ES52	19,847.64	0.057905813
2014	ES21	31,346.93	0.056031440
2011	ES21	30,636.64	0.055712076
2016	ES21	33,504.82	0.054956054
2012	ES21	30,486.45	0.054951346
2019	ES21	36,542.41	0.054303467
2001	ES61	14,131.31	0.053948401
2006	ES70	21,947.91	0.053896526
2009	ES21	30,849.17	0.053431297
2005	ES61	17,533.87	0.052645749
2000	ES21	21,982.81	0.051436753
2015	ES42	19,552.37	0.051102010
2020	ES24	28,087.75	0.050874335
2017	ES42	21,364.96	0.050632136

year	NUTS2	gdp_per_capita	gini_nuts2
2015	ES70	20,633.11	0.050166870
2018	ES42	21,928.53	0.049953062
2003	ES61	15,717.71	0.049802617
2009	ES11	21,361.04	0.048600012
2016	ES42	20,267.75	0.048465697
2007	ES70	22,691.71	0.048274349
2010	ES21	30,871.57	0.048069396
2015	ES21	32,509.95	0.048005489
2004	ES61	16,524.41	0.046970951
2012	ES42	18,668.59	0.046617518
2020	ES42	20,300.71	0.046330980
2019	ES43	20,740.29	0.046238021
2010	ES11	21,321.30	0.046121944
2019	ES42	22,385.26	0.045951683
2020	ES61	18,496.67	0.045865885
2014	ES70	20,076.97	0.044076621
2016	ES24	28,081.54	0.043925278
2006	ES61	18,880.10	0.043654396
2019	ES61	21,013.09	0.043266783
2007	ES61	19,815.34	0.043169799
2016	ES70	21,193.08	0.043045909
2013	ES42	18,573.08	0.042931955
2011	ES11	20,993.30	0.042175576
2008	ES61	19,789.06	0.041912094
2013	ES70	19,818.66	0.041752296
2017	ES70	22,251.35	0.041663713
2012	ES61	17,489.09	0.041626029

year	NUTS2	gdp_per_capita	gini_nuts2
2015	ES11	22,499.95	0.040480885
2017	ES61	20,134.44	0.040477041
2012	ES70	19,902.27	0.040376957
2017	ES24	29,492.09	0.040343336
2011	ES42	18,978.15	0.039552891
2014	ES42	18,520.63	0.039285663
2016	ES11	23,214.31	0.039283750
2011	ES70	20,422.78	0.039003109
2018	ES61	20,502.86	0.038954242
2008	ES11	22,601.64	0.038870723
2020	ES51	29,249.65	0.038759203
2009	ES61	18,461.31	0.038227415
2007	ES11	22,017.27	0.037987703
2013	ES11	20,961.14	0.037984163
2018	ES70	22,476.98	0.037581201
2012	ES11	20,731.16	0.037259730
2014	ES61	17,781.81	0.037185152
2019	ES51	33,705.37	0.036887369
2018	ES43	20,209.00	0.036837063
2016	ES61	19,085.47	0.036701163
2018	ES51	32,710.99	0.035079867
2010	ES61	18,148.99	0.034890825
2002	ES51	24,155.07	0.034785546
2004	ES11	17,415.20	0.034685123
2014	ES11	21,422.67	0.034464460
2010	ES70	20,693.46	0.034257556
2019	ES70	22,874.46	0.033952181

year	NUTS2	gdp_per_capita	gini_nuts2
2018	ES11	24,822.30	0.033818321
2019	ES24	30,926.05	0.032604065
2008	ES70	22,500.87	0.032464320
2005	ES11	18,770.85	0.032409778
2003	ES42	16,428.79	0.032391302
2011	ES61	17,910.84	0.031926456
2020	ES43	18,800.47	0.031874197
2015	ES61	18,653.32	0.031618379
2003	ES11	16,452.02	0.031505182
2001	ES42	14,899.92	0.031281469
2004	ES51	25,534.49	0.030768228
2002	ES11	15,740.67	0.030752361
2006	ES11	20,571.37	0.030697847
2017	ES51	32,365.24	0.030430383
2003	ES51	24,671.90	0.030047137
2018	ES24	30,071.29	0.029602279
2004	ES42	17,043.58	0.029213797
2010	ES42	19,274.25	0.029016603
2017	ES11	24,323.84	0.028977658
2000	ES11	13,920.65	0.028968640
2000	ES42	13,976.08	0.028942131
2020	ES70	18,131.91	0.027792623
2013	ES61	17,366.82	0.027318527
2004	ES43	14,036.89	0.026426089
2002	ES42	15,789.57	0.026324517
2019	ES11	25,612.10	0.026213503
2017	ES43	19,822.91	0.025395624

year	NUTS2	gdp_per_capita	gini_nuts2
2005	ES42	18,136.62	0.023750834
2013	ES24	25,732.21	0.022984379
2009	ES70	20,811.92	0.022966465
2020	ES11	22,907.89	0.022934335
2001	ES11	14,805.32	0.022523040
2007	ES42	20,646.89	0.022089135
2011	ES24	25,896.04	0.021768481
2006	ES24	26,177.70	0.021125656
2014	ES43	16,860.09	0.020962512
2005	ES24	24,134.74	0.020619417
2015	ES24	26,914.58	0.019910241
2008	ES42	20,813.37	0.019484963
2008	ES43	17,872.20	0.019309031
2006	ES42	19,581.47	0.019066406
2010	ES24	26,206.94	0.018792026
2004	ES24	22,832.37	0.018310400
2016	ES43	18,500.02	0.018286738
2001	ES51	23,196.47	0.017780767
2000	ES24	18,766.02	0.017141375
2013	ES51	27,610.31	0.016923532
2009	ES43	17,073.23	0.016480183
2014	ES24	26,326.88	0.016191407
2012	ES43	16,451.34	0.016130599
2007	ES24	27,867.77	0.016017144
2014	ES51	28,595.18	0.016008058
2012	ES24	25,325.58	0.015383620
2013	ES43	16,627.48	0.015163454

