Assignment1 msb104 group5

Frida Alendal og Katinca Valvatne

Table of contents

1 Del A: Subnasjonalt BNP og BNP per innbygger

1.1 Data og deskriptiv statistikk

Vi analyserer perioden **2000-2023** for **Spania (ES), Østerrike (AT), Tsjekkia (CZ), Slovenia (SI)** og **Estland (EE)** på **NUTS-3**-nivå. Data ble lastet ned fra Eurostat som tilpassede "Spreadsheet CSV"-filer den **28.10.25**. Vi beregner **BNP per innbygger (GDPC)** som BNP i euro delt på totalt innbyggertall. Eventuelle datamerker/"flagg" (f.eks. p for foreløpig) beholdes som referanse, mens elve beregningen bruker tallverdiene.

1.1.1 BNP (nama_10r_3gdp)

Regionalt BNP hentes fra `nama_10r_3gdp` med `na_item = B1GQ` (BNP i markedspriser) og `unit = MIO_EUR` (millioner euro). Tallene er årlige og rapporteres på NUTS-3. Eurostat fordeler nasjonale skatter/subsidier på produkter til regioner i tråd med metodikken i regionale nasjonalregnskaper. Aktivitet som ikke kan knyttes til en bestemt region håndteres som **extra-regio**. Serien kan inneholde **revisjoner** og mindre **brudd** (bl.a. oppdateringer i NUTS-klassifikasjon), så små sprang mellom år kan forekomme.

1.1.2 Befolkning (demo_r_pjanaggr3)

Befolning hentes fra `demo_r_pjanaggr3` med `sex = T` (totalt) og `age = TOTAL` (alle aldre). Populasjonsbegrepet er **vanlig bosted** (*ususal residence*) og referansetidspunktet er **1. januar** hvert år. Verdiene er oppgitt i **personer**. Kombinert med BNP-tabellen gir dette et konsistent grunnlag for å beregne **GDPC** på tvers av land og regioner i hele analyseperioden

