

Assignment 1:

Regional GDP Inequality in 4 Selected European Economies

Kristoffer Tufta

Monday 3 Nov, 2025

```
library (tidyverse)

-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.4      v readr     2.1.5
vforcats    1.0.0      v stringr   1.5.1
v ggplot2    3.5.2      v tibble    3.3.0
v lubridate  1.9.4      v tidyrr    1.3.1
v purrr     1.1.0

-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
```

```
library (PxWebApiData)
library(readxl)
library(purrr)
library(dineq)
library(psych)
```

Attaching package: 'psych'

The following objects are masked from 'package:ggplot2':

%+%, alpha

1 Data Acquisition

In this assignment we will use four selected countries data from Eurostat to process it and analyse the sub-national GDP (gross domestic product) and population data from the years 2000-2023. Eurostat serves as the statistical office of the European Union, and their work is to collect and provide statistics on EU countries, through reliable, impartial and comparable data. The countries

in question are Germany, Switzerland, Croatia and Ireland. In these datasets we encountered missing values which we decided to keep. These NA, or missing data came from different reason for each country.

- Germany which has the most observations, lacks data for GDP from the time 2023 in a lot of its regions. This can be because of late reporting of its data to Eurostat. There are also some missing data on population during 2000-2010 and a few other regions during 2000-2023 which may be the emergence of new regions or change in their districts that require their own data.
- Ireland lacks data from the early 2000 to 2011 in population due to changes in NUTS 3 level in their regions. When it comes to the GDP, Ireland only misses data from 2015-2017 in Mid-West and South-West. This was due to confidentiality concerns.
- Croatia only have NA values on population but its spread by different regions. Same as Germany, here the lack of data can be explained by the changes in regions and districts, which may be the cause of the spread in NA values. It has also been shown that the NUTS2 regions have changed from 2007 to 2021, and the data has been reported using several different NUTS-definitions.
- Switzerland is a non.EU, but EFTA country, and have not had a data-sharing agreement with Eurostat for NUTS3 GDP from 2000-2007, while in the 2008 the NUTS classification was updated and it was standardized across all regions. Switzerland also lacks the data from 2022-2023 which may be they are waiting to finalize the data before releasing it.

We will then calculate the GDP per capita and explore regional inequality through a EDA (exploratory data analysis).

1.1 GDP per Region

In the code chunk below, we use the “read_excel”-function from the package “readxl” to fetch the downloaded GDP data from Eurostat, and put it away as a table into a function we have called “raw_econ”. We then use the table stored in the function “raw_econ” to create a tidy dataset, which we have called “tidy_econ”.

```
#Removing the metadata from the top and bottom by defining the range.
raw_econ <- read_excel("./Data/GDP_noFlag.xlsx", sheet = "Sheet 1", range = "A8:Y464", col
#Dropping the first row to align time with years
  raw_econ <- raw_econ[-1,]
  names(raw_econ)[1] <- "Geo_Labels"
  print(raw_econ)

# A tibble: 455 x 25
  Geo_Labels    `2000` `2001` `2002` `2003` `2004` `2005` `2006` `2007` `2008` 
  <chr>        <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr> 
  1 Stuttgart, St~ 35273~ 38408~ 39723~ 41115~ 40680~ 39624~ 42668~ 44532~ 42082~
```

```

2 Böblingen      13867~ 15260~ 14664~ 14967~ 14478~ 12993~ 14904~ 17899~ 17315~
3 Esslingen      14404~ 15465~ 14816~ 15216~ 15205~ 15237~ 16253~ 17230~ 17752~
4 Göppingen       6000.~ 6048.~ 6099.~ 6216.~ 6158.~ 6077.~ 6366.~ 6695.4 6850.~
5 Ludwigsburg    14657~ 15566~ 15731~ 15797~ 15851~ 16209~ 17243~ 18206~ 18580~
6 Rems-Murr-Kre~ 10367~ 10434~ 10516~ 10416~ 10602~ 10642~ 11418~ 11588~ 11752~
7 Heilbronn, St~ 5273.~ 5447.~ 5279.~ 4742.~ 4796.~ 4900.~ 5104.~ 5138.~ 5258.~
8 Heilbronn, La~ 8453.~ 8816.~ 8749.~ 9212.~ 9436.~ 9842.~ 11048~ 11650~ 11985~
9 Hohenlohekreis 3083.~ 3182.~ 3192.6 3208.~ 3302.0 3402.4 3630.~ 3894.~ 3941.~
10 Schwäbisch Ha~ 4503.~ 4525.~ 4716.~ 4634.3 4795.~ 5080.~ 5338.~ 5694.~ 5890.~

# i 445 more rows
# i 15 more variables: `2009` <chr>, `2010` <chr>, `2011` <chr>, `2012` <chr>,
#   `2013` <chr>, `2014` <chr>, `2015` <chr>, `2016` <chr>, `2017` <chr>,
#   `2018` <chr>, `2019` <chr>, `2020` <chr>, `2021` <chr>, `2022` <chr>,
#   `2023` <chr>

# Making raw_econ into long format.
tidy_econ <- pivot_longer(data = raw_econ,
                           cols = -Geo_Labels,
                           names_to = "Time",
                           values_to = "GDP Million EUR")