year	NUTS2	gdp_per_capita	gini_nuts2
2008	ES24	28,107.43	0.014994261
2015	ES51	29,898.35	0.014056184
2009	ES42	19,525.68	0.013381015
2015	ES43	17,813.89	0.013157427
2009	ES24	26,226.55	0.012810622
2003	ES24	21,929.70	0.012505802
2016	ES51	30,990.49	0.011615462
2000	ES51	21,826.67	0.011417385
2012	ES51	27,414.87	0.011344103
2009	ES51	28,233.15	0.009398151
2008	ES51	30,096.92	0.008524479
2011	ES51	27,655.15	0.008497482
2005	ES51	26,726.02	0.008408687
2006	ES51	28,841.72	0.007862421
2007	ES51	30,164.22	0.006181856
2005	ES43	15,152.72	0.006124312
2011	ES43	16,771.04	0.005544380
2010	ES51	28,082.50	0.005418339
2007	ES43	17,439.14	0.004961829
2002	ES43	12,798.42	0.004016902
2002	ES24	21,190.49	0.003779675
2001	ES24	19,853.71	0.002903706
2000	ES43	11,394.68	0.002857910
2006	ES43	16,337.25	0.002705474
2001	ES43	12,046.94	0.001670964
2003	ES43	13,354.18	0.001000062
2010	ES43	17,156.82	0.000404504

year	NUTS2	gdp_per_capita	gini_nuts2
2000	ES12	15,064.42	
2001	ES12	16,038.01	
2002	ES12	16,944.14	
2003	ES12	17,517.32	
2004	ES12	18,377.97	
2005	ES12	19,871.61	
2006	ES12	21,888.90	
2007	ES12	23,321.64	
2008	ES12	23,731.55	
2009	ES12	22,000.86	
2010	ES12	21,990.54	
2011	ES12	21,714.49	
2012	ES12	21,198.45	
2013	ES12	20,895.53	
2014	ES12	21,224.82	
2015	ES12	22,155.82	
2016	ES12	22,609.08	
2017	ES12	23,884.17	
2018	ES12	24,337.60	
2019	ES12	25,021.68	
2020	ES12	22,082.58	
2000	ES13	16,765.03	
2001	ES13	17,848.09	
2002	ES13	18,852.40	
2003	ES13	19,224.11	
2004	ES13	19,967.44	
2005	ES13	21,242.28	

year	NUTS2	gdp_per_capita	gini_nuts2
2006	ES13	22,853.51	
2007	ES13	24,133.58	
2008	ES13	24,388.58	
2009	ES13	22,896.44	
2010	ES13	22,665.01	
2011	ES13	22,261.31	
2012	ES13	21,867.70	
2013	ES13	21,585.25	
2014	ES13	22,285.35	
2015	ES13	22,863.65	
2016	ES13	23,675.71	
2017	ES13	24,793.20	
2018	ES13	25,388.47	
2019	ES13	26,230.93	
2020	ES13	23,244.74	
2000	ES22	22,815.47	
2001	ES22	23,808.93	
2002	ES22	24,964.43	
2003	ES22	25,551.62	
2004	ES22	26,549.33	
2005	ES22	28,087.41	
2006	ES22	30,158.68	
2007	ES22	31,406.23	
2008	ES22	31,822.40	
2009	ES22	29,838.84	
2010	ES22	29,466.80	
2011	ES22	29,313.83	

year	NUTS2	gdp_per_capita	gini_nuts2
2012	ES22	28,714.87	
2013	ES22	28,956.51	
2014	ES22	29,800.11	
2015	ES22	30,852.98	
2016	ES22	31,742.76	
2017	ES22	33,169.66	
2018	ES22	33,327.83	
2019	ES22	34,427.54	
2020	ES22	30,385.61	
2000	ES23	20,044.88	
2001	ES23	21,015.10	
2002	ES23	21,756.42	
2003	ES23	22,541.88	
2004	ES23	23,059.43	
2005	ES23	24,254.03	
2006	ES23	26,247.26	
2007	ES23	27,418.95	
2008	ES23	27,687.41	
2009	ES23	25,929.46	
2010	ES23	25,959.92	
2011	ES23	25,701.56	
2012	ES23	25,343.64	
2013	ES23	25,471.18	
2014	ES23	26,499.87	
2015	ES23	27,543.86	
2016	ES23	27,708.29	
2017	ES23	28,911.72	

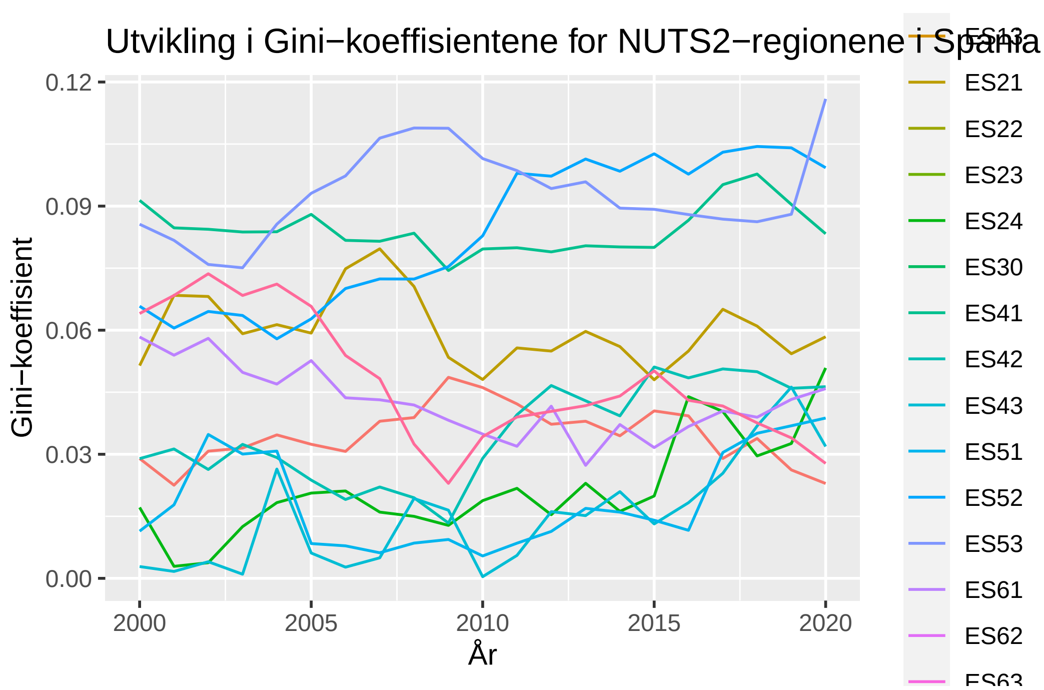
year	NUTS2	gdp_per_capita	gini_nuts2
2018	ES23	29,524.85	
2019	ES23	30,204.10	
2020	ES23	27,060.77	
2000	ES30	24,003.03	
2001	ES30	25,516.18	
2002	ES30	26,524.94	
2003	ES30	27,088.26	
2004	ES30	28,201.74	
2005	ES30	29,834.48	
2006	ES30	32,530.83	
2007	ES30	33,944.82	
2008	ES30	34,232.05	
2009	ES30	32,798.65	
2010	ES30	32,146.15	
2011	ES30	32,308.34	
2012	ES30	32,253.51	
2013	ES30	32,139.62	
2014	ES30	32,990.29	
2015	ES30	34,490.60	
2016	ES30	35,609.71	
2017	ES30	37,251.62	
2018	ES30	37,708.43	
2019	ES30	38,971.35	
2020	ES30	34,086.02	
2000	ES62	15,003.64	
2001	ES62	15,923.12	
2002	ES62	16,807.51	

