Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Finanziato nell'ambito del Piano Nazionale di Ripresa e Resilienza PNRR. Missione 4, Componente 2, Investimento 1.3 Creazione di "Partenariati estesi alle università, ai centri di ricerca, alle aziende per il finanziamento di progetti di ricerca di base"

GRINS FOUNDATION

DELIVERABLE 5.3.1

Indicators of trade, FDI, GVCs and migration flows in firms and territories

Document data	
Title	Spoke 5 Work Package 3 D5.3.1 Indicators of trade, FDI, GVCs and migration flows in firms and territories
Owner	University of Torino
Contributor/s	University of Torino Bocconi University University of Bologna Polytechnic of Milan
Document version	D5.3.1 – Final version
Last version date	01/07/2024

Executive summary

Despite the lack of uniform literature in the area, including the standards, definitions and classification of the Circular Economy (CE), it can be seen that the concept and practice of CE has emerged to overcome the drawbacks of the linear economy, which is based on take-use-dispose. On the other hand, CE is based on reuse, reduce and recycle, which is similar to a closed supply chain.

The European Commission, for example, made the circular economy a key pillar in its European Green Deal with the launch of the Circular Economy Action Plan (CEAP). Italy has been in top positions in terms of CE in Europe with actions also based on its National Recovery and Resilience Plan (NRRP), as part of the Next generation EU program, with the aim of advancing a green and sustainable transition.

In addition to the domestic plans, the international trade plays an important role in the success of such actions. The World Economic Forum argues that supply chains are the key unit of action with regard to CE implementation and success, and it will be the foundation for driving needed change. However, this new dynamic can affect the geography of trade, most probably concentrating power in large buyers and manufactures in the Global North.

This document presents the description of Indicators of trade, FDI, and GVCs flows in firms and territories developed by Bocconi University, Polytechnic of Milan, University of Bologna and University of Turin, as part of their contribution to GRINS. The indicators span a wide range of dimensions around wich the interplay between the innovation-based circular economy transition and international economic flows revolves.

First, these indicators offer insights into the susceptibility of Italian regions, at the Local Labor Systems (LLS) level, to external shocks and policies related to the circular economy, transmitted to Italian regions

through: (1) ownership linkages, and (2) Global Value Chains (GVC) linkages. Several sources of data are used, namely, Orbis and ISTAT.

Second, these indicators offer a perspective on regional participation in trade related to Circular Economy (CE) sectors through a series of indicators, namely: (1) Regional Domestic Value Added of CE sectors embedded in exports, (2) Regional Foreign Value Added of CE sectors embedded in exports, (3) Regional exports, (4) Regional backward participation of CE sectors in GVCs, (5) Regional forward participation of CE sectors in GVCs, and (6) Regional overall participation of CE sectors in GVCs. The data source used are OECD interregional input-output matrices and ISTAT.

Third, these indicators allow to: (a) assess circular economy innovation at the firm level (and at the aggregated regional level), and to (b) analyse the contribution of foreign actors in circular innovation.

Fourth, these indicators allow to gauge engagement in the circular economy and how these investments relate to international trade. By leveraging various measures, such as firm surveys, it offers a unique approach to measuring circular economy investments from the micro to the macro level. Conducted through firm-level surveys in Italy, the research provides indicators for the circular economy at both the individual firm and regional levels, offering insight into the extent of a region's involvement in circular practices. These indicators encompass four key types: circular economy indicators, green economy indicators, indicators for a broad definition of the circular economy, and indicators for willingness to invest in circular practices. Furthermore, the study delves into estimating the connection between participation in global value chains (GVC) and investment in the circular economy.

Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | GRINS FOUNDATION

TABLE OF CONTENTS

Sommario

1. Indicators of exposure to FDI by Italian Regions

We develop a set of indicators related to the exposure of territories to foreign direct investments (FDI), and their connections to the Circular Economy (CE). The indicators will cover:

i) The overall exposure of Italian regions to FDI, disaggregating by sector and economic activity;
ii) The exposure of Italian regions to FDI in *sectors* that can be identified as connected to the CE;
iii) The exposure of Italian regions to FDI in *business activities* that are directly related to the CE.

Data on FDI come from the Financial Times' fDi Markets database[1]. This database, which has been extensively used in academic research and policy, provides granular evidence at the project level on global *greenfield* FDI flows, covering all countries in the world with the data available since 2003 and up to the most recent period.

These data are collected from various sources, including the media and national investment promotion agencies. This database includes information on the location of each project, the sector and the business activity of the affiliate firm, as well as the investor's name and country of origin. The database reports information on capital expenditure and jobs created at the project level. However, most of these figures are estimated using a proprietary econometric model. Therefore, we base all the indicators presented in this report on the presence of a project and its number, allowing—with the caveats above—for replication of the analysis with capital expenditures and employment levels.

In what follows, we illustrate the construction of the main indicators focusing on inward FDI (IFDI) to Italian regions. Note that the attached code allows to replicate the same set of indicators for outward FDI (OFDI) from Italian regions. The "Readme" included as an Appendix of this report provides details on the content of the code and the replication package.

1.1 FDI at the regional level

Our initial focus is broad, presenting data on FDI to Italian regions by calculating the number of IFDI, irrespective of sector, for each year and region (see Figure 1). We rely on the Eurostat R package to visualise the data on a map.[2] Figures A1 and A2 in the Appendix replicate the same map using the amount of jobs and capital expenditures received by Italian regions.

Figure 1. FDI penetration in Italian regions.

[1] https://www.fdimarkets.com
[2] https://ropengov.github.io/eurostat/

2003

Note. The map reports the total number of IFDI projects received by each Italian region from 2003 to 2021.

Next, we calculate the number of projects received by each region in each sector (Figure 2), using the sectoral classification provided by fDiMarkets. The description of the Sectors as reported by fDiMarkets can be found in Table A1 in the Appendix. fDiMarkets also provide a more granular definition, based on a list of sub-sectors, whose description is summarised in Table A2 in the Appendix.

Figure 2 Distribution of projects by sector for each Italian region.

Note: Each dot represents one of the sectors. The number reported in each dot is the number of IFDI projects in that region and sector.

1.1.1 Indicators of FDI penetration:

We construct three main indicators:

Indicator 1. A measure of the concentration/distribution of projects across different sectors within each region:

$$I_{sr} = \frac{IFDI_{sr}}{IFDI_r}$$

Where:

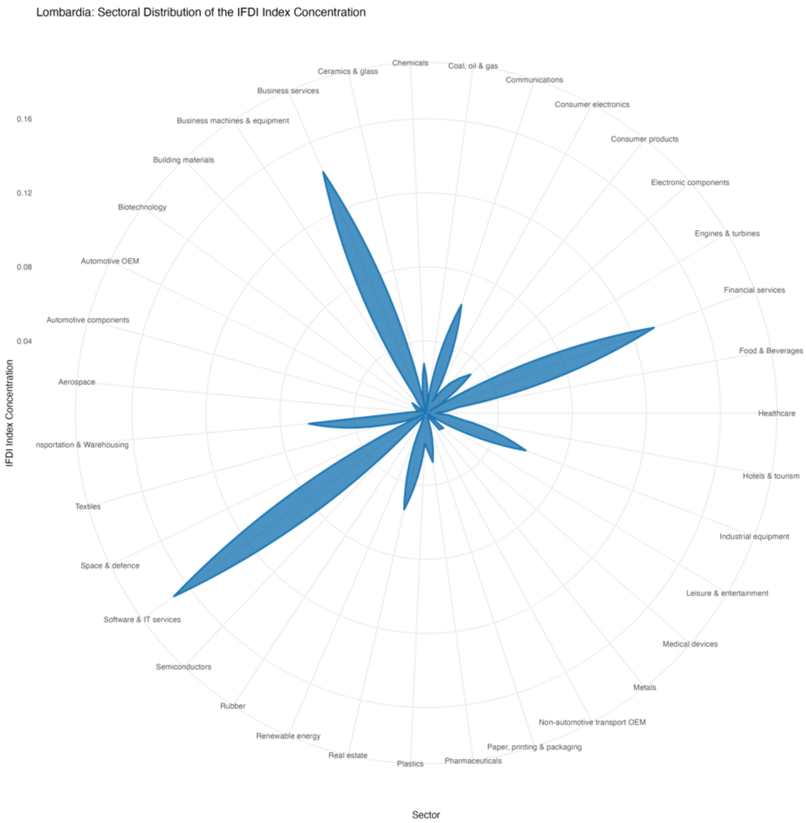
$IFDI_{sr}$ = total number of projects in sector s within region r

$IFDI_r$ = total number of projects in region r

The indicator I_{sr} represents the proportion of projects in sector s within region r. A higher value of I_{sr} for a specific sector s in a region r indicates that a larger proportion of the region's projects are concentrated in that sector. Comparing I_{sr} across different sectors within a single region allows to identify the main sectors within that region. Similarly, comparing I_{sr} values across regions for a specific sector helps identify regions focused on that sector.

We provide an example of each indicators using the case of the Lombardia region, the most attractive region in Italy. Figure 3 displays the I_{sr} for Lombardia.

Figure 3. I_{sr} for Lombardia.



Indicator 2. A measure of the proportion of projects in a given sector within a region relative to the total number of projects in that sector across all regions. This indicator explains the distribution and concentration of projects within sectors across different regions:

$$Idx_{sr} = \frac{IFDI_{sr}}{IFDI_s}$$

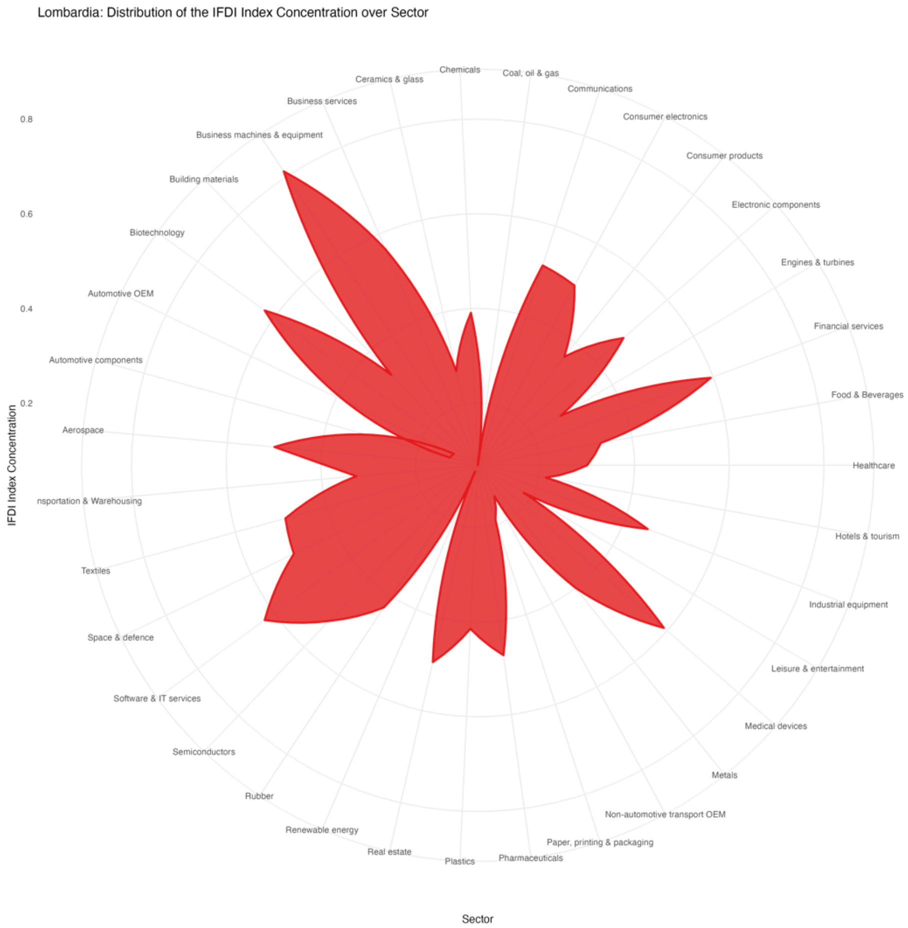
Where

$IFDI_{sr}$ = total number of projects in sector s within region r

$IFDI_s$ = total number of projects in sector s across all regions

A higher value of Idx_{sr} for a particular sector s in a region r indicates that a larger proportion of projects in that sector are concentrated within that specific region. Figure 4 reports an example for Lombardia.

Figure 4. Idx_{sr} for the case of Lombardia



Indicator 3. A measure which assesses the relative importance of IFDI projects within a specific region and sector, compared to the average values across all regions and sectors:

$$Index\_mean_{sr} = \frac{IFDI_{sr}}{\overline{IFDI_{sr}}}$$

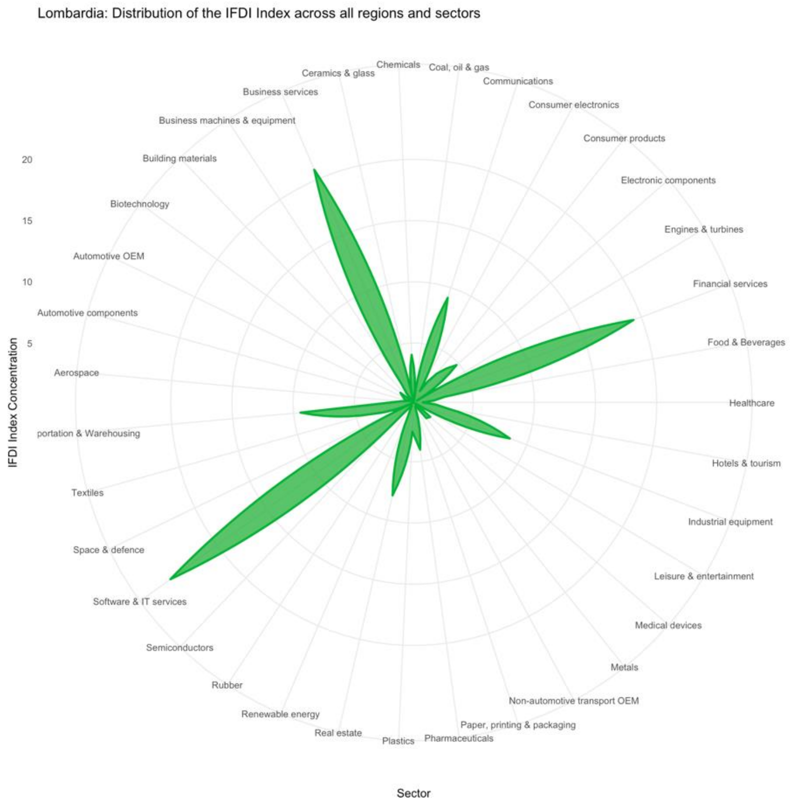
Where:

$IFDI_{sr}$ = total number of projects in sector s within region r

$\overline{IFDI_{sr}}$ = the average total number of projects in sector s within region r

This indicator provides a measure of how IFDI activity in a specific sector s and region r compares to the average across different regions and sectors. A higher value suggests a higher concentration of FDI compared to the average. Figure 5 reports an example of this indicator on the data for Lombardia.

Figure5. $Index\_mean_{sr}$ for Lombardia.



Note that the same set of indicators are also calculated using a more detailed sectoral description based on more detailed sub-sectors, as made available by the fDiMarkets data. Again, all the indicators can be replicated using the codes made available for this project using capital expenditure and employment levels in place of the number of projects.

1.1.2 FDI and Circular Economy – Sectoral Analysis

To develop FDI indicators incorporating the circular economy (CE) dimension, we mapped each project's sectoral definition with a CE definition based on the indicators used in the EU monitoring framework on the circular economy[3]. The European Commission provides documentation reporting information at the sector level (based on the NACE classification). It identifies goods and services as CE based on economic activity or the purpose of the product produced as the result of that activity. The resulting classification allows to map such activities into sectors, defined at the NACE level. Based on the latter classification, we attribute FDI projects as CE or not. Note that, also by the admission of the

[3] https://ec.europa.eu/eurostat/web/circular-economy . The applied definition of CE adopted by the EU for the specific purpose of this exercise is the following: "The circular economy goods and services sector is a sub-set of the whole economy. Economic goods and services of the circular economy sector are those that maintain the value of products and materials as long as possible and minimise waste and resource use, thereby, closing or narrowing the [raw] material cycle."

proponent of this classification, as there is no consensus on a definition of CE, this has not to be intended as an exhaustive list of activities and sectors related to the CE.

Next, we derive the first two sets of indicators:

1) The ratio of circular economy (CE) projects to total FDI projects by region. A higher ratio suggests that a larger proportion of IFDI projects in a region are aligned with circular economy principles (Index_1);
2) The proportion of CE projects relative to the average project reception within each region and subsector (Index_2).

We then use basic text analysis methods to link each sector included in the fDiMarkets dataset with its corresponding sector in the European Commission's definition of CE. We employ the quantida packages to calculate the Jaccard similarity coefficients between each pair of strings representing fDiMarkets sector/subsector and circular economic activity sectors from the European Commission's dataset. The Jaccard similarity function enables us to set a threshold for identifying potential matches with some flexibility. Figure presents the output using a similarity threshold set at 0.29, showing that only a limited number of sectors can be matched to the CE classification[4].

Figure 7 Result of the Jaccard Similarity Process

Matched Columns

IFDI	EC on Circular Economy
Subsector	CE Description
Waste management & remediation services	Remediation activities and other waste management services
Waste management & remediation services	Architectural services for wastewater and waste management projects
Waste management & remediation services	Engineering services for waste management projects;

Figure 8 provides an example based on the case of the Trentino Alto Adige region.

Figure 8 Example of Indicator of Circular economy activity at the sector level

[4] Note that this is mostly due to the fact that the definition of sectors and sub-sectors provided by fDiMarkets (Tables A1 and A2 in the Appendix) is not as disaggregated as it is the classification of the CE provided by the European Commission. fDiMarkets adopt a classification based on a proprietary scheme, which can be mapped to the NAICS system, but most of the entries correspond to 2-3 digits, and it is rare to have more detailed entries. For instance, despite raising up the threshold, we had to manually exclude the "Tyres" subsector even if it was suggested as a match to the "Retreading of Tyres", which is instead an activity listed as a CE sector by the files of the commission.

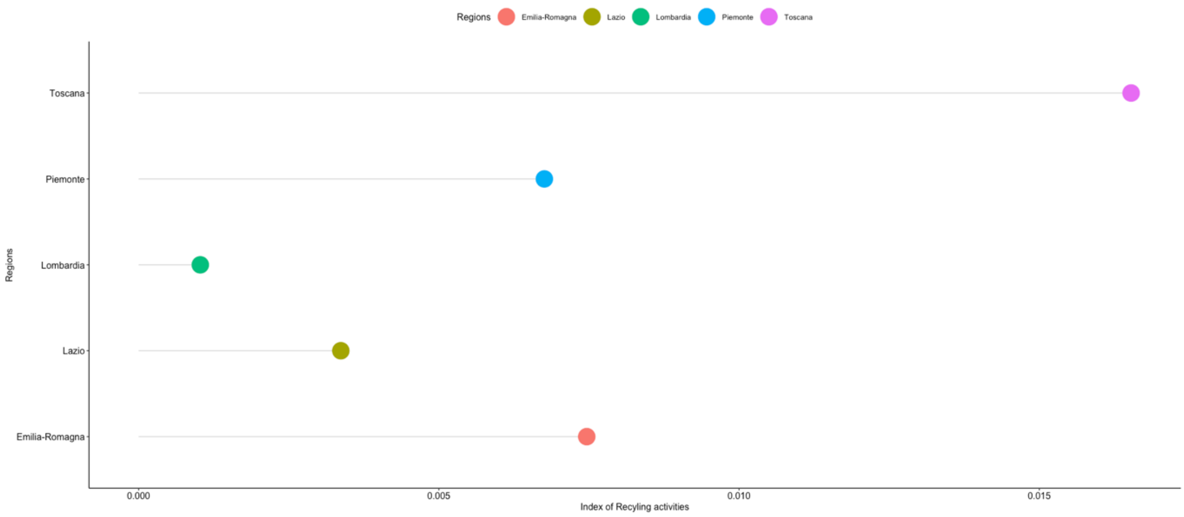
Regions	Num. of WMRS IFDI	Num. of IFDI	Index_1	Index_2
Trentino-Alto Adige	1	39	0.02564103	0.02352941

Note: 1 The 'Num. of WMRS IFDI' denotes the number of projects related to the subsector Waste management & remediation services within each region. Meanwhile, 'Num. of IFDI' reflects the total number of projects for each region.

1.1.3 FDI and Circular Economy – Analysis at the Activity level

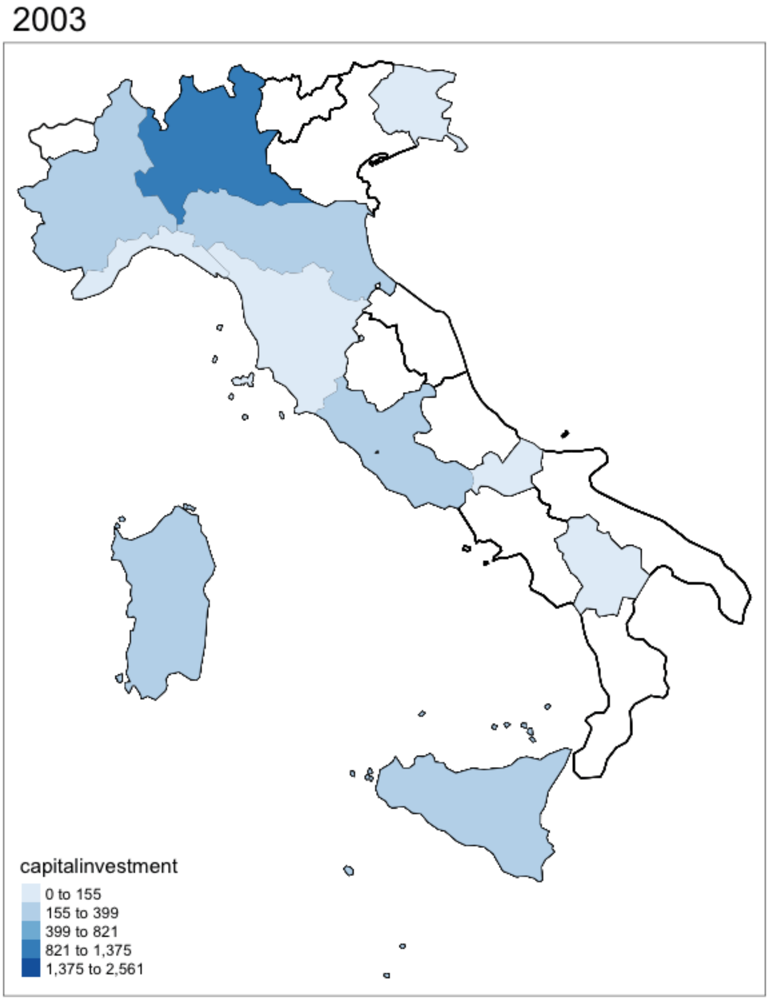
One of the strengths of the fDiMarkets database is the possibility of knowing the exact business activity of a foreign affiliate and so of an FDI project. Based on the list of activities included in the data, we identify "Recycling" as the one more closely related to the scope of the CE. Hence, we follow a procedure similar to the previous subsection and develop indicators of FDI in CE penetration based on the number of projects with recycling as the main activity. Figure 9 provides an example.

Figure 9. Indicator of Recycling activities expressed as a proportion of the total number of IFDI projects for each region



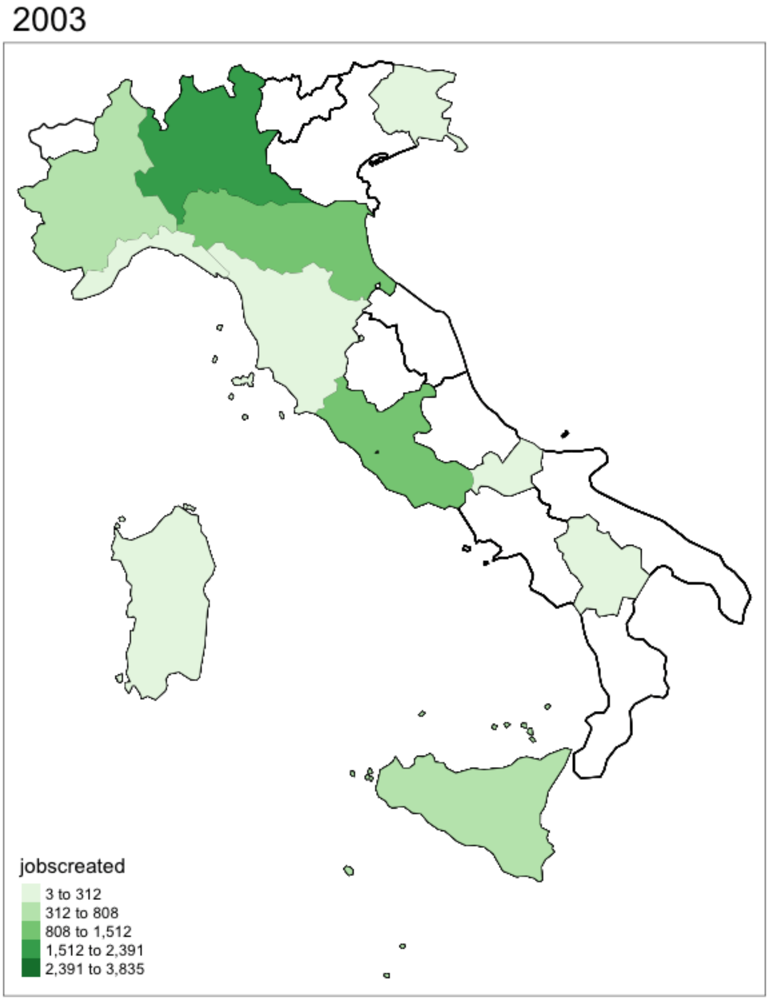
1.2 Appendix

Figure A1. Capital expenditures by FDI projects in Italian regions.



Note: The map illustrates the sum of capital expenditure (in million USD) related to inward FDI investments received by each Italian region from 2003 to 2021.

Figure A2. Jobs created by IFDI in Italian Regions.



Note: The map illustrates the total number of jobs generated by IFDI investments received by each Italian region from 2003 to 2021.