1.2 Beregning av BNP per innbygger (GDP per Capita Calculation)

```
# Globale innstillinger og pakker
knitr::opts_chunk$set(fig.align = "center", fig.width = 7, fig.height = 4.5, dpi = 300)

# install.packages(c("tidyverse","janitor","eurostat","dineq")) # kjør én gang ved behov
library(tidyverse)
library(janitor)
library(stringr)
```

```
# Lastet ned data og bytter navn til gdp:

#df1 <- read.csv("demo_r_pjanaggr3__custom_18648204_spreadsheet.csv", sep = ";")
#df2 <- read.csv("nama_10r_3gdp__custom_18648452_spreadsheet.csv", sep = ";")
# Kaller den heller for gdp
gdp <- readxl::read_xlsx(
  path = "nama_10r_3gdp__custom_18648452_spreadsheet_uten_flags.xlsx",
  sheet = 'Sheet 1'
  )
```

```
head(gdp)
```

```
# A tibble: 6 x 26
  TIME...1    TIME...2   `2000` `2001` `2002` `2003` `2004` `2005` `2006` `2007`
  <chr>       <chr>       <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 GEO (Codes) GEO (Labe~    NA     NA     NA     NA     NA     NA     NA     NA
2 CZ010       Hlavní mě~ 15173. 17419. 21005. 21749. 23860. 27790. 31428. 36242.
3 CZ020       Středočes~  7366.  8312   9768.  9668. 10528. 11627. 13576. 15736.
```

```
4 CZ031       Jihočeský~  3873.  4297.  4886.  4860.  5250.  6046.  6735.  7067.
5 CZ032       Plzeňský ~  3382.  3836.  4376.  4488.  5042.  5531.  6385.  6303.
6 CZ041       Karlovars~  1713.  1860.  2123.  2140.  2227.  2470.  2669   2834.
# i 16 more variables: `2008` <dbl>, `2009` <dbl>, `2010` <dbl>, `2011` <dbl>,
#   `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>, `2016` <dbl>,
#   `2017` <dbl>, `2018` <dbl>, `2019` <dbl>, `2020` <dbl>, `2021` <dbl>,
#   `2022` <dbl>, `2023` <chr>
```

```r
# Fikser litt
names(gdp)[1] <- "geo_codes"
names(gdp)[2] <- "geo_names"
# dropper første rekken
gdp <- gdp[-1,]
# Fikser år 2023 som er character
gdp <- gdp |>
  mutate(
    `2023` = as.numeric(`2023`)
  )
```

```r
# Gjør gdp land (tidy)
gdp_long <- gdp |>
  pivot_longer(
    cols = `2000`:`2023`,
    names_to = "Year",
    values_to = "gdp"
  )
```

```r
#Lager NUTS2, NUTS1 og NUTSc (Country) variabler

gdp <- gdp_long |>
  mutate(
    NUTS2 = str_sub(geo_codes, start = 1L, end = 4L),
    NUTS1 = str_sub(geo_codes, start = 1L, end = 3L),
    NUTSc = str_sub(geo_codes, start = 1L, end = 2L)
  )
```

```r
# laste inn datasett 2, befolkning per 1. januar:

Pop <- readxl::read_excel ("demo_r_pjanaggr3__custom_18648204_spreadsheet_uten_flags.xlsx")
```

```r
library(readxl)
library(dplyr)
library(tidyr)
library(stringr)
library(readr)
library(janitor)
```

```r
# 1) Les inn på nytt slik at vi får den brede tabellen igjen
Pop <- read_excel(
  "demo_r_pjanaggr3__custom_18648204_spreadsheet_uten_flags.xlsx",
  col_types = "text"
) |>
  clean_names() |>
  rename(geo_kode = 1, geo_navn = 2)

# 2) Finn år-kolonner robust (tåler både '2000' og 'x2000')
year_cols <- names(Pop)[str_detect(names(Pop), "^\\d{4}$|^x\\d{4}$")]

# 3) Gjør alle år-kolonner til tall og smelt langt
Pop_long <- Pop |>
  filter(!str_detect(geo_kode, "^GEO"), geo_navn != "GEO (Labels)") |>
  mutate(across(all_of(year_cols), readr::parse_number)) |>
  pivot_longer(
    cols = all_of(year_cols),
    names_to = "år",
    values_to = "befolkning",
    names_transform = list(år = ~ as.integer(str_remove(.x, "^x")))
  )

# Sjekk
Pop_long |>
  slice_head(n = 10)
```

```
# A tibble: 10 x 4
   geo_kode geo_navn              år befolkning
   <chr>    <chr>              <int>      <dbl>
 1 CZ010    Hlavní město Praha  2000    1186855
 2 CZ010    Hlavní město Praha  2001    1170476
 3 CZ010    Hlavní město Praha  2002    1157876
 4 CZ010    Hlavní město Praha  2003    1157443
 5 CZ010    Hlavní město Praha  2004    1158738
 6 CZ010    Hlavní město Praha  2005    1161334
 7 CZ010    Hlavní město Praha  2006    1169892
 8 CZ010    Hlavní město Praha  2007    1173933
 9 CZ010    Hlavní město Praha  2008    1195521
10 CZ010    Hlavní město Praha  2009    1214300
```

```r
# Sjekk
gdp_long |>
  slice_head(n = 10)
```

```
# A tibble: 10 x 4
   geo_codes geo_names          Year     gdp
```

```
   <chr>     <chr>              <chr>  <dbl>
 1 CZ010     Hlavní město Praha 2000  15173.
 2 CZ010     Hlavní město Praha 2001  17419.
 3 CZ010     Hlavní město Praha 2002  21005.
 4 CZ010     Hlavní město Praha 2003  21749.
 5 CZ010     Hlavní město Praha 2004  23860.
 6 CZ010     Hlavní město Praha 2005  27790.
 7 CZ010     Hlavní město Praha 2006  31428.
 8 CZ010     Hlavní město Praha 2007  36242.
 9 CZ010     Hlavní město Praha 2008  42639.
10 CZ010     Hlavní město Praha 2009  39269.
```

```
# Viser de første radene
Pop |> dplyr::slice_head(n = 10)
```

```
# A tibble: 10 x 26
   geo_kode geo_navn x2000 x2001 x2002 x2003 x2004 x2005 x2006 x2007 x2008 x2009
   <chr>    <chr>    <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
 1 GEO (Co~ GEO (La~ <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
 2 CZ010    Hlavní ~ 1186~ 1170~ 1157~ 1157~ 1158~ 1161~ 1169~ 1173~ 1195~ 1214~
 3 CZ020    Středoč~ 1111~ 1121~ 1123~ 1128~ 1135~ 1143~ 1157~ 1174~ 1201~ 1230~
 4 CZ031    Jihočes~ :     6252~ 6242~ 6243~ 6244~ 6243~ 6260~ 6278~ 6308~ 6336~
 5 CZ032    Plzeňsk~ :     5508~ 5494~ 5491~ 5497~ 5491~ 5509~ 5538~ 5602~ 5687~
 6 CZ041    Karlova~ :     3044~ 3033~ 3035~ 3031~ 3031~ 3024~ 3023~ 3048~ 3054~
 7 CZ042    Ústecký~ :     8203~ 8187~ 8182~ 8185~ 8190~ 8191~ 8184~ 8255~ 8294~
 8 CZ051    Liberec~ :     4282~ 4271~ 4268~ 4270~ 4266~ 4278~ 4293~ 4323~ 4355~
 9 CZ052    Králové~ :     5508~ 5492~ 5483~ 5473~ 5470~ 5480~ 5492~ 5517~ 5540~
10 CZ053    Pardubi~ :     5082~ 5070~ 5062~ 5050~ 5047~ 5053~ 5069~ 5105~ 5142~
# i 14 more variables: x2010 <chr>, x2011 <chr>, x2012 <chr>, x2013 <chr>,
#   x2014 <chr>, x2015 <chr>, x2016 <chr>, x2017 <chr>, x2018 <chr>,
#   x2019 <chr>, x2020 <chr>, x2021 <chr>, x2022 <chr>, x2023 <chr>
```

```
# Sjekker begge to gdp_long og Pop_long
gdp_long |> dplyr::slice_head(n = 10)
```

```
# A tibble: 10 x 4
   geo_codes geo_names          Year     gdp
   <chr>     <chr>              <chr>  <dbl>
 1 CZ010     Hlavní město Praha 2000  15173.
 2 CZ010     Hlavní město Praha 2001  17419.
 3 CZ010     Hlavní město Praha 2002  21005.
 4 CZ010     Hlavní město Praha 2003  21749.
 5 CZ010     Hlavní město Praha 2004  23860.
 6 CZ010     Hlavní město Praha 2005  27790.
 7 CZ010     Hlavní město Praha 2006  31428.
 8 CZ010     Hlavní město Praha 2007  36242.
```

```
 9 CZ010     Hlavní město Praha 2008  42639.
10 CZ010     Hlavní město Praha 2009  39269.
```

```
Pop_long |> dplyr::slice_head(n = 10)
```

```
# A tibble: 10 x 4
   geo_kode geo_navn              år befolkning
   <chr>    <chr>              <int>      <dbl>
 1 CZ010    Hlavní město Praha  2000    1186855
 2 CZ010    Hlavní město Praha  2001    1170476
 3 CZ010    Hlavní město Praha  2002    1157876
 4 CZ010    Hlavní město Praha  2003    1157443
 5 CZ010    Hlavní město Praha  2004    1158738
 6 CZ010    Hlavní město Praha  2005    1161334
 7 CZ010    Hlavní město Praha  2006    1169892
 8 CZ010    Hlavní město Praha  2007    1173933
 9 CZ010    Hlavní město Praha  2008    1195521
10 CZ010    Hlavní město Praha  2009    1214300
```

```
library(dplyr)
library(readr)  # for parse_number

# 1) Standardiser kolonnenavn og typer -------------------------
# Anta at du har:
#   gdp_long:  geo_kode, geo_navn, år, gdp_mio_eur  (dbl)
#   Pop_long:  geo_kode, geo_navn, år, befolkning   (dbl)

gdp <- gdp_long %>%
  rename(geo = geo_codes, geo_name = geo_names, year = Year, gdp_mio_eur = gdp) %>%
  mutate(year = as.integer(year))

pop <- Pop_long %>%
  rename(geo = geo_kode, geo_name = geo_navn, year = år, pop = befolkning) %>%
  mutate(year = as.integer(year))

# 2) Slå sammen med left_join ---------------------------------
gdp_pop <- gdp %>%
  left_join(pop %>% select(geo, year, pop), by = join_by(geo, year))

# 3) Regn ut BNP per innbygger --------------------------------
# NB: gdp_mio_eur er i millioner euro, så gang med 1e6 før du deler.
gdp_pop <- gdp_pop %>%
  mutate(
    gdpc_eur = (gdp_mio_eur * 1e6) / pop,      # euro per innbygger
    gdpc_thousand = gdpc_eur / 1000            # tusen euro (ofte penere tall)
  )
```

```r
# Kjapp sanity check
gdp_pop %>% slice_head(n = 10)
```

```
# A tibble: 10 x 7
   geo   geo_name            year gdp_mio_eur     pop gdpc_eur gdpc_thousand
   <chr> <chr>              <int>       <dbl>   <dbl>    <dbl>          <dbl>
 1 CZ010 Hlavní město Praha  2000      15173. 1186855   12784.           12.8
 2 CZ010 Hlavní město Praha  2001      17419. 1170476   14882.           14.9
 3 CZ010 Hlavní město Praha  2002      21005. 1157876   18141.           18.1
 4 CZ010 Hlavní město Praha  2003      21749. 1157443   18790.           18.8
 5 CZ010 Hlavní město Praha  2004      23860. 1158738   20591.           20.6
 6 CZ010 Hlavní město Praha  2005      27790. 1161334   23930.           23.9
 7 CZ010 Hlavní město Praha  2006      31428. 1169892   26864.           26.9
 8 CZ010 Hlavní město Praha  2007      36242. 1173933   30872.           30.9
 9 CZ010 Hlavní město Praha  2008      42639. 1195521   35666.           35.7
10 CZ010 Hlavní město Praha  2009      39269. 1214300   32339.           32.3
```

```r
# 2023-oversikt for våre land (ES, AT, CZ, SI, EE)
oversikt_2023 <- gdp_pop %>%
  filter(year == 2023, grepl("^(ES|AT|CZ|SI|EE)", geo)) %>%
  arrange(desc(gdpc_eur)) %>%
  select(geo, geo_name, year, gdp_mio_eur, pop, gdpc_eur)

oversikt_2023 %>% slice_head(n = 10)
```

```
# A tibble: 10 x 6
   geo   geo_name               year gdp_mio_eur     pop gdpc_eur
   <chr> <chr>                 <int>       <dbl>   <dbl>    <dbl>
 1 CZ010 Hlavní město Praha     2023      85494. 1357326   62987.
 2 SI041 Osrednjeslovenska      2023      25144.  561407   44788.
 3 EE001 Põhja-Eesti            2023      23575.  638076   36947.
 4 CZ064 Jihomoravský kraj      2023      34587. 1217200   28415.
 5 SI044 Obalno-kraška          2023       3377.  119056   28368.
 6 SI037 Jugovzhodna Slovenija  2023       4004.  147088   27224.
 7 SI043 Goriška                2023       3217.  118436   27164.
 8 CZ020 Středočeský kraj       2023      37757. 1439391   26231.
 9 SI034 Savinjska              2023       6817.  260132   26206.
10 SI042 Gorenjska              2023       5493.  210188   26134.
```