year	NUTS2	gdp_per_capita	gini_nuts2
2003	ES62	17,377.27	
2004	ES62	17,977.29	
2005	ES62	19,017.06	
2006	ES62	20,481.54	
2007	ES62	21,472.99	
2008	ES62	21,728.03	
2009	ES62	20,068.71	
2010	ES62	19,950.47	
2011	ES62	19,448.74	
2012	ES62	19,271.93	
2013	ES62	19,460.55	
2014	ES62	19,776.06	
2015	ES62	21,058.98	
2016	ES62	21,611.05	
2017	ES62	22,564.33	
2018	ES62	22,563.72	
2019	ES62	23,267.64	
2020	ES62	20,760.82	
2000	ES63	16,084.93	
2001	ES63	16,305.60	
2002	ES63	17,275.71	
2003	ES63	17,969.80	
2004	ES63	18,636.31	
2005	ES63	19,503.46	
2006	ES63	20,918.01	
2007	ES63	21,853.62	
2008	ES63	22,040.24	

year	NUTS2	gdp_per_capita	gini_nuts2
2009	ES63	20,949.36	
2010	ES63	20,391.56	
2011	ES63	19,825.81	
2012	ES63	19,243.82	
2013	ES63	19,656.09	
2014	ES63	19,754.85	
2015	ES63	20,455.62	
2016	ES63	20,883.59	
2017	ES63	21,263.65	
2018	ES63	21,748.65	
2019	ES63	22,609.59	
2020	ES63	20,860.00	
2000	ES64	15,864.26	
2001	ES64	15,967.17	
2002	ES64	16,742.90	
2003	ES64	17,626.62	
2004	ES64	18,630.14	
2005	ES64	19,289.46	
2006	ES64	20,508.69	
2007	ES64	20,824.68	
2008	ES64	20,777.70	
2009	ES64	19,695.30	
2010	ES64	19,041.47	
2011	ES64	18,444.76	
2012	ES64	17,596.86	
2013	ES64	17,791.99	
2014	ES64	18,065.52	

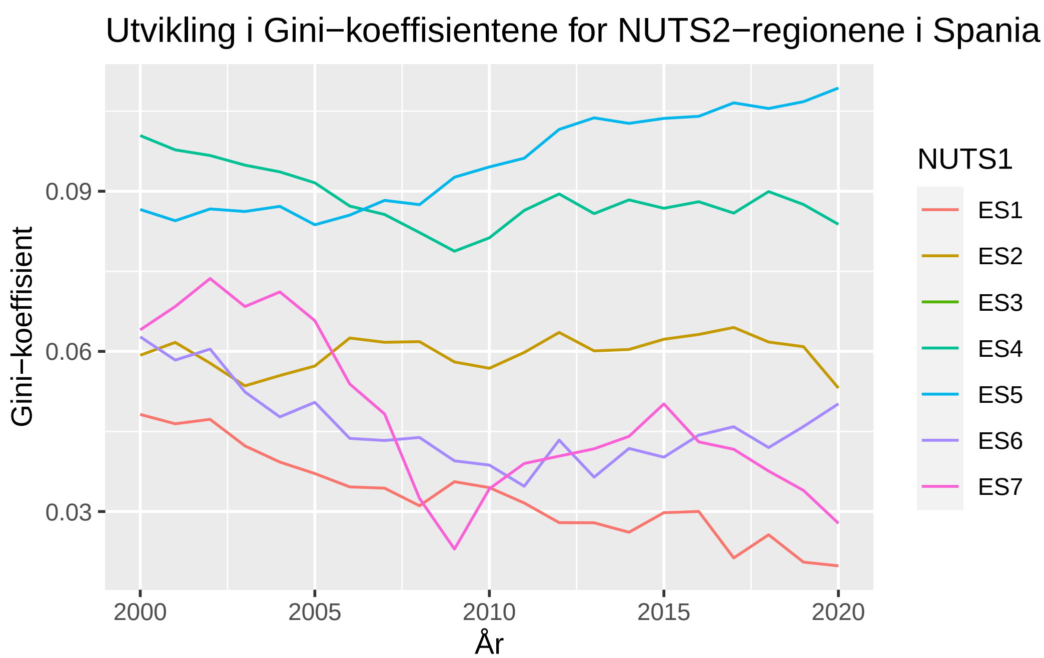
year	NUTS2	gdp_per_capita	gini_nuts2
2015	ES64	18,669.54	
2016	ES64	19,218.65	
2017	ES64	19,516.51	
2018	ES64	20,017.38	
2019	ES64	20,725.36	
2020	ES64	19,036.77	

```
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "ES") %>%
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
  ggplot(aes(x = year, y = gini_nuts2, group = NUTS2, color = NUTS2)) +
  geom_line() +
  labs(title = "Utvikling i Gini-koeffisientene for NUTS2-regionene i Spania",
       x = "År",
       y = "Gini-koeffisient")
```



Oppgave 26

```
eu_data_nested %>%
  unnest(NUTS1_data) %>%
  filter(country == "ES") %>%
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
  ggplot(aes(x = year, y = gini_nuts1, group = NUTS1, color = NUTS1)) +
  geom_line() +
  labs(title = "Utvikling i Gini-koeffisientene for NUTS2-regionene i Spania",
       x = "År",
       y = "Gini-koeffisient")
```