```

1.2 Demographic Data

Similarly as before, we do the same for the downloaded demographic dataset from Eurostat, again using the “read_excel”-function. This time we input the data into a function we have called “raw_demo”, before we make it tidy and input that into “tidy_demo”.

```

#Removing the metadata from the top.
raw_demo <- read_excel("./Data/demo_noFlag.xlsx", sheet = "Sheet 1", range = "A10:Z487", c

New names:
* `TIME` -> `TIME...1`
* `TIME` -> `TIME...2`

#Dropping the first row to align time with years
raw_demo <- raw_demo[-1,]
names(raw_demo)[1] <- "Geo_Codes"
names(raw_demo)[2] <- "Geo_Labels"
print(raw_demo)

# A tibble: 476 x 26
  Geo_Codes Geo_Labels `2000` `2001` `2002` `2003` `2004` `2005` `2006` `2007` 
  <chr>     <chr>      <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr> 
1 DE111    Stuttgart,~ 582443 583874 587152 588477 589161 590657 592569 593923
2 DE112    Böblingen   362048 364987 367830 370337 371678 372113 372155 372228
3 DE113    Esslingen   497826 500666 505340 509495 511564 513105 514245 514108

```

```

4 DE114      Göppingen  256136 256792 257651 258488 258707 258492 257783 256967
5 DE115      Ludwigsburg 495443 497764 503229 507043 509681 511830 513317 513998
6 DE116      Rems-Murr~- 407213 409296 412959 415764 416635 417463 417697 417609
7 DE117      Heilbronn,~ 119526 119305 120163 120683 120705 121320 121613 121384
8 DE118      Heilbronn,~ 317578 320955 324043 326229 327540 328866 329503 329979
9 DE119      Hohenlohe~- 106930 107754 108920 109519 109755 109756 109718 109717
10 DE11A     Schwäbisch~- 184819 185728 186967 188229 188563 189041 189580 189346
# i 466 more rows
# i 16 more variables: `2008` <chr>, `2009` <chr>, `2010` <chr>, `2011` <chr>,
#   `2012` <chr>, `2013` <chr>, `2014` <chr>, `2015` <chr>, `2016` <chr>,
#   `2017` <chr>, `2018` <chr>, `2019` <chr>, `2020` <chr>, `2021` <chr>,
#   `2022` <chr>, `2023` <chr>

# Making raw_demo into long format.
tidy_demo <- pivot_longer(data = raw_demo,
                           cols = c(-Geo_Codes, -Geo_Labels),
                           names_to = "Time",
                           values_to = "Population")

```

2 GDP Per capita:

To calculate GDP per capita we have used the NUTS-3 column as a primary key to join the tidied demographic and economic tables together. The code chunk below joins the two datasets and adds a new column called “GDP_Capita”, calculated by multiplying the “GDP Million EUR”-column by a million and dividing it by the reported population in the same year. We also add two more columns called “Country” and “NUTS2” by Extracting the first letters (which indicate country and NUTS2-region) from the NUTS3-column.

3 Descriptive analysis of the “GDP per capita”-table

```

# Descriptive Analysis grouped by country code
describeBy(
  tidyjoined,
  tidyjoined$Country
)

```

Descriptive statistics by group
group: CH

	vars	n	mean	sd	median	trimmed	mad
Geo_Codes	1	624	13.50	7.51	13.50	13.50	9.64
NUTS2	2	624	4.00	1.86	5.00	4.08	1.48