1.3 Beskrivende analyse

1.3.1 Datagrunnlag og definisjoner

Vi analyserer BNP per innbygger (GDPC) på NUTS-3-nivå for Østerrike (AT), Tsjekkia (CZ), Estland (EE), Spania (ES) og Slovenia (SI) i perioden 2000-2023. GDPC er beregnet som regionalt

BNP i millioner euro delt på regional befolkning per 1. januar, og deretter omregnet til euro per person. Tallene er i løpende euro (ikke prisjustert og ikke i PPS), så sammenlikninger over tid og på tvers av land kan påvirkes av prisnivå og valutakurser. Datagrunnlaget kommer fra Eurostat (nama_10r_3gdp og demo_r_pjanaggr3).

```
library(dplyr)
library(ggplot2)

# Definer landene vi skal ha med (ISO2-landkoder), brukes til å filtrere datasettet vårt

land <- c("ES","AT","CZ","SI","EE")

# Trekker ut landkode fra NUTS-koden og filtrer
# Substr(geo, 1, 2): tar de 2 første tegnene i geo (landkode)
# Filter(country %in% land): beholder de landene vi trenger

gdpc <- gdp_pop %>%
  mutate(country = substr(geo, 1, 2)) %>%
  filter(country %in% land)
```

```
# Sammendrag av GDPC per land (hele perioden), grupperer i land, beregner sentraltendens og spr

stats_land_all <- gdpc %>%
  group_by(country) %>%
  summarise(
    n_obs   = n(),                                    # antall observasjoner
    mean    = mean(gdpc_eur, na.rm = TRUE),           # gjennomsnitt
    sd      = sd(gdpc_eur, na.rm = TRUE),             # standardavvik
    q1      = quantile(gdpc_eur, 0.25, na.rm = TRUE), # nedre kvartil
    median  = median(gdpc_eur, na.rm = TRUE),         # median
    q3      = quantile(gdpc_eur, 0.75, na.rm = TRUE), # øvre kvartil
    min     = min(gdpc_eur, na.rm = TRUE),            # minimum
    max     = max(gdpc_eur, na.rm = TRUE),            # maksimum
    cv      = sd/mean  # variasjonskoeffisient
  )
stats_land_all
```

```
# A tibble: 5 x 10
  country n_obs   mean    sd     q1 median     q3    min    max    cv
  <chr>   <int>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>
1 AT        864 32729. 9925. 24998. 31587. 39759. 14068. 65721. 0.303
2 CZ        336 14781. 8011. 10369. 12898. 16749.  5067. 62987. 0.542
3 EE        120 13705. 7778.  8572. 11860. 17926.  2990. 36947. 0.568
4 ES       1440 21047. 5028. 17614. 20041. 23904. 10426. 39681. 0.239
5 SI        288 16832. 5523. 12948. 15839. 19219.  8237. 44788. 0.328
```

Tabellen viser fordelingen av BNP per innbygger (GDPC, EUR) på NUTS-3-nivå for fem land,

med antall observasjoner, gjennomsnitt, standardavvik, kvartiler (Q1–median–Q3), min/maks og variasjonskoeffisient (sd/mean).

Østerrike (AT) ligger klart høyest i nivå (gj.sn. 32 700; median 31 600; Q1–Q3 25 000–39 800). Spredningen er stor (sd 9 900, CV 0,30), noe som peker mot betydelige forskjeller mellom rike byregioner og mer perifere områder. Spania (ES) har nest høyest nivå (gj.sn. 21 000, median 20 000) og den laveste relative spredningen (sd 5 000, CV 0,24). Med 1 440 region–år-observasjoner er mønsteret også mest stabilt. Slovenia (SI) ligger midt på treet (gj.sn. 16 800, median 15 800; CV 0,33) med moderat spredning. Tsjekkia (CZ) (14 800, median 12 900) og Estland (EE) (13 700, median 11 900) har lavere nivåer og klart høyere relative forskjeller (CV 0,54–0,57). Det at gjennomsnittet ligger over medianen i flere land tyder på høyreskjeve fordelinger – noen få svært produktive regioner (typisk hovedstadsområder) løfter snittet og maksverdiene.