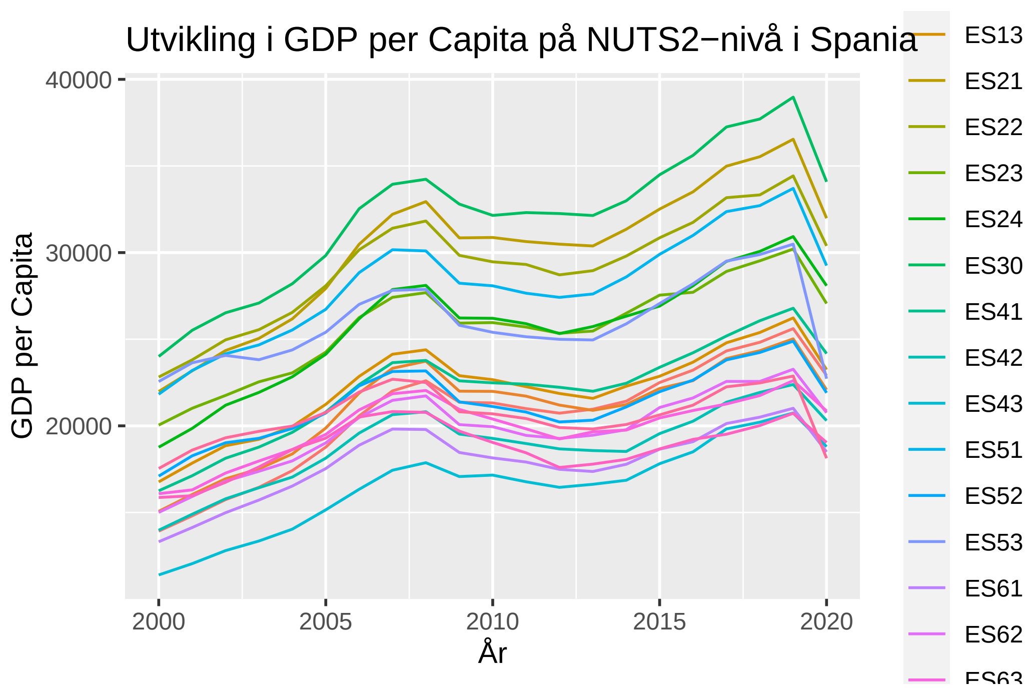
Oppgave 27

Basert på figurene over som viser Gini-koeffisientene for de spanske NUTS2-regionene, observerer vi betydelige variasjoner i inntektsulikheten mellom regionene og over tid. Dette indikerer at fordelingen av økonomisk aktivitet og inntekt i Spania har endret seg gjennom årene. Noen regioner fremviser en trend mot økende ulikhet, mens andre har mer stabile eller avtagende nivåer av ulikhet. ES7 Canarias utmerker seg med en sterk nedgang. De varierende

Gini-koeffisientene reflekterer regionsspesifikke økonomiske forhold og er en indikator på at økonomisk vekst og velstand ikke er jevnt fordelt over landet.

Oppgave 28

```
# ag_comment: Jeg vil foreslå å kjøre alt i en pipe. Gnerere plot direkte fra eu_data_nest
# Filtrer på Spania (ES) og NUTS2-nivå
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "ES") |>
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
# Plotting av gdp_per_capita for de ulike NUTS2-regionene i Spania
ggplot(
  aes(x = year, y = gdp_per_capita, group = NUTS2, color = NUTS2)
  ) +
  geom_line() +
  labs(title = "Utvikling i GDP per Capita på NUTS2-nivå i Spania",
       x = "År",
       y = "GDP per Capita")
```

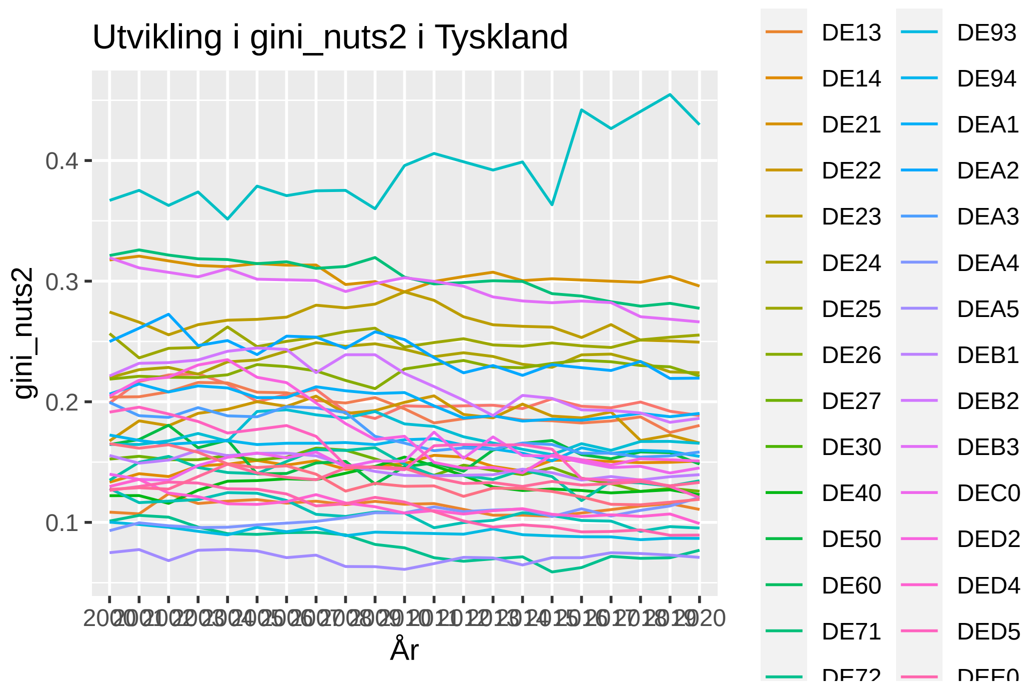


Knekken skyldes covid-19 som slo hardt ned i økonomien i Spania.