Table A1 Sectors in fDiMarkets Data

Unique Sectors in fDiMarkets
Business services
Metals
Electronic components
Software & IT services
Financial services
Business machines & equipment
Transportation & Warehousing
Pharmaceuticals
Chemicals
Hotels & tourism
Real estate
Medical devices
Paper, printing & packaging
Industrial equipment
Plastics
Aerospace
Consumer products
Renewable energy
Automotive components
Consumer electronics
Automotive OEM
Engines & turbines
Semiconductors
Communications
Textiles
Leisure & entertainment
Food & Beverages
Coal, oil & gas
Healthcare
Biotechnology
Non-automotive transport OEM
Building materials
Ceramics & glass
Minerals
Rubber
Space & defence

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
GRINS FOUNDATION

Table A2 Subsectors in fDiMarkets Data

Subsector			
Legal services	Insurance	Residential building construction	Leather & hide tanning and finishing
Gold ore & silver ore mining	Alumina & aluminium production and processing	Bakeries & tortillas	Electromedical and Electrotherapeutic Apparatus
Electric lighting equipment	All other electrical equipment & components	Measuring & control instruments	Other rubber products
Software publishers, except video games	Natural, liquefied and compressed gas	Power transmission equipment	Coating, engraving, heat treating, & allied activities
Retail banking	Advertising, PR, & related	Electrical equipment	Navigational instruments
Investment management	Wireless telecommunication carriers	Glass & glass products	Other (Aerospace)
Computer & peripheral equipment	Petroleum bulk stations & terminals	Specialised design services	Tobacco
Warehousing & storage	Converted paper products	Video games, applications and digital content	Clay product & refractory
Pharmaceutical preparations	Outpatient care centres & medical & diagnostic laboratories	Resin & artificial synthetic fibres & filaments	Other (Business machines & equipment)
Basic chemicals	Sporting goods, hobby, books & music	Other fabricated metal products	Offices of physicians, dentists, & other healthcare practitioners
Employment services	Plastics packaging materials & unlaminated film & sheets	Food & beverage Stores	Motor vehicle seating & interior trim
Accommodation	Wired telecommunication carriers	Metalworking machinery	Paper industry machinery
Freight/Distribution Services	Business support services	Aircraft engines, other parts & auxiliary equipment	Motor vehicle transmission & power train parts
Internet publishing & broadcasting & web search	Other (Consumer electronics)	Other chemical products & preparation	Food services
Commercial & institutional building construction	Steel products	Business schools, computer & management training	Coffee & tea
Professional, scientific & technical services	Other (Consumer products)	Radio & TV broadcasting	Other (Software & IT services)
Medical equipment & supplies	Data processing, hosting, & related services	Pipeline transportation of natural gas	Commercial & service industry machinery
Pulp, paper, & paperboard	Biological products (except diagnostic)	Environmental consulting services	Artificial & synthetic fibres
All other industrial machinery	Oil & gas extraction	Spring & wire products	Health & personal care stores
Urethane, foam products & other compounds	Accounting, tax preparation, bookkeeping, & payroll services	Building material & garden equipment & supplies dealers (Consumer products)	Other (Building materials)
Aircraft	Iron & steel mills & ferroalloy	Motorcycle, bicycle, & parts	All other transportation (Non-Automotive OEM)
Jewellery & silverware	Biomass power	Other (Food & Beverages)	Other (Real estate)
Wind electric power	Medicinal & botanical	Architectural, engineering, & related services	Boiler, tank, & shipping container
Printing & related activities	Motor vehicle body & trailers	Breweries & distilleries	Rental & leasing services
Motor vehicle steering & suspension components	Ships & boats	Motion picture & sound recording industries	Other (Textiles)
Household appliances	Plastic bottles	Grains & oilseed	General merchandise stores
Plastics & rubber industry machinery	Other electric power generation (Coal, oil and gas)	Nonmetallic mineral mining & quarrying	Other leather & allied products
Newspaper, periodical, book, & directory publishers	Fossil fuel electric power	Nursing & residential care facilities	Asphalt paving, roofing, & saturated materials
Custom computer programming services	Travel arrangement & reservation services	Postal service	Other (Paper, printing & packaging)
Air transportation	Soap, cleaning compounds, & toilet preparation	Other non-metallic mineral products	Other (Space & defence)
Geothermal electric power	Sugar & confectionary products	Tyres	Other telecommunications
Light trucks & utility vehicles	Industrial building construction	Wineries	Seafood products
Engines & Turbines	Textiles & Textile Mills	In-Vitro diagnostic substances	General medical & surgical hospitals
Automobiles	Schools, colleges, universities, & professional schools	Cosmetics, perfume, personal care & household products	Gambling industries
Semiconductors & other electronic components	Computer systems design services	Ventilation, heating, air conditioning, and commercial refrigeration equipment manufacturing	Other plastics products
General purpose machinery	Rail transportation	Animal food	Batteries
Paints, coatings, additives & adhesives	Support activities for mining & energy (Coal, oil & gas)	Rubber hoses & belting	Waste management & remediation services
Communications equipment	Apparel accessories & other apparel	Satellite telecommunications	Technical, trade & other schools
Other motor vehicle parts	Soft drinks & ice	Footwear	Textile machinery
Corporate & investment banking	Real estate services	Audio & video equipment (Electronic components)	Heavy duty trucks
Clothing & clothing accessories	Water transportation	Agriculture, construction, & mining machinery	
Machine shops, turned products, screws, nuts & bolts	Other support services	Dolls, toy, & games	
Motor vehicle gasoline engines & engine parts	Dairy products	Truck transportation	
Motor vehicle electrical & electronic equipment	Management consulting services	Hydroelectric power	
Printing machinery & equipment	Other petroleum & coal products	Heavy & civil engineering	
Furniture, homeware & related products (Consumer Products)	Solar electric power	Amusement & theme parks	
Performing arts, spectator sports, & related	Cement & concrete products	Food product machinery	
Motor vehicle brake systems	Railroad rolling stock	Pesticide, fertilisers & other agricultural chemicals	
Fruits & vegetables & specialist foods	Wiring devices	Support activities for transportation	
Laminated plastics plates, sheets & shapes	Audio & video equipment (Consumer electronics)	Nonstore retailers	

Table A3. Activities in fDiMarkets Data

Activity
Business Services
Extraction
Sales, Marketing & Support
Logistics, Distribution & Transportation
Manufacturing
Construction
Research & Development
Education & Training
Electricity
Customer Contact Centre
Maintenance & Servicing
Headquarters
ICT & Internet Infrastructure
Technical Support Centre
Recycling

2. Environmental Performance Exposure Gap

We propose four different indicators of the exposure gap. All the indicators use firm-level information downloaded from Moody's ORBIS database. We accessed the database in July 2023 and, due to the constraints of the standard ORBIS subscription, we had backward access to firm-level data for only a reference period of 10 years prior to the access date.

To complement the firm-level dataset, which is later aggregated, we use direct CO2 input requirements from EXIOBASE, a Multi-Regional Environmentally Extended Input-Output table. These tables provide emission intensity and resource usage at the industry level (NACE Rev. 2 – 2 digits) for a large number of countries. We assign to each firm and its subsidiaries (owners) the direct CO2 emissions estimated to be prevalent in the industry in the country where those subsidiaries (owners) operate. Aggregating across ownership groups allows us to obtain an indicator of the environmental performance of the group.

Two caveats are worth pointing out:

(i) Since EXIOBASE provides the description of the NACE sector but not the code identifiers, we did some basic encoding to assign a NACE 2-digit code to each sector listed in EXIOBASE.

(ii) We exploit the geographical detail of the firm data to build indicators at the provincial, regional, and ISTAT-consistent macro-regional levels. To do so, we had to match and harmonize province and regional details using ISTAT shapefiles, which are available on their website.

These two caveats lead to a disclaimer: firm-level data from ORBIS might be incomplete and may not report information on either NACE codes or the location of either a subsidiary or an owner firm. If either of these cases arises, we chose to drop the observation from the sample. When constructing the exposure gap indicators, we focused solely on CO2 emissions. However, EXIOBASE provides other information that can be readily incorporated into the estimator. Given their relevance to the circular economy, we identified Water Withdrawal, Infrastructure Land Use, and Total Energy Inputs from Nature as potential inputs of interest. In a further refinement, we compute the same indicator focusing on those sectors included in the list of circular-economy-sensitive activities by the European Commission.

We developed a Python web scraper to download the necessary data for the task. Data management and the construction of the indicator have been coded in R. It is important to note that the code we provide, especially the part dedicated to data management, is tailored to the format we obtained from ORBIS. However, the indicator can be computed using any type of firm-level ownership data available, as long as the format remains consistent with the one we adopted.

The Exposure gap indicator:

The exposure gap is a measure used to assess the disparity or difference in environmental performance between domestic and foreign owners of Italian subsidiaries, particularly regarding their impact on CO2 emissions. We generically define the exposure gap as

$$ExposureGap_t = \sum_j \frac{I_{jt}Y_{jt}}{\sum_k I_{kt}} - \sum_j \frac{F_{jt}Y_{jt}}{\sum_k F_{kt}} \qquad (1)$$

Where $I_{jt}$and F_{jt} are the Italian and Foreign owners that are active in country j at time t respectively. Y_{jt} is a measure of environmental impact of the firm's sector.

We further refine the indicator in (1) to express it as a ratio of its two components. This allows us to obtain an Exposure Gap where a value of 1 indicates a perfect balance between the activities of national and foreign owners on Italian territory. Consequently, any value below 1 would indicate that business groups headed by a local owner have a lesser impact on CO2 emissions compared to groups with foreign ownership.

$$ExposureGap_t = \sum_j \frac{\frac{I_{jt}Y_{jt}}{\sum_k I_{kt}}}{\sum_j \frac{F_{jt}Y_{jt}}{\sum_k F_{kt}}} \qquad (2)$$

2.1 Exposure gap: Focus on Owners

The algorithm reported here only refers to the creation of the Exposure Gap indicator for Owner firms. We report the cleaning and manipulation routine for the raw ORBIS and EXIOBASE data in the annex to this file.

2.1.1 Algorithm

```
# ========================================
# Indicator at Country level (NUTS0) ----
# ========================================

Data_municipalities_firms$Co2_Emission_Air <-
as.numeric(Data_municipalities_firms$Co2_Emission_Air)

#at numerator we sum all the emission of all the owners that have a bvd id number that
#start ,with IT so the pollution sum of italian ones

numerator <- sum(Data_municipalities_firms$Co2_Emission_Air[grep("^IT",
Data_municipalities_firms$`BvD ID number`)], na.rm = TRUE)

# we count how many of them we have
numerator_count <- sum(grepl("^IT", Data_municipalities_firms$`BvD ID number`))

# same thing at the denominator but we sum and count the non italian, so the ones which
do not start with IT
denominator <- sum(Data_municipalities_firms$Co2_Emission_Air[!grepl("^IT",
Data_municipalities_firms$`BvD ID number`)], na.rm = TRUE)

denominator_count <- sum(!grepl("^IT", Data_municipalities_firms$`BvD ID number`))
exposure_gap_national <- numerator / numerator_count / (denominator / denominator_count)

# ==================================
# Indicator at Macro-Region level (NUTS 1) ----
# ==================================

Data_municipalities_firms$Co2_Emission_Air <-
as.numeric(Data_municipalities_firms$Co2_Emission_Air)

exposure_gaps <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {

      current_firms <- Data_municipalities_firms[grep(current_macroregion,
Data_municipalities_firms$`macroregion`), ]

      numerator <- sum(current_firms$Co2_Emission_Air, na.rm = TRUE)
      numerator_count <- sum(!is.na(current_firms$Co2_Emission_Air))

      other_firms <- Data_municipalities_firms[!grepl(current_macroregion,
Data_municipalities_firms$`macroregion`), ]
      denominator <- sum(other_firms$Co2_Emission_Air, na.rm = TRUE)
      denominator_count <- sum(!is.na(other_firms$Co2_Emission_Air))

      exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
      exposure_gaps[[current_macroregion]] <- exposure_gap
      cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")
}
```

```r
exposure_gaps_macroregion <- data.frame(Macroregion = names(exposure_gaps), ExposureGap =
unlist(exposure_gaps))


# ============================
# Indicator at Region level (NUTS 2) ----
# ============================

Data_municipalities_firms$region <-
italian_regions$region[match(Data_municipalities_firms$state_code,
italian_regions$state_code)]

exposure_gaps_regions <- list()

for (current_region in unique(italian_regions$region)) {

       current_firms <- Data_municipalities_firms[grep(current_region,
Data_municipalities_firms$`region`), ]
       numerator <- sum(current_firms$Co2_Emission_Air, na.rm = TRUE)
       numerator_count <- sum(!is.na(current_firms$Co2_Emission_Air))

       other_firms <- Data_municipalities_firms[!grepl(current_region,
Data_municipalities_firms$`region`), ]
       denominator <- sum(other_firms$Co2_Emission_Air, na.rm = TRUE)
       denominator_count <- sum(!is.na(other_firms$Co2_Emission_Air))

       exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
       exposure_gaps_regions[[current_region]] <- exposure_gap
       cat("Exposure gap for", current_region, ":", exposure_gap, "\n")
}

exposure_gaps_reg <- data.frame(region = names(exposure_gaps_regions), ExposureGap =
unlist(exposure_gaps_regions))

# =============================
# Indicator at Province level (NUTS 3) ----
# =============================

exposure_gaps_municipalities <- list()

for (current_city in unique(Data_municipalities_firms$City_Lower)) {

       current_firms <- Data_municipalities_firms[grep(current_city,
Data_municipalities_firms$`City_Lower`), ]
       numerator <- sum(current_firms$Co2_Emission_Air, na.rm = TRUE)
       numerator_count <- sum(!is.na(current_firms$Co2_Emission_Air))
       other_firms <- Data_municipalities_firms[!grepl(current_region,
Data_municipalities_firms$`City_Lower`), ]

       denominator <- sum(other_firms$Co2_Emission_Air, na.rm = TRUE)
       denominator_count <- sum(!is.na(other_firms$Co2_Emission_Air))
       exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
       exposure_gaps_municipalities[[current_city]] <- exposure_gap
       cat("Exposure gap for", current_city, ":", exposure_gap, "\n")
}

exposure_gaps_cit <- data.frame(municipalities = names(exposure_gaps_municipalities),
ExposureGap = unlist(exposure_gaps_municipalities))


# POST COMPUTATION CLEANING OF THE R ENVIRONMENT
# Notice: we can remove the exposure gap measures for municipalities that have less than
10 firms if we want to

#city_counts <- table(Data_municipalities_firms$City_Lower)
#municipalities_to_remove <- names(city_counts[city_counts > 10])
#exposure_gaps_cit <- exposure_gaps_cit[!exposure_gaps_cit$municipalities %in%
municipalities_to_remove, ]
```

```
remove(Balance_2011)
remove(copy, copy2)
remove(current_firms)
sector_to_nation_co2 <- data
remove(data)
remove(exposure_gaps_reg, exposure_gaps)
remove(exposure_gaps_municipalities, exposure_gaps_regions)
remove(exposure_gaps_macroreg)
remove(other_firms)
remove(current_city, current_macroregion, current_region)
remove(denominator, denominator_count, numerator, numerator_count)
remove(exposure_gap)
remove(municipalities_to_remove, city_counts, exposure_gap_national,
exposure_gap_ratioofratio, exposure_ratio)

#RENAMING OUTPUT OBJECT

exposure_gaps_macroreg_owners<- exposure_gaps_macroreg
remove(exposure_gaps_macroreg)

exposure_gaps_reg_owners<- exposure_gaps_reg
remove(exposure_gaps_reg)

exposure_gaps_city_owners<- exposure_gaps_cit
remove(exposure_gaps_cit)
```

2.1.2 Selected Exhibits

These are the results for our first indicator. As we can see, they are all disaggregated at the macro-regional, regional, and provincial/sub-provincial levels. Starting from NUTS 1, we can conclude that owners in the North-West operate in more polluting sectors compared to all other owners in Italy, while North-East owners operate in the least polluting sectors. The same comparison within Italy has been done for regions and municipalities. Similarly, a value above one would mean that the firms (owners) in a city or region operate in more polluting sectors compared to the firms in all other regions or municipalities.

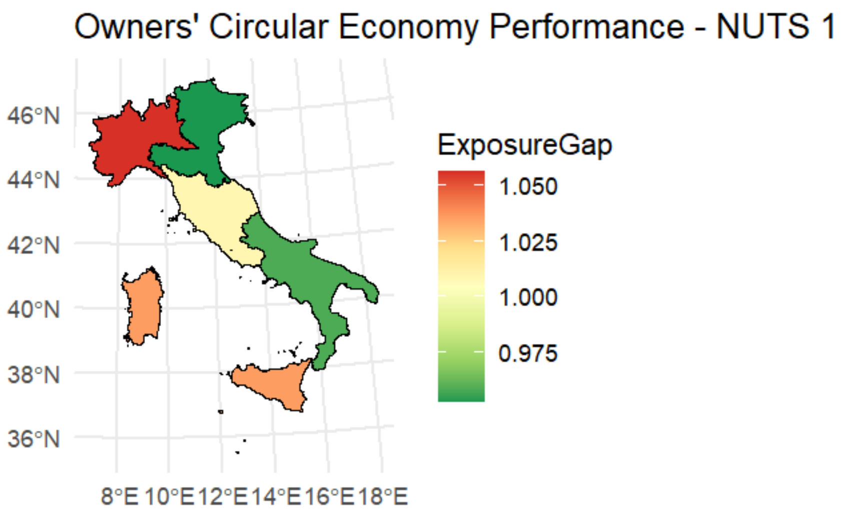
At the national level, we obtained an exposure gap of 0.763875. This means that Italian owners operate in sectors that pollute around 25% less compared to foreign owners. As we already mentioned, foreign owners are differentiated based on their emission data by sector and nation.

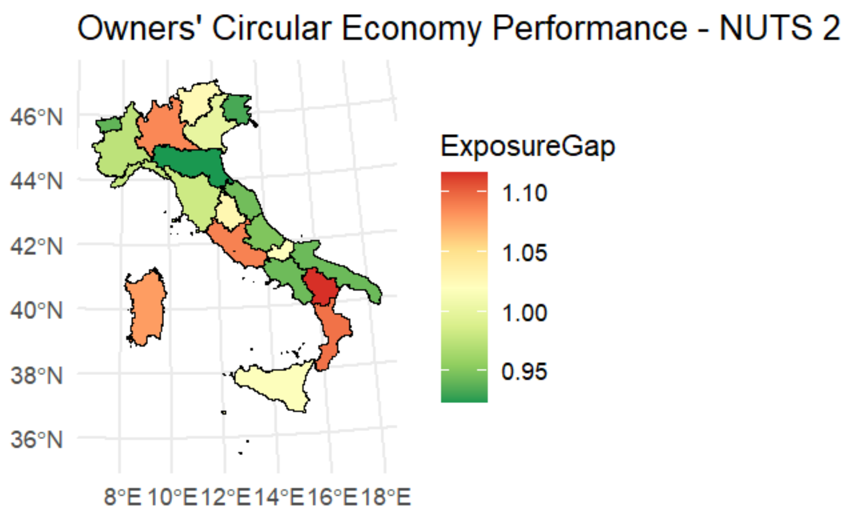
Using firm-level data from the European Union Emission Trading on CO2 emissions, we conducted another analysis comparing Italian firms against European firms and found an exposure gap of 0.9521136. This confirms Italy's positive result, even if it is less than the original exposure gap. Furthermore, we have to consider the fact that we are comparing Italy with other European countries, while the previous analysis compared Italy with all OECD countries. Therefore, it is reasonable to think that European countries have better environmental performance. Since the denominator is CO2 emissions for OECD countries excluding Italy in the previous exposure gap, and European countries excluding Italy for the latter, it is likely that the exposure gap was lower because the denominator included more countries with better environmental performance.

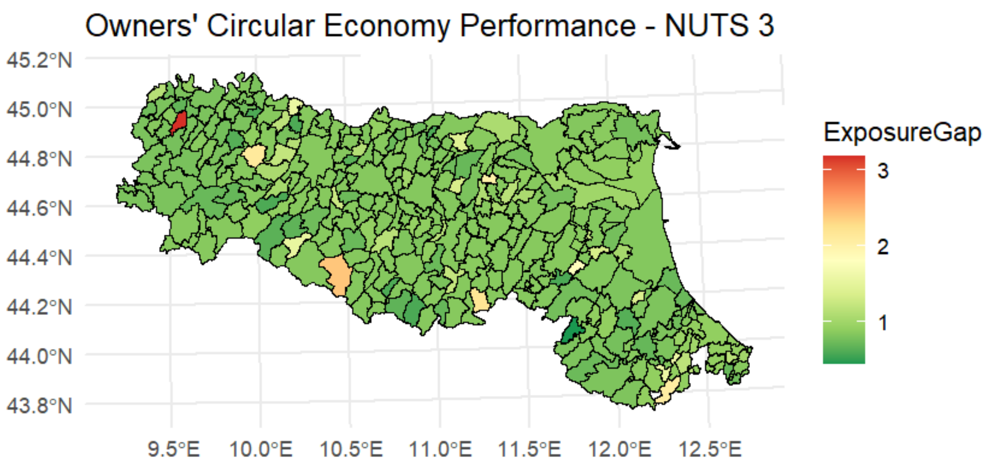
This is the reason why we chose to compare the NUTS 1, 2, and 3 within Italy. Otherwise, by adding foreign owners, we would have most values below one because foreign firms operate in more polluting sectors (considering also the differentiation of pollution between nations), so the comparison to every Italian territorial unit would favor the average firm in Italy, which operates in sectors that pollute less.

Below, we provide a few examples of the exposure gap in CO2 emissions.

	Ownership exposure gap
Nord-Ovest	1.05559
Nord-Est	0.9526528
Centro	1.0088914
Sud	0.9583500
Isole	1.0353323




Owners' Circular Economy Performance - NUTS 2
46°N
44°N
42°N
40°N
38°N
36°N
8°E 10°E 12°E 14°E 16°E 18°E
ExposureGap
1.10
1.05
1.00
0.95


Owners' Circular Economy Performance - NUTS 3
45.2°N
45.0°N
44.8°N
44.6°N
44.4°N
44.2°N
44.0°N
43.8°N
9.5°E 10.0°E 10.5°E 11.0°E 11.5°E 12.0°E 12.5°E
ExposureGap
3
2
1

2.2 Exposure Gap on Subsidiaries

The algorithm reported here only refers to the creation of the Exposure Gap indicator for Subsidiary firms. We report the cleaning and manipulation routine for the raw ORBIS and EXIOBASE data in the annex to this file.

2.2.1 Algorithm

```
# ========================================
# Indicator at Country level (NUTS0) ----
# ========================================


FirmsNace$`Co2 emission subsidiaries` <- as.numeric(FirmsNace$`Co2 emission
subsidiaries`)

numerator <- sum(FirmsNace$`Co2 emission subsidiaries`[FirmsNace$`owner state` == "IT"],
na.rm = TRUE)

numerator_count <- sum(!is.na(FirmsNace$`Co2 emission subsidiaries`[FirmsNace$`owner
state` == "IT"]))

denominator <- sum(FirmsNace$`Co2 emission subsidiaries`[FirmsNace$`owner state` !=
"IT"], na.rm = TRUE)

denominator_count <- sum(!is.na(FirmsNace$`Co2 emission subsidiaries`[FirmsNace$`owner
state` != "IT"]))

exposure_gap_national <- numerator / numerator_count / (denominator / denominator_count)

#print(exposure_gap_national)


# =============================================
# Indicator at Macro-Region level (NUTS1) ----
# =============================================


exposure_gaps_macro <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {


       current_firms <- FirmsNace[grep(current_macroregion, FirmsNace$`macroregion`), ]

       numerator <- sum(current_firms$`Co2 emission subsidiaries`[current_firms$`owner
state` == "IT"], na.rm = TRUE)

       numerator_count <- sum(!is.na(current_firms$`Co2 emission
       subsidiaries`[current_firms$`owner state` == "IT"]))

       other_firms <- FirmsNace[!grepl(current_macroregion, FirmsNace$`macroregion`) &
FirmsNace$`owner state` == "IT", ]

       denominator <- sum(other_firms$`Co2 emission subsidiaries`, na.rm = TRUE)

       denominator_count <- sum(!is.na(other_firms$`Co2 emission subsidiaries`))


       exposure_gap <- (numerator / numerator_count) / (denominator / denominator_count)

       exposure_gaps_macro[[current_macroregion]] <- exposure_gap
```

```
        cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")

}

exposure_gaps_macroreg <- data.frame(Macroregion = names(exposure_gaps_macro),
ExposureGap = unlist(exposure_gaps_macro))


remove(current_macroregion, denominator_count, denominator, exposure_gap, numerator,
numerator_count, total_denominator, total_denominator_count, total_numerator,
total_numerator_count)


# ========================================
# Indicator at Region level (NUTS2) ----
# ========================================


exposure_gaps_regions <- list()

for (current_region in unique(italian_regions$region)) {

        current_firms <- FirmsNace[grep(current_region, FirmsNace$`region`), ]

        numerator <- sum(current_firms$`Co2 emission subsidiaries`[current_firms$`owner
state` == "IT"], na.rm = TRUE)

        numerator_count <- sum(!is.na(current_firms$`Co2 emission
subsidiaries`[current_firms$`owner state` == "IT"]))

        other_firms <- FirmsNace[!grepl(current_region, FirmsNace$`region`), ]


        denominator <- sum(other_firms$`Co2 emission subsidiaries`, na.rm = TRUE)

        denominator_count <- sum(!is.na(other_firms$`Co2 emission subsidiaries`))


        exposure_gap <- numerator / numerator_count / (denominator / denominator_count)

        exposure_gaps_regions[[current_region]] <- exposure_gap

        cat("Exposure gap for", current_region, ":", exposure_gap, "\n")

}

exposure_gaps_reg <- data.frame(region = names(exposure_gaps_regions), ExposureGap =
unlist(exposure_gaps_regions))


# ========================================
# Indicator at Province level (NUTS3) ----
# ========================================


exposure_gaps_city <- list()

for (current_city in unique(FirmsNace$`owner city`)) {

        current_firms <- FirmsNace[grep(current_city, FirmsNace$`owner city`), ]

        numerator <- sum(current_firms$`Co2 emission subsidiaries`[current_firms$`owner
        state` == "IT"], na.rm = TRUE)

        numerator_count <- sum(!is.na(current_firms$`Co2 emission
        subsidiaries`[current_firms$`owner state` == "IT"]))
```

```
        other_firms <- FirmsNace[!grepl(current_city, FirmsNace$`owner city`) &
        FirmsNace$`owner state` == "IT", ]


        denominator <- sum(other_firms$`Co2 emission subsidiaries`, na.rm = TRUE)

        denominator_count <- sum(!is.na(other_firms$`Co2 emission subsidiaries`))

        exposure_gap <- (numerator / numerator_count) / (denominator / denominator_count)

        exposure_gaps_city[[current_city]] <- exposure_gap

        cat("Exposure gap for", current_city, ":", exposure_gap, "\n")

}


exposure_gaps_city_df <- data.frame(City = names(exposure_gaps_city), ExposureGap =
unlist(exposure_gaps_city))


remove(current_macroregion, denominator, denominator_count, exposure_gap, numerator,
numerator_count)

remove(current_city, copy, copy1, copy2)

remove(current_firms, exposure_gaps_macro)

remove(other_firms)


# RENAMING OUTPUT OBJECTS

exposure_gaps_macroreg_subs<- exposure_gaps_macroreg

remove(exposure_gaps_macroreg)

exposure_gaps_reg_subs<- exposure_gaps_reg

remove(exposure_gaps_reg)

exposure_gaps_city_subs<-exposure_gaps_city

remove(exposure_gaps_city)


df <- data.frame(

        city = names(exposure_gaps_city_subs),

        value = unlist(exposure_gaps_city_subs),

        row.names = NULL

)


exposure_gaps_city_subs<- df

remove(df)
```

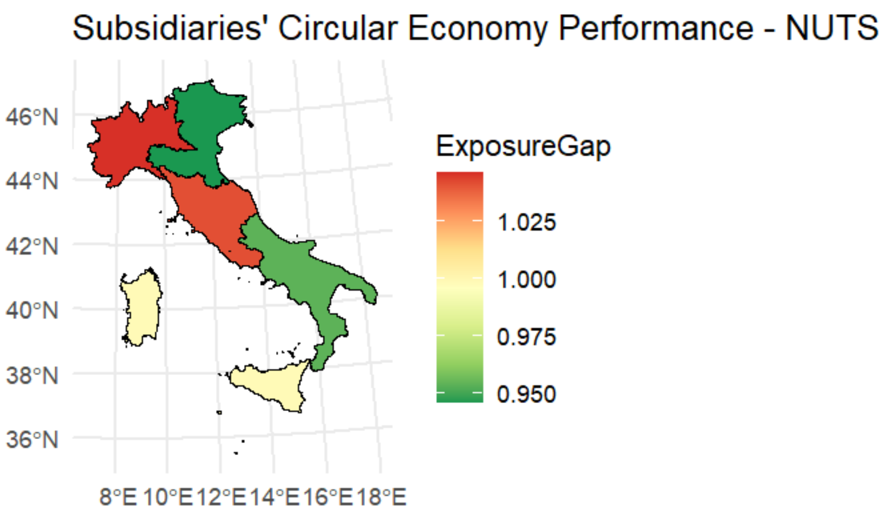
2.2.2 Selected exhibits

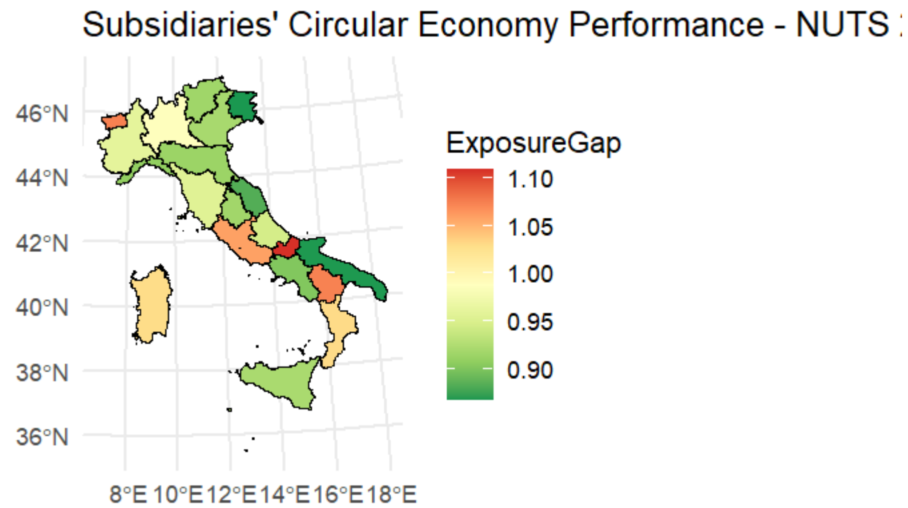
These are the results for our second indicator. As we can see, they are all disaggregated at the macro-regional, regional, and provincial/sub-provincial levels. Starting from NUTS 1, we can conclude that subsidiaries with owners located in the North-West are the ones which operate in more polluting sectors compared to all the other subsidiaries in Italy, while subsidiaries with owners located in the North-East operate in less polluting sectors compared to the rest of the subsidiaries owned by firms operating outside of the North-East. The same comparison within Italy has been done for regions and municipalities. Similarly, a value above one would mean that the city or region's firms (subsidiaries) with owners from that region operate in more polluting sectors compared to the subsidiaries of all the other regions or municipalities.