Kort sagt: AT høyt nivå og stor ulikhet, ES jevnere fordeling, SI middels, mens CZ/EE har lavere nivå og størst relativ spredning.

```
# Behold bare observasjoner fra 2023

gdpc_2023 <- gdpc %>% filter(year == 2023)

# Sammendrag per land (2023)

stats_land_2023 <- gdpc_2023 %>%
  group_by(country) %>%
  summarise(
    n_regions = n(),
    mean = mean(gdpc_eur, na.rm = TRUE),
    sd   = sd(gdpc_eur, na.rm = TRUE),
    min  = min(gdpc_eur, na.rm = TRUE),
    max  = max(gdpc_eur, na.rm = TRUE)
  )

# Topp 10 regioner i 2023 (høyest GDPC)

topp10_2023 <- gdpc_2023 %>%
  arrange(desc(gdpc_eur)) %>%
  select(geo, geo_name, country, gdpc_eur) %>%
  slice_head(n = 10)

# Bunn 10 regioner i 2023 (lavest GDPC)

bunn10_2023 <- gdpc_2023 %>%
  arrange(gdpc_eur) %>%
  select(geo, geo_name, country, gdpc_eur) %>%
  slice_head(n = 10)

# Kjapp visning

stats_land_2023
```

```
# A tibble: 5 x 6
  country n_regions   mean     sd    min    max
  <chr>       <int>  <dbl>  <dbl>  <dbl>  <dbl>
1 AT             36   NaN     NA    Inf   -Inf
2 CZ             14 26669. 10764. 17772. 62987.
3 EE              5 22918.  8012. 17938. 36947.
4 ES             60   NaN     NA    Inf   -Inf
5 SI             12 25711.  6978. 16508. 44788.
```

```
topp10_2023
```

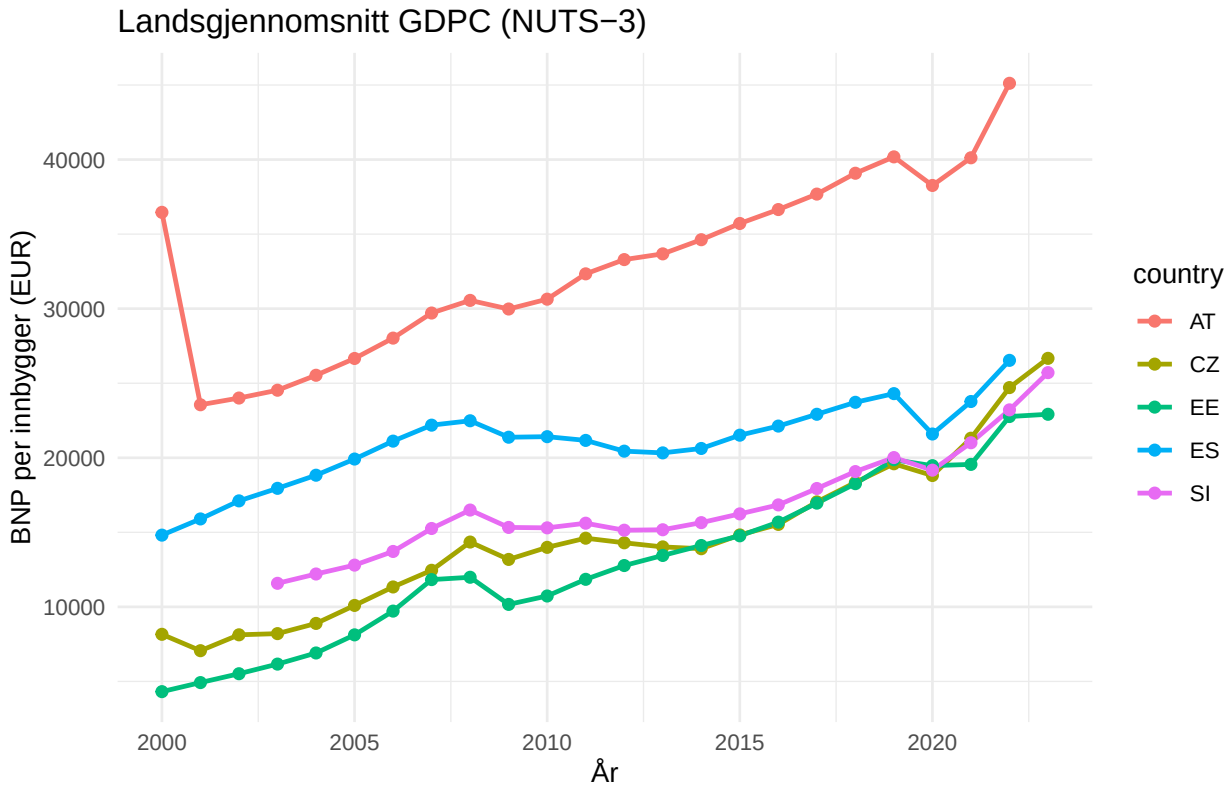
```
# A tibble: 10 x 4
   geo   geo_name              country gdpc_eur
   <chr> <chr>                 <chr>      <dbl>
 1 CZ010 Hlavní město Praha    CZ        62987.
 2 SI041 Osrednjeslovenska     SI        44788.
 3 EE001 Põhja-Eesti           EE        36947.
 4 CZ064 Jihomoravský kraj     CZ        28415.
 5 SI044 Obalno-kraška         SI        28368.
 6 SI037 Jugovzhodna Slovenija SI        27224.
 7 SI043 Goriška               SI        27164.
 8 CZ020 Středočeský kraj      CZ        26231.
 9 SI034 Savinjska             SI        26206.
10 SI042 Gorenjska             SI        26134.
```

```
bunn10_2023
```

```
# A tibble: 10 x 4
   geo   geo_name             country gdpc_eur
   <chr> <chr>                <chr>      <dbl>
 1 SI035 Zasavska             SI        16508.
 2 CZ041 Karlovarský kraj     CZ        17772.
 3 EE009 Kesk-Eesti           EE        17938.
 4 EE004 Lääne-Eesti          EE        18083.
 5 EE00A Kirde-Eesti          EE        19608.
 6 SI038 Primorsko-notranjska SI        19793.
 7 SI031 Pomurska             SI        20331.
 8 CZ051 Liberecký kraj       CZ        21306.
 9 CZ042 Ústecký kraj         CZ        21491.
10 EE008 Lõuna-Eesti          EE        22014.
```