Table 1

```
# Joining the two datasets
tidyjoined <- left_join(tidy_demo, tidy_econ, by = join_by(Geo_Labels, Time), keep = FALSE)
# Mutating to add column for GDP per capita.
tidyjoined <- tidyjoined %>%
  mutate(
    `GDP Million EUR` = as.numeric(`GDP Million EUR`),
    Population = as.numeric(Population),
    GDP_Capita = (`GDP Million EUR` * 1000000) / Population
  )

Warning: There were 2 warnings in `mutate()` .
The first warning was:
i In argument: `GDP Million EUR = as.numeric(`GDP Million EUR)` .
Caused by warning:
! NAs introduced by coercion
i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.

tidyjoined <- tidyjoined %>%
  mutate(Country = substr(Geo_Codes, 1, 2), .before = 2)
tidyjoined <- tidyjoined %>%
  mutate( NUTS2 = (str_sub(Geo_Codes, start= 1L, end = 4L)), .before = 2)
print(tidyjoined)

# A tibble: 11,424 x 8
  Geo_Codes NUTS2 Country Geo_Labels      Time Population `GDP Million EUR`
  <chr>     <chr> <chr>   <chr>       <chr>     <dbl>           <dbl>
1 DE111     DE11   DE     Stuttgart, Stadtk~ 2000     582443        35274.
2 DE111     DE11   DE     Stuttgart, Stadtk~ 2001     583874        38409.
3 DE111     DE11   DE     Stuttgart, Stadtk~ 2002     587152        39723.
4 DE111     DE11   DE     Stuttgart, Stadtk~ 2003     588477        41116.
5 DE111     DE11   DE     Stuttgart, Stadtk~ 2004     589161        40680.
6 DE111     DE11   DE     Stuttgart, Stadtk~ 2005     590657        39624.
7 DE111     DE11   DE     Stuttgart, Stadtk~ 2006     592569        42668.
8 DE111     DE11   DE     Stuttgart, Stadtk~ 2007     593923        44533.
9 DE111     DE11   DE     Stuttgart, Stadtk~ 2008     597176        42082.
10 DE111    DE11   DE     Stuttgart, Stadtk~ 2009    600068        38337.
# i 11,414 more rows
# i 1 more variable: GDP_Capita <dbl>
```

	3	624	1.00	0.00	1.00	1.00	0.00
Geo_Labels	4	624	249.35	162.82	273.00	252.42	209.79
Time	5	624	12.50	6.93	12.50	12.50	8.90
Population	6	624	305461.88	319131.49	214911.50	242215.72	215700.51
GDP Million EUR	7	364	21507.00	25817.17	14450.17	16282.75	16671.43
GDP_Capita	8	364	65662.47	26094.69	59181.38	61016.74	13876.25
			min	max	range	skew	kurtosis
Geo_Codes			1.00	26.0	25.0	0.00	-1.21
NUTS2			1.00	7.0	6.0	-0.25	-1.37
Country			1.00	1.0	0.0	NaN	NaN
Geo_Labels			1.00	474.0	473.0	-0.18	-1.49
Time			1.00	24.0	23.0	0.00	-1.21
Population			14946.00	1579967.0	1565021.0	1.88	3.55
GDP Million EUR			526.26	140799.6	140273.4	2.56	7.58
GDP_Capita			30506.73	187233.7	156727.0	2.24	5.70
<hr/>							
group: DE							
	vars	n	mean	sd	median	trimmed	mad
Geo_Codes	1	9768	230.00	117.50	230.00	230.00	151.23
NUTS2	2	9768	26.54	11.78	27.00	26.43	16.31
Country	3	9768	2.00	0.00	2.00	2.00	0.00
Geo_Labels	4	9768	233.76	135.21	233.00	232.99	171.98
Time	5	9768	12.50	6.92	12.50	12.50	8.90
Population	6	9331	203245.16	235682.57	146751.00	166898.85	80591.17
GDP Million EUR	7	9202	7022.39	11430.89	4243.68	4979.97	2878.05
GDP_Capita	8	8765	32287.57	15350.99	28584.35	29874.51	10166.43
			min	max	range	skew	kurtosis
Geo_Codes			27.00	433.0	406.0	0.00	-1.20
NUTS2			8.00	45.0	37.0	0.04	-1.30
Country			2.00	2.0	0.0	NaN	NaN
Geo_Labels			2.00	476.0	474.0	0.03	-1.18
Time			1.00	24.0	23.0	0.00	-1.20
Population			33264.00	3677472.0	3644208.0	8.35	100.64
GDP Million EUR			806.67	197516.7	196710.0	7.35	72.01
GDP_Capita			10984.41	199296.2	188311.8	2.58	11.81
<hr/>							
group: HR							
	vars	n	mean	sd	median	trimmed	mad
Geo_Codes	1	840	451.00	10.11	451.00	451.00	13.34
NUTS2	2	840	47.66	1.26	48.00	47.57	1.48
Country	3	840	3.00	0.00	3.00	3.00	0.00
Geo_Labels	4	840	279.03	135.22	300.00	284.21	169.02
Time	5	840	12.50	6.93	12.50	12.50	8.90
Population	6	595	198678.28	163448.17	141186.00	164221.88	52927.34
GDP Million EUR	7	504	2166.74	3226.54	1146.18	1459.15	727.21
GDP_Capita	8	315	10596.37	4177.19	9365.78	10035.89	2752.18
			min	max	range	skew	kurtosis
Geo_Codes			434.00	468.00	34.00	0.00	-1.21
NUTS2			46.00	50.00	4.00	0.41	-0.59
Country			3.00	3.00	0.00	NaN	NaN