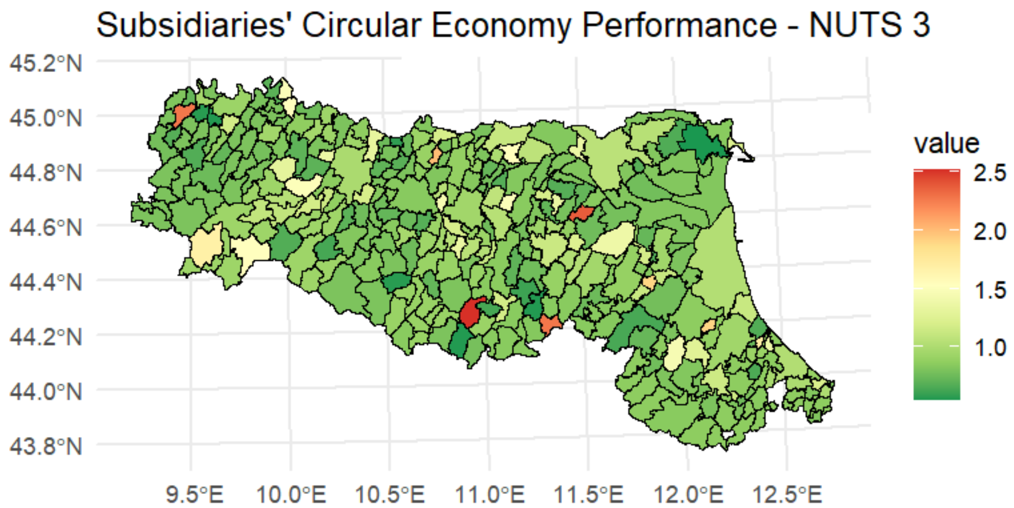
At the national level, we have an exposure gap of 0.86715. This means that Italian subsidiaries owned domestically operate in sectors which are around 15% less polluting compared to the foreign-owned Italian subsidiaries.

This is the reason why we chose to compare the NUTS 1, 2, and 3 within Italy, namely with subsidiaries all domestically owned. Otherwise, by adding foreign-owned firms, we would have most values below one because foreign-owned subsidiaries operate in more polluting sectors. So, comparing every Italian territorial unit would favor the average domestically-owned firm in Italy, which operates in sectors that pollute less.

Also, it is better to distinguish subsidiaries geographically at the owners' level and not at the subsidiaries' level because we have already analyzed the performance of a subset of firms (the owners) depending on their location. In this indicator, we wanted to make a more sophisticated analysis by concentrating on the ownership effect of having owners from a certain location on Italian subsidiaries. The main indicator that does this job is the national indicator which compares foreign-owned firms with domestically-owned firms. The disaggregation is useful if we want to observe the effect of being owned by owners in different macro-regions, regions, and municipalities in Italy.







1.3 Importing Pollution

This indicator offers insight into the extent to which foreign owners "export" their emissions through Italian subsidiaries. In contrast to the algorithm outlined in section 1.1, we have selectively retained foreign owners who possess at least one Italian subsidiary alongside one foreign subsidiary. Subsequently, we compare the identified Italian subsidiary(-ies) against other subsidiaries within the same ownership group. This indicator evaluates the degree to which business groups, whose ultimate owner is situated outside of Italy, gravitate towards more (or less) polluting sectors in Italy in comparison to their operations abroad.

The algorithm reported here only refers to the creation of the indicator on pollution imports. We report the cleaning and manipulation routine for the raw ORBIS and EXIOBASE data in the annex to this file.

1.3.1 Algorithm

```
# ========================================
# Indicator at Country level (NUTS0) ----
# ========================================

copy<-Foreign2
Foreign2 <- Foreign2[!is.na(Foreign2$subsidiary_emission), ]
Foreign2$subsidiary_emission <- as.numeric(Foreign2$subsidiary_emission)

numerator <- sum(Foreign2$subsidiary_emission[grep("^IT", Foreign2$`Country ISO
code.y`)], na.rm = TRUE)
numerator_count <- sum(grepl("^IT", Foreign2$`Country ISO code.y`))
denominator <- sum(Foreign2$subsidiary_emission[!grepl("^IT", Foreign2$`Country ISO
code.y`)], na.rm = TRUE)

denominator_count <- sum(!grepl("^IT", Foreign2$`Country ISO code.y`))

exposure_gap_uncontrolled_foreign <- numerator / numerator_count / (denominator /
denominator_count)

#print(exposure_gap_uncontrolled_foreign)

# Weighted by sales:

copy2 <- Foreign2
Foreign2 <- Foreign2[Foreign2$`Sales\nth USD 2019` != "n.a.", ]
Foreign2$`Sales\nth USD 2019` <- as.numeric(Foreign2$`Sales\nth USD 2019`)

numerator <- sum(Foreign2$subsidiary_emission[grep("^IT", Foreign2$`Country ISO
code.y`)], na.rm = TRUE)
numerator_sales <- numerator * sum(Foreign2$`Sales\nth USD 2019`[grep("^IT",
Foreign2$`Country ISO code.y`)], na.rm = TRUE)

denominator <- sum(Foreign2$subsidiary_emission[!grepl("^IT", Foreign2$`Country ISO
code.y`)], na.rm = TRUE)
denominator_sales <- denominator * sum(Foreign2$`Sales\nth USD 2019`[!grepl("^IT",
Foreign2$`Country ISO code.y`)], na.rm = TRUE)

total_sales <- sum(Foreign2$`Sales\nth USD 2019`, na.rm = TRUE)
numerator_count <- sum(grepl("^IT", Foreign2$`Country ISO code.y`)) * total_sales
denominator_count <- sum(!grepl("^IT", Foreign2$`Country ISO code.y`)) * total_sales

exposure_gap_foreign <- numerator_sales / numerator_count / (denominator_sales /
denominator_count)

# ============================================
# Indicator at Macro-region level (NUTS0) ----
# ============================================

Foreign2<-copy2
Foreign2 <- Foreign2[complete.cases(Foreign2$city_subsidiary), ]
Foreign2 <- Foreign2[Foreign2$`Country ISO code.y` == "IT", ]

Foreign2 <- Foreign2 %>% left_join(Municipalities, by = c("city_subsidiary" =
"name_clean"))

italian_regions <- italian_regions %>% mutate(state_code = as.character(state_code))

Foreign2 <- Foreign2 %>% left_join(italian_regions, by = c("state_code" = "state_code"))

copy2<- Foreign2
```

```
exposure_gaps <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {
       current_firms <- Foreign2[grep(current_macroregion, Foreign2$`macroregion`), ]
       numerator <- sum(current_firms$subsidiary_emission, na.rm = TRUE)
       numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

       other_firms <- Foreign2[!grepl(current_macroregion, Foreign2$`macroregion`), ]
       denominator <- sum(other_firms$subsidiary_emission, na.rm = TRUE)
       denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

       exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
       exposure_gaps[[current_macroregion]] <- exposure_gap
       cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")
}

exposure_gaps_macroreg_foreign <- data.frame(Macroregion = names(exposure_gaps),
ExposureGap = unlist(exposure_gaps))

# Weighted by sales

exposure_gaps <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {

       current_firms <- Foreign2[grep(current_macroregion, Foreign2$`macroregion`), ]
       numerator <- sum(current_firms$subsidiary_emission * current_firms$`Sales\nth USD
2019`, na.rm = TRUE)
       numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

       other_firms <- Foreign2[!grepl(current_macroregion, Foreign2$`macroregion`), ]
       denominator <- sum(other_firms$subsidiary_emission * other_firms$`Sales\nth USD
2019`, na.rm = TRUE)
       denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

       exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
       exposure_gaps[[current_macroregion]] <- exposure_gap
       cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")
}

exposure_gaps_macroreg_foreign_sales <- data.frame(Macroregion = names(exposure_gaps),
ExposureGap = unlist(exposure_gaps))

# =======================================
# Indicator at Region level (NUTS2) ----
# =======================================

exposure_gaps_regions <- list()

for (current_region in unique(italian_regions$region)) {

       current_firms <- Foreign2[grep(current_region, Foreign2$`region`), ]
       numerator <- sum(current_firms$subsidiary_emission, na.rm = TRUE)
       numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

       other_firms <- Foreign2[!grepl(current_region, Foreign2$`region`), ]
       denominator <- sum(other_firms$subsidiary_emission, na.rm = TRUE)
       denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

       exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
       exposure_gaps_regions[[current_region]] <- exposure_gap
       cat("Exposure gap for", current_region, ":", exposure_gap, "\n")
}

exposure_gaps_reg_foreign <- data.frame(region = names(exposure_gaps_regions),
ExposureGap = unlist(exposure_gaps_regions))
```

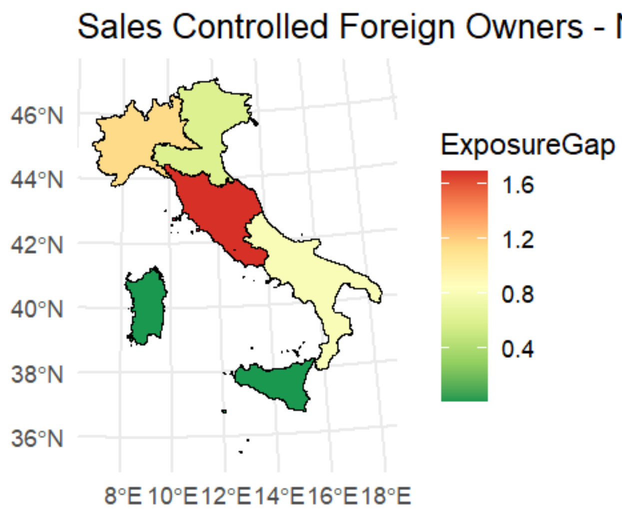
```r
# Weighted by sales

exposure_gaps_regions <- list()

for (current_region in unique(italian_regions$region)) {

	current_firms <- Foreign2[grep(current_region, Foreign2$`region`), ]
	numerator <- sum(current_firms$subsidiary_emission * current_firms$`Sales\nth USD 
2019`, na.rm = TRUE)
	numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

	other_firms <- Foreign2[!grepl(current_region, Foreign2$`region`), ]
	denominator <- sum(other_firms$subsidiary_emission * other_firms$`Sales\nth USD 
2019`, na.rm = TRUE)
	denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

	exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
	exposure_gaps_regions[[current_region]] <- exposure_gap
	cat("Exposure gap for", current_region, ":", exposure_gap, "\n")
}

exposure_gaps_reg_foreign_sales <- data.frame(region = names(exposure_gaps_regions), 
ExposureGap = unlist(exposure_gaps_regions))

# =======================================
# Indicator at Province level (NUTS3) ----
# =======================================

exposure_gaps_municipalities <- list()

for (current_city in unique(Foreign2$city_subsidiary)) {

	current_firms <- Foreign2[grep(current_city, Foreign2$city_subsidiary), ]
	numerator <- sum(current_firms$subsidiary_emission, na.rm = TRUE)
	numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

	other_firms <- Foreign2[!grepl(current_city, Foreign2$city_subsidiary), ]
	denominator <- sum(other_firms$subsidiary_emission, na.rm = TRUE)
	denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

	exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
	exposure_gaps_municipalities[[current_city]] <- exposure_gap
	cat("Exposure gap for", current_city, ":", exposure_gap, "\n")
}

exposure_gaps_cit_foreign <- data.frame(municipalities = 
names(exposure_gaps_municipalities), ExposureGap = unlist(exposure_gaps_municipalities))

# Wighted by sales

exposure_gaps_municipalities <- list()

for (current_city in unique(Foreign2$city_subsidiary)) {

	current_firms <- Foreign2[grep(current_city, Foreign2$city_subsidiary), ]
	numerator <- sum(current_firms$subsidiary_emission * current_firms$`Sales\nth USD 
2019`, na.rm = TRUE)
	numerator_count <- sum(!is.na(current_firms$subsidiary_emission))

	other_firms <- Foreign2[!grepl(current_city, Foreign2$city_subsidiary), ]
	denominator <- sum(other_firms$subsidiary_emission * other_firms$`Sales\nth USD 
2019`, na.rm = TRUE)
	denominator_count <- sum(!is.na(other_firms$subsidiary_emission))

	exposure_gap <- numerator / numerator_count / (denominator / denominator_count)
	exposure_gaps_municipalities[[current_city]] <- exposure_gap
	cat("Exposure gap for", current_city, ":", exposure_gap, "\n")
}
```

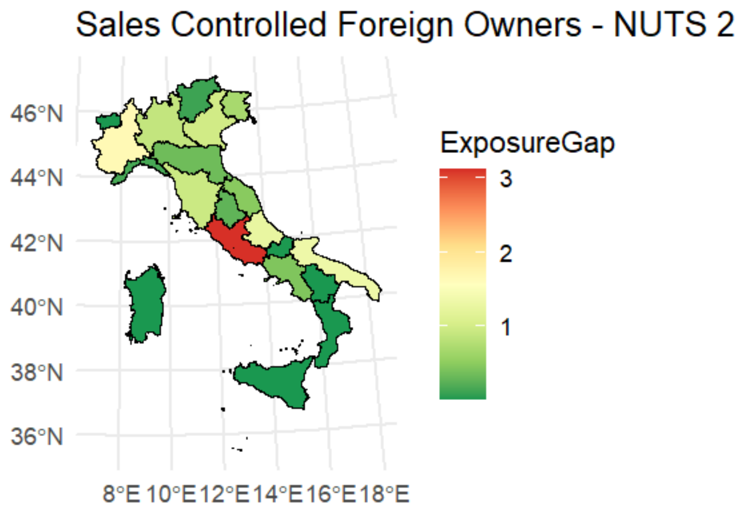
```
exposure_gaps_cit_foreign_sales <- data.frame(municipalities =
names(exposure_gaps_municipalities), ExposureGap = unlist(exposure_gaps_municipalities))

# cleaning the working space

remove(other_firms)
remove(current_city, current_macroregion, current_region)
remove(denominator, denominator_count, numerator, numerator_count)
remove(numerator_count2, numerator_sales, numerator_sales2, numerator2, start_IT_group,
start_non_IT_group, subsidiary_bvd_id, total_sales, unique_bvd_ids)
remove(non_missing_matches, nace_code, na_count, match_rows, keep_rows, keep_rows_non_IT,
iso_code, i, denominator2, denominator_count2, denominator_sales, denominator_sales2,
count_IT_rows, count_IT_rows_bvd, column_11_value, bvd_id_number)
remove(copy, copy2, current_firms)
remove(exposure_gaps, exposure_gaps_regions_uncontrolled, exposure_gaps_regions,
exposure_gaps_macro_uncontrolled, exposure_gaps_macro, exposure_gaps_city,
exposure_gaps_municipalities)
remove(Firms_filtered)
remove(match_row, merged_data)
remove(Subs, Subs3)
remove(city_names)
remove(exposure_gap, exposure_gaps)
```

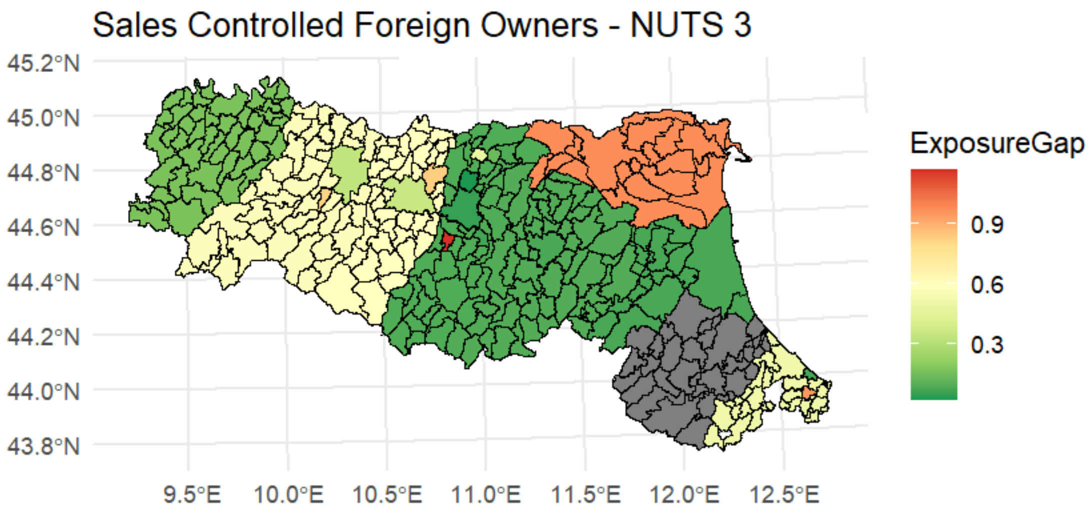
1.3.2 Selected Exhibits

We find an exposure gap of 0.7641015, indicating that Italian subsidiaries of foreign owners emit less pollution than the foreign-owned subsidiaries. As evident from the caption, we also controlled for sales. This step was taken to assess the extent of emission export these firms engaged in. By controlling for sales, the situation for Italian subsidiaries improved. The exposure gap, controlled for sales, is 0.4575345. This signifies that Italian subsidiaries of foreign owners both produce and sell less compared to the foreign-owned subsidiaries. Consequently, their absolute performance is even better when sales are taken into consideration.

Regarding the geographic breakdown into NUTS 1, 2, and 3, we scrutinized the performance of Italian subsidiaries owned by foreign owners. We excluded subsidiaries owned by Italian owners to discern the geographical locations where foreign owners controlled more polluting subsidiaries in Italy.







1.4 Exporting Pollution

This indicator complements the one described in section 1.3, providing a measure of how much Italian owners "export" their emissions through foreign subsidiaries. We consider Italian owners with at least one Italian and one foreign subsidiary. Then, we compare the performance of the identified foreign subsidiary(-ies) to the Italian ones. The indicator evaluates the extent to which Italian owners opt to relocate more (or less) polluting activities abroad or if they retain them within the Italian territory. This analysis helps us understand whether Italian firms are transferring their higher-emission operations to countries with potentially less stringent environmental regulations, or whether they are keeping these operations within Italy, which may have stricter environmental standards.

The algorithm reported here only refers to the creation of the exporting pollution indicator. We report the cleaning and manipulation routine for the raw ORBIS and EXIOBASE data in the annex to this file.

1.4.1 Algorithm

```
# =======================================
# Indicator at Country level (NUTS0) ----
# =======================================

copy<-Italy2

Italy2 <- Italy2[!is.na(Italy2$subsidiary_emission), ]
Italy2$subsidiary_emission <- as.numeric(Italy2$subsidiary_emission)

numerator <- sum(Italy2$subsidiary_emission[grep("^IT", Italy2$`Country ISO code.y`)],
na.rm = TRUE)
numerator_count <- sum(grepl("^IT", Italy2$`Country ISO code.y`))

denominator <- sum(Italy2$subsidiary_emission[!grepl("^IT", Italy2$`Country ISO
code.y`)], na.rm = TRUE)
denominator_count <- sum(!grepl("^IT", Italy2$`Country ISO code.y`))

exposure_gap_national_uncontrolled <- numerator / numerator_count / (denominator /
denominator_count)

#Weighted by sales

copy2<-Italy2
Italy2 <- Italy2[Italy2$`Sales\nth USD 2019` != "n.a.", ]
Italy2$`Sales\nth USD 2019` <- as.numeric(Italy2$`Sales\nth USD 2019`)

numerator <- sum(Italy2$subsidiary_emission[grep("^IT", Italy2$`Country ISO code.y`)],
na.rm = TRUE)
numerator_sales <- numerator * sum(Italy2$`Sales\nth USD 2019`[grep("^IT",
Italy2$`Country ISO code.y`)], na.rm = TRUE)

denominator <- sum(Italy2$subsidiary_emission[!grepl("^IT", Italy2$`Country ISO
code.y`)], na.rm = TRUE)
denominator_sales <- denominator * sum(Italy2$`Sales\nth USD 2019`[!grepl("^IT",
Italy2$`Country ISO code.y`)], na.rm = TRUE)

total_sales <- sum(Italy2$`Sales\nth USD 2019`, na.rm = TRUE)
numerator_count <- sum(grepl("^IT", Italy2$`Country ISO code.y`)) * total_sales
denominator_count <- sum(!grepl("^IT", Italy2$`Country ISO code.y`)) * total_sales

exposure_gap_national <- numerator_sales / numerator_count / (denominator_sales /
denominator_count)

# ============================================
# Indicator at Macro.region level (NUTS1) ----
# ============================================

Italy2 <- copy2

Italy2 <- Italy2 %>% left_join(Municipalities, by = c("city_owner" = "name_clean"))

italian_regions <- italian_regions %>% mutate(state_code = as.character(state_code))

Italy2 <- Italy2 %>% left_join(italian_regions, by = c("state_code" = "state_code"))

Italy2<-copy2

exposure_gaps_macro_uncontrolled <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {
```

```
        current_firms <- Italy2[grep(current_macroregion, Italy2$`macroregion`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))
        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"]))

        other_firms <- Italy2[!grepl(current_macroregion, Italy2$`macroregion`), ]  #
        Added missing comma here
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))
        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` !=
        "IT"]))

        exposure_gap <- ((numerator / numerator_count) / (denominator2 /
        denominator_count2)) / ((numerator2 / numerator_count2) / (denominator /
        denominator_count ))
        exposure_gaps_macro_uncontrolled[[current_macroregion]] <- exposure_gap
        cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")
}

exposure_gaps_macroreg_uncontrolled <- data.frame(Macroregion =
names(exposure_gaps_macro_uncontrolled), ExposureGap =
unlist(exposure_gaps_macro_uncontrolled))

# Weighted by sales

copy2<-Italy2
Italy2 <- Italy2[Italy2$`Sales\nth USD 2019` != "n.a.", ]
Italy2$`Sales\nth USD 2019` <- as.numeric(Italy2$`Sales\nth USD 2019`)

exposure_gaps_macro <- list()

for (current_macroregion in unique(italian_regions$macroregion)) {

        current_firms <- Italy2[grep(current_macroregion, Italy2$`macroregion`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_sales <- numerator * sum(current_firms$`Sales\nth USD
        2019`[current_firms$`Country ISO code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_sales2 <- denominator2 * sum(current_firms$`Sales\nth USD
        2019`[current_firms$`Country ISO code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"]))

        other_firms <- Italy2[!grepl(current_macroregion, Italy2$`macroregion`), ] # Added
missing comma here
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
```

```
        numerator_sales2 <- numerator2 * sum(other_firms$`Sales\nth USD
        2019`[other_firms$`Country ISO code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))

        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_sales <- denominator * sum(other_firms$`Sales\nth USD
        2019`[other_firms$`Country ISO code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` !=
        "IT"]))

        exposure_gap <- ((numerator_sales / numerator_count) / (denominator_sales2 /
        denominator_count2)) / ((numerator_sales2 / numerator_count2) / (denominator_sales
        / denominator_count ))
        exposure_gaps_macro[[current_macroregion]] <- exposure_gap
        cat("Exposure gap for", current_macroregion, ":", exposure_gap, "\n")
}

exposure_gaps_macroreg <- data.frame(Macroregion = names(exposure_gaps_macro),
ExposureGap = unlist(exposure_gaps_macro))

# =======================================
# Indicator at Region level (NUTS2) ----
# =======================================

Italy2<-copy2
exposure_gaps_regions_uncontrolled <- list()

for (current_region in unique(italian_regions$region)) {

        current_firms <- Italy2[grep(current_region, Italy2$`region`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"]))

        other_firms <- Italy2[!grepl(current_region, Italy2$`region`), ]
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))

        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` !=
        "IT"]))

        exposure_gap <- ((numerator / numerator_count) / (denominator2 /
        denominator_count2))/(( numerator2 / numerator_count2) / (denominator /
        denominator_count ))
        exposure_gaps_regions_uncontrolled[[current_region]] <- exposure_gap
        cat("Exposure gap for", current_region, ":", exposure_gap, "\n")
}
```

```
exposure_gaps_reg_uncontrolled <- data.frame(region =
names(exposure_gaps_regions_uncontrolled), ExposureGap =
unlist(exposure_gaps_regions_uncontrolled))

# Weighted by sales

copy2<-Italy2
Italy2 <- Italy2[Italy2$`Sales\nth USD 2019` != "n.a.", ]
Italy2$`Sales\nth USD 2019` <- as.numeric(Italy2$`Sales\nth USD 2019`)

exposure_gaps_regions <- list()

for (current_region in unique(italian_regions$region)) {

        current_firms <- Italy2[grep(current_region, Italy2$`region`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_sales <- numerator * sum(current_firms$`Sales\nth USD
        2019`[current_firms$`Country ISO code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_sales2 <- denominator2 * sum(current_firms$`Sales\nth USD
        2019`[current_firms$`Country ISO code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"]))

        other_firms <- Italy2[!grepl(current_region, Italy2$`region`), ]
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_sales2 <- numerator2 * sum(other_firms$`Sales\nth USD
        2019`[other_firms$`Country ISO code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))

        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_sales <- denominator * sum(other_firms$`Sales\nth USD
        2019`[other_firms$`Country ISO code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` !=
        "IT"]))

        exposure_gap <- ((numerator_sales / numerator_count) / (denominator_sales2 /
        denominator_count2)) / ((numerator_sales2 / numerator_count2) / (denominator_sales
        / denominator_count ))
        exposure_gaps_regions[[current_region]] <- exposure_gap
        cat("Exposure gap for", current_region, ":", exposure_gap, "\n")
}

exposure_gaps_reg <- data.frame(region = names(exposure_gaps_regions), ExposureGap =
unlist(exposure_gaps_regions))

# ========================================
# Indicator at Province level (NUTS0) ----
# ========================================

Italy2<-copy2

exposure_gaps_city <- list()

for (current_city in unique(Italy2$`city_owner`)) {
```

```
        current_firms <- Italy2[grep(current_city, Italy2$`city_owner`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        other_firms <- Italy2[!grepl(current_city, Italy2$`city_owner`), ]  # Include all
        necessary columns here
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))
        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))

        exposure_gap <- ((numerator / numerator_count) / (denominator2 /
        denominator_count2))/(( numerator2 / numerator_count2) / (denominator /
        denominator_count ))
        exposure_gaps_city[[current_city]] <- exposure_gap
        cat("Exposure gap for", current_city, ":", exposure_gap, "\n")
}

exposure_gaps_city_df <- data.frame(City = names(exposure_gaps_city), ExposureGap =
unlist(exposure_gaps_city))

#Weighted by sales

Italy2$`Sales\nth USD 2019` <- as.numeric(Italy2$`Sales\nth USD 2019`)

exposure_gaps_city <- list()

for (current_city in unique(Italy2$`city_owner`)) {

        current_firms <- Italy2[grep(current_city, Italy2$`city_owner`), ]
        numerator <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` == "IT"] * current_firms$`Sales\nth USD 2019`[current_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO code.y`
        == "IT"]))

        denominator2 <- sum(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"] * current_firms$`Sales\nth USD 2019`[current_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count2 <-
        sum(!is.na(current_firms$`subsidiary_emission`[current_firms$`Country ISO
        code.y` != "IT"]))

        other_firms <- Italy2[!grepl(current_city, Italy2$`city_owner`), ]  # Include all
        necessary columns here
        numerator2 <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` == "IT"] * other_firms$`Sales\nth USD 2019`[other_firms$`Country ISO
        code.y` == "IT"], na.rm = TRUE)
        numerator_count2 <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))
```