Oppgave 29

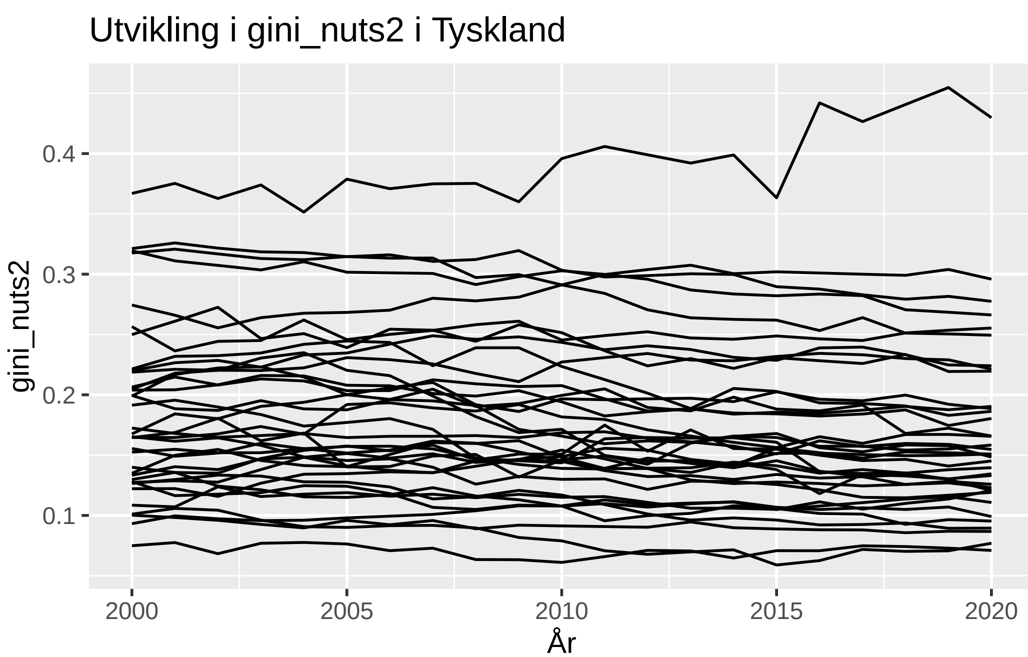
```
# Filtrer på Tyskland (DE) og NUTS2-nivå
eu_data_Germany_nuts2 <- eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "DE")

# Plotting av gdp_per_capita for de ulike NUTS2-regionene i Tyskland
ggplot(eu_data_Germany_nuts2, aes(x = year, y = gini_nuts2, group = NUTS2, color = NUTS2))
  geom_line() +
  labs(title = "Utvikling i gini_nuts2 i Tyskland",
       x = "År",
       y = "gini_nuts2")
```

```
Warning: Removed 42 rows containing missing values (`geom_line()`).
```



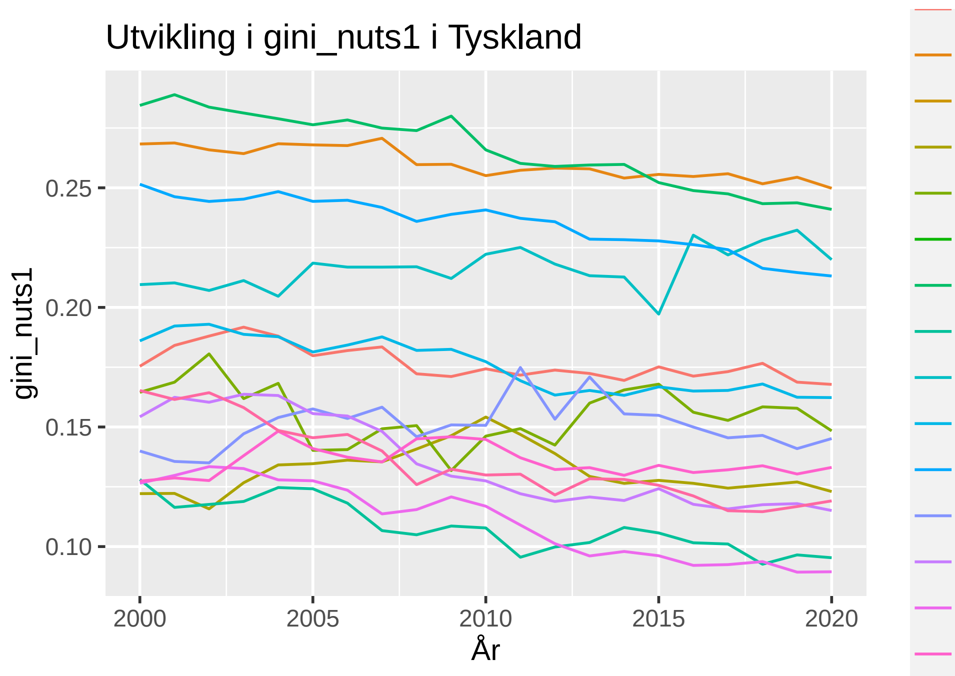
ag_comment: Jeg vil foreslå å droppe fargekoding. Tror ikke mange har sterkt nok fargesyn til å identifisere de ulike linjene.

```r
# Filtrer på Tyskland (DE) og NUTS2-nivå
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "DE") |>
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
  # Plotting av gdp_per_capita for de ulike NUTS2-regionene i Tyskland
  ggplot(
    mapping = aes(x = year, y = gini_nuts2, group = NUTS2)
    ) +
  geom_line() +
  labs(title = "Utvikling i gini_nuts2 i Tyskland",
       x = "År",
       y = "gini_nuts2")
```



Oppgave 30

```
# ag_comment: Igjen vil jeg anbefale å lage figurene on-the-fly
# Dette er ikke et objekt vi trenger å lagre for senere bruk
# Filtrer på Tyskland (DE) og NUTS1-nivå
eu_data_nested %>%
  unnest(NUTS1_data) %>%
  filter(country == "DE") |>
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
  # Plotting av gdp_per_capita for de ulike NUTS2-regionene i Tyskland
ggplot(aes(x = year, y = gini_nuts1, group = NUTS1, color = NUTS1)) +
  geom_line() +
  labs(title = "Utvikling i gini_nuts1 i Tyskland",
       x = "År",
       y = "gini_nuts1"
       )
```