Geo_Labels	44.00	471.00	427.00	-0.18	-1.20	4.67
Time	1.00	24.00	23.00	0.00	-1.21	0.24
Population	42469.00	809235.00	766766.00	2.46	5.96	6700.72
GDP Million EUR	237.32	25658.09	25420.77	4.08	18.25	143.72
GDP_Capita	4141.43	33381.85	29240.42	1.75	4.54	235.36
<hr/>						
group: IE						
	vars	n	mean	sd	median	trimmed
Geo_Codes	1	192	472.50	2.30	472.50	472.50
NUTS2	2	192	52.12	0.78	52.00	52.16
Country	3	192	4.00	0.00	4.00	4.00
Geo_Labels	4	192	266.88	134.78	254.50	270.30
Time	5	192	12.50	6.94	12.50	12.50
Population	6	96	607720.80	318232.23	472295.50	552867.55
GDP Million EUR	7	185	30186.15	42352.46	13769.34	19981.87
GDP_Capita	8	89	51718.66	41164.89	35327.62	43961.75
		min	max	range	skew	kurtosis
Geo_Codes		469.00	476.0	7	0.00	-1.26
NUTS2		51.00	53.0	2	-0.22	-1.35
Country		4.00	4.0	0	NaN	NaN
Geo_Labels		51.00	455.0	404	-0.28	-1.14
Time		1.00	24.0	23	0.00	-1.22
Population		286326.00	1499179.0	1212853	1.55	1.36
GDP Million EUR		3765.33	248326.3	244561	2.83	8.57
GDP_Capita		16456.08	193838.1	177382	1.73	2.03
		se				
Geo_Codes						0.17
NUTS2						0.06
Country						0.00
Geo_Labels						9.73
Time						0.50
Population						32479.44
GDP Million EUR						3113.81
GDP_Capita						4363.47

Using the following code, we see that we have a total of 3838 NA-values in our dataset. Most due to different ways of reporting demographic and economic data, making the datasets hard to pair and leading to more NA-values in the GDP_Capita-column.

```
#NA for all countries
sum(is.na(tidyjoined))
```

[1] 3838

```
# Simple statistics for all countries combined
summary(tidyjoined)
```

Geo_Codes	NUTS2	Country	Geo_Labels
Length:11424	Length:11424	Length:11424	Length:11424
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Time	Population	GDP Million EUR	GDP_Capita
------	------------	-----------------	------------

Length:11424	Min. : 14946	Min. : 237.3	Min. : 4141
Class :character	1st Qu.: 105932	1st Qu.: 2544.7	1st Qu.: 22243
Mode :character	Median : 150867	Median : 4220.9	Median : 28681
	Mean : 212629	Mean : 7715.8	Mean : 33027
	3rd Qu.: 247409	3rd Qu.: 7596.0	3rd Qu.: 38111
	Max. :3677472	Max. :248326.3	Max. :199296
	NA's :778	NA's :1169	NA's :1891

4 Regional GDP inequality - Using light levels as a predictor of economic development

In this assignment we use reported GDP and demographic data from Eurostat to determine regional inequality in a selection of countries, but what can you do when regional income data isn't readily available? The paper "Regional inequality, convergence, and its determinants – A view from outer space" by Christian Lessmann and André Seidel aimed to find a new way of finding regional inequalities in areas without economic data – estimating regional income using satellite images of nighttime light intensity.

Their method involved using luminosity data taken from meteorological satellites from the U.S air force, and existing income data to estimate a relationship between the two variables. They then used this estimate to predict regional income for other regions where economic data was not available, and to calculate inequality indicators such as the Gini coefficient. The main takeaway from the study would be that yes – it is possible to use light as an indicator of GDP. Findings also showed that for about 70% of countries, regional gaps got smaller, while other countries saw inequality grow. They also discovered an "n-shaped" link between development and regional inequality: in early stages of growth inequality is low, for mid-income regions it rises, before it falls again in rich economies.