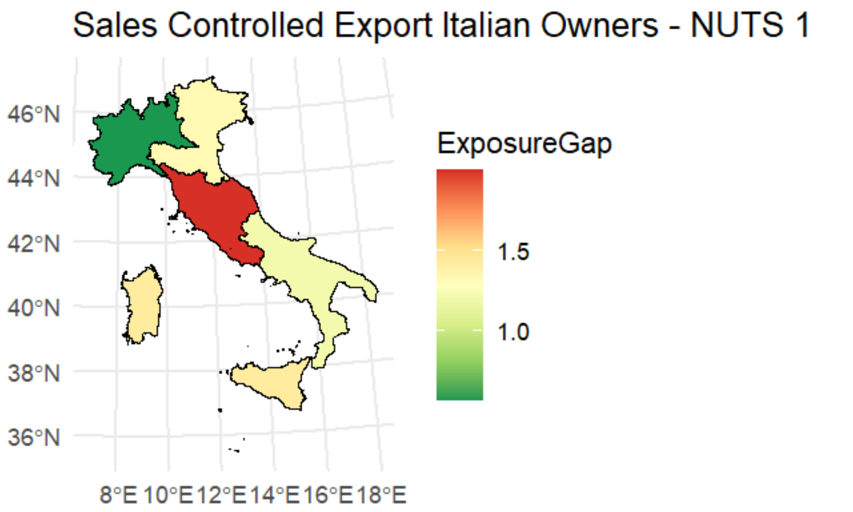
```
        denominator <- sum(other_firms$`subsidiary_emission`[other_firms$`Country ISO
        code.y` != "IT"] * other_firms$`Sales\nth USD 2019`[other_firms$`Country ISO
        code.y` != "IT"], na.rm = TRUE)
        denominator_count <-
        sum(!is.na(other_firms$`subsidiary_emission`[other_firms$`Country ISO code.y` ==
        "IT"]))

        exposure_gap <- ((numerator / numerator_count) / (denominator2 /
        denominator_count2)) / (( numerator2 / numerator_count2) / (denominator /
        denominator_count ))
        exposure_gaps_city[[current_city]] <- exposure_gap
        cat("Exposure gap for", current_city, ":", exposure_gap, "\n")
}

exposure_gaps_city_df_sales <- data.frame(City = names(exposure_gaps_city), ExposureGap =
unlist(exposure_gaps_city))
```

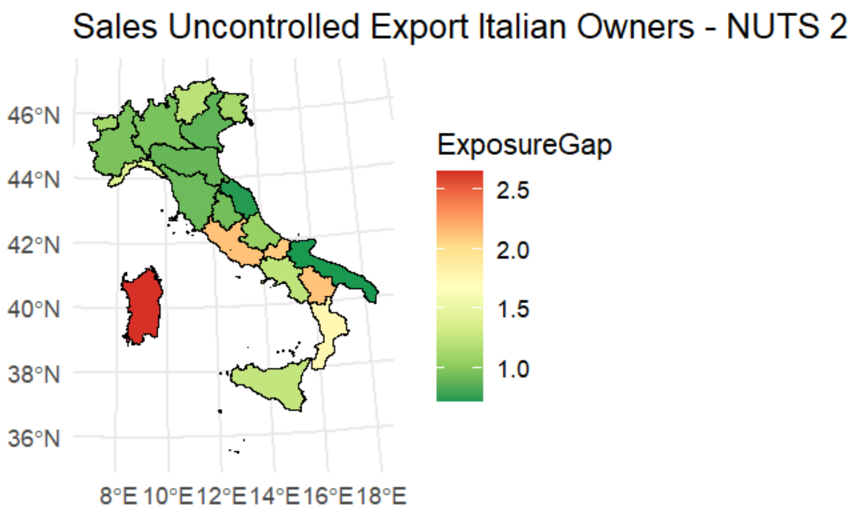
1.4.2 Selected Exhibits

The national exposure gap without sales control is 0.630465, indicating that foreign subsidiaries operate in more polluting sectors compared to Italian subsidiaries.

As depicted in the caption, we also controlled for sales. We did this to gauge the magnitude of emission export these firms were engaged in. The result changes when we account for sales; the exposure gap, controlled for sales, spikes to 1.572769. The final interpretation for this indicator is that foreign subsidiaries of Italian firms operate in more polluting sectors. However, they produce and sell less than their Italian counterparts. Therefore, Italian subsidiaries pollute more in absolute terms but not in relative terms (relative to their sector).

In the NUTS 1, 2, and 3 disaggregation, we compared the emission export of different Italian owners depending on their location. For example, in the macro regional table, we can observe how much a certain region (in this case, the North-West) has owners who export less pollution through foreign subsidiaries. This is particularly interesting because the North-West performed less well in all our other measures. However, a certain degree of this poorer performance is attributable to lower pollution export, which is more pronounced in previously better-performing regions such as the North-East. Therefore, the geographic disaggregation is at the owner level and creates a ratio of ratios of ratios, by computing how much the owners in a certain location own subsidiaries that operate in polluting sectors outside of Italy compared to the domestic subsidiaries, and then compares this pollution export with the one made by the owners of all the other Italian regions. To ensure the robustness of our analysis, we computed these tables both controlled and uncontrolled for sales.


Sales Controlled Export Italian Owners - NUTS 1
46°N
44°N
42°N
40°N
38°N
36°N
8°E 10°E 12°E 14°E 16°E 18°E
ExposureGap
1.5
1.0


Sales Uncontrolled Export Italian Owners - NUTS 2
46°N
44°N
42°N
40°N
38°N
36°N
8°E 10°E 12°E 14°E 16°E 18°E
ExposureGap
2.5
2.0
1.5
1.0

2.5 Data Cleaning and Management

Here, we also provide the code related to data cleaning and management. It is required to have physically downloaded the ORBIS data.

1.5.1 DATA MANIPULATION: GEOGRAPHY/OWNERSHIP

```
# Preamble: recall libraries and set path to data ----

library(readxl)
library(tidyverse)
library(dplyr)

path.orbis <- "PATH to ORBIS OWNERSHIP DATA"
setwd(path)

# Load files and Cleaning incomplete observations ----
# Notice: Files have been downloaded from ORBIS using a scraper. All downloaded files
must be stored accordingly.

files <- list.files(pattern = "\\.xlsx$")
fn.import <- function(x) read_excel(x, sheet = "Results")

data_frame <- lapply(files, fn.import)
data_frame <- bind_rows(data_frame)


Firms <- data_frame
remove(data_frame)

FirmsNace <- Firms[!is.na(Firms$`BvD ID number`), ]
FirmsNace <- Firms[!is.na(Firms$`NACE Rev. 2, core code (4 digits)`), ]

remove(files, path, fn.import)

# Quality check on missing variables of interest ----
#      (fortunately the standardized city variable has low incidence of missing values)
#      (uncomment if needed)

#      variables <- c("Country ISO code.x", "NACE Rev. 2, core code (4 digits).x", "NACE
Rev. 2, #     secondary code(s)", "City\nLatin Alphabet", "NUTS1", "NUTS2", "NUTS3",
"Standardized city", #"Subsidiary - Total %", "Subsidiary - Direct %", "Subsidiary -
Status")

#      missing_variables <- variables[!(variables %in% names(FirmsBalance))]
#      missing_percentage <- colSums(is.na(FirmsBalance[variables])) / nrow(FirmsBalance)
* 100

#      print(missing_percentage)
#      remove(read_results_sheet, variables,missing_variables,missing_percentage)

### now we need a dataset to link city names to their region, after we have that we can
run the #exposure gap by region (or big municipalities)
# dataset uploaded, here is the region code, we can also upload and match them if
necessary
# "City\nLatin Alphabet"
# here we use another dataset which gives different codes for every region and
municipalities, so that we can #connect every city (NUTS 3) to its region (NUTS 2)
# and macro region (NUTS 1)

municipalities <- read_csv("FILE PATH to municipalities.csv")

# Delete duplicate municipalities
```

```
filtered_municipalities <- municipalities %>%
       arrange(desc(country_code == "IT")) %>%
       distinct(name, .keep_all = TRUE)

# Harmonize municipalities name variable to lower-case for matching with
filtered_municipalities

FirmsNace$City_Lower <- tolower(FirmsNace$`City\nLatin Alphabet`)
filtered_municipalities$name_lower <- tolower(filtered_municipalities$name)

# Merge the datasets based on the lowercase columns,

merged <- merge(FirmsNace, filtered_municipalities,
                 by.x = "City_Lower", by.y = "name_lower",
                 all.x = TRUE)

remove(municipalities)

FirmsMunicipalities <- merged
remove(merged)

# Creating the ISTAT-consistent macro region variable, to be matched to each region code

italian_regions <- data.frame(
                          region = c(
                          "Abruzzo", "Aosta Valley", "Apulia", "Basilicata", "Calabria",
                          Campania", "Emilia-Romagna", "Friuli-Venezia Giulia", "Lazio",
                          "Liguria", "Lombardy", "Marche", "Molise", "Piedmont",
                          "Sardinia", "Sicily", "Trentino-South Tyrol", "Tuscany",
                          "Umbria", "Veneto"),
                          state_code = c(65, 23, 75, 77, 78, 72, 45, 36, 62, 42, 25, 57,
                          67, 21, 88, 82, 32, 52, 55, 34),
                          country_code = rep("IT", 20)
                          )

region_to_macroregion <- c("Sud", "Nord-ovest", "Sud", "Sud", "Sud", "Sud", "Nord-est",
                          "Nord-est", "Centro", "Nord-ovest", "Nord-ovest", "Centro",
                    "Sud",
                          "Nord-ovest", "Isole", "Isole", "Nord-est", "Centro", "Centro",
                          "Nord-est")

italian_regions$macroregion <- region_to_macroregion
remove(macroregion_data)

# Merge NUTS info to main FirmsMunicipalities dataset ----
# NUTS1 = macroregion, NUTS2 = region, NUTS3 standardized municipality

merged <- merge(FirmsMunicipalities, italian_regions, by = "state_code", all.x = TRUE)

View(merged)
FirmsMunicipalities<- merged
View(FirmsMunicipalities)
remove(merged)
remove(region_to_macroregion, variables)
```

1.5.2 DATA MANIPULATION: EXIOBASE

```
# Load EXIOBASE direct requirement matrix ----
# Notice: we extract CO2 emission data, but EXIOBASE includes other CE related info

M <- read_csv("PATH to EXIOBASE DIRECT REQUIREMENT MATRIX")

# We have replicated the Exiobase methodology in another file
# Notice: it is a sectoral to country variable. We select row 26 because it is CO2
emission air. If we focus on a different pollutant, change the code accordingly (e.g.
M[38,] - water consumption)
```

```
#we need these 3 rows because the first 2 are nations and sector and the 26th is co2
emission air

first_row <- M[1, ]
second_row <- M[2, ]
measure_of_interest <- M[26, ]
new_dataset <- rbind(first_row, second_row, measure_of_interest)
View(new_dataset)
remove(first_row, second_row, measure_of_interest)

Co2 <- new_dataset
remove(new_dataset)

# Manual conversion of the sector x nation data from EXIOBASE to to nace and nation
variables from the ORBIS data (firmsmunicipalities). Here, we map every industry code
from EXIOBASE to NACE: for example (IT87)Manufacture of electrical machinery and
apparatus to its nace corresponding (nace from 2823 to 3100)

# Create a dataframe with numbers from 1000 to 3300
data <- data.frame(Numbers = 1000:3300)

# Create a new column based on the given ranges and names
data$Category <- NA
data$Category[data$Numbers >= 1000 & data$Numbers <= 1011] <- "IT34"
data$Category[data$Numbers == 1012] <- "IT36"
data$Category[data$Numbers == 1013] <- "IT37"
data$Category[data$Numbers >= 1014 & data$Numbers <= 1019] <- "IT38"
data$Category[data$Numbers >= 1020 & data$Numbers <= 1039] <- "IT44"
data$Category[data$Numbers >= 1040 & data$Numbers <= 1061] <- "IT39"
data$Category[data$Numbers >= 1062 & data$Numbers <= 1080] <- "IT40"
data$Category[data$Numbers == 1081] <- "IT41"
data$Category[data$Numbers >= 1082 & data$Numbers <= 1089] <- "IT42"
data$Category[data$Numbers >= 1090 & data$Numbers <= 1109] <- "IT43"
data$Category[data$Numbers >= 1110 & data$Numbers <= 1299] <- "IT45"
data$Category[data$Numbers >= 1300 & data$Numbers <= 1400] <- "IT46"
data$Category[data$Numbers >= 1401 & data$Numbers <= 1499] <- "IT47"
data$Category[data$Numbers >= 1500 & data$Numbers <= 1599] <- "IT48"
data$Category[data$Numbers >= 1600 & data$Numbers <= 1699] <- "IT49"
data$Category[data$Numbers >= 1700 & data$Numbers <= 1720] <- "IT51"
data$Category[data$Numbers >= 1721 & data$Numbers <= 1730] <- "IT53"
data$Category[data$Numbers >= 1731 & data$Numbers <= 1899] <- "IT54"
data$Category[data$Numbers >= 1900 & data$Numbers <= 1910] <- "IT55"
data$Category[data$Numbers >= 1911 & data$Numbers <= 1999] <- "IT56"
data$Category[data$Numbers >= 2000 & data$Numbers <= 2200] <- "IT62"
data$Category[data$Numbers >= 2200 & data$Numbers <= 2299] <- "IT63"
data$Category[data$Numbers >= 2300 & data$Numbers <= 2330] <- "IT64"
data$Category[data$Numbers >= 2331 & data$Numbers <= 2340] <- "IT66"
data$Category[data$Numbers == 2341] <- "IT67"
data$Category[data$Numbers == 2352] <- "IT68"
data$Category[data$Numbers >= 2353 & data$Numbers <= 2410] <- "IT71"
data$Category[data$Numbers >= 2410 & data$Numbers <= 2441] <- "IT73"
data$Category[data$Numbers == 2442] <- "IT75"
data$Category[data$Numbers == 2443] <- "IT77"
data$Category[data$Numbers == 2444] <- "IT79"
data$Category[data$Numbers == 2445] <- "IT81"
data$Category[data$Numbers == 2446] <- "IT57"
data$Category[data$Numbers >= 2447 & data$Numbers <= 2455] <- "IT83"
data$Category[data$Numbers >= 2456 & data$Numbers <= 2800] <- "IT84"
data$Category[data$Numbers >= 2801 & data$Numbers <= 2822] <- "IT85"
data$Category[data$Numbers == 2823] <- "IT86"
data$Category[data$Numbers >= 2823 & data$Numbers <= 3100] <- "IT87"
data$Category[data$Numbers >= 3101 & data$Numbers <= 3300] <- "IT92"


# Now we create this dataset with only Italian municipalities related to regions and
macroregions only for NACE manufacturing sectors (between 1000 and 3300). We focus on
manufacturing for the greater comparability between countries
```

```
category_mapping <- setNames(paste0("IT.", sprintf("%02d", 34:92)), paste0("IT",
sprintf("%02d", 34:92)))
NACEtoCo2 <- NACEtoCo2 %>% mutate(Matching_Column = category_mapping[Category])
NACEtoCo2$Co2 <- sapply(NACEtoCo2$Matching_Column, function(category) matching[1,
as.character(category)])

NACEtoCo2$Numbers <- as.character(NACEtoCo2$Numbers)

FirmsMunicipalities <- merged_dataset[, -c(49, 50)]
remove(merged_dataset)

# this mapping is necessary for the matching that we are going to do now

working <- FirmsMunicipalities[complete.cases(FirmsMunicipalities$macroregion) &
                        grepl("^\\d+$", FirmsMunicipalities$`NACE Rev. 2, core code (4
digits).x`) &
                        as.numeric(FirmsMunicipalities$`NACE Rev. 2, core code (4
digits).x`) >= 1000 &
                        as.numeric(FirmsMunicipalities$`NACE Rev. 2, core code (4
digits).x`) <= 3300, ]

working$Co2 <- as.numeric(as.character(working$Co2))

# ……………………………………………………………………………………………..
# TEST ----
# Exposure gap for nord-ovest macroregion: after we attached pollution values to sector
we create a ratio of italian firms #times their pollution and foreign firms.
# Notice: uncomment if needed

#numerator <- sum(working$Co2[working$macroregion == "Nord-ovest"], na.rm = TRUE) /
             #sum(!is.na(working$Co2[working$macroregion == "Nord-ovest"]))
#denominator <- sum(working$Co2[working$macroregion != "Nord-ovest"], na.rm = TRUE) /
             #sum(!is.na(working$Co2[working$macroregion != "Nord-ovest"]))
#exposure_ratio <- numerator / denominator

#print(exposure_ratio)
#[1] 1.088189

##check
#numerator_rows <- sum(!is.na(working$Co2[working$macroregion == "Nord-ovest"]))
#denominator_rows <- sum(!is.na(working$Co2[working$macroregion != "Nord-ovest"]))
#remove(numerator, numerator_rows, denominator, denominator_rows)

# End of test ……….………………………..………………………..………………………..

# Notice: Indicator might benefit from data on sectoral emission for all other countries.
We could wait #for better data or change it now
# Here a way to simulate firm level data but it would be better to have them. Let's
remove duplicates of #bvd id number

# Keep one randomly if there are multiple rows with the same "BvD ID number"

FirmsMunicipalities_unique <-
       FirmsMunicipalities %>%
       group_by(`BvD ID number`) %>%
              filter(!any(!is.na(`City\nLatin Alphabet`)) | is.na(`City\nLatin
Alphabet`)) %>%
             slice_sample(n = 1)

# filter to keep the row with no missing value for "City\nLatin Alphabet" that we know
are not there. If there are multiple rows with the same id after filtering, it randomly
selects one row to keep. Just to be sure that we don't miss municipalities

duplicate_count <- sum(duplicated(FirmsMunicipalities$`BvD ID number`))
print(paste(duplicate_count))
remove(duplicate_count)
```

```
# Keep one randomly for each group of duplicates
Data_municipalities_firms <-
       FirmsMunicipalities %>%
              group_by(`BvD ID number`) %>%
              slice_sample(n = 1)

#print(Data_municipalities_firms) # Print the resulting dataset

#Now, keep the observation with NACE code, but now we include country x sectors dyad
beyond Italy

Data_municipalities_firms <-
       Data_municipalities_firms[
              grepl("^\\d+$", Data_municipalities_firms$`NACE Rev. 2, core code (4
       digits).x`) &
              as.numeric(Data_municipalities_firms$`NACE Rev. 2, core code (4
digits).x`) >= 1000 &
              as.numeric(Data_municipalities_firms$`NACE Rev. 2, core code (4 digits).x`)
       <= 3300, ]

# Keep just the manufacturing sectors for all the 44 countries reported in EXIOBASE
# Identify manufacturing NACE codes from 1000 to 3300 for which we have a mapping

data <- data.frame(Numbers = 1000:3300)

# Create a new column based on the given ranges and names
data$Category <- NA

data$Category[data$Numbers >= 1000 & data$Numbers <= 1011] <- "34"
data$Category[data$Numbers == 1012] <- "36"
data$Category[data$Numbers == 1013] <- "37"
data$Category[data$Numbers >= 1014 & data$Numbers <= 1019] <- "38"
data$Category[data$Numbers >= 1020 & data$Numbers <= 1039] <- "44"
data$Category[data$Numbers >= 1040 & data$Numbers <= 1061] <- "39"
data$Category[data$Numbers >= 1062 & data$Numbers <= 1080] <- "40"
data$Category[data$Numbers == 1081] <- "41"
data$Category[data$Numbers >= 1082 & data$Numbers <= 1089] <- "42"
data$Category[data$Numbers >= 1090 & data$Numbers <= 1109] <- "43"
data$Category[data$Numbers >= 1110 & data$Numbers <= 1299] <- "45"
data$Category[data$Numbers >= 1300 & data$Numbers <= 1400] <- "46"
data$Category[data$Numbers >= 1401 & data$Numbers <= 1499] <- "47"
data$Category[data$Numbers >= 1500 & data$Numbers <= 1599] <- "48"
data$Category[data$Numbers >= 1600 & data$Numbers <= 1699] <- "49"
data$Category[data$Numbers >= 1700 & data$Numbers <= 1720] <- "51"
data$Category[data$Numbers >= 1721 & data$Numbers <= 1730] <- "53"
data$Category[data$Numbers >= 1731 & data$Numbers <= 1899] <- "54"
data$Category[data$Numbers >= 1900 & data$Numbers <= 1910] <- "55"
data$Category[data$Numbers >= 1911 & data$Numbers <= 1999] <- "56"
data$Category[data$Numbers >= 2000 & data$Numbers <= 2200] <- "62"
data$Category[data$Numbers >= 2200 & data$Numbers <= 2299] <- "63"
data$Category[data$Numbers >= 2300 & data$Numbers <= 2330] <- "64"
data$Category[data$Numbers >= 2331 & data$Numbers <= 2340] <- "66"
data$Category[data$Numbers == 2341] <- "67"
data$Category[data$Numbers == 2352] <- "68"
data$Category[data$Numbers >= 2353 & data$Numbers <= 2410] <- "71"
data$Category[data$Numbers >= 2410 & data$Numbers <= 2441] <- "73"
data$Category[data$Numbers == 2442] <- "75"
data$Category[data$Numbers == 2443] <- "77"
data$Category[data$Numbers == 2444] <- "79"
data$Category[data$Numbers == 2445] <- "81"
data$Category[data$Numbers == 2446] <- "57"
data$Category[data$Numbers >= 2447 & data$Numbers <= 2455] <- "83"
data$Category[data$Numbers >= 2456 & data$Numbers <= 2800] <- "84"
data$Category[data$Numbers >= 2801 & data$Numbers <= 2822] <- "85"
data$Category[data$Numbers == 2823] <- "86"
data$Category[data$Numbers >= 2823 & data$Numbers <= 3100] <- "87"
data$Category[data$Numbers >= 3101 & data$Numbers <= 3300] <- "92"
```

```r
column_names <- colnames(Co2)
desired_columns <- c()

for (column_name in column_names) {
	#extract the fourth and fifth characters from the column name
	numeric_part <- as.numeric(substr(column_name, 4, 5), na.rm = TRUE)

	#check if the numeric part is not missing and between 34 and 92
	if (!is.na(numeric_part) && numeric_part >= 34 && numeric_part <= 92) {
		# Add the column name to the vector of desired column names
		desired_columns <- c(desired_columns, column_name)
	}
}

# As before, first two letters refer to the country, the last two refer to the sector. We
do this for all EXIOBASE sectors that we want to map to NACE.

Manufacturing <- Co2[, desired_columns]

manufacturing_column_names <- colnames(Manufacturing) # Keep initial letters

unique_initial_letters <- c()

for (column_name in manufacturing_column_names) {
	initial_letters <- substr(column_name, 1, 2)
	if (!(initial_letters %in% unique_initial_letters)) {
		 print(initial_letters)
		unique_initial_letters <- c(unique_initial_letters, initial_letters)
	}
}

# Remove the world related to "World"

manufacturing_column_names <- colnames(Manufacturing)
columns_to_remove <-
	manufacturing_column_names[startsWith(manufacturing_column_names, "W")]

Manufacturing <- Manufacturing[, !(colnames(Manufacturing) %in% columns_to_remove)]

# Remove useless objects from environment

remove(unique_initial_letters, numeric_part, manufacturing_column_names, initial_letters,
desired_columns, column_name, column_names, columns_to_remove,)

# Replicate the measure_of_interest dataset for all countries to match the CO2 emission
for each country x sector dyad

num_replications <- 44 # Number of countries in EXIOBASE. Potentially, other datasets
might serve the same purpose.

data <- data[rep(seq_len(nrow(data)), times = num_replications), ]

rownames(replicated_data) <- NULL

remove(num_replications, num_rows_data, pairs_of_letters, rows_per_replication)

states <- c("AT", "BE", "BG", "CY", "CZ", "DE", "DK", "EE", "ES", "FI",
             "FR", "GR", "HR", "HU", "IE", "IT", "LT", "LU", "LV", "MT",
             "NL", "PL", "PT", "RO", "SE", "SI", "SK", "GB", "US", "JP",
             "CN", "CA", "KR", "BR", "IN", "MX", "RU", "AU", "CH", "TR",
              "TW", "NO", "ID", "ZA")

data$States <- rep(states, each = nrow(data) / length(states))

# Match CO2 data for all partner countries. ----
data_subset <- data[, c("States", "Category")]

data$Co2 <- NA
```

```
for (col_name_manufacturing in colnames(Manufacturing)) {
       state_from_manufacturing <- substr(col_name_manufacturing, 1, 2)
       category_from_manufacturing <- as.numeric(substr(col_name_manufacturing, 4, 5))

       match_rows <- which(
                             data_subset$States == state_from_manufacturing &
                             data_subset$Category == category_from_manufacturing)
       data$Co2[match_rows] <- Manufacturing[3, col_name_manufacturing]
}

remove(state_from_manufacturing, pairs_of_letters, match_rows, col_name_manufacturing,
       category_from_manufacturing)

remove(data_subset)

# Match the country x sector CO2 data it with the firm dataset

Data_municipalities_firms$Co2_Emission_Air <- NA

for (i in 1:nrow(Data_municipalities_firms)) {
       # extract the first two letters of the 'BvD ID number' column in
       'Data_municipalities_firms'
       state_Data_municipalities_firms <- substr(Data_municipalities_firms$`BvD ID
       number`[i], 1, 2)
       # extract the 'NACE Rev. 2, core code (4 digits).x' value in
       'Data_municipalities_firms'
       number_Data_municipalities_firms <-
              as.numeric(Data_municipalities_firms$`NACE Rev. 2, core code (4
       digits).x`[i])

       match_rows <- which(
                     data$States == state_Data_municipalities_firms & data$Numbers ==
                     number_Data_municipalities_firms)

       if (length(match_rows) > 0) {
              Data_municipalities_firms$Co2_Emission_Air[i] <- data$Co2[match_rows]
       }
}

remove(data_subset, FirmsBalance, merged_datay, bvd_id_from_firmsmunicipalities,
       col_name_firmsmunicipalities, i, match_rows)

remove(merged_data, number_Data_municipalities_firms, state_Data_municipalities_firms)

names(Data_municipalities_firms)[names(Data_municipalities_firms) == "Co2"] <-
"Co2_Italy"

# Check missing obs
#na_count <- sum(is.na(Data_municipalities_firms$Co2_Emission_Air))
#remove(na_count)

Data_municipalities_firms <-
       Data_municipalities_firms[complete.cases(Data_municipalities_firms$Co2_Emission_Ai
r), ]

# ok now we will make use of the city to region to macro region relation, for now we did
not use it
# as we just used the first two letters of the bvd id number
# now we are going to need that we do not have rows which have either a bvd number or a
city missing

Data_municipalities_firms <- Data_municipalities_firms[grepl("^IT",
Data_municipalities_firms$`BvD ID number`)
& !is.na(Data_municipalities_firms$`City\nLatin Alphabet`), ]

# we missed some municipalities before so now we rematch them, missing in the sense that
we do not have the match between municipality and region
```

```
Municipalities <- read.csv("D:\\Orbis\\municipalities.csv")

Municipalities <- Municipalities[Municipalities$country_code == "IT", ]
Municipalities$name[Municipalities$name == "Città metropolitana di Roma Capitale" &
rownames(Municipalities) == "54576"] <- "Roma"
Municipalities$name[Municipalities$name == "Città metropolitana di Milano" &
rownames(Municipalities) == "55733"] <- "Milano"
rownames(Municipalities) <- NULL

# again other formatting issues we put them to lower and to ascii format
library(stringi)

Municipalities$name_clean <- stri_trans_general(Municipalities$name, "Latin-ASCII")
Municipalities$name_clean <- tolower(Municipalities$name_clean)

merged_data <- merge(Data_municipalities_firms, Municipalities, by.x = "City_Lower", by.y
= "name_clean", all.x = TRUE)

unique_merged_data <- merged_data %>% distinct(City_Lower, .keep_all = TRUE)

Data_municipalities_firms$state_code <-
unique_merged_data$state_code.y[match(Data_municipalities_firms$City_Lower,
unique_merged_data$City_Lower)]

remove(merged_data, unique_merged_data)

missing_count <- sum(is.na(Data_municipalities_firms$state_code))
remove(missing_count)
#we see that we have some missing municipalities that do not match, so now we manually
add them

Data_municipalities_firms$state_code[Data_municipalities_firms$City_Lower == "genova"] <-
"42"
Data_municipalities_firms$state_code[Data_municipalities_firms$City_Lower == "firenze"]
<- "52"
Data_municipalities_firms$state_code[Data_municipalities_firms$City_Lower == "venezia"]
<- "34"
Data_municipalities_firms$state_code[Data_municipalities_firms$City_Lower == "carpi"] <-
"45"
Data_municipalities_firms$state_code[Data_municipalities_firms$City_Lower == "fiorano
modenese"] <- "34"

missing_city_table <-
table(Data_municipalities_firms$City_Lower[is.na(Data_municipalities_firms$state_code)])

sorted_missing_table <- sort(missing_city_table, decreasing = TRUE)

top_missing_values <- names(head(sorted_missing_table, 10))

cat(top_missing_values, "\n")
remove(missing_city_table, sorted_missing_table, top_missing_values)


Data_municipalities_firms <-
Data_municipalities_firms[complete.cases(Data_municipalities_firms$state_code), ]

Data_municipalities_firms$macroregion <-
italian_regions$macroregion[match(Data_municipalities_firms$state_code,
italian_regions$state_code)]

missing_macroregion_count <- sum(is.na(Data_municipalities_firms$macroregion))
      remove(missing_macroregion_count)
```