1.3.2 Tidsutvikling (lands-gjennomsnitt 2000-2023)

```r
# Tidsutvikling av landsgjennomsnitt (2000-2023)
# Aggreger til land x år: gjennomsnittlig GDPC

gdpc_country_mean <- gdpc %>%
  group_by(country, year) %>%
  summarise(mean_gdpc = mean(gdpc_eur, na.rm = TRUE), .groups = "drop")

# Plot tidsserier per land

library(ggplot2)

ggplot(gdpc_country_mean,
       aes(x = year, y = mean_gdpc, color = country, group = country)) +
  geom_line(linewidth = 0.9) +
  geom_point(size = 1.8) +
  labs(x = "År", y = "BNP per innbygger (EUR)",
       title = "Landsgjennomsnitt GDPC (NUTS-3)") +
  theme_minimal()
```



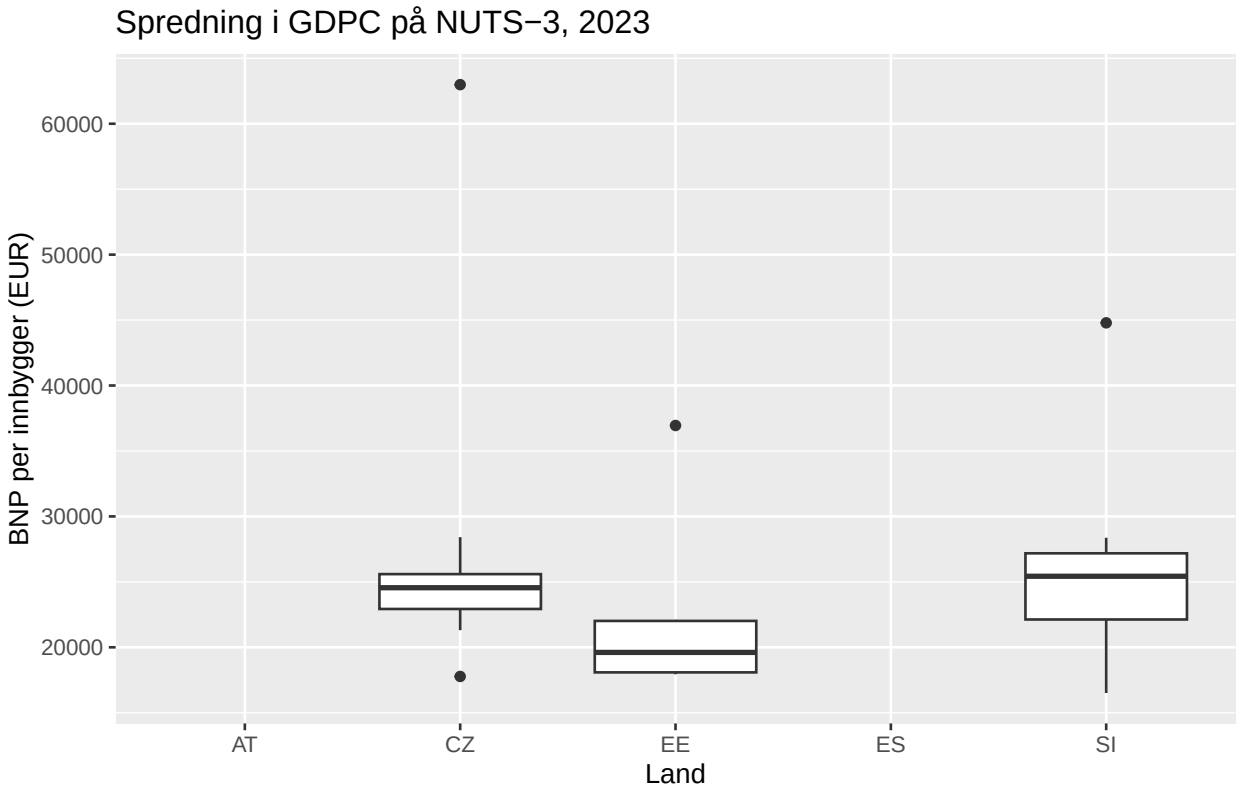
Linjeplottet med landsgjennomsnitt (NUTS-3 snitt innen hvert land) viser et tydelig mønster. Alle land har stigende GDPC over tid. AT ligger høyest gjennom hele perioden, mens EE og SI viser

tydelig innhenting. CZ og ES vokser jevnt. Eventuelle "knekkpunkt" rundt 2008-2009 samsvarer med finanskrisen før vekstbanen fortsetter.

Bildet er konsistent med økonomisk innhenting i Sentral- og Øst-Europa, samtidig som AT holder et stabilt høyere nivå.

1.3.3 Spredning internt i land - 2023

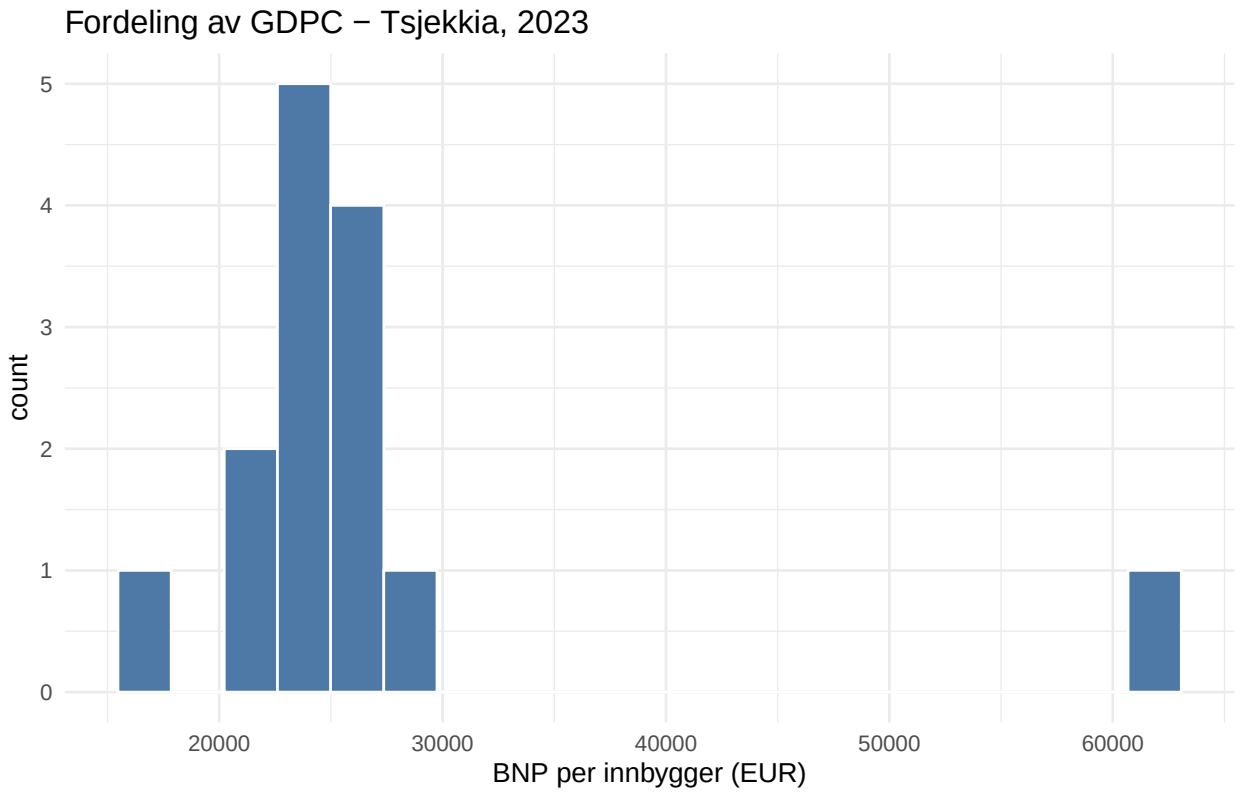
```
# Spredning per land i 2023 (boksplott på regionnivå)

ggplot(gdpc_2023, aes(country, gdpc_eur)) +
  geom_boxplot() +
  labs(x = "Land", y = "BNP per innbygger (EUR)", title = "Spredning i GDPC på NUTS-3, 2023")
```