5 Regional GDP inequality - Calculating the Gini coefficient

To calculate the Gini coefficient for our selected countries (weighted for population) we first insert our data including GDP per capita Table 1 into a new function we have called "ginigdp" Table 2. To be able to calculate the Gini coefficient, we then have to remove NA-values from our dataset. This can be done by using the "na.omit"-function. The output is then grouped by year (variable "Time") and region (NUTS2), and sent to a summarise-function which includes our Gini calculation (done by the "gini.wtd"-function). We have also included a count-column that shows the amount of NUTS3-regions in each of the NUTS2-regions. This is to provide clarity in case we get "strange" Gini-values like 0, which we would represent ultimate equality. We will get this in all cases where there is only one NUTS3-region per NUTS2-region. The code chunk below does all this and prints the first 10 results.

Table 2

```
ginigdp <- tidyjoined %>%
  na.omit %>%
  group_by(NUTS2, Time)%>%
  summarise( Count = n(),
             gini = gini.wtd(GDP_Capita, weights = Population))

`summarise()` has grouped output by 'NUTS2'. You can override using the
`.groups` argument.

print(ginigdp)

# A tibble: 1,066 x 4
# Groups:   NUTS2 [52]
  NUTS2 Time  Count  gini
  <chr> <chr> <int> <dbl>
1 CH01  2008     3 0.139
2 CH01  2009     3 0.130
3 CH01  2010     3 0.131
4 CH01  2011     3 0.130
5 CH01  2012     3 0.128
6 CH01  2013     3 0.134
7 CH01  2014     3 0.132
8 CH01  2015     3 0.127
9 CH01  2016     3 0.121
10 CH01 2017    3 0.128
# i 1,056 more rows
```

6 Visualizing the Gini coefficient

In the sections below, we will use the newly created “ginigdp”-function where all our gini-coefficients have been stored. To do this, we first filter down to the specific country, before we send the data to ggplot, group it by NUTS2-region and visualize using geom_point and geom_line graphs.

6.1 Switzerland

```
#Gini Switzerland
#Plot for Switzerland
ginigdp %>%
  filter(startsWith(NUTS2, "CH")) %>%
ggplot(mapping = aes(x = Time, y = gini, colour = NUTS2, group = NUTS2)) +
  geom_point(mapping = aes()) +
  geom_line() +
  labs(x = "Year", y = "Gini-Coefficient", title = "Regional GDP Inequity in Switzerland",
```

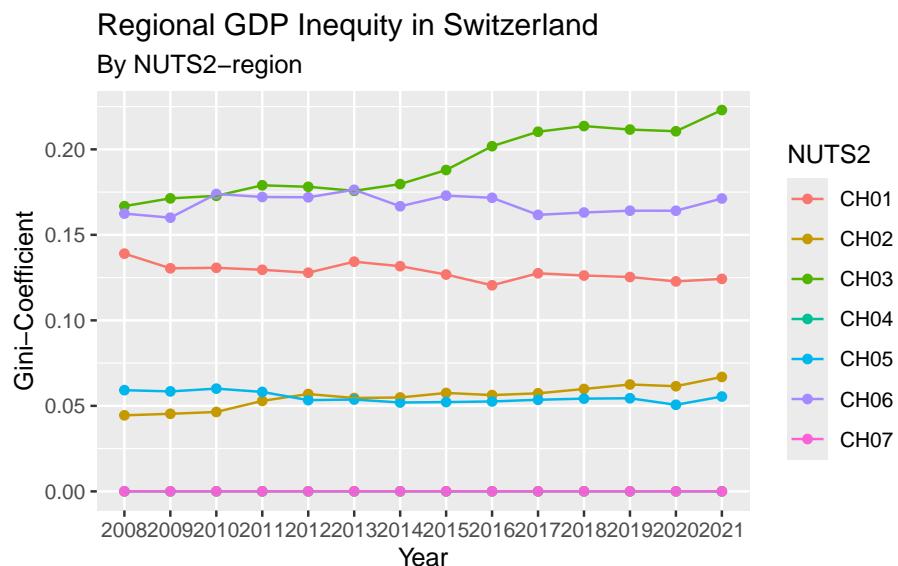


Figure 1: Regional GDP Inequity in Switzerland

Figure 1 shows low Gini-coefficients for each region, with a tendency to stay below 0.20. There is seemingly a divide into two groups one of which has a higher gini-coefficient than the other. While the overall inequality seems to remain low and consistent, we do see a divergence from the rest by CH03 who has seen growing regional inequality the last 10+ years. This seems to be because CH03 Basel-Stadt has a much higher GDP per capita growth than the surrounding areas. CH07 stays at 0 for the whole period due to only having one region.