1.5.3 DATA MANIPULATION: SUBSIDIARIES (IND 2)

```
# In the first extraction we only downloads GUO data, not their subsidiaries. If
subsidiaries data is extracted alongside the owners, the following code is not necessary.

extracted_column <- Firms$`Subsidiary - BvD ID number`

output_data <- data.frame("Subsidiary - BvD ID number" = extracted_column)

output_data <- unique(output_data)

output_data1 <- output_data[grepl("^IT", output_data[, 1]), , drop = FALSE]

write.csv(output_data1, file = "PATH TO SUBSIDIARY ORBIS DATA", row.names = FALSE)

remove(output_data,output_data1)


# Here we extract data on subsidiaries such as sales and city for subsidiaries, to be
downloaded again with the scraper described on top.

setwd(PATH to ORBIS SUBSIDIARIES DATA)

files <- list.files(pattern = "\\.xlsx$")

fn.import <- function(x) {
       df <- read_excel(x, sheet = "Results")
       return(df)
}

data_frame <- lapply(files, fn.import)
data_frame <- bind_rows(data_frame)

Subs <- data_frame
remove(data_frame)
remove(files, fn.import)

# let's take out the owners with non-italian subsidiaries only.

Data_municipalities_firms <-
       Data_municipalities_firms[grep("^IT", Data_municipalities_firms$`Subsidiary - BvD
ID number`), ]


FirmsNace <- subset(FirmsNace, grepl("^IT", `Subsidiary - BvD ID number`))

# remove duplicate owners and their subsidiaries too, so we see there is already a
necessary relation between owners and their subsidiaries

duplicates <- duplicated(FirmsNace$`BvD ID number`)

rows_to_remove <- logical(nrow(FirmsNace))

for (i in seq_along(duplicates)) {

       if (duplicates[i]) {
              # If it's a duplicate, mark the row and subsequent rows without a BvD ID
              number
              rows_to_remove[i:(i + sum(is.na(FirmsNace$`BvD ID
              number`[i:(length(duplicates))])))] <- TRUE
       }
}

FirmsNace <- FirmsNace[!rows_to_remove, ]

remove(duplicates, rows_to_remove, true_count)


library(dplyr)
```

```r
FirmsNace <- FirmsNace %>% filter(
       !((is.na(`Subsidiary - Direct %`) | as.numeric(gsub("[^0-9.]", "", `Subsidiary -
       Direct %`)) < 50) & (is.na(`Subsidiary - Total %`) | as.numeric(gsub("[^0-9.]",
       "", `Subsidiary - Total %`)) < 50)
       ))

#match info subsidiaries Subs is the same dataset as before with the info of all the
subsidiaries

FirmsNace <- FirmsNace %>%
       left_join(Subs, by = c("Subsidiary - Name" = "Company name Latin alphabet"))

# column for owner's nations and city
# we are attaching static values, which are other values that we missed from the first
download such as the sales for a certain firm. We merge them by bvd id number

Static_non_na <- na.omit(Static$`BvD ID number`)
FirmsNace_non_na <- na.omit(FirmsNace$`BvD ID number`)
common_ids <- intersect(Static_non_na, FirmsNace_non_na)

FirmsNace$`Company City` <- NA

for (id in common_ids) {

       static_indices <- which(Static$`BvD ID number` == id)
       firmsnace_indices <- which(FirmsNace$`BvD ID number` == id)

       if (length(static_indices) > 0 && length(firmsnace_indices) > 0) {

              FirmsNace$`Company City`[firmsnace_indices] <- Static$`City\nLatin
              Alphabet`[static_indices]
       }

}

remove(common_ids, firmsnace_indices, FirmsNace_non_na, static_indices, Static_non_na)
remove(id)

# now we are attaching to the subsidiaries the owner’s state and city, so that we can
disaggregate between nuts 1 2 3 again and see the influence on subsidiaries to have a
owner from a #certain nuts compared to another

FirmsNace$`owner state` <- NA
FirmsNace$`owner city` <- NA

for (i in 1:nrow(FirmsNace)) {

       # check if the current row has a non-missing BvD ID number
       if (!is.na(FirmsNace$`BvD ID number`[i])) {

              owner_state <- substr(FirmsNace$`BvD ID number`[i], 1, 2)
              # Assign the "owner state" to the current and subsequent rows with missing
              BvD ID number
              FirmsNace$`owner state`[i:nrow(FirmsNace)] <- owner_state
              # Assign the "owner city" to the current and subsequent rows with missing
              BvD ID number
              FirmsNace$`owner city`[i:nrow(FirmsNace)] <- FirmsNace$`Company City`[i]
       }
}

#now match co2 (or other CE related indicators) sectoral italian with the subsidiaries
sector

subset_co2 <- sector_to_nation_co2[sector_to_nation_co2$States == "IT", ]

FirmsNace$`Co2 emission subsidiaries` <- NA

for (i in 1:nrow(FirmsNace)) {
```

```
        nace_code_FirmsNace <- as.numeric(FirmsNace$`NACE Rev. 2, core code (4
        digits).y`[i])
        #Matching rows in subset_co2
        match_rows <- which(subset_co2$Numbers == nace_code_FirmsNace)

        if (length(match_rows) > 0) {

                FirmsNace$`Co2 emission subsidiaries`[i] <- subset_co2$Co2[match_rows[1]]
        }
}

# we adjust the owner city and do the matching as we did before

FirmsNace$`owner city` <- tolower(iconv(FirmsNace$`owner city`, to = "ASCII//TRANSLIT"))

remove(merged_data, subset_co2, denominator, denominator_count, i, exposure_gap_national,
match_rows)
remove(nace_code_FirmsNace, numerator, numerator_count, owner_state)
remove(CityGdp, filtered_municipalities)

# we also take out accents with ascii translit and so now we can marge the owner city to
the name clean

result <- FirmsNace %>% left_join(Municipalities, by = c("owner city" = "name_clean"))

FirmsNace <- result
remove(result)

italian_regions <- italian_regions %>% mutate(state_code = as.character(state_code))
result <- FirmsNace %>% left_join(italian_regions, by = c("state_code" = "state_code"))

FirmsNace <- result
remove(result)
```

1.5.4 DATA MANIPULATION: FOREIGN EXPORT (IND 3)

```
keep_rows <- logical(nrow(Firms))

# initialize a flag to keep track of whether the current group starts with "IT"

start_IT_group <- FALSE

for (i in 1:nrow(Firms)) {

        bvd_id_number <- Firms$`BvD ID number`[i]

        # Check if the current row starts with "IT"
        if (!is.na(bvd_id_number) && !grepl("^IT", bvd_id_number)) {

                start_IT_group <- TRUE
        }

        keep_rows[i] <- start_IT_group

        # if 'BvD ID number' is starting with "IT" or is missing, reset the flag
        if (!is.na(bvd_id_number) && grepl("^IT", bvd_id_number)) {

                start_IT_group <- FALSE
        }
}

Firms_filtered <- Firms[keep_rows, ]
```

```
Firms_filtered <- Firms_filtered %>%
       filter(is.na(`Country ISO code`) | `Country ISO code` != 'IT')

Firms_filtered <- Firms_filtered[!(grepl("^IT", Firms_filtered$`BvD ID number`)
& !is.na(Firms_filtered$`BvD ID number`)), ]

count_IT_rows_bvd <- sum(!is.na(Firms_filtered$`BvD ID number`) & grepl("^IT",
Firms_filtered$`BvD ID number`))

cat("Number of rows with 'BvD ID number' starting with 'IT':", count_IT_rows_bvd, "\n")

# add city subsidiary by using the previous database, no city owners because foreign

Firms_filtered$city_subsidiary <- NA

match_rows <- match(Firms_filtered$`Subsidiary - BvD ID number`, FirmsMunicipalities$`BvD
ID number`)

non_missing_matches <- !is.na(match_rows)

Firms_filtered$city_subsidiary[non_missing_matches] <-
FirmsMunicipalities$City_Lower[match_rows[non_missing_matches]]

Firms_filtered <- Firms_filtered %>% filter(
       !((is.na(`Subsidiary - Direct %`) | as.numeric(gsub("[^0-9.]", "", `Subsidiary -
       Direct %`)) < 50) & (is.na(`Subsidiary - Total %`) | as.numeric(gsub("[^0-9.]",
       "", `Subsidiary - Total %`)) < 50)))

na_count <- sum(is.na(Firms_filtered$`Subsidiary - Bvd ID number`))

subsidiary_bvd_id <- Firms_filtered$`Subsidiary - BvD ID number`

write.csv(subsidiary_bvd_id, file = "Subsidiary_BvD_ID.csv", row.names = FALSE)

library(readxl)
library(tidyverse)

path <- "PATH TO SUBSIDIARY FOLDER"
setwd(path)

files <- list.files(pattern = "\\.xlsx$")

fn.import <- function(x) {

       df <- read_excel(x, sheet = "Results")
       return(df)
}

data_frame <- lapply(files, fn.import)

data_frame <- bind_rows(data_frame)

Subs3 <- data_frame
remove(data_frame)
remove(files, path, fn.import)

Firms_filtered <- Firms_filtered %>% left_join(Subs2, by = c("Subsidiary - BvD ID number"
= "BvD ID number"))

Foreign2<-Firms_filtered

# again keep kust the codes between 1000 and 3300 for which we have the matching, we can
change #this
Foreign2$NACE_numeric <- as.numeric(Foreign2$`NACE Rev. 2, core code (4 digits).y`)
Foreign2 <- Foreign2 %>%  filter(!is.na(NACE_numeric) & NACE_numeric >= 1000 &
NACE_numeric <= 3300)

# now we also match the pollution values for the new subs that are also foreign
```

```
Foreign2 <- Foreign2 %>% mutate(subsidiary_emission = NA)

# Loop through each row in Italy2 to fill subsidiary_emission

for (i in 1:nrow(Foreign2)) {

       iso_code <- Foreign2[i, "Country ISO code.y"]
       nace_code <- Foreign2[i, "NACE Rev. 2, core code (4 digits).y"]

       # Find matching row in sector_to_nation_co2
       match_row <- sector_to_nation_co2 %>% filter(States == iso_code, Numbers ==
       as.numeric(nace_code)) %>%  slice(1)  # Take the first match if multiple
       # Fill subsidiary_emission with Co2 value if match is found

       if (nrow(match_row) > 0) {

              Foreign2[i, "subsidiary_emission"] <- match_row$Co2
       }
}
```

1.5.5 DATA MANIPULATION: ITALY EXPORT (IND 4)

```
keep_rows <- logical(nrow(Firms))

# initialize a flag to keep track of whether the current group starts with "IT"

start_IT_group <- FALSE

for (i in 1:nrow(Firms)) {

       bvd_id_number <- Firms$`BvD ID number`[i]

       # Check if the current row starts with "IT"
       if (!is.na(bvd_id_number) && grepl("^IT", bvd_id_number)) {

              start_IT_group <- TRUE
       }

       keep_rows[i] <- start_IT_group

       if (!is.na(bvd_id_number) && !grepl("^IT", bvd_id_number)) {

              start_IT_group <- FALSE
       }
}

Firms_filtered <- Firms[keep_rows, ]

##### now take out duplicates bvd

cat("duplicates not missing:", sum(duplicated(Firms_filtered$`BvD ID
number`[!is.na(Firms_filtered$`BvD ID number`)])), "\n")

keep_rows <- logical(nrow(Firms_filtered))
unique_bvd_ids <- c()

# iterate through each row
for (i in 1:nrow(Firms_filtered)) {

       bvd_id_number <- Firms_filtered$`BvD ID number`[i]

       # Check if the BvD ID number is missing or not in the vector
       if (!is.na(bvd_id_number) && !(bvd_id_number %in% unique_bvd_ids)) {
```

```
        unique_bvd_ids <- c(unique_bvd_ids, bvd_id_number)
    }

    # Update the 'keep_rows' vector
    keep_rows[i] <- !duplicated(bvd_id_number) || is.na(bvd_id_number)
}

Foreign2 <- Firms_filtered[keep_rows, ]

# add city owner by using the previous database firmsmunicipalities, the subsidiary
municipalities will be matched later

Italy2$city_owner <- NA

match_rows <- match(Italy2$`BvD ID number`, FirmsMunicipalities$`BvD ID number`)
non_missing_matches <- !is.na(match_rows)

Italy2$city_owner[non_missing_matches] <-
FirmsMunicipalities$City_Lower[match_rows[non_missing_matches]]

Italy2 <- Italy2 %>% filter(
    !((is.na(`Subsidiary - Direct %`) | as.numeric(gsub("[^0-9.]", "", `Subsidiary -
    Direct %`)) < 50) & (is.na(`Subsidiary - Total %`) | as.numeric(gsub("[^0-9.]",
    "", `Subsidiary - Total %`)) < 50)))

na_count <- sum(is.na(Italy2$`Subsidiary - Bvd ID number`))

subsidiary_bvd_id <- Italy2$`Subsidiary - BvD ID number`

write.csv(subsidiary_bvd_id, file = "Subsidiary_BvD_ID.csv", row.names = FALSE)


remove(current_firms)
remove(other_firms)
remove(bvd_id_number, i, keep_rows, match_rows, na_count, non_italian_count,
non_italian_rows, non_missing_matches, start_IT_group)
remove(match_row, subsidiary_bvd_id, unique_bvd_ids)

# Import the data back in

library(readxl)
library(tidyverse)

path <- "PATH TO SUBSIDIARY FOLDER
setwd(path)

files <- list.files(pattern = "\\.xlsx$")

fn.import <- function(x) {

    df <- read_excel(x, sheet = "Results")
    return(df)
}

data_frame <- lapply(files, fn.import)
data_frame <- bind_rows(data_frame)

Subs2 <- data_frame
remove(data_frame)
remove(files, path, fn.import)

# now join subsidiaries with ownership data
#Italy2 <- Italy2 %>%
#  left_join(Subs2, by = c("Subsidiary - Name" = "Company name Latin alphabet"))

Italy2 <- Italy2 %>%
    left_join(Subs2, by = c("Subsidiary - BvD ID number" = "BvD ID number"))
```

```r
library(zoo)

#fill missing values in 'city_owner' column with the last non-missing value
Italy2$city_owner <- na.locf(Italy2$city_owner)

#filter again for non italian owners

start_IT_group <- FALSE
keep_rows <- logical(nrow(Italy2))

for (i in 1:nrow(Italy2)) {

	country_iso_code <- Italy2$`Country ISO code.x`[i]

	# Check if the current row has 'Country ISO code.x' equal to 'IT'
	if (!is.na(country_iso_code) && country_iso_code == 'IT') {

		start_IT_group <- TRUE
	}

	keep_rows[i] <- start_IT_group

	# If 'Country ISO code.x' is not equal to 'IT' and is not missing, reset the flag
	if (!is.na(country_iso_code) && country_iso_code != 'IT') {

		start_IT_group <- FALSE
	}
}

Italy2_filtered <- Italy2[keep_rows, ]

Italy2_filtered <- Italy2_filtered %>%
	filter(is.na(`Country ISO code.x`) | `Country ISO code.x` == 'IT')

non_italian_rows <- Italy2_filtered %>%
	filter(!is.na(`BvD ID number`) & !grepl("^IT", `BvD ID number`, ignore.case =
	TRUE))
cat("fine:", nrow(non_italian_rows), "\n")

Italy2<-Italy2_filtered
remove(Italy2_filtered)
remove(bvd_id_number, i, keep_rows, match_rows, na_count, non_italian_count,
non_italian_rows, non_missing_matches, start_IT_group)
remove(match_row, subsidiary_bvd_id, unique_bvd_ids)
remove(country_iso_code)

# now we have nace and nationality of the subsidiaries (foreign and domestic) of italian
owners, we attach pollution value for their sector + state

Italy2 <- Italy2 %>% mutate(subsidiary_emission = NA)

# Loop through each row in Italy2 to fill subsidiary_emission
for (i in 1:nrow(Italy2)) {

	iso_code <- Italy2[i, "Country ISO code.y"]
	nace_code <- Italy2[i, "NACE Rev. 2, core code (4 digits).y"]

	# Find matching row in sector_to_nation_co2
	match_row <- sector_to_nation_co2 %>%
		filter(States == iso_code, Numbers == as.numeric(nace_code)) %>% slice(1)
		# Take the first match if multiple

	# Fill subsidiary_emission with Co2 value if match is found
	if (nrow(match_row) > 0) {

		Italy2[i, "subsidiary_emission"] <- match_row$Co2
	}
}
```

3. Circular Economy and Trade

3.1 Data

The data from COEWEB corresponds to customs data recorded at the transaction level by the firm reporting the trade flow at the 8-digit level, defined at the CN8 classification, which corresponds to the 6-digit international harmonized system classification plus 2 additional national digits, and reporting the partner country, destination for export and origin for imports.

We have available data on the annual frequency of exports and imports at the firm-product-partner country triplet which allows us to explore several dimensions of the trade flow behaviour. The data goes from 2010 to 2021. Moreover, we have the data reported at the province level which allows us to provide indicators at this geographical level.

To be able to link our data to other datasets and create a consistent dataset over time, we aggregate the data at the HS 6-digit level and use concordance tables available from UNCTAD to consistently classify goods over time[5].

3.2 Definitions:

Eurostat has created a list of circular economy industries based on NACE/ISIC classification. The list includes the 'traditional' sectors involved in the circular economy, such as recycling and waste management. However, the circular economy is present in all stages of the product life cycle from design to use, recycling etc. To drive a circular economy, all activities leading to circularity should be measured. For example, sustainable design and efforts to lengthen product life cycle to minimize waste are essential parts of circularity.

[5] These concordances tables are available at https://unstats.un.org/unsd/classifications/Econ.

3.2.1 Recyclable Raw Materials

The first set of analysis focuses on recyclable raw materials (waste) as defined by "List of CN-codes used for the calculation of Trade in recyclable raw materials" from the Eurostat, from February 2023. This includes residues and waste from diverse industries: paper, plastic, rubber, wood, textiles, glass, organic materials, minerals, and metals. They are residues that could be recycled or reused. It includes 212[6] product categories from several industries as mentioned above.

The indicators we construct are:

(a) Share of exporting (importing) firms that export (import) circular goods, by province.

This corresponds to the ratio between the number of exporting (importing) firms that export (import) circular goods and the total number of firms, by province and year, over the 2010 and 2021 period, which is expressed in the expression below:

$$Sh.FirmsExportCirc._{rt} = \frac{NumberFirmsExportCirc._{rt}}{Number\ of\ Exporting\ Firms_{rt}}$$

where r is the province and t the year.
An analogous expression holds for the importing firms.

(b) Share of the value of exports (imports) of circular goods over the total exports, by province.

This corresponds to the ratio between the value of exports (imports) of circular goods and the total number of firms, by province and over the 2010 and 2021 period which is expressed in the expression below:

[6] At the 6-digit level, our unit of analysis, this number reduces to around 105 goods.

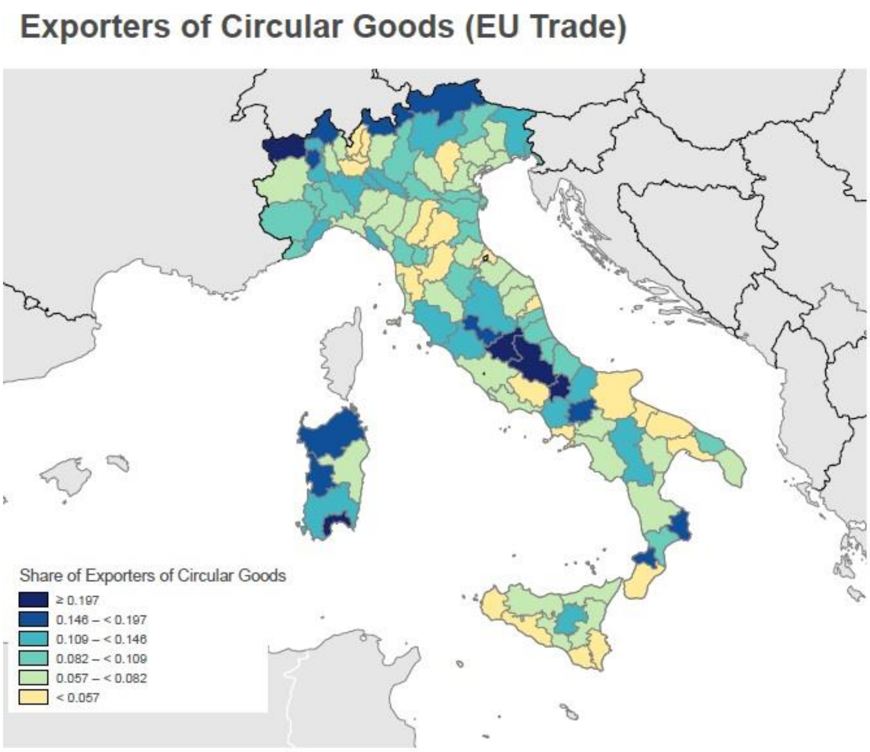
$$Sh.ValueExportCirc._{rt} = \frac{Total\,Value\ of\ Export\ Circ._{rt}}{Total\,Value\ of\ Exports_{rt}}$$

An analogous expression holds for the importing firms.

Noteworthy, these shares are calculated from firm-level data and correspond to the contribution of the fraction of firms described in the previous item to the export value.

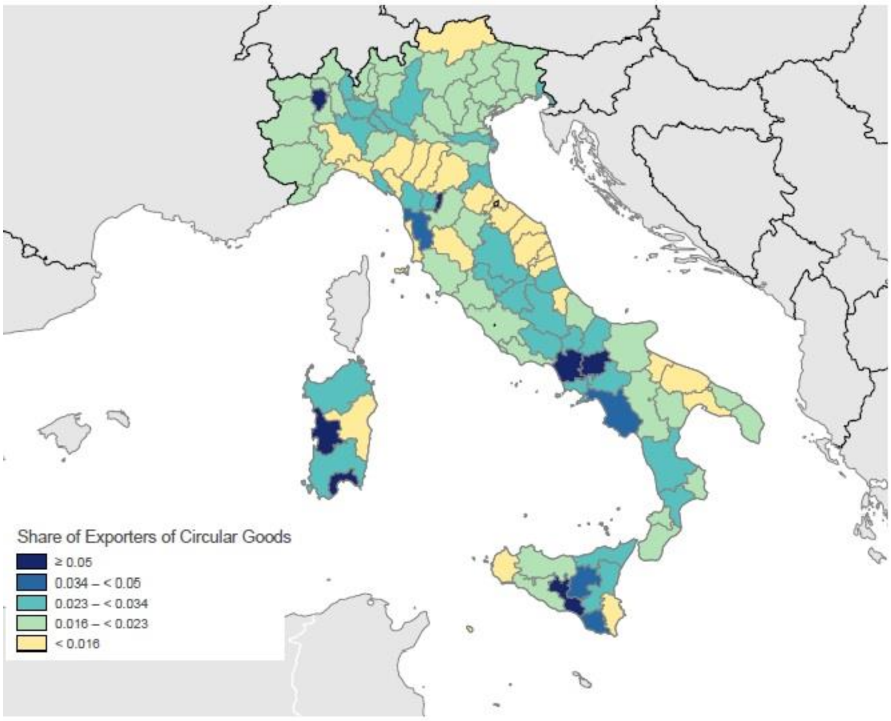
3.3 Overview of the results

With these indicators one can construct maps that show the relative importance of the circular economy for each province of Italy. Below we show the maps representing the average share over the entire period of the sample. They are the maps with the share of exporting firms that export circular goods for the EU and non-EU separately; and the maps showing the share of export value of circular goods relative to the total export value of firms for the EU and non-EU trade.