Boksskjemaet for 2023 (GDPC på NUTS-3 innen hvert land) viser at forskjellene mellom regioner varierer på tvers av land. SI har en relativt høy median og tydelig spredning, mens noen regioner trekker opp. CZ har moderat spredning, men ett tydelig høyt punkt som ligger godt over resten. EE har lavere median enn CZ og SI, og litt smalere spredning.

1.3.4 Fordeling for ett land - Tsjekkia

```
# Fordeling for ett land (Tsjekkia) - fast farge
ggplot(
  filter(gdpc_2023, country == "CZ"),       # filtrer til Tsjekkia
  aes(gdpc_eur)                             # x-variabel: BNPI per innbygger (EUR)
) +
  geom_histogram(
    bins = 20,                              # antall «bøtter»
    color = "white",                        # kantfarge på stolpene
    fill  = "#4E79A7"                       # fyllfarge (velg hvilken som helst hex/navn)
  ) +
  labs(
    x = "BNP per innbygger (EUR)",
    title = "Fordeling av GDPC - Tsjekkia, 2023"
  ) +
  theme_minimal()
```



Histogrammet for Tsjekkia 2023 viser en høyreskjev fordeling. De fleste regioner ligger samlet i et bånd rundt 20-30 000 euro per innbygger, mens Praha er et tydelig høyt punkt som trekker maksimum opp.

1.3.5 konklusjon

Østerrike ligger høyest i GDPC gjennom hele perioden, mens Slovenia tar mest innpå. Tsjekkia og Estland har lavere nivå men større relativ spredning mellom regioner (høy CV), noe som tyder på sterk hovedstads-/kjerneeffekt. Spania er jevnere med moderat spredning. Samlet peker mønstrene på både økonomisk vekst og økende konvergens, men uten prisjustering/PPP må nivåforskjeller tolkes varsomt.

2 Del B

2.1 Litteraturoversikt

Artikkelen Lessmann Seidel fra 2017 bygger på et globalt datasett for regional inntektsulikhet ved å predikere regionale inntekter fra nattlys-satellittdata, og bruker disse til å beregne standard mål som befolkningsvektet Gini og generaliserte entropi-mål. Poenget er at lysintensitet henger stabilt sammen med regional økonomisk aktivitet, og at predikert inntekt (korrigert for målefeil som «top-coding» og «zero-coding») fanger reelle (ikke bare nominelle) forskjeller bedre enn lys alene.

De finner at om lag 67–70 % av landene opplever sigma-konvergens (fallende spredning mellom regioner) over tid, men at en ikke-triviell andel har økende ulikhet. Samtidig estimeres et N-formet (Kuznets-liknende) forhold mellom utviklingsnivå og regional ulikhet: svært fattige land får økende ulikhet i starten, mellominntektsland ser fallende ulikhet, mens de aller rikeste kan ha en svak ny økning [@lessmann_seidel_2017].

I panelanalyser finner de at naturressurser, handelsåpenhet, transportkostnader, bistand og etnisk ulikhet henger positivt sammen med regional ulikhet, mens dyrkbar mark-andel, føderalisme og humankapital (utdanning) henger negativt sammen. Praktisk for vår oppgave er også begrunnelsen for befolkningsvekting: når regioner har svært ulik størrelse/befolkning, må ulikhetsmål gi små regioner lavere vekt for å si noe meningsfullt om «inter-gruppe» ulikhet innen landet.

2.2 Beregning: Gini (befolkningsvektet) for NUTS-2 per år

Vi beregner befolkningsvektede Gini-koeffisienter for GDPC på NUTS-2 ved å bruke NUTS-3 som fordelingsenheter innen hver region og år. Dette gir et mål på regional ulikhet innad i NUTS-2, ikke mellom land, og lar oss studere hvordan intern skjevhet utvikler seg over tid.

```
# Pakker
library(dplyr)
library(stringr)
library(dineq)     # gini.wtd(x, weights = ...)

# 0) Sørg for kodekolonner
gdpc <- gdpc %>%
  mutate(
    NUTS2 = str_sub(geo, 1, 4),      # NUTS-2 fra NUTS-3 koden
    NUTS1 = str_sub(geo, 1, 3)
  )
```

```r
# 1) Kvalitetsfilter
#    - gyldige tall for gdpc_eur og pop
#    - minst 3 NUTS-3-enheter per (country, NUTS2, year)
gdpc_clean <- gdpc %>%
  filter(is.finite(gdpc_eur), is.finite(pop), pop > 0) %>%
  group_by(country, NUTS2, year) %>%
  filter(dplyr::n() >= 3) %>%
  ungroup()

# 2) Gini pr (land, NUTS2, år) - befolkningsvektet
gini_nuts2_year <- gdpc_clean %>%
  group_by(country, NUTS2, year) %>%
  summarise(
    n_nuts3 = dplyr::n(),
    mean_y  = weighted.mean(gdpc_eur, w = pop, na.rm = TRUE),
    gini_w  = gini.wtd(gdpc_eur, weights = pop),   # befolkningsvektet Gini
    .groups = "drop"
  )

# 3) Lag også 2023-snitt og ev. rangeringer
gini_2023 <- gini_nuts2_year %>%
  filter(year == 2023) %>%
  arrange(country, desc(gini_w))

gini_nuts2_year |> dplyr::slice_head(n = 10)
```

```
# A tibble: 10 x 6
   country NUTS2  year n_nuts3 mean_y gini_w
   <chr>   <chr> <int>   <int>  <dbl>  <dbl>
 1 AT      AT11   2001       3 18237. 0.0246
 2 AT      AT11   2002       3 18816. 0.0321
 3 AT      AT11   2003       3 19021. 0.0335
 4 AT      AT11   2004       3 19989. 0.0506
 5 AT      AT11   2005       3 20060. 0.0462
 6 AT      AT11   2006       3 20896. 0.0521
 7 AT      AT11   2007       3 22232. 0.0459
 8 AT      AT11   2008       3 22208. 0.0480
 9 AT      AT11   2009       3 22435. 0.0440
10 AT      AT11   2010       3 23470. 0.0469
```