6.2 Ireland

```
#Gini Ireland
ginigdp %>%
  filter(startsWith(NUTS2, "IE")) %>%
  ggplot(mapping = aes(x = Time, y = gini, colour = NUTS2, group = NUTS2, )) +
  geom_point(mapping = aes()) +
  geom_line() +
  labs(x = "Year", y = "Gini-Coefficient", title = "Regional GDP Inequity in Ireland", sub
```

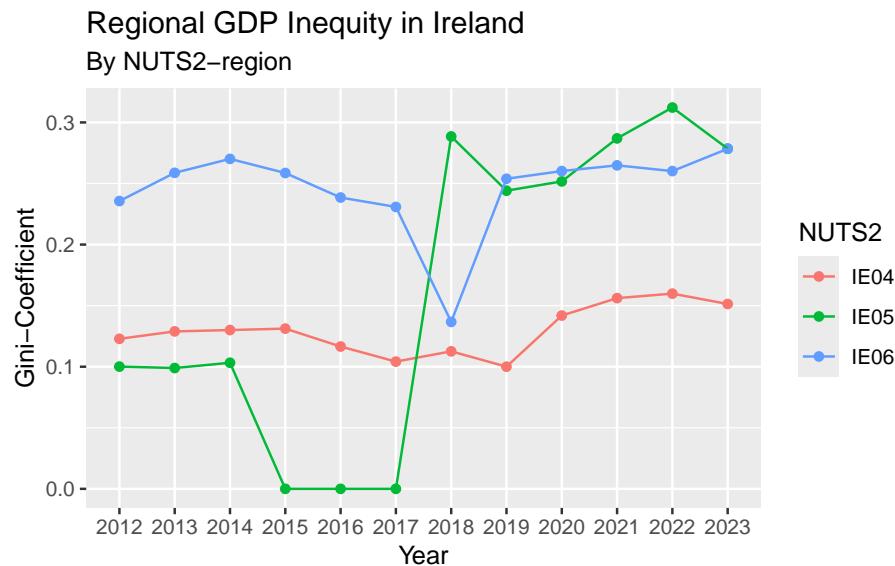


Figure 2: Regional GDP Inequity in Ireland

The Irish graph seen in Figure 2 shows a bigger spread and more “movement” than the Swiss graph. We only have population data from 2012 onwards, hence our starting point. As we can see in the graph, IE05 and IE04 start out with similarly low Gini coefficients, both lying around 0.1, whereas IE06, containing the capital Dublin, has a much less even distribution of GDP per capita. If we take a look at Table 1, we can see that for the period 2015-2017 no GDP data was reported for the Mid-West and South-West NUTS3-regions, leaving only one NUTS3-region remaining in the IE05 group, giving us a Gini coefficient of 0,0 for those years. When GDP data returned in 2018, IE053 (South-West) had grown in GDP per capita in a big way, pulling away from the rest of the group and increasing the Gini coefficient. The graph seems to show a trend towards slowly growing regional inequality of development in Ireland.

6.3 Germany

```
#Gini Germany
ginigdp %>%
  filter(startsWith(NUTS2, "DE")) %>%
  ggplot(mapping = aes(x = Time, y = gini, colour = NUTS2, group = NUTS2)) +
  geom_point(mapping = aes()) +
  geom_line() +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(x = "Year", y = "Gini-Coefficient", title = "Regional GDP Inequity in Germany", sub
```