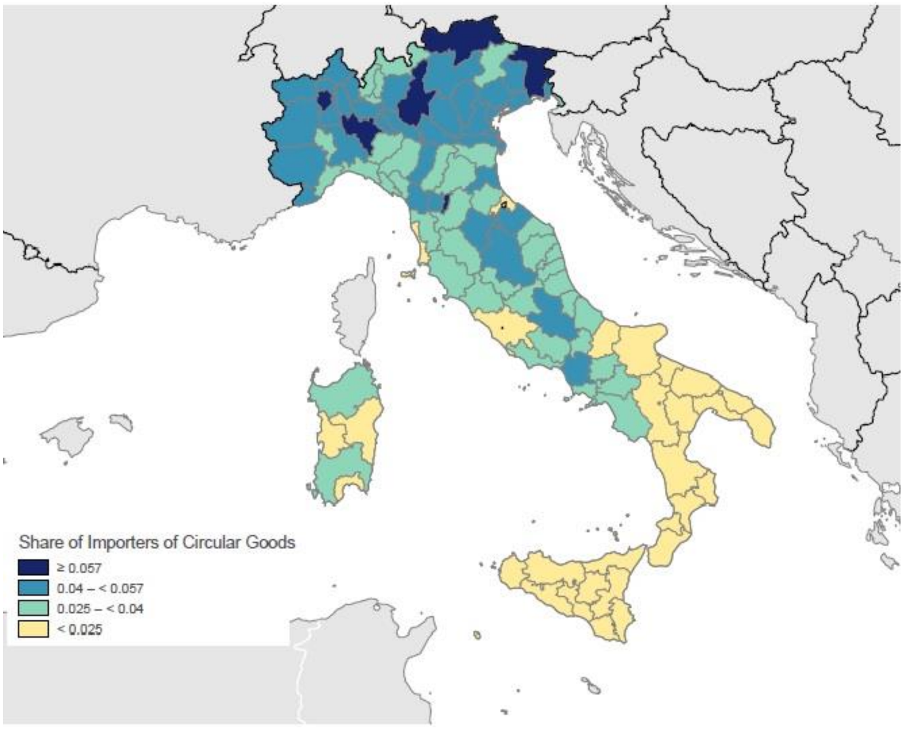
Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

Exporters of Circular Goods (non-EU Trade)



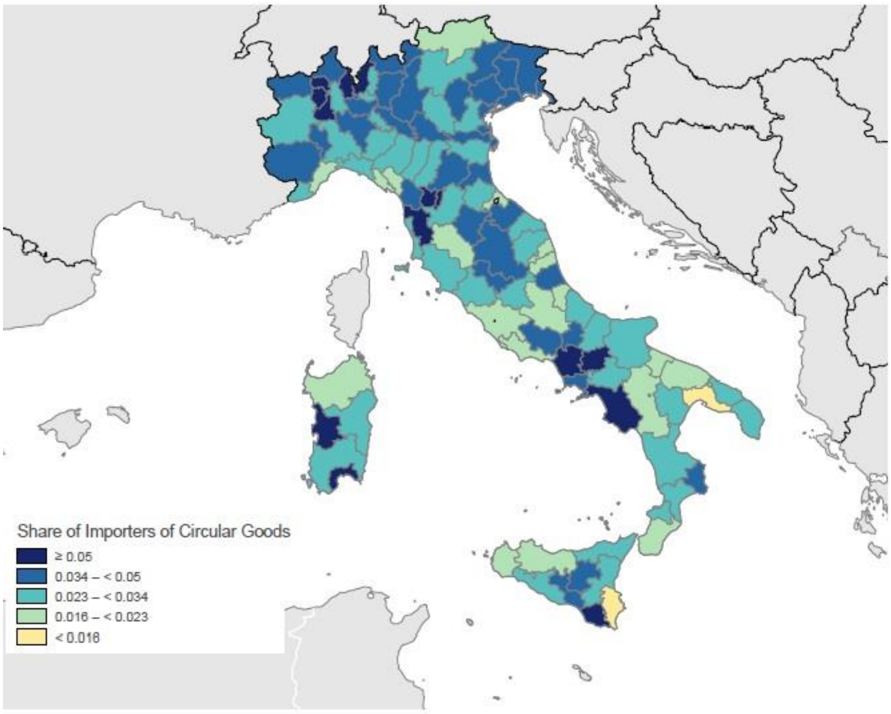
Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

Importers of Circular Goods (EU Trade)



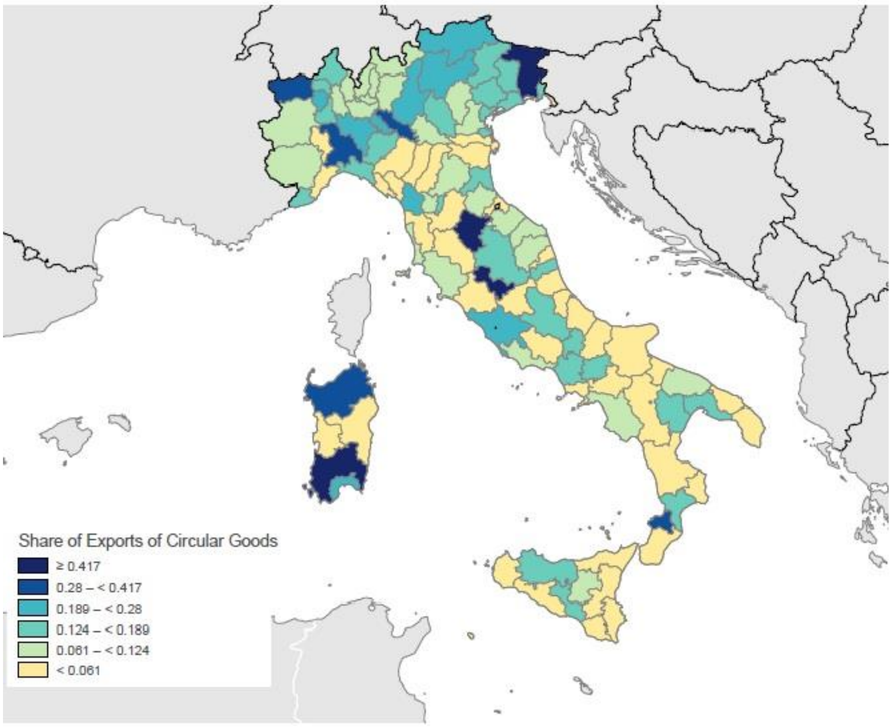
Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

Importers of Circular Goods (non-EU Trade)

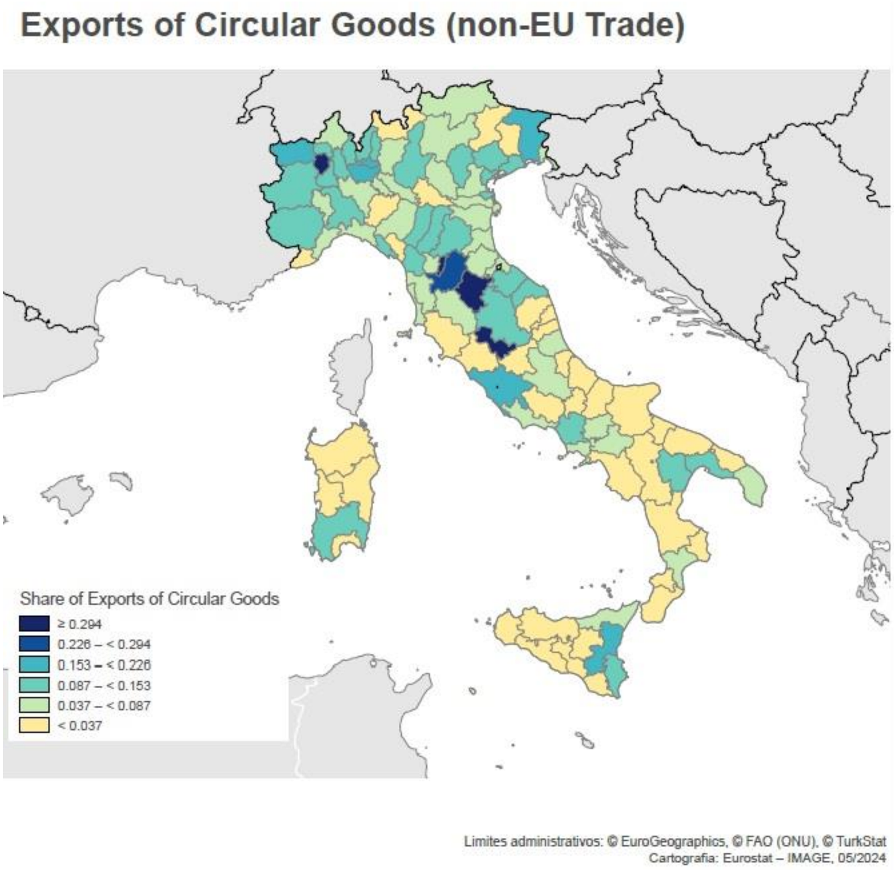


Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

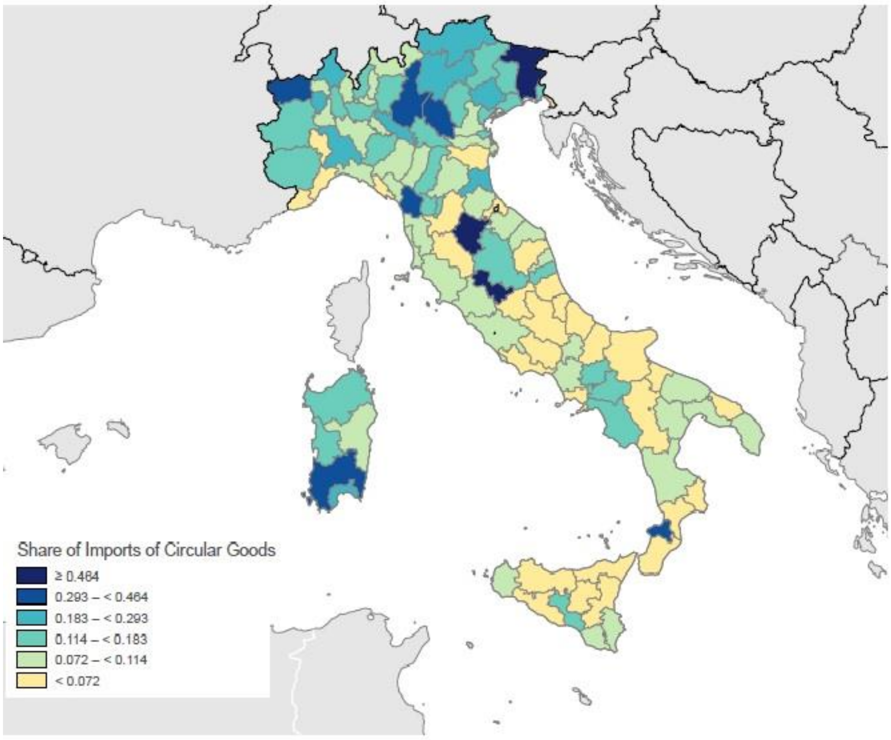
Share of Exports of Circular Goods (EU Trade)



Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

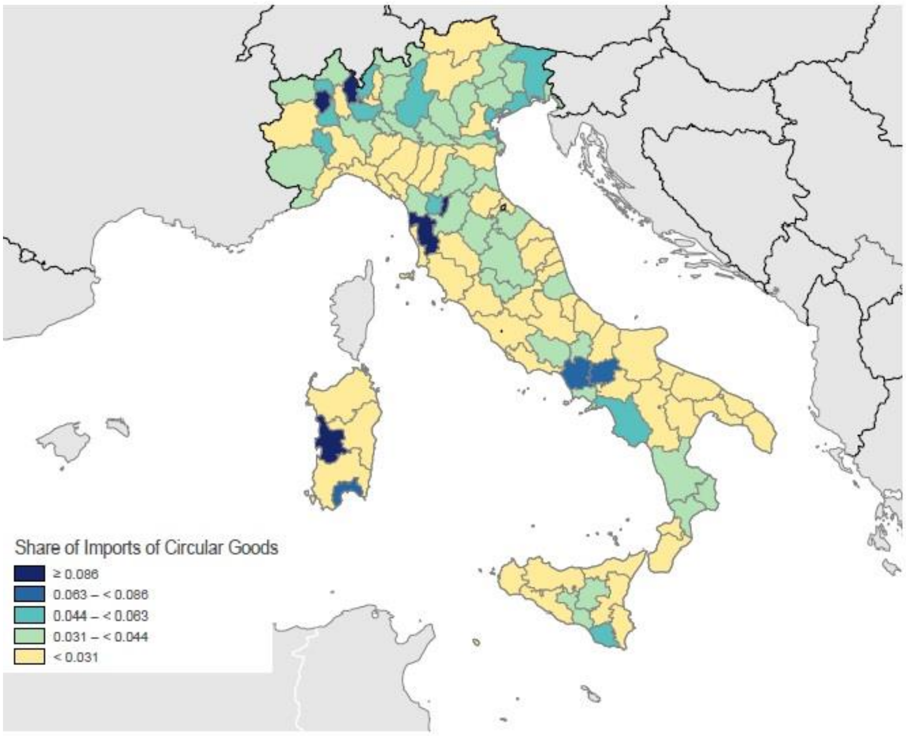

Exports of Circular Goods (non-EU Trade)
Share of Exports of Circular Goods
≥ 0.294
0.226 – < 0.294
0.153 – < 0.226
0.087 – < 0.153
0.037 – < 0.087
< 0.037
Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

Share of Imports of Circular Goods (EU Trade)



Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

Imports of Circular Goods (non-EU Trade)



Limites administrativos: © EuroGeographics, © FAO (ONU), © TurkStat
Cartografia: Eurostat – IMAGE, 05/2024

4. Firms and Circular Sectors of the Economy

Another set of analysis focuses on what is defined by Eurostat as "circular economy goods and services". We use the list that includes mainly two categories of goods and services. Sectors that serve primarily the circular economy, such as waste collection, treatment, equipment for recycling activities and secondary raw material. Additional sectors that are not primarily a circular economy industry but serve a secondary circular economy purpose are included. They contribute to increasing the efficiency of use of resources such as e-books, leasing and renting.

There are in total 80 industries classified as involved in the circular economy at the 6-digit level ATECO classification. They comprise sub-sectors of mining and manufacturing industries.

We link the ATECO 2007 classification to a correspondence table available from Eurostat that links NACE Revision 2 sectors with CPA 2008 product classification and PRODCOM 2208 product

classification. The first 4 digits of the ATECO 2007 classification corresponds to the NACE Revision 2 sector with the last 2 digits being specific.

We did the match by identifying the characteristics of each product and industry and comparing those with ATECO 2007's description. In most cases, the NACE sector already defined the ATECO2007 industry since it included only one subsector. In other cases, a more careful match had to be done. We did it case-by-case with the additional help from the descriptions available from Istat.[7]

[7] See the website codiceateco.it. We also provide in the Appendix the list with the corresponding concordances.

4.1 Firm-Level Data in Circular Sectors (AIDA)

We develop indicators that explore balance sheet information from the AIDA firm-level dataset on the Italian firms to obtain information

on the number of firms operating in the circular economy sectors and to build the corresponding indicators. The data comes from AIDA-Orbis from 2010 to 2021 and it covers the entire Italian economy.

We match the firm-level data with the circular economy sectors, as described above, and construct the indicators on firms' participation in the circular economy.

4.2 Definitions

The indicators constructed are:

(c) Share of firms operating in the circular economy sectors, by province and year.

This corresponds to the ratio between the number of firms operating in the circular economy sectors and the total number of firms, by province and over the 2010 and 2021 period, as the expression below shows:

$$Sh.FirmsCirc._{rt} = \frac{Number\ of\ Firms\ Circ._{rt}}{Total\ Number\ of\ Firms_{rt}}$$

(d) Share of manufacturing firms operating in the circular economy sectors, by province and year.

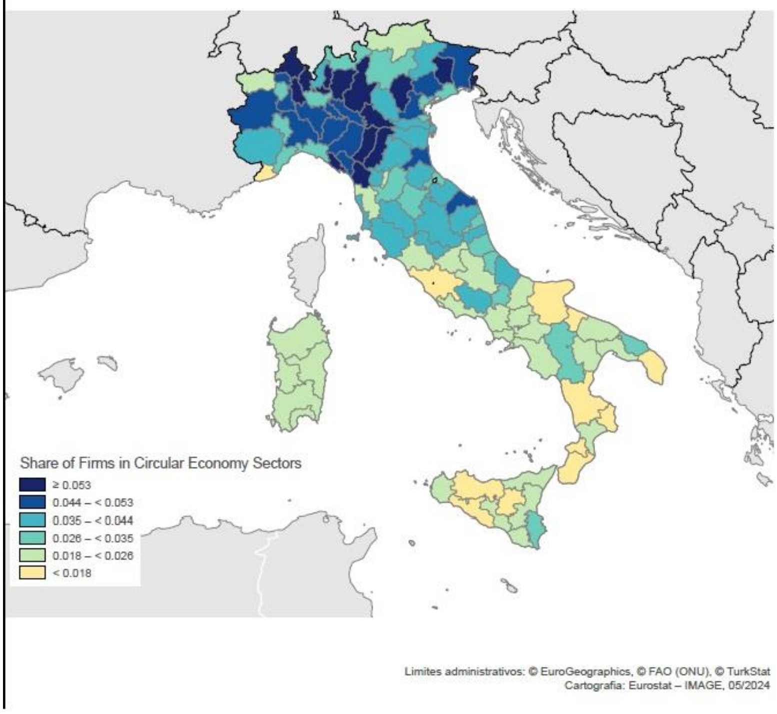
$$Sh.Manuf.FirmsCirc._{rt} = \frac{Number\ of\ Manuf.Firms\ Circ._{rt}}{Total\ Number\ of\ Manuf.Firms_{rt}}$$

This corresponds to the ratio between the number of manufacturing firms operating in the circular economy sectors and the total number of manufacturing firms, by province and over the 2010 and 2021 period.

4.3 Overview of the results

With these indicators one can construct maps like the ones presented in the previous section. They also show the relative importance of the circular economy for each province of Italy, however, they represent the importance of firms operating in circular economy sectors for the entire sample and constrained to the set of manufacturing firms. Below we show the maps representing the average share over the entire period of the sample.

Firms in Circular Economy Sectors



5. Circular Economy and GVCs: From Micro-to-Macro

5.1 Introduction

This part of the project seeks to comprehend how trade and value chains influence the circular, horizontal, and linear structures of the economy. Global value chains are constructed through the horizontal and vertical integration of various firms worldwide, creating input-output linkages. Nevertheless, the growing awareness of environmental and ethical considerations is prompting firms in these value chains to adopt a new approach. Industries are now shifting their focus towards responsible sourcing, aligning with the principles of the circular economy.

This section enhances existing literature by identifying the factors influencing engagement in the circular economy. Through firm-level surveys conducted in Italy, the research offers indicators for the circular economy at both the individual firm level and the aggregated region level. These indicators help gauge the extent of a region's involvement in the circular economy. Additionally, the study goes a step further by estimating the relationship between participation in global value chains (GVC) and investment in the circular economy.

Furthermore, it introduces a novel measurement guide for circular economy investment. The project develops indicators based on firm-level participation measures. These measures are then aggregated at the region level, each given equal weight, offering a new approach to measuring investment in both the circular and green economy.

5.2 Dataset

The dataset builds on MET Surveys conducted in 2021 and 2023 at the firm level. These surveys offer insights into firm characteristics, measurements, and their involvement in global trade. Additionally, they provide information on the proportion of investment allocated to both the green economy and the circular economy, contributing to the understanding of their relationship.

I. GREEN ECONOMY

43. In quali aree "green" ha investito o intende investire nel prossimo biennio:

	Investimento già effettuato	Programmato per il prossimo biennio	Né effettuato né programmato
Risparmio ed efficienza energetica	☐	☐	☐
Sistemi a valle del ciclo produttivo per riduzione dell'inquinamento	☐	☐	☐
Economia circolare (recupero e riutilizzo/riciclo e nuovi materiali)	☐	☐	☐
Durabilità e riparabilità (cambiamento dei prodotti e allungamento del loro ciclo di vita)	☐	☐	☐
Riduzione/Ottimizzazione dell'uso di materie prime non energetiche e semilavorati nella produzione (anche dematerializzazione prodotto finito)	☐	☐	☐
Cambiamento delle componenti per ridurre l'impatto ambientale in smaltimento	☐	☐	☐
Organizzazione della logistica per ridurre trasporti merci e persone	☐	☐	☐
Cambiamento significativo nei prodotti/servizi realizzati per seguire orientamenti di sostenibilità ambientale	☐	☐	☐

Figure 1. Circular and Green Economy Survey Questions

Notes: The columns indicate whether the firms have already made investments, are planning to invest, or have not invested.

These surveys are designed considering various firm characteristics, including substantial sample size, with a notable emphasis on factors such as regionality, sector, and dimensionality. Deliberate unbalancing of the sample ensures a more representative inclusion of companies more likely to undertake noteworthy and uncommon investments. Specific survey questions explore the reasons behind firm status transformations, particularly decisions related to green economy investment and their engagement in trade.

In Figure 1, the survey question illustrating participation in the circular and green economy is presented. The columns in the figure depict whether firms have already made investments, are planning to invest, or have not invested. The survey specifically inquires about various aspects, including efficient energy investment, circular economy investment, investments in durability and repairability, optimization of non-energy use, organization of logistics to reduce transportation, and significant changes in products/services to align with environmental sustainability deadlines.

(Nota per Format: risponde a questa domanda solo chi ha selezionato almeno un "investimento già effettuato", ovvero una risposta alla colonna 1 nella domanda 43)

45. Può indicare (anche orientativamente) a quanto ammonta l'investimento "green" effettuato, in percentuale del fatturato, nell'ultimo triennio 2019-21?

________ %

Figure 2. Green Investment as Percentage of Turnover

Moreover, the indicators for the circular economy extend beyond binary variables. Figure 2 displays the survey question inquiring about the percentage of green investment in terms of turnover over the last three years.

5.3 Circular Economy Indicators

Leveraging this dataset, the project generates various indicators at the region level to distinguish their connection to the circular economy. There are four primary indicators to assess the region's association with the circular economy. The four indicators encompass the circular investment indicator, the percentage of green investment indicator, the planning to invest in the circular economy indicator, and the broad circular economy investment indicator.

5.3.1 3.1 Circular Economy Investment

This indicator is based on a binary variable, taking a value of one if the firm is already invested in the circular economy; otherwise, it takes the value of zero. The construction of this variable is informed by the question in Figure 1 of the MET Surveys 2021 and 2023. Descriptive statistics for this variable are presented in Table 1.

	Observations	Mean	Std. Dev	Min	Max
CircularEconomyInvestment	24,433	10.43	30.56	0	100
By year, 2021	53,192	9.42	29.21	0	100
By year, 2023	28,759	11.29	31.64	0	100

Table 1: Descriptive Statistics for the Circular Economy Indicator

The circular economy proxy is depicted in the dataset, covering 53,192 anonymous firms with an average of 10.43%. In summary, the microdata evidence indicates that only 5,548 firms out of the total invest in the circular economy.

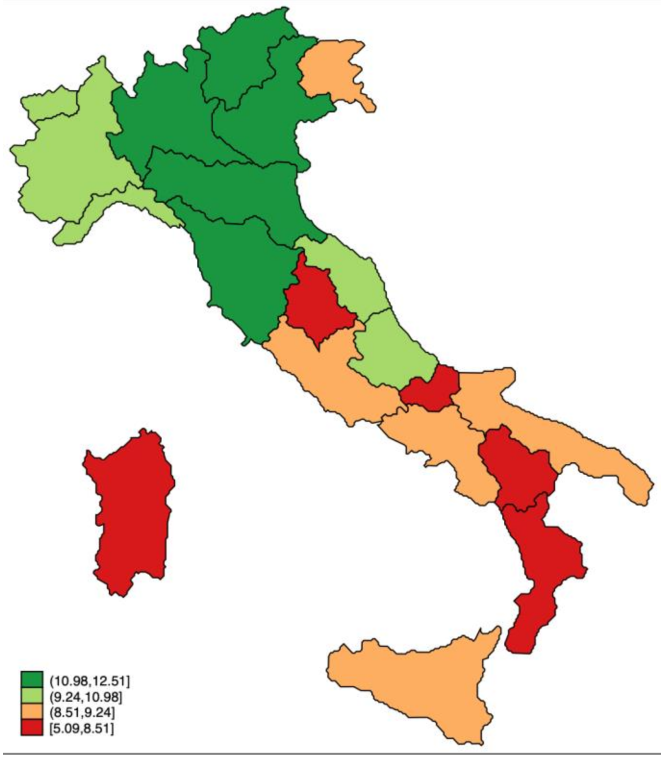


Figure 3: Circular Investment

Notes: This map displays distinct colors representing four quantiles of the circular economy investment measure at the regional level.

5.3.1.1 Regional Level

In this section, the micro-level measures of circular economy investment are aggregated at the regional level by utilizing the regions associated with the firms in the MET surveys. For each region, the average circular economy investment is calculated with equal weights assigned to the firms. The ensuing figure illustrates the circular economy investment at the regional level. The color spectrum in each region transitions from red to green, and these colors are further divided into four percentiles with equal weighting. Figure 3 represents the circular investment indicator for regions, where the color of each region is distributed across different percentiles. This map distinctly highlights variations in circular economy investment between North and South regions.

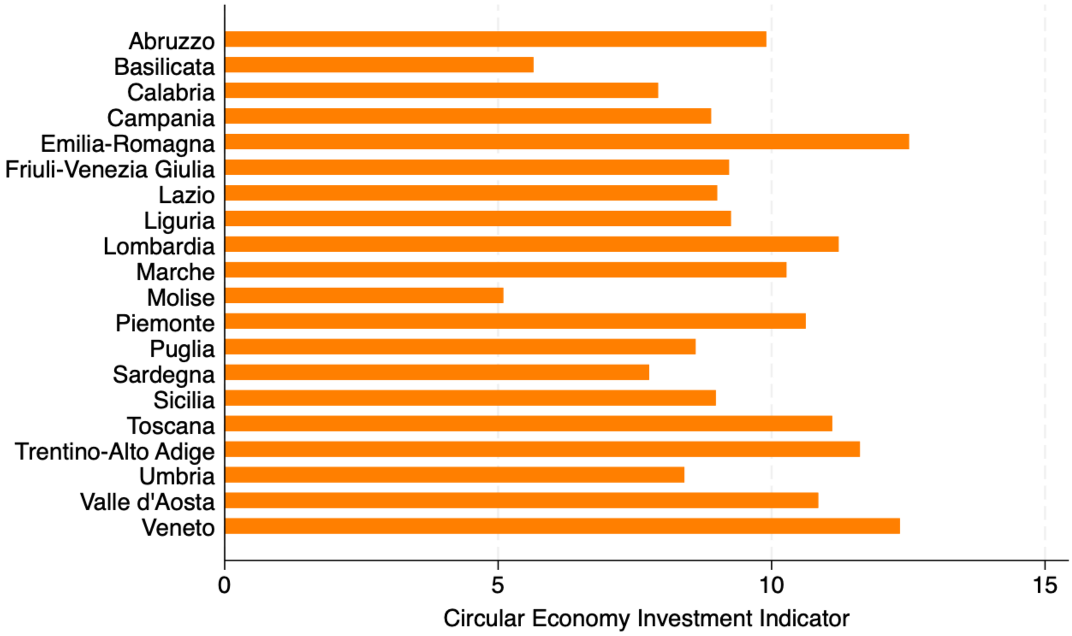


Figure 4: Circular Investment Indicator

Figure 4 illustrates the average circular economy investment for different regions. As depicted in the figure, the Emilia-Romagna region exhibits the highest circular economy investment, with 12.51% of firms engaged in circular production. Conversely, the Molise region has the lowest circular economy investment, with only 5.09% of firms participating. Additionally, there is considerable heterogeneity observed across regions.