I tabellen viser hver rad én NUTS-2-region per år, med antall NUTS-3-enheter (n_nuts3), befolkningsvektet gjennomsnittlit BNP per innbygger (mean_y, EUR) og befolningsvektet Gini-koefiisient (gini_w). Gini måler ulikhet i GDPC mellom NUTS-3 inni regionen (0 = helt jevnt; høyere = større forskjeller). Eksempelvis øker AT11 sin mean_y fra ~18,2 til ~23,5 tusen EUR i 2001-2010, samtidig som gini_w stiger fra 0,025 til 0,047, som tyder på økende interne forskjeller i regionen. Beregningene er gjort på gyldige (endelige) tall med befolkningsvekter, og rader med færre enn tre NUTS-3-enheter er filtrert bort for robusthet.

2.3 Befolkningsvektet Gini for BNP per innbygger (GDPC) på NUTS-2

Tabellen nedenfor viser en oppsummering per land, og i løpet av alle årene. Tabellen viser antall observasjoner (n_obs), gjennomsnitt (mean), median, standardavvik (sd) og kvartiler (q1-q3) for den årlige, befolkningsvektede Gini-koeffisient beregnet innen hver NUTS-2 (ulikhet mellom underliggende NUTs-3-regioner). Høyere Gini indikerer større interne forksjeller i GPDC.

```
# Sammendrag per land over hele perioden
gini_summary_country <- gini_nuts2_year %>%
  group_by(country) %>%
  summarise(
    n_obs   = dplyr::n(),
    mean    = mean(gini_w, na.rm = TRUE),
    median  = median(gini_w, na.rm = TRUE),
    sd      = sd(gini_w, na.rm = TRUE),
    q1      = quantile(gini_w, 0.25, na.rm = TRUE),
    q3      = quantile(gini_w, 0.75, na.rm = TRUE),
    .groups = "drop"
  )

gini_summary_country
```

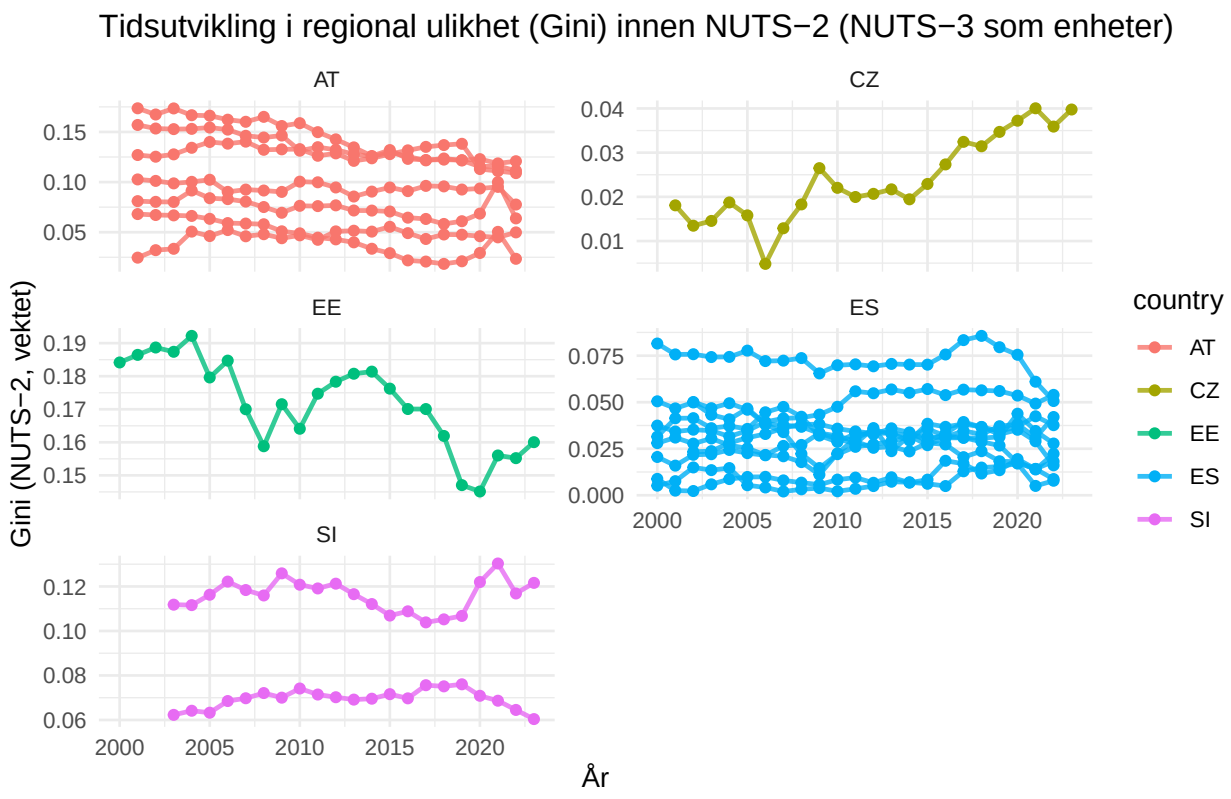
```
# A tibble: 5 x 7
  country n_obs   mean median      sd     q1     q3
  <chr>   <int>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>
1 AT        154 0.0955 0.0949 0.0412  0.0588 0.128
2 CZ         23 0.0239 0.0217 0.00952 0.0182 0.0319
3 EE         24 0.172  0.173  0.0134  0.161  0.182
4 ES        226 0.0326 0.0315 0.0190  0.0197 0.0405
5 SI         42 0.0927 0.0899 0.0243  0.0698 0.116
```

Funnene våre viser et tydelig mønster i regional ulikhet målt med befolkningsvektet Gini. Estland (EE) skiller seg ut med klart høyest intern ulikhet (gjennomsnitt ~0,172) og et relativt stabilt nivå over tid (sd ~0,013; IQR ~0,161–0,183). Østeriket (AT) og Slovenia (SI) ligger på et middels-høyt nivå (AT ~0,096; SI ~0,093), men med mer markert tidsvariasjon (AT sd ~0,041; SI sd ~0,024), som antyder perioder med både økende og avtakende regional skjevhet. Spania (ES) og Tsjekkia (CZ) har derimot lav intern ulikhet (ES ~0,033; CZ ~0,024) og smal kvartilbredde, noe som peker mot jevnere fordeling mellom NUTS-3-regioner.