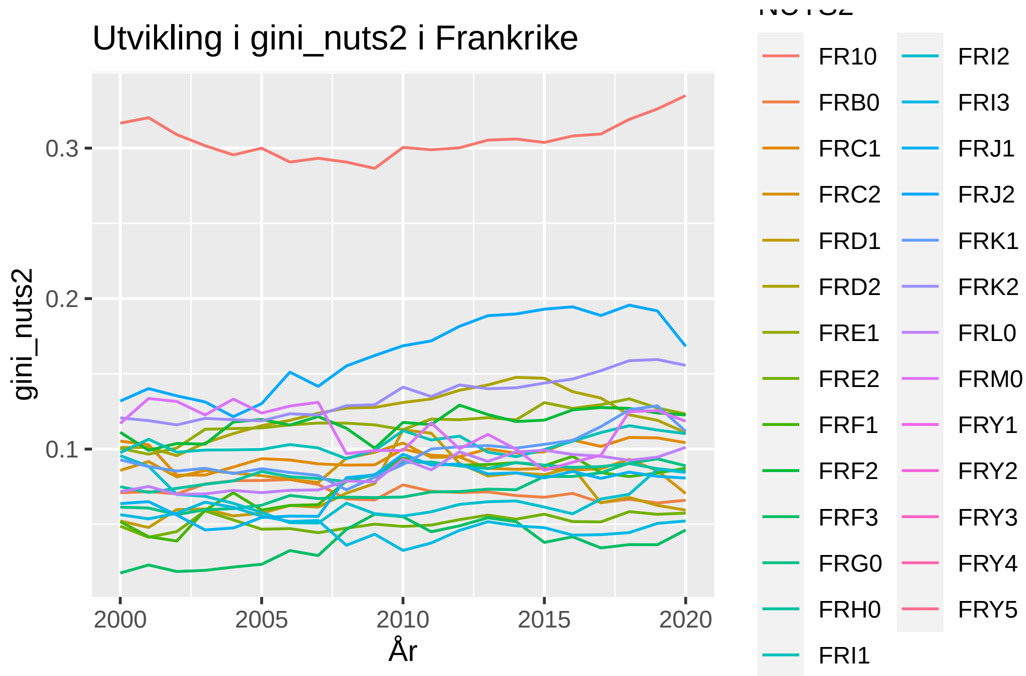
Det er nesten samme tilfelle når vi ser på de større regioner (NUTS1) i Tyskland, men vi ser at spredningen ikke er like stor.

ag_comment: Svak negativ tendens. Det vil si at forskjellene mellom NUTS2 regionene i

Tyskland har blitt redusert over tid. Altså den samme utviklingen vi så for de mindre NUTS 3 regionene. Merker oss også at gini-koeffisientene fremdeles er relativt høye for Tyskland. Dette tyder på at mye av verdiskapningen i Tyskland er konsentrert til noen regioner. En ting vi burde sjekket er hvordan disse NUTS sonene er definert. Er det f.eks slik at noen soner består hovedsaklig av landbruksområder, mens andre er industri og by-områder?

Oppgave 31

```
# Filtrer på Frankrike (FR) og NUTS2-nivå
# ag_comment: Skrevet om til on-the-fly
eu_data_nested %>%
  unnest(NUTS2_data) %>%
  filter(country == "FR") |>
  # ag_comment: Et lite triks med year sikrer fin horisontal akse
  mutate(year = make_date(year)) |>
  # Plotting av gdp_per_capita for de ulike NUTS2-regionene i Frankrike
  ggplot(
    mapping = aes(x = year, y = gini_nuts2, group = NUTS2, color = NUTS2)
    ) +
  geom_line() +
  labs(
    title = "Utvikling i gini_nuts2 i Frankrike",
    x = "År",
    y = "gini_nuts2"
    )
```




```
eu_data_nested |>
    unnest(NUTS2_data) |>
    filter(country_name == "Frankrike") |>
    filter(year == 2020) |>
    select(NUTS2, gini_nuts2) |>
    arrange(desc(gini_nuts2)) |>
    flextable()  |>
    width(1, width = 1.5) |>
    width(2, width = 1.5)
```

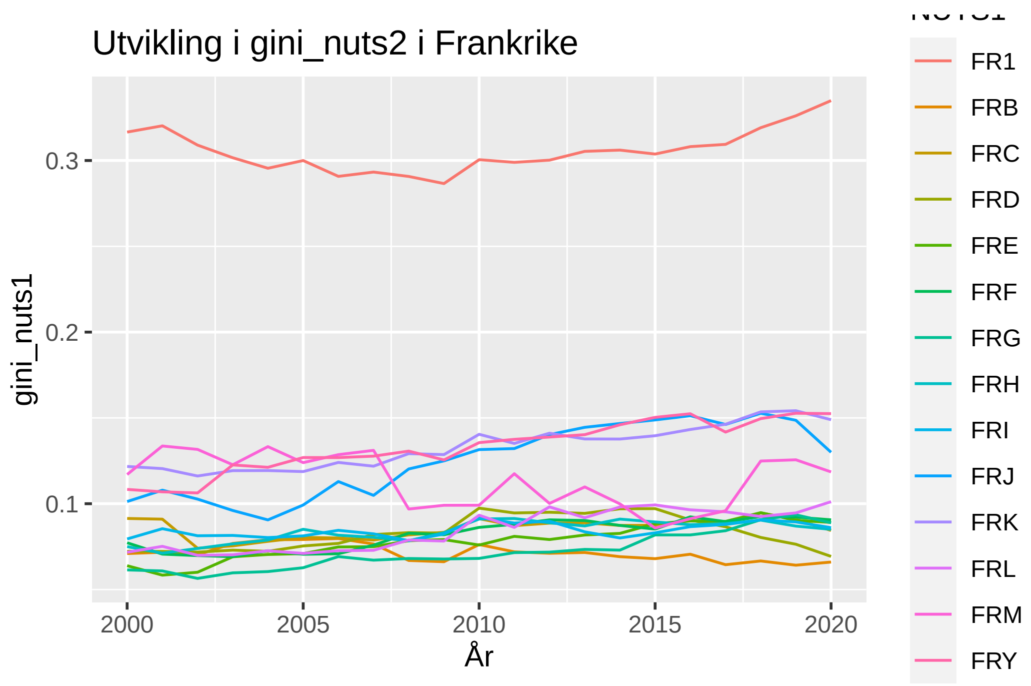
NUTS2	gini_nuts2
FR10	0.33489950
FRJ2	0.16829469
FRK2	0.15564658
FRE1	0.12329433
FRF2	0.12263434
FRM0	0.11852362
FRK1	0.11184046

NUTS2	gini_nuts2
FRD2	0.11091262
FRI1	0.11028406
FRC1	0.10422935
FRL0	0.10117125
FRG0	0.08885419
FRF1	0.08728930
FRH0	0.08506753
FRI2	0.08448774
FRJ1	0.08076450
FRC2	0.07047775
FRB0	0.06601480
FRD1	0.05941256
FRE2	0.05744966
FRI3	0.05224556
FRF3	0.04621701
FRY1	
FRY2	
FRY3	
FRY4	
FRY5	

Oppgave 32

```
# Filtrer på Frankrike (FR) og NUTS2-nivå
eu_data_Frankrike_nuts1 <- eu_data_nested %>%
  unnest(NUTS1_data) %>%
  filter(country == "FR") |>
  mutate(year = make_date(year))
```

```r
# Plotting av gininuts1 for de ulike NUTS1-regionene i Frankrike
ggplot(eu_data_Frankrike_nuts1,
       aes(x = year, y = gini_nuts1, group = NUTS1, color = NUTS1)) +
  geom_line() +
  labs(title = "Utvikling i gini_nuts2 i Frankrike",
       x = "År",
       y = "gini_nuts1")
```