5.3.2 Percentage of Green Economy Investment

This indicator is built on the question in Figure 2 and provides a direct measure of firms' investments in the green economy in terms of percentage of their total turnover. If the answer is not available, this part assumes zero investment. According to Table 2, firms invest 3.40% of their total turnover in the green economy.

	Observations	Mean	Std. Dev	Min	Max

PercentageofGreenInvestment	53,192	3.40	10.74	0	100
By year, 2021	24,433	2.91	9.55	0	100
By year, 2023	28,759	3.82	11.63	0	100

Table 2: Descriptive Statistics for the Percentage of Green Investment Indicator

5.3.2.1 Regional Level

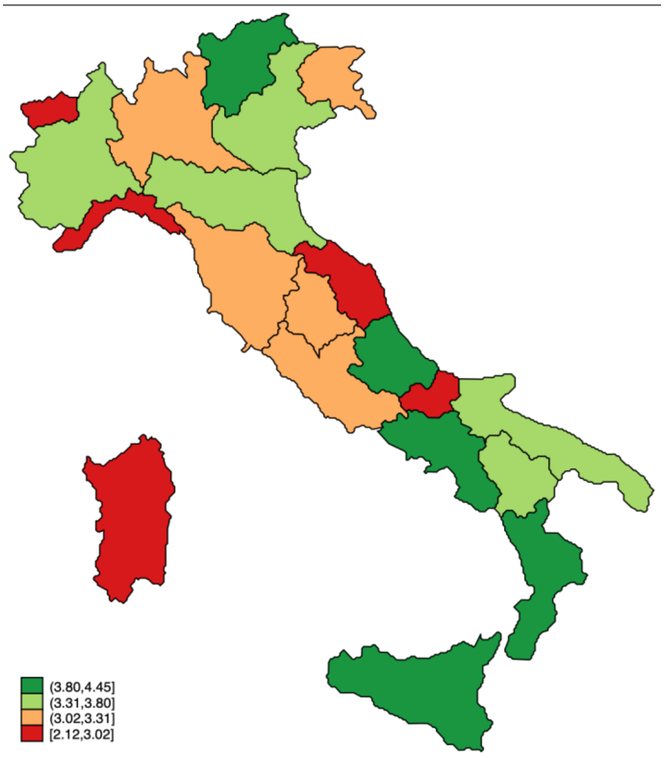


Figure 5: Green Investment (%)

Notes: This map displays distinct colors representing three quantiles of the percentage of green economy measured at the regional level.

In the second indicator, this study offers an estimate that is not reliant on dummy variables but is numerical. To accomplish this, the focus is on regions determined by the firms' average reported values of investment in the green economy relative to their turnover, expressed as percentages. Consequently, this section directly presents the percentage of investments for various regions.

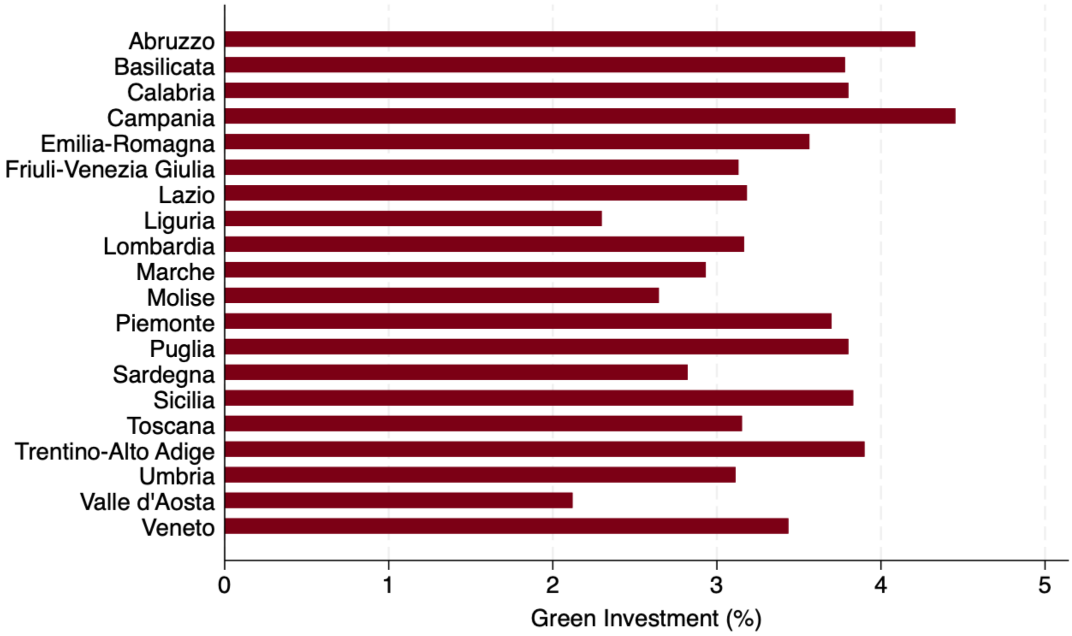


Figure 6: Green Investment (%)

Figures 5 and 6 showcase the investments made by different regions. The regions with the highest reported investments in the green economy include Trentino-Alto Adige, Abruzzo, and Campania. Conversely, Valle d'Aosta, Molise, and Liguria exhibit the lowest percentage of green economy investments relative to their turnover.

5.3.3 Planning to Invest in Circular Economy

Another aspect of the question presented in Figure 1 pertaining to the circular economy is the response indicating whether the firm plans to invest in the circular economy within the next two years. Therefore, it is crucial to distinguish between those already invested and those planning to invest to assess the future dynamics of firms in relation to the circular economy.

	Observations	**Mean**	**Std. Dev**	**Min**	**Max**
PlanningtoInvestinCircularEconomy	53,192	10.45	30.59	0	100
By year, 2021	24,433	10.74	30.96	0	100
By year, 2023	28,759	10.21	30.29	0	100

Table 3: Descriptive Statistics for the Planning to Invest in Circular Economy Indicator

The noteworthy discovery from Table 3 is that the intention to invest in the circular economy is relatively higher than the number of firms that have already invested. Interestingly, firms planning future investments in the circular economy are not much different from those that have already made investments. A total of 5,562 firms have reported their intention to invest in the circular economy in the future.

5.3.3.1 Regional Level

It is equally important to illustrate the planning-to-invest indicator at the regional level. The results presented in Figure 4 highlight the variation across regions in terms of those planning to invest compared to those who have already invested.

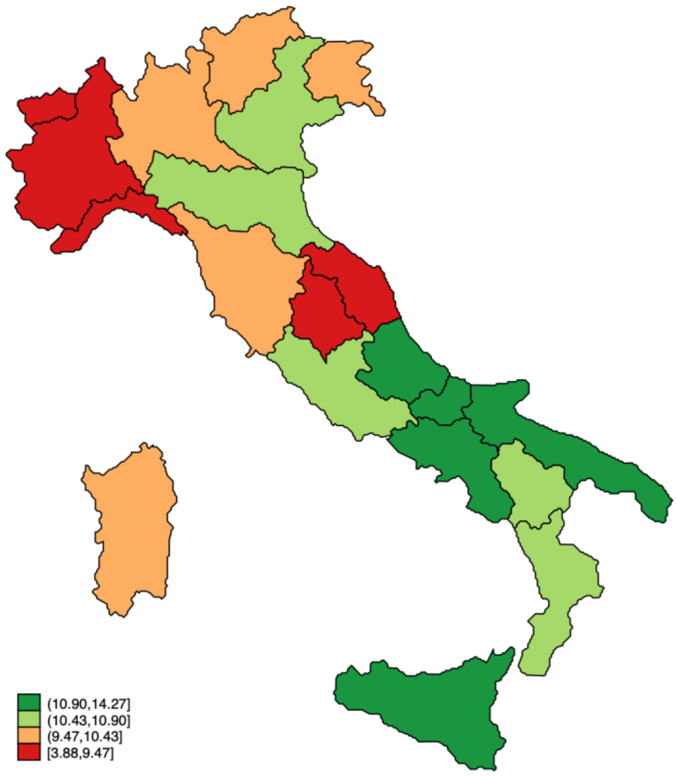


Figure 5: Planning to Invest in Circular Economy

Notes: This map displays distinct colors representing four quantiles of the circular economy investment measure at the regional level.

The indicator measuring the willingness to invest in the circular economy reveals notable regional variations. Notably, Southern regions such as Campania and Puglia show the highest inclination to invest, marking a significant contrast. Conversely, the trend of lower investments in the circular economy persists in regions like Valle d'Aosta.

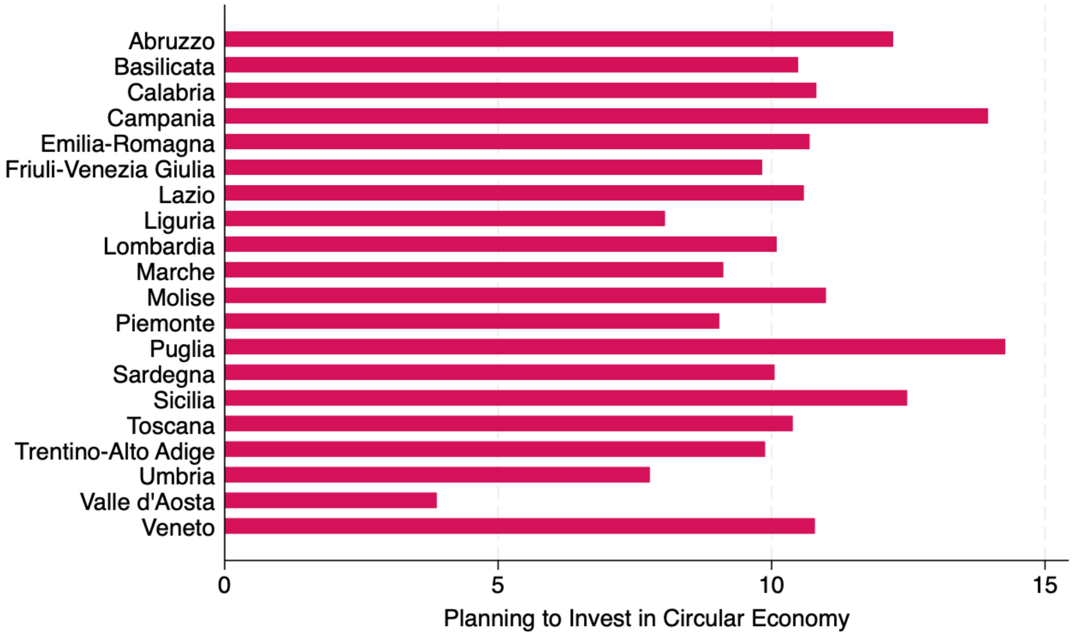


Figure 8: Planning to Invest in Circular Economy

Based on these findings, Figure 8 has been created for a detailed exploration of regional variations in future investments. On the y-axis of Figure 8, we observe the percentage of current investments, while the x-axis represents the planned future investments. While most regions closely align with the forty-five-degree line, there are notable outliers. For instance, Valle d'Aosta has invested more than the average among Italian regions but does not have significant plans for future investments compared to others. Conversely, Molise and Basilicata, despite having weaker past investments in the circular economy, are planning to invest more in the future compared to their counterparts.

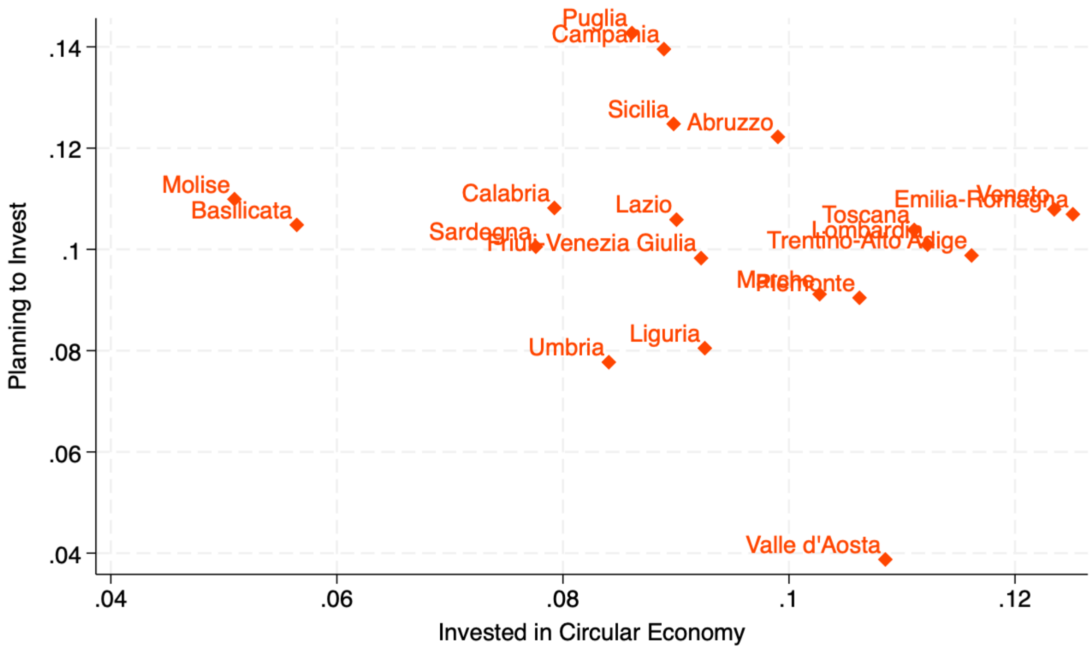


Figure 7: Planning to Invest vs. Invested

Moreover, regions positioned above the forty-five-degree line indicate higher planned investments compared to their current levels in the circular economy. The majority of these regions are located in the Southern part of the country.

5.3.4 Broad Definition of Circular Economy

A comprehensive definition of the circular economy can also encompass a broad perspective, and it is essential to formulate such a definition by referring to question 43 in Figure 1. This expansive definition relies on the firm's responses regarding investments in the circular economy, considerations for durability and repairability, optimization of non-energy raw material use, and the significant change in products or services to follow environmental sustainability guidelines.

	Observations	Mean	Std. Dev	Min	Max
BroadDefinitionofCircularEconomy	53,192	23.03	42.10	0	100
By year, 2021	24,433	21.53	41.11	0	100
By year, 2023	28,759	24.31	42.89	0	100

Table 4: Descriptive Statistics for the Broad Definition of Circular Economy Indicator

The extensive definition of the circular economy results in the creation of Table 4, showcasing that the average investment by firms, based on this broad definition, is 23.03%, surpassing the narrower definition of the circular economy indicator. This table emphasizes that firms' engagement with a broader circular economy definition is notably higher compared to previous figures. In total, 12,261 firms are actively participating in the circular economy based on this inclusive definition.

5.3.4.1 Regional Level

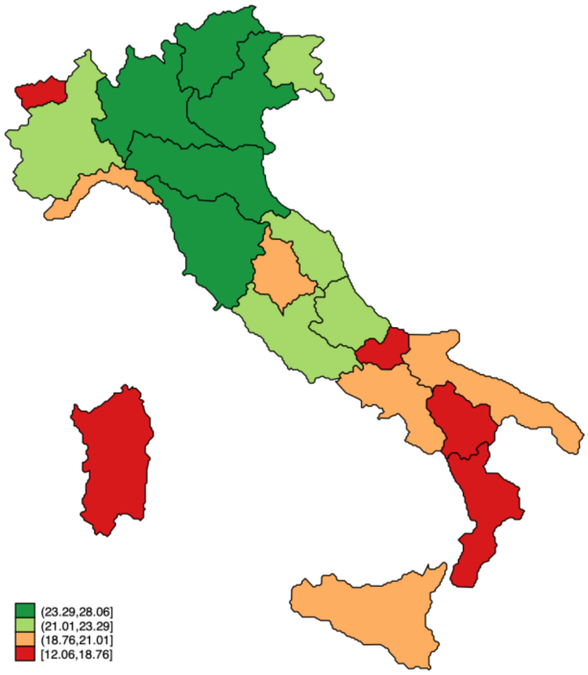


Figure 10: Broad Circular Economy Definition

Notes: This map displays distinct colors representing four quantiles of the circular economy investment measure at the regional level.

The regional distribution of the broad circular economy definition is illustrated in Figure 10. When compared to the narrow definition calculated at the regional level, the two indicators exhibit only minor differences.

However, with the exceptions of Valle d'Aosta and Sardegna, the alternative definition yields similar results. For the narrow definition discussed in the earlier section, there was a clear north and south heterogeneity based on the circular economy indicator. But in the broad definition, where durability and optimization is considered as a circular investment some regions become outlier. Notably, a comparison between Figure 3 and 10 highlights significant variations in regional-level investments in the circular economy, as exemplified by Valle d'Aosta. On the other hand, there is not any green or light-green region displayed even with the broad definition of the circular economy. These findings highlight the robustness of the circular economy indicator for different types of definitions and perceptions of the firms.

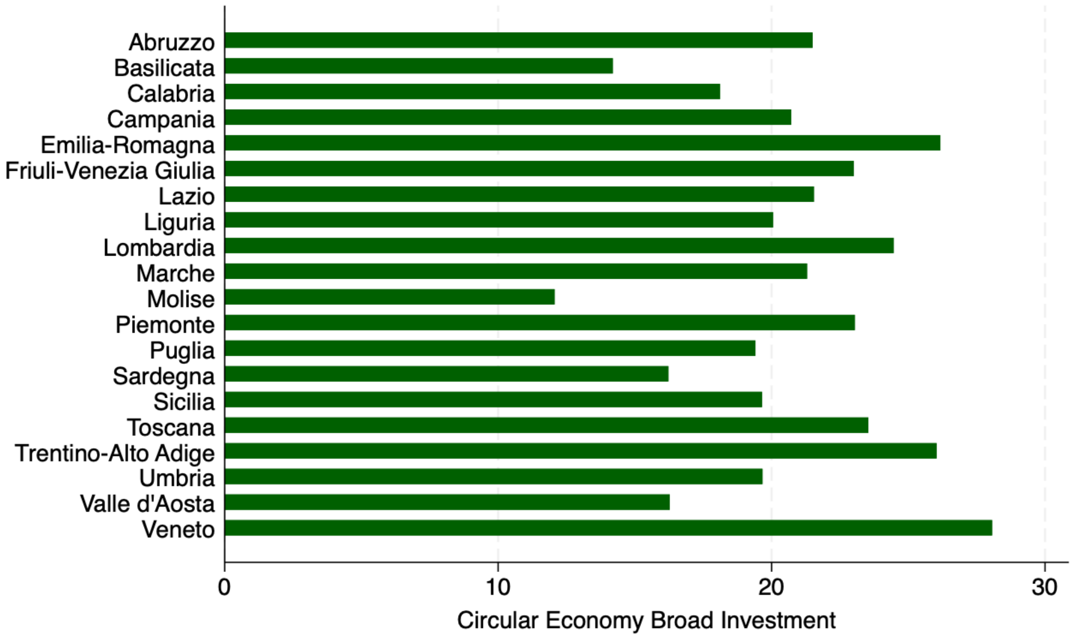


Figure 11: Broad Circular Economy Definition

5.4 Global Value Chain Indicators

In this section, we delve into the regional contributions to global trade by constructing various indicators that gauge the extent of trade. These indicators progress from a general overview to more specific aspects, building on micro-level evidence. The focus is on understanding the role of global value chains. The process of constructing these indices begins with defining indicators such as Multinational Enterprise indicator, Global Value Chain indicator, Arm-Length GVC indicator, Hierarchical indicator, Quasi-Hierarchical indicator, Modular indicator, and Relational indicator.

5.4.1 MNE Indicator

This section centers on firms identified as part of multinational enterprises (MNEs). Utilizing data from the MET Surveys allows for the identification of such firms. Consequently, a firm is categorized as part of an MNE if it belongs to a group, and notably, this group is foreign owned.

	Observations	**Mean**	**Std. Dev**	**Min**	**Max**
MNE	53,192	5.11	22.02	0	100
By year, 2021	24,433	5.02	21.84	0	100
By year, 2023	28,759	5.18	22.17	0	100

Table 5: Descriptive Statistics for the MNE Firms

As per the descriptive statistics provided by the firms, only 5.11% of them reported being part of foreign-owned groups. The subsequent section delves into the regional aspects of foreign ownership.

5.4.1.1 Regional Level

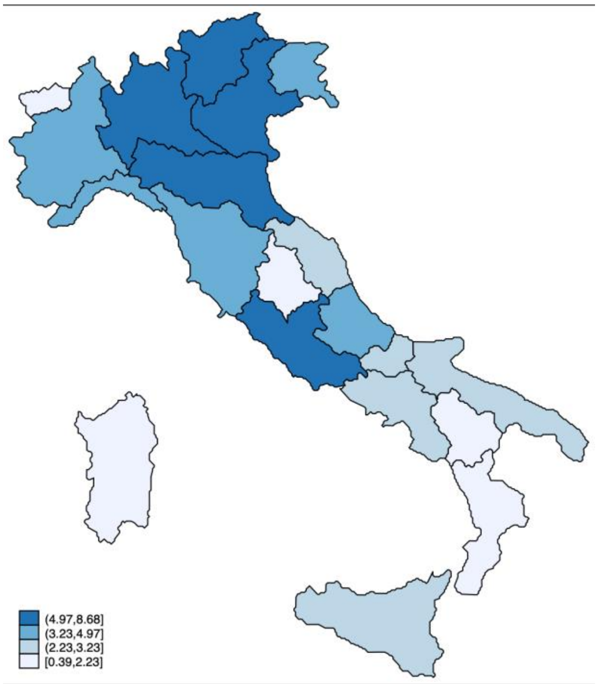


Figure 12: MNE Indicator across Regions

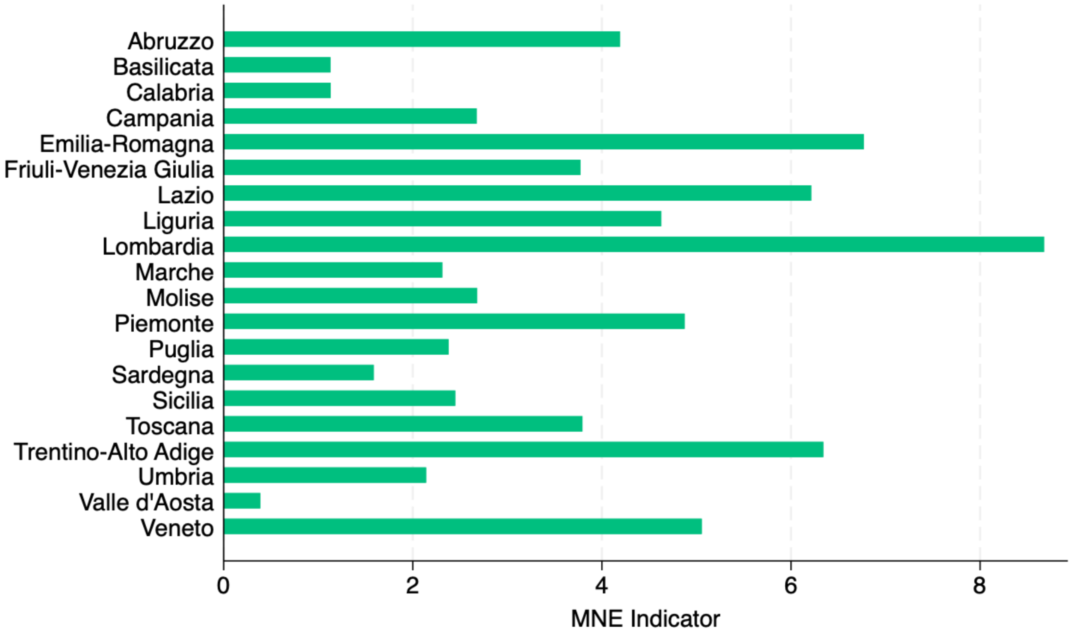


Figure 13: MNE Indicator

5.4.2 GVC Indicator

This section follows the alternative definition proposed by Brancati (2017) to compare the results discussed in the GVC section. There are several different aspects to this definition. This part identifies a firm as part of a global value chain if it is involved in both importing and exporting activities, exports intermediate goods, or has a long-lasting relationship with a foreign firm while simultaneously exporting or importing.

	Observations	**Mean**	**Std. Dev**	**Min**	**Max**
GVC	53,192	22.75	41.92	0	100
By year, 2021	24,433	23.50	42.40	0	100
By year, 2023	28,759	22.11	41.51	0	100

Table 6: Descriptive Statistics for Alternative GVC Definition

5.4.2.1 Regional Level

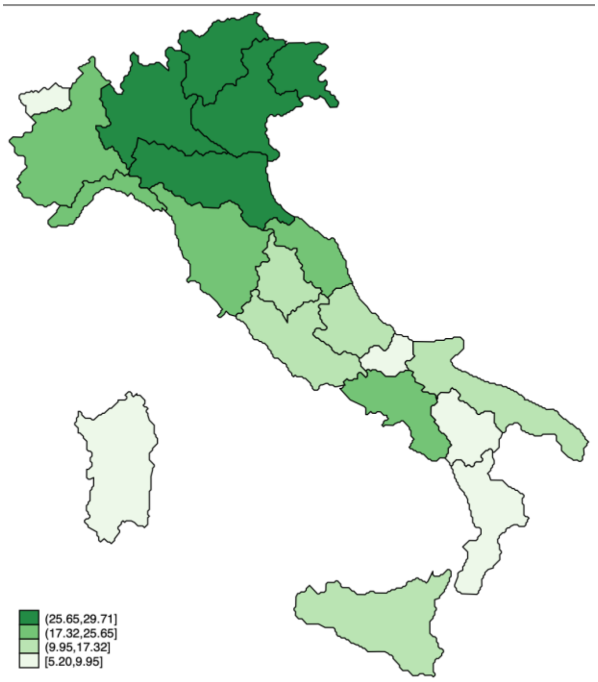


Figure 14: GVC Indicators by Brancati (2017)

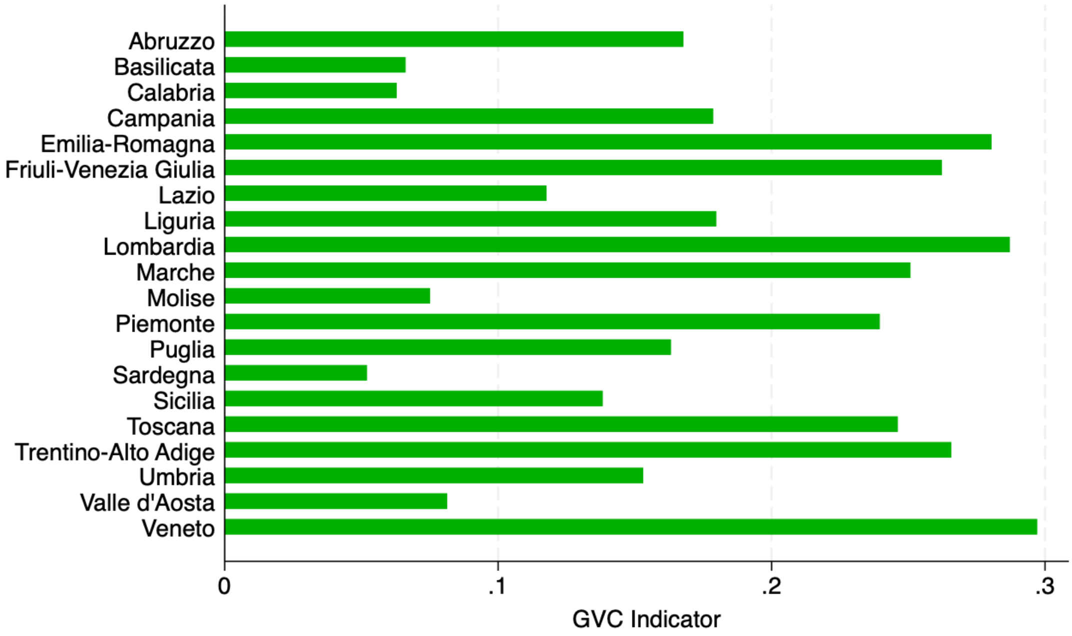


Figure 15: GVC Indicators by Region

5.4.3 Modes of Participation in GVCs

Following Brancati (2017), this part focuses on four detailed participation measures for GVCs. Based on the increasing complexity of the relationship, we divide these GVC firms into four categories: arm-length, quasi-hierarchy, hierarchy, and relational scopes.