2.4 Tidsutvikling i regional ulikhet

Figurene under viser tidsutviklingen i regional ulikhet målt som befolkningsvektet Gini-koeffisient for NUTS-3 innen hver NUTS-2 i de fem landene vi har observert i perioden 2000-2023. Hver liten rute er ett land, og hver linje er en NUTS-2-region i det landet. Merk at y-aksen har egen skala per land (free_y), så nivåer kan ikke sammenlignes direkte på tvers av land. Vi leser først og fremst trender og spredning.

```r
# Tidsutvikling i regional ulikhet (Gini) innen NUTS-2
# Viser utviklingen i befolkningsvektet Gini (gini_w) over tid
# En panel-figur (facet) per land, men én linje per NUTS-2 region

library(ggplot2)

ggplot(gini_nuts2_year,
       aes(x = year, y = gini_w, group = NUTS2, color = country)) +
  geom_line(linewidth = 0.9, alpha = 0.8) +
  geom_point(size = 1.6) +
  facet_wrap(~ country, ncol = 2, scales = "free_y") +
  labs(
    x = "År",
    y = "Gini (NUTS-2, vektet)",
    title = "Tidsutvikling i regional ulikhet (Gini) innen NUTS-2 (NUTS-3 som enheter)"
  ) +
  theme_minimal()
```



Mønstrene varierer tydelig. Østeriket (AT) starter høyt og faller svakt over tid,som tyder på gradvis konvergens, men ulikheten forblir relativt høy. Tsjekkia (CZ) går motsatt vei, fra lavt nivå rundt 2000 til en jevn økning i flere NUTS-2-regioner, altså økende regional divergens. Estland (EE) ligger høyt hele perioden, med et tydelig fall rundt 2008-2012 og ny oppgang de siste årene, et syklisk

mønster. Spania (ES) har lav-moderat og stabil ulikhet; linjene ligger tett og beveger seg lite, noe som peker mot en jevn fordeling mellom regioner. Slovenia (SI) ligger midt imellom, med nedgang gjennom 2010-tallet og et nylig oppsving i enkelte regioner. Samlet bekrefter dette at landene har ulike dynamikker: EE og til dels AT/SI viser mer variasjon over tid, mens ES er stabilt lavt og CZ er klart stigende.

2.5 Rangering 2023, "topp 10" og "bunn 10" NUTS-2 (befolkningsvektet Gini)

Tabellen rangerer NUTS-2-regioner i 2023 etter befolkningsvektet Gini for BNP per innbygger (høyere = mer ulikhet mellom NUTS-3 i samme NUTS-2). Kolonnen "n_nuts3" viser hvor mange NUTS-3 som inngår i beregningen. Dette sier noe om hvor robust målet er (få underenheter –> mer følsomt).

```
# Rangering 2023: "topp 10" og "bunn 10" NUTS-2 (befolkningsvektet Gini)

topp10_gini_2023 <- gini_2023 %>%
  arrange(desc(gini_w)) %>%
  slice_head(n = 10)                          # Sorterer fallende på Gini

bunn10_gini_2023 <- gini_2023 %>%
  arrange(gini_w) %>%
  slice_head(n = 10)

topp10_gini_2023
```

```
# A tibble: 4 x 6
  country NUTS2  year n_nuts3 mean_y gini_w
  <chr>   <chr> <int>   <int>  <dbl>  <dbl>
1 EE      EE00   2023       5 27958. 0.160
2 SI      SI04   2023       4 36897. 0.122
3 SI      SI03   2023       8 24117. 0.0604
4 CZ      CZ05   2023       3 23777. 0.0398
```

```
bunn10_gini_2023
```

```
# A tibble: 4 x 6
  country NUTS2  year n_nuts3 mean_y gini_w
  <chr>   <chr> <int>   <int>  <dbl>  <dbl>
1 CZ      CZ05   2023       3 23777. 0.0398
2 SI      SI03   2023       8 24117. 0.0604
3 SI      SI04   2023       4 36897. 0.122
4 EE      EE00   2023       5 27958. 0.160
```

Rangeringen bekrefter hovedmønstrene vi har sett i tidsseriene. EE00 ligger helt i toppen i 2023 (Gini 0,16), noe som peker på markert intern ulikhet mellom etiske NUTS-3. Også SI04 fremstår som

relativt ulik (Gini 0,12), mens andre slovenske NUTS-2, som SI03, ligger mer moderat (Gini 0,06). I den andre enden finner vi CZ05 med lav ulikhet (Gini 0,04). Samlet tyder dette på at Estland fortsatt er mest polarisert, Slovenia er mellomnivå men heterogent på tvers av NUTS-2, mens tsjekkiske regioner ligger stabilt lavt. Merk at vurderingen bør ta hensyn til antall underregioner (n_nuts3). Få NUTS-3 kan gi mer volatile Gini-anslag.

3 Del C: Bruk av KI-verktøy i arbeidet

Denne rapporten er utarbeidet med bistand fra KI-verktøy. Det primære verktøyet var ChatGPT (GPT-5, OpenAI), som ble brukt til å støtte både tekniske og språklige deler av arbeidet. Vi tok i bruk KI-verktøyet til å tydliggjøre oppgaveinstruksene og tolke datakrav fra Eurostat. Det ble også tatt i bruk for å foreslå struktur og formuleringer i drøftingsdelen.

Alle numeriske resultater og figurer ble generert lokalt i RStudio med R versjon 4.5.1 og Quarto-rammeverket. KI-systemet hadde ikke direkte tilgang til datasett og kjørte ikke kode på egen hånd. Endelig kode, resultater og tolkninger ble manuelt verifisert av oss selv før innlevering.

4 Kilder

Lessmann, C., & Seidel, A. (2017). Regional inequality, convergence, and its determinants – A view from outer space. European Economic Review, 92, 110–132. https://doi.org/10.1016/j.euroecorev.2016.11.009

European Commission, Eurostat. (2025). Regional gross domestic product (by NUTS 3) [nama_10r_3gdp]. Hentet 28. oktober 2025 fra https://ec.europa.eu/eurostat

European Commission, Eurostat. (2025). Population on 1 January by age group, sex and NUTS 3 region [demo_r_pjanaggr3]. Hentet 28. oktober 2025 fra https://ec.europa.eu/eurostat