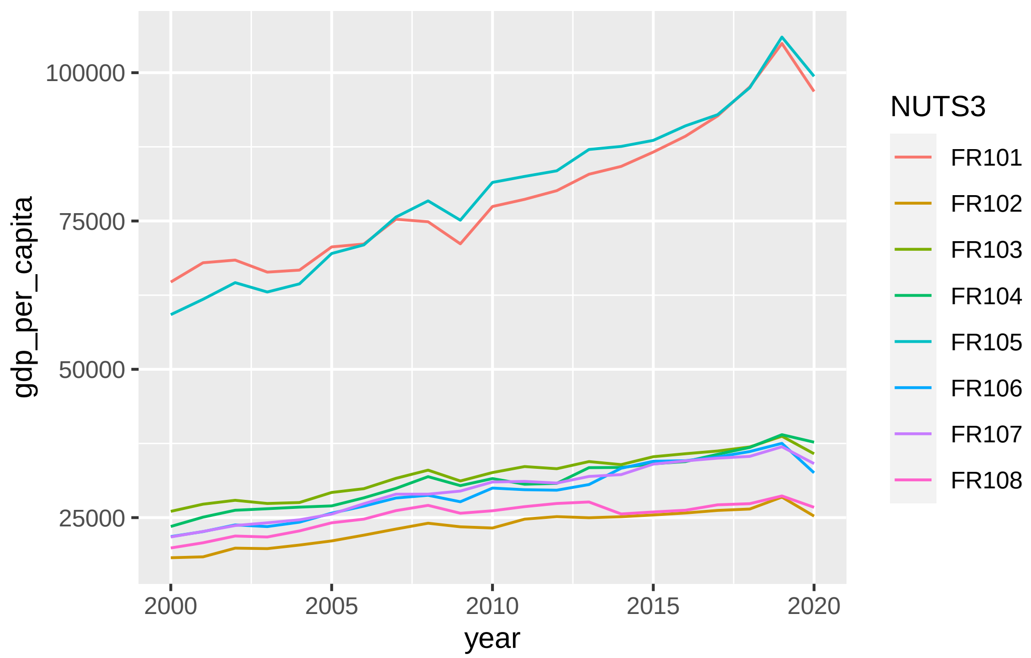
Oppgave 33

Regionen FR1 er Ile-de-France og inkluderer Paris, og er mye høyere pga. det er mye økonomisk aktivitet i denne regionen.

Oppgave 34

```r
eu_data_nested |>
  filter(country == "FR") |>
  unnest(NUTS3_data) |>
  filter(NUTS1 == "FR1") |>
```

```
  mutate(year = make_date(year)) |>
 ggplot(
   aes(
     x = year,
     y = gdp_per_capita,
     group = NUTS3,
     color = NUTS3,
   )
 ) +
 geom_line()
```



Oppgave 35

Regionen FR1 er Ile-de-France og inkluderer Paris, og er mye høyere pga. det er mye økonomisk aktivitet i denne regionen.

Oppgave 36

```
NUTS2_diff <- eu_data_nested |>
  unnest(NUTS2_data) |>
  mutate(
    # Når vi tar diff får vi en obs. mindre. Legger derfor inn en NA først
    # i vektoren
    diff_gdp_per_capita = c(NA, diff(gdp_per_capita)),
    diff_gini_nuts2 = c(NA, diff(gini_nuts2))
    ) |>
  select(country_name, country, NUTS2, year, diff_gdp_per_capita, diff_gini_nuts2) %>%
  # Fjerner obs. der vi har NA
  filter(complete.cases(.)) |>
  group_by(country_name, country, NUTS2) |>
  nest(.key = "NUTS2_diff")
```

Oppgave 37

```
NUTS2_diff <- NUTS2_diff %>%
  mutate(
    modell = map(NUTS2_diff, ~ lm(diff_gini_nuts2 ~ diff_gdp_per_capita, data = .))
  )
```

Oppgave 38

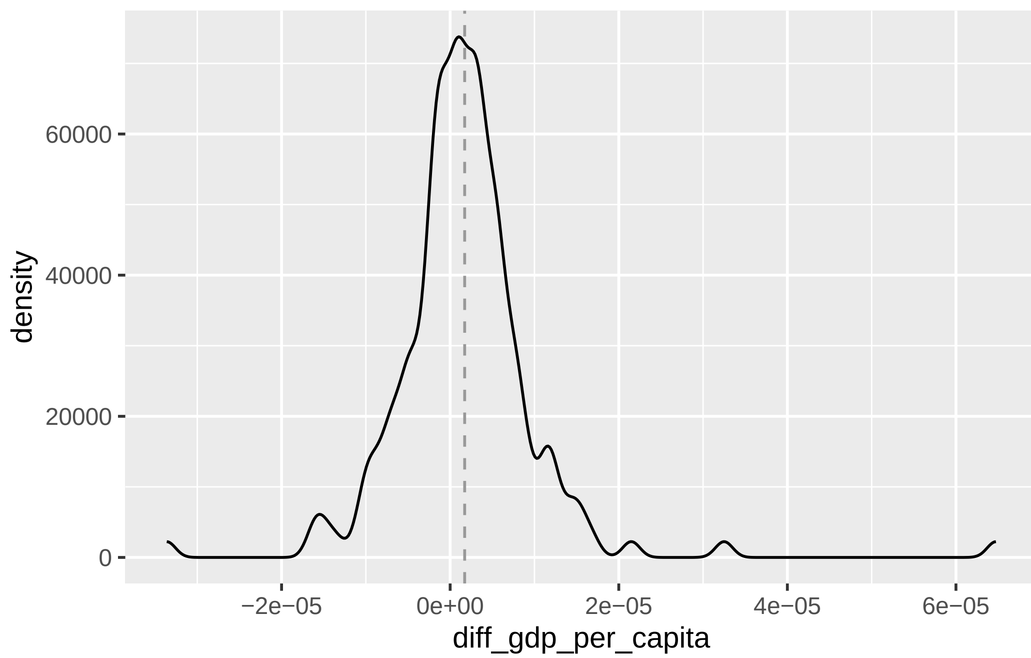
```
NUTS2_diff <- NUTS2_diff  |>
  group_by(country_name, country, NUTS2) |>
  mutate(
    mod_coeff = map_df(
      .x = modell,
      .f = coef
      )
    )
```