The first form is arm-length, which represents the most basic type of Global Value Chain (GVC) participation. In this form, standard production is involved, and the production process is not contingent on specific buyer requirements that necessitate deep linkages. Arm-length is defined as if the firm is regarded as a global value chain, further emphasizing its enduring relationships driven by competitive pricing. This study defines the most basic scope as arm-length. A firm is classified as arm-length if it is part of a global value chain (GVC) but does not have a long-lasting relationship with a foreign firm.

The second indicator is the quasi-hierarchical GVC, wherein firms exert greater control over production decisions compared to others in the chain. For this reason, firms are also required to have a long-lasting trade relationship with another foreign firm and the underlying relationship must be based on trade.

	Type of GVC	Underlying Relationship with Foreign Firm
1	*Arm length*	Competitive Pricing
2	*Quasi-hierarchy*	Sales and Purchases
3	*Hierarchy*	Foreign Subsidiary
4	*Relational*	Research and Development

Table 7: Participation Modes to Global Value Chains

The third indicator for GVC participation is the hierarchical mode, where firms carry out in-house production and integrate into the value chain in a linear fashion. However, the firm has less control over the decision-making process. Hence, this form requires the firms to be part of a global value chain and also be subsidiaries of foreign groups.

The fourth and final type of GVC participation is the relational GVC, which involves the production of complex products and entails knowledge diffusion across counterparts within the value chain. This is the most complex form of GVCs, with these firms also involved in product development and possessing technological abilities to contribute to the conception and definition of the product.

	Observations	Mean	Std. Dev	Min	Max
GVCarmlength	53,192	7.18	25.81	0	100

GVCquasihierarchy	53,192	6.07	23.88	0	100
GVChierarchy	53,192	2.68	16.15	0	100
GVCrelational	53,192	5.96	23.68	0	100

Table 7: Descriptive Statistics for the Different Levels of GVC Participation

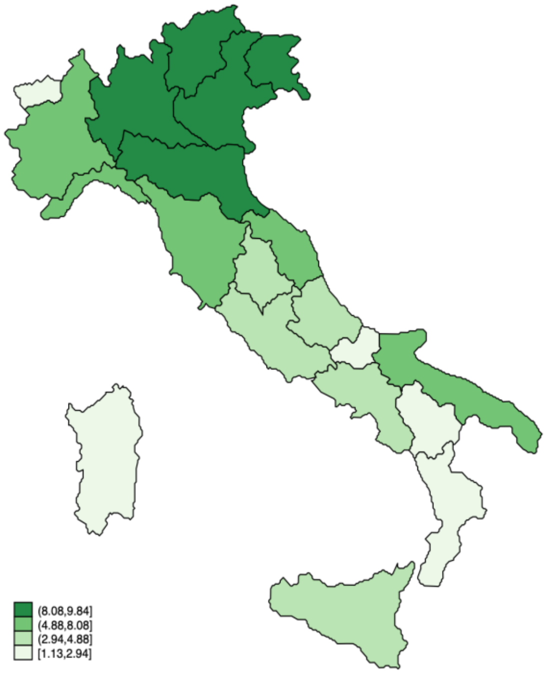


Figure 15(a): GVC Participation: Arm-Length

Notes: Each indicator is grounded in distinct modes of Global Value Chain (GVC) participation at the firm level.

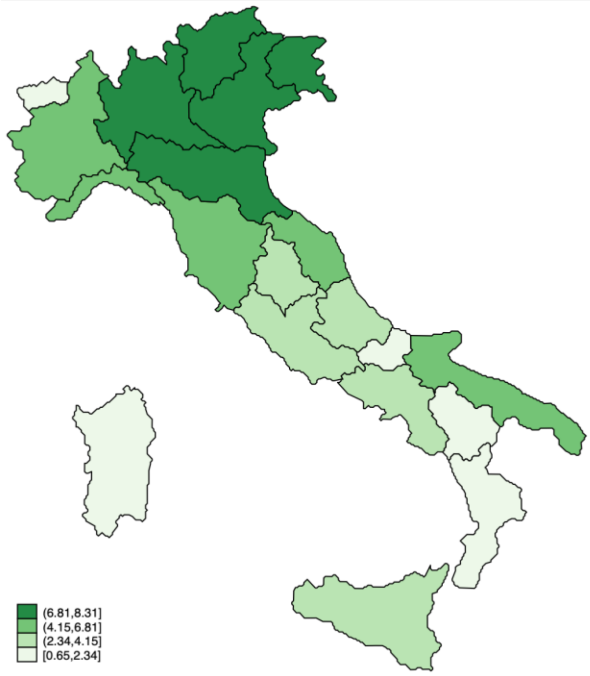


Figure 15(b): GVC Participation: Quasi-hierarchy

Notes: Each indicator is grounded in distinct modes of Global Value Chain (GVC) participation at the firm level.

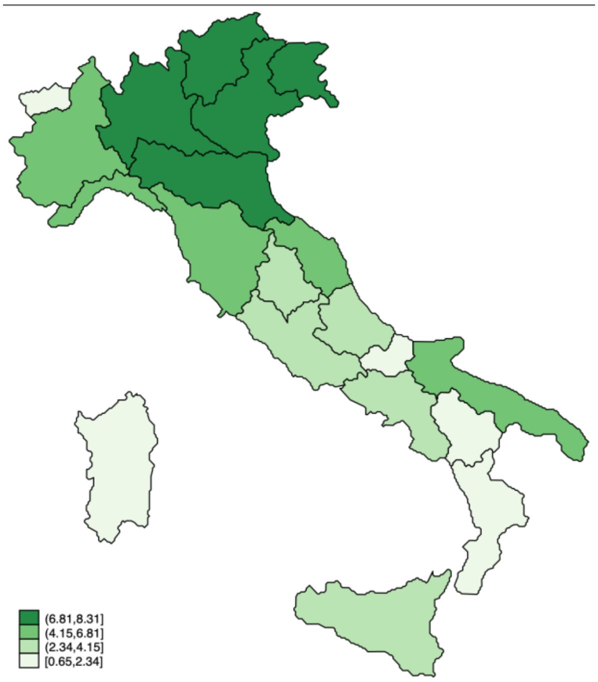


Figure 15(c): GVC Participation: Hierarchy

Notes: Each indicator is grounded in distinct modes of Global Value Chain (GVC) participation at the firm level.

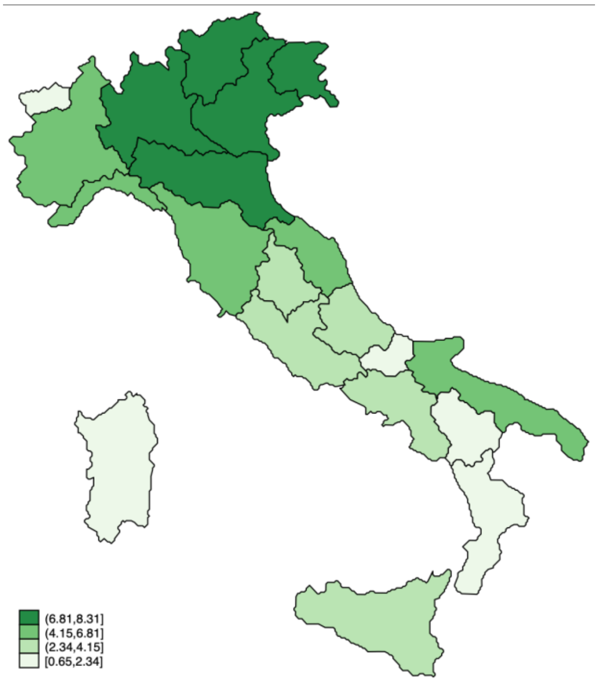


Figure 15(d): GVC Participation: Relational

Notes: Each indicator is grounded in distinct modes of Global Value Chain (GVC) participation at the firm level.

5.4.4 GVC Participation and Circular Economy

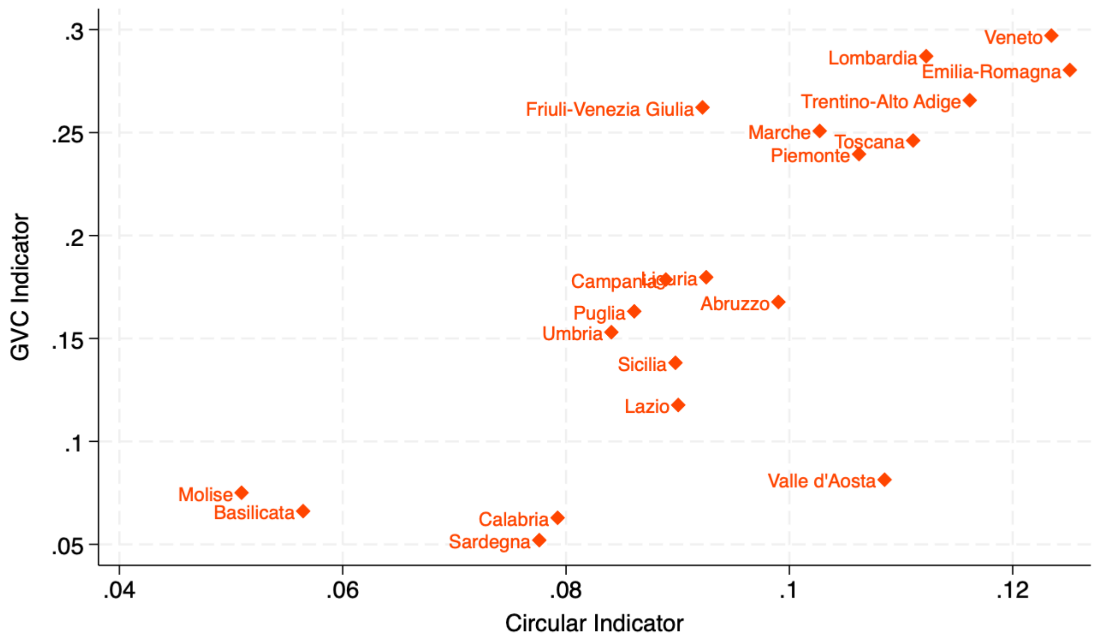


Figure 16: GVC and Circular Economy Indicators at the Regional Level

5.5 Conclusion

In a globalized world where production spans across borders, this project introduces a fresh approach to managing the circular economy within production processes. It begins at the micro level by offering circular economy measures for individual firms, and expands to the macro level by developing regional indicators for assessing the circular economy.

To ensure the reliability of our indicators, this report introduces four distinct measures. The first measure is a circular economy indicator derived from direct responses at the firm level regarding their participation in circular economy practices. The second measure assesses green economy investments, with firms reporting the percentage of their investments allocated to green initiatives. The third measure evaluates future investments in the circular economy. In Figure 7, the graph illustrates both current and projected investments across regions, showing a near-perfect alignment, except for three regions: Basilicata, Molise, and Valle d'Aosta. Lastly, the fourth measure expands upon the concept of the circular economy and evaluates the reliability of our indicators.

Even more intriguing, this indicator brings to light regional differences, as demonstrated in Figure 3. The figure distinctly delineates a North-South divide. Given this pronounced regional contrast, the report takes an additional step by constructing global value chain indicators. This decision is motivated by the extensive study and documentation of the North-South phenomenon in international trade literature.

To explore the disparity between North and South regions, this report establishes two indicators: one regarding participation in multinational groups and the other concerning involvement in global value chains (GVCs). Moreover, it breaks down the GVC indicators into four categories—arm-length, quasi-hierarchy, hierarchy, and relational—based on the scope of international trade. These GVC indicators, constructed using trade metrics for the same firms, exhibit consistent patterns across regions, a noteworthy discovery outlined in this report. Taking this analysis a step further, it also compares GVC and circular economy indicators in Figure 16. As depicted in the figure, these two indicators display a high correlation at the regional level.

5.6 Appendix

5.6.1 Alternative Indicators for Global Value Chains

The firm's participation in Global Value Chains (GVC) is broadly defined in this section. It hinges on a general definition of a GVC firm, whereby being part of a Multinational Enterprise (MNE) or engaging in export or import activities categorizes a firm as part of a value chain.

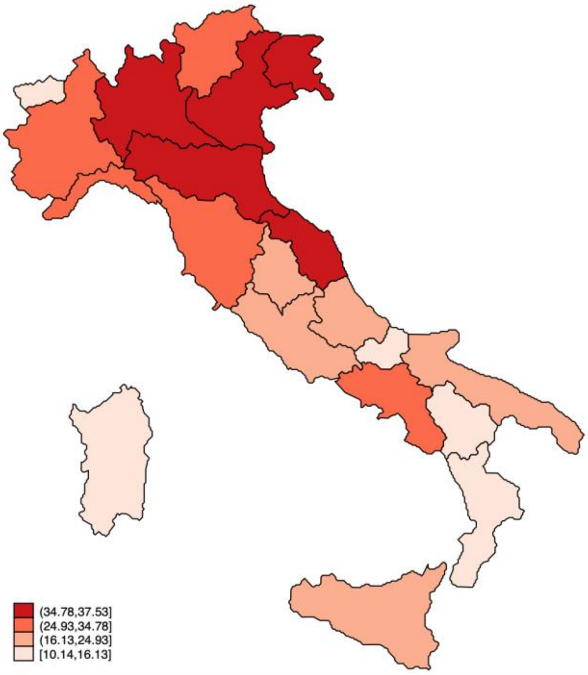


Figure A.1. Alternative Definition of GVCs

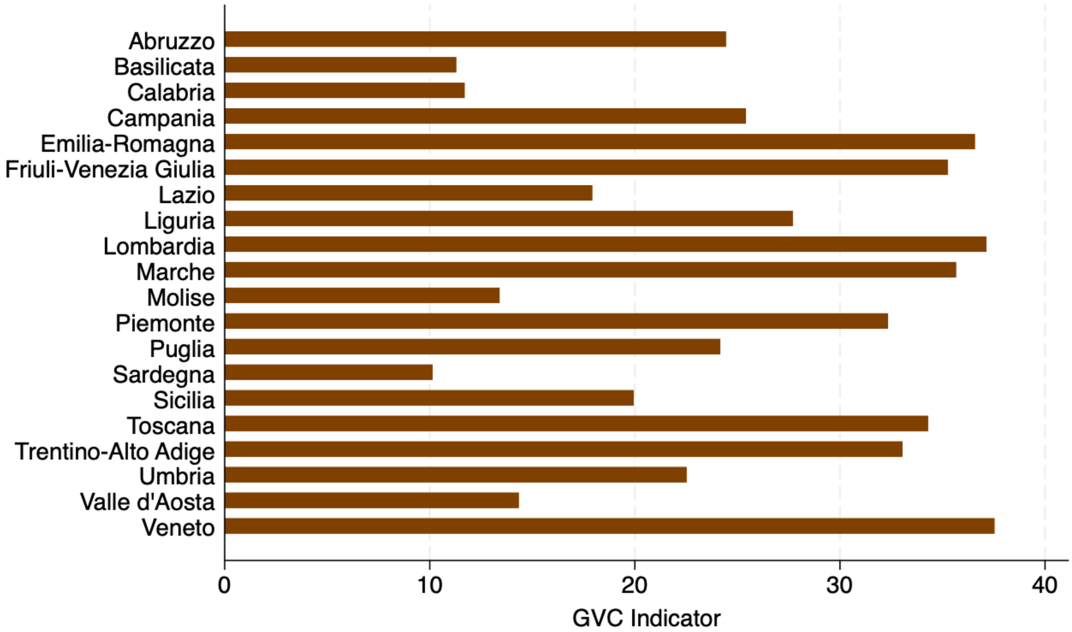


Figure A.2. Alternative Definition of GVCs and Regional Distribution

5.6.2 Indicators over Time

Figure B.1. Circular Investment Indicator by Years

Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

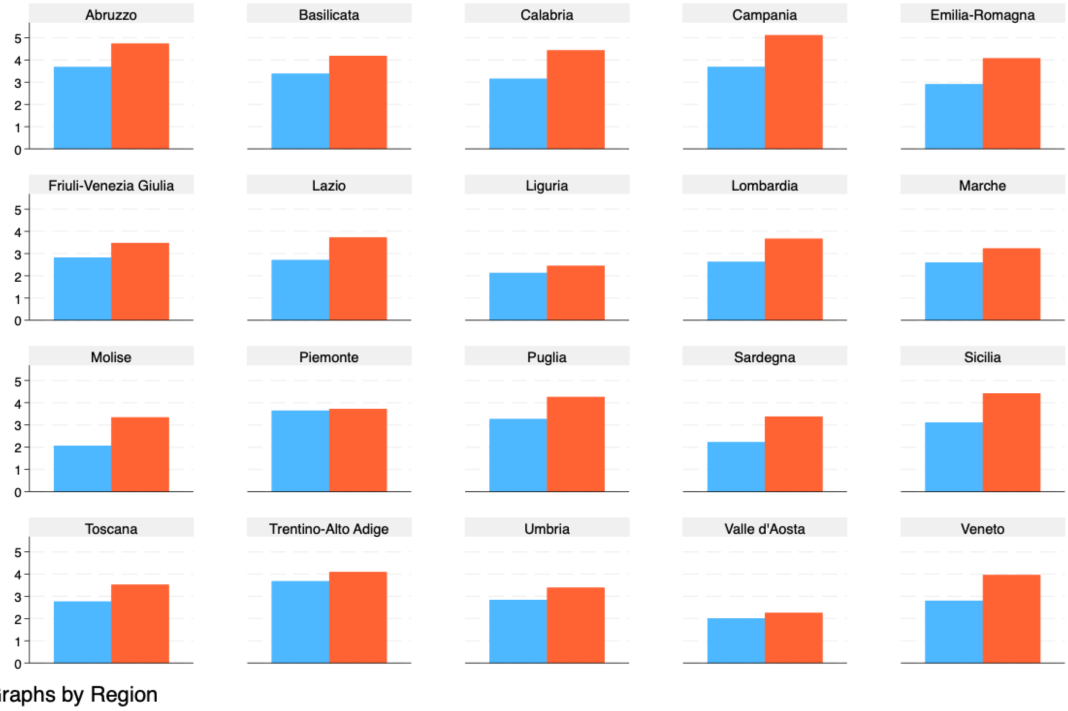


Figure B.2. Green Investment Indicator by Years

Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

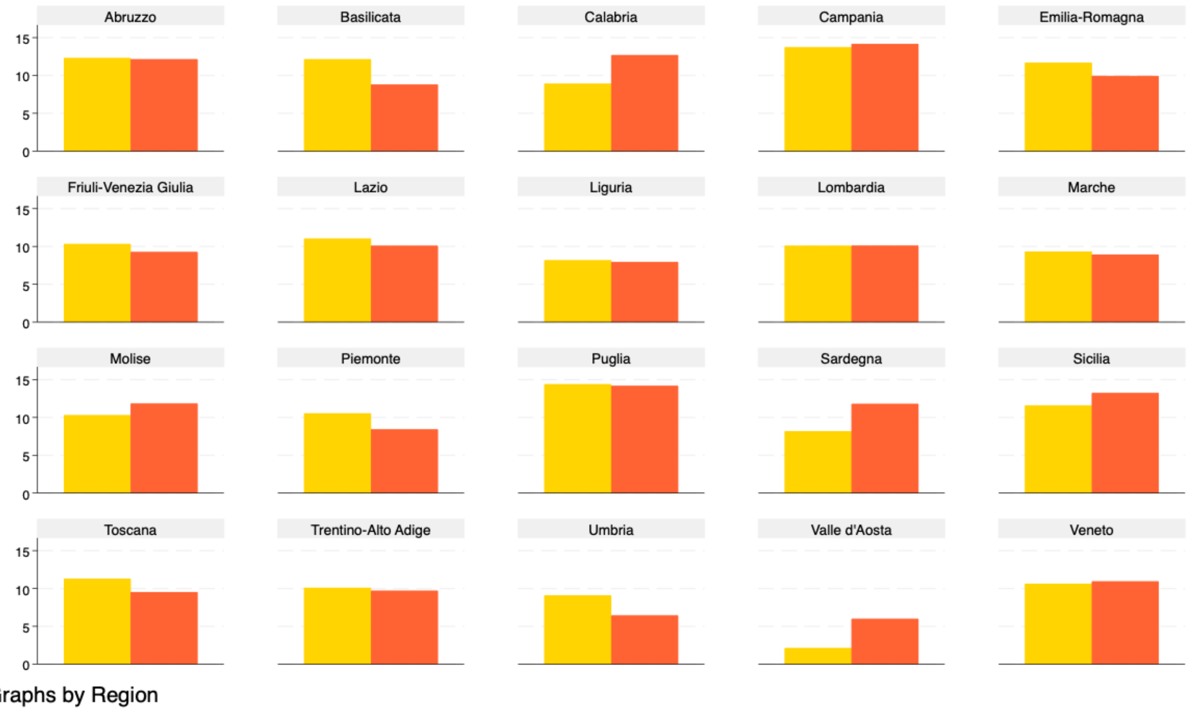


Figure B.3. Willingness to Invest in Circular Economy Indicator by Years

Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

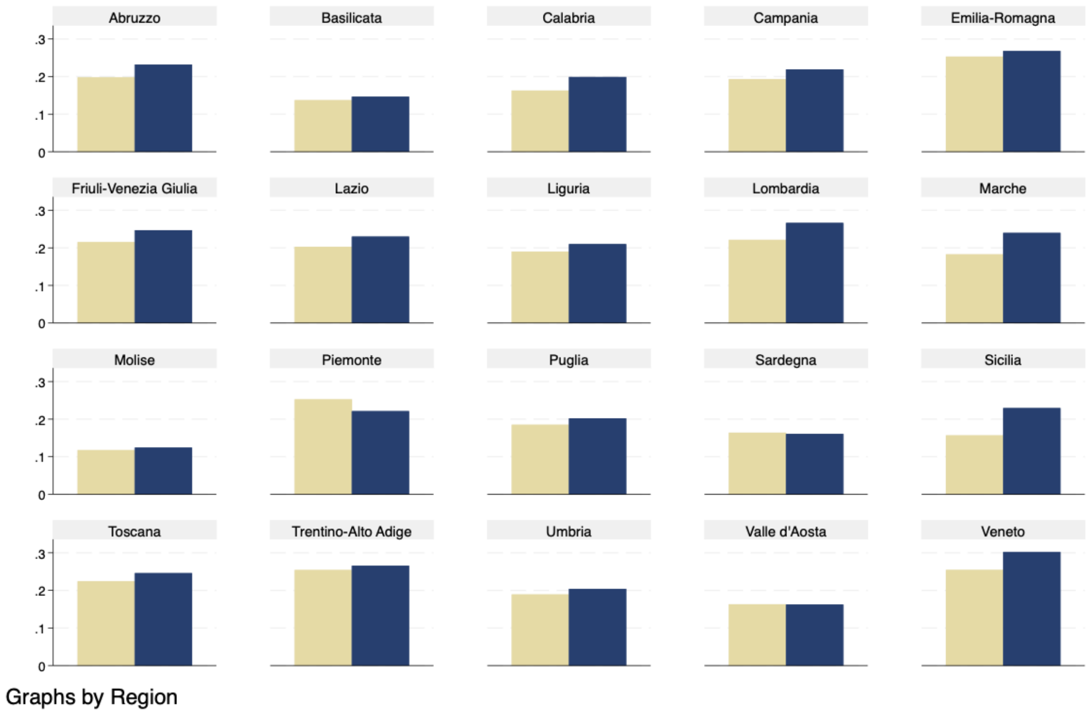


Figure B.4. Broad Circular Economy Indicator

Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

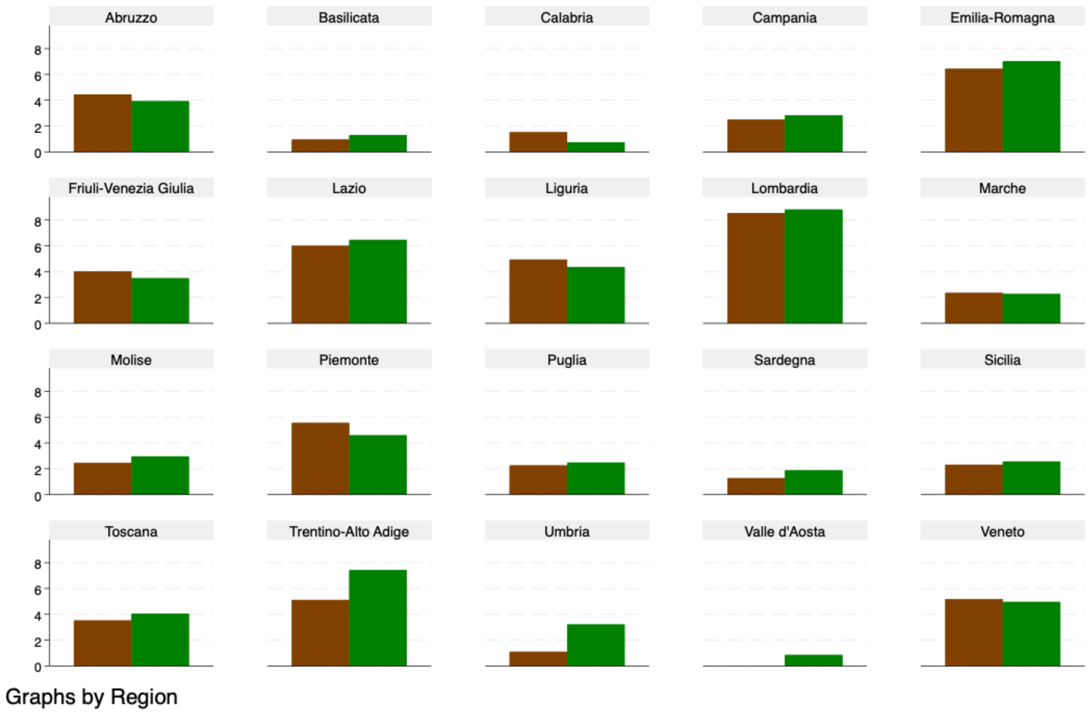


Figure B.5. MNE Indicator

Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

Figure B.6. GVC Indicator