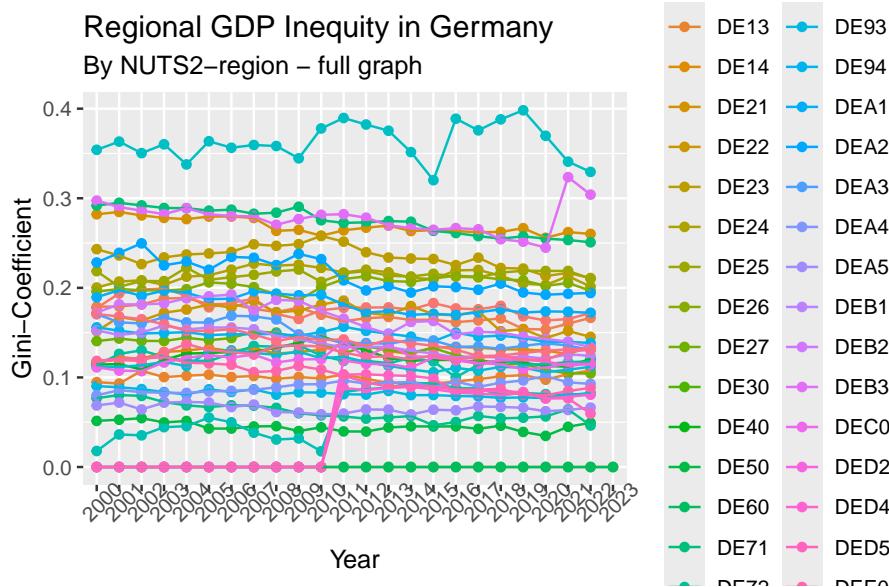


Figure 3: Regional GDP Inequity in Germany - Full graph

Germany has been very consistent at reporting data and we have data for the full period, but as Figure 3 shows, we have a little bit of an information overload on our hands. Germany consists of up to 38 NUTS2 regions which is crowding the graph, making it very hard to read. Our dataset gives a spread from 0 to 0.4 in gini-coefficient. To combat the information overload we have chosen to extract 10 random regions to get a better picture of Germany's regional inequalities.

```
# Picking out 10 different regions for Germany
# Set.seed to not get random every time one runs the codes
set.seed(123)

ginigdp %>%
  filter(startsWith(NUTS2, "DE")) %>%
  group_by(NUTS2) %>%
  summarise()
```

```

sample_n(10) %>%
pull(NUTS2) -> sampled_regions

ginigdp %>%
filter(NUTS2 %in% sampled_regions) %>%
ggplot(mapping = aes(x = Time, y = gini, colour = NUTS2, group = NUTS2)) +
geom_point() +
geom_line() +
theme(axis.text.x = element_text(angle = 45)) +
labs(x = "Year", y = "Gini-Coefficient", title = "Regional GDP Inequity in Germany", sub

```

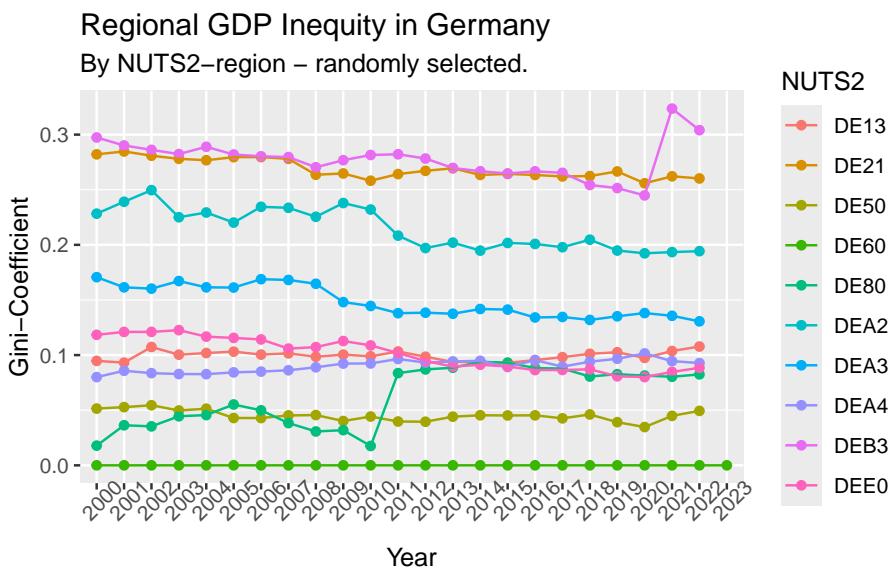


Figure 4: Regional GDP Inequity in Germany - Random selection

Figure 4 takes 10 randomly selected NUTS2 regions and shows the same spread as the one with all regions. The graph Figure 3 containing all regions from Germany showed one region with a Gini of 0,4 while most other regions lie between 0,1 and 0,2. The pattern seems to be the Gini coefficient remains stable over time, but we do have two significant “jumps”. DEB3 shows a significant jump from 0,24 in 2020 to 0,32 in 2021. This seems to be because DEB3 Mainz doubles in reported GDP, with their population remaining stable. The other regions in the group do not see a similar jump, hence inequality grows. The other jump can be seen in DE80, where we have a jump from 0,017 (extremely low) in 2010 to 0,083 in 2011. If we refer to the tidyjoined-table Table 1 , we see that population data was only reported for two of the NUTS3-regions in DE80 until 2011, giving us artificially low data for the preceeding period. The Gini coefficient for DE80 is therefore not comparable between the periods before 2010 and after 2011. DE60 only contains one NUTS3-region (Hamburg) and is therefore stable at 0.