Oppgave 39

```
NUTS2_diff <- NUTS2_diff  |>
  group_by(country_name, country, NUTS2) |>
  mutate(
    mod_sum = map_df(
      .x = modell,
      .f = glance
      )
    )
```

Oppgave 40

```
NUTS2_diff$mod_coeff |>
  ggplot() +
  geom_density(mapping = aes(x = diff_gdp_per_capita), adjust = 0.6) +
  geom_vline(mapping = aes(xintercept = mean(diff_gdp_per_capita, na.rm = TRUE)),
                           colour = "gray60",
                           linetype = "dashed")
```



Oppgave 41

```
# Antall positive regresjonskoeffisienter
antall_positive_koef <- sum(NUTS2_diff$modell %>%
                              map_dbl(~ coef(.)["diff_gdp_per_capita"] > 0))

cat("Antall positive regresjonskoeffisienter for diff_gdp_per_capita:", antall_positive_ko
```

```
Antall positive regresjonskoeffisienter for diff_gdp_per_capita: 105
```

Oppgave 42

```
# Gjennomsnitt av regresjonskoeffisientene for diff_gdp_per_capita
gjennomsnitt_koef <- mean(NUTS2_diff$modell %>% map_dbl(~ coef(.)["diff_gdp_per_capita"]))

cat("Gjennomsnitt av regresjonskoeffisientene for diff_gdp_per_capita:", gjennomsnitt_koef
```

```
Gjennomsnitt av regresjonskoeffisientene for diff_gdp_per_capita: 1.720057e-06
```

Oppgave 43

```
t.test(NUTS2_diff$mod_coeff$diff_gdp_per_capita, y = NULL,
       alternative = c("greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95)
```

```

    One Sample t-test

data:  NUTS2_diff$mod_coeff$diff_gdp_per_capita
t = 2.6226, df = 172, p-value = 0.004755
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 6.35423e-07         Inf
sample estimates:
   mean of x
1.720057e-06
```

Ettersom p-verdien (0,004755) er mindre enn det vanlige signifikansnivået på 0.05, har vi tilstrekkelig bevis til å avvise nullhypotesen om at gjennomsnittet av diff_gdp_per_capita er lik null. Derfor kan vi konkludere med at endringene i gdp_per_capita er signifikant større enn null.

Oppgave 44

```
# Lager et nytt datasett som inneholder de nødvendige variablene vi trenger.
new_dataset <- eu_data_nested %>%
  unnest(NUTS2_data) %>%
  mutate(
    diff_gdp_per_capita = c(NA, diff(gdp_per_capita)),
    diff_gini_nuts2 = c(NA, diff(gini_nuts2))
  ) %>%
  select(country_name, country, NUTS2, year, diff_gdp_per_capita, diff_gini_nuts2) %>%
  filter(!is.na(diff_gdp_per_capita))  # Fjern rader med NA-verdier

# Panel
p_mod <- plm(diff_gini_nuts2 ~ diff_gdp_per_capita, data = new_dataset, index = c("NUTS2",
```

Oppgave 45

```
summary(p_mod)
```

```
Pooling Model

Call:
plm(formula = diff_gini_nuts2 ~ diff_gdp_per_capita, data = new_dataset,
    model = "pooling", index = c("NUTS2", "year"))

Unbalanced Panel: n = 173, T = 7-21, N = 3465

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.2881766 -0.0061217 -0.0012859  0.0041276  0.3106728

Coefficients:
                       Estimate  Std. Error t-value Pr(>|t|)
(Intercept)         -4.0028e-04  3.8405e-04 -1.0422   0.2974
```

```
diff_gdp_per_capita  3.1165e-06  1.3032e-07 23.9140   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    2.0614
Residual Sum of Squares: 1.7692
R-Squared:      0.14173
Adj. R-Squared: 0.14149
F-statistic: 571.882 on 1 and 3463 DF, p-value: < 2.22e-16
```

Panelet over viser sammenhengen mellom endringer i BNP per innbygger (**`diff_gdp_per_capita`**) og endringer i Gini-koeffisienten på NUTS2-nivå (**`diff_gini_nuts2`**). Resultatene indikerer at økonomisk vekst, representert ved **`diff_gdp_per_capita`**, har en signifikant positiv sammenheng med endringer i Gini-koeffisienten. Koeffisienten for **`diff_gdp_per_capita`** er tilnærmet lik null (3.1165e-06), og p-verdien er svært lav, noe som tyder på at økt økonomisk aktivitet er assosiert med økninger i inntektsulikheter på NUTS2-nivå. Modellen forklarer 14.17% av variansen i Gini-koeffisienten.

Oppgave 46

```r
summary(p_mod, vcov = function(x) vcovHC(x, method = "white2"))
```

```
Pooling Model

Note: Coefficient variance-covariance matrix supplied: function(x) vcovHC(x, method = "white2

Call:
plm(formula = diff_gini_nuts2 ~ diff_gdp_per_capita, data = new_dataset,
    model = "pooling", index = c("NUTS2", "year"))

Unbalanced Panel: n = 173, T = 7-21, N = 3465

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.2881766 -0.0061217 -0.0012859  0.0041276  0.3106728

Coefficients:
                       Estimate  Std. Error t-value Pr(>|t|)
(Intercept)         -4.0028e-04  3.8401e-04 -1.0424   0.2973
diff_gdp_per_capita  3.1165e-06  1.6572e-07 18.8059   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    2.0614
Residual Sum of Squares: 1.7692
R-Squared:      0.14173
Adj. R-Squared: 0.14149
F-statistic: 353.662 on 1 and 172 DF, p-value: < 2.22e-16
```

I den alternative måten å generere summary() for p_mod blir det brukt heteroskedastisitetskonsistent (HC) standardfeil beregnet ved White's metode (method = "white2"). Dette gjøres ved å inkludere vcovHC(x, method = "white2") som et argument i summary()-funksjonen. White's HC-estimator justerer for mulig heteroskedastisitet i feilleddene og gir mer robuste standardfeil, spesielt i tilfeller der det er mistanke om at feilleddene ikke har konstant varians.

Sammenlignet med den ordinære summary () gir den alternative metoden litt forskjellige standardfeil og justerte p-verdier. Dette skyldes bruken av robuste standardfeil i den alternative tilnærmingen. Koeffisienten for diff_gdp_per_capita forblir signifikant, og hovedfunnene i analysen er i stor grad ganske lik. Den robuste tilnærmingen gir mer pålitelige standardfeil når det er bekymringer for heteroskedastisitet.