6.4 Croatia

```
#Gini Croatia
ginigdp %>%
  filter(startsWith(NUTS2, "HR")) %>%
  ggplot(mapping = aes(x = Time, y = gini, colour = NUTS2, group = NUTS2)) +
  geom_point(mapping = aes()) +
  geom_line() +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(x = "Year", y = "Gini-Coefficient", title = "Regional GDP Inequity in Croatia", sub
```

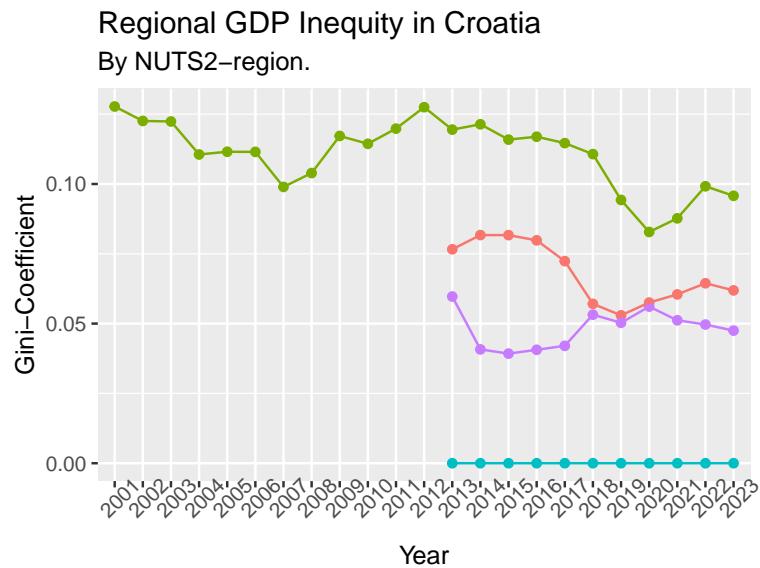


Figure 5: Regional GDP Inequity in Croatia

The graph of Croatia Figure 5 shows us a bit of a different picture from the other countries. Demographic data seems to be missing in the “ginigdp”-table for all regions but HR03 before 2013, hence the early start for HR03. Croatia has had its NUTS regions change several times between 2000 to 2023, and the data available from Eurostat contains both the old and new definitions, making it hard to pair the two datasets. For instance, the capital Zagreb appears as both HR041 and HR050 using different NUTS-definitions. Because we joined the two Eurostat-datasets by NUTS3-code, we end up getting a graph that looks incomplete, but this appears to be the best way to pair the available datasets with eachother while still limiting room for error.

If we take a look at the calculated Gini coefficients in Table 2 , we see that we get very low values for all available croatian regions. This could usually mean that we have very few NUTS3-regions per group, but if we take a look at the data, we see that this is not the case. HR06 for instance, has 5 NUTS3-regions inside of it, all with similar GDP per capita. The fact that all of Croatia’s regions have

such a low Gini coefficient could indicate that economic development is evenly spread inside the NUTS2-regions. HR05 only contains the capital Zagreb, and is therefore shown as 0.

7 Implications of our findings

In this assignment we have calculated Gini coefficients inside each available NUTS2-region using Eurostat data for Ireland, Germany, Croatia and Switzerland. The calculated variations in the Gini coefficient tell us something about the variations of gdp per capita inside each NUTS2-region. A high Gini could indicate that we have a case of a highly productive city-region inside a greater region containing a lot of less productive land.

Overall, the the calculated Gini coefficients seem to be stable over time, except for a few outliers as commented on previously. This could mean that economic development in our selected countries is mostly stable, and not especially concentrated in a handful of highly productive cities inside larger regions.

Our findings do however not entirely dismiss the idea that growth and economic development is mostly centered around cities. If we take a look at the data for Croatia in Table 1, we can see that Zagreb has been designated as its' own region, with a GDP per capita much higher than the other regions in Croatia. Had Zagreb instead been a part of any of the other regions, the calculated Gini coefficient would have been much higher than what Figure 5 showed. The same is the case for Berlin, Hamburg and many other highly productive cities, but because they are designated to their own NUTS2-regions, they end up as “blind spots” for this specific assignment and end up with a Gini of 0. To further examine this theory, it would be interesting to look at the calculated Gini coefficients based on variations of GDP per capita per NUTS2 region, which might be able to pick up these disparities.

8 The use of AI

In this assignments there was used AI to confirm through controlling questions and constructive judging the text and the codes used, to provide a constructive feedback. The AI was used as a sparring partner to help with the wording of the writing and testing of the codes to provide an explanation of how each function in the code works. The software of AI we used was ChatGPT and the model was 3.5 which is the free version. This is a quick and basic model, which means that the text produced by this type of model must be checked. It was only used to check the functioning of our codes and recommend spellings of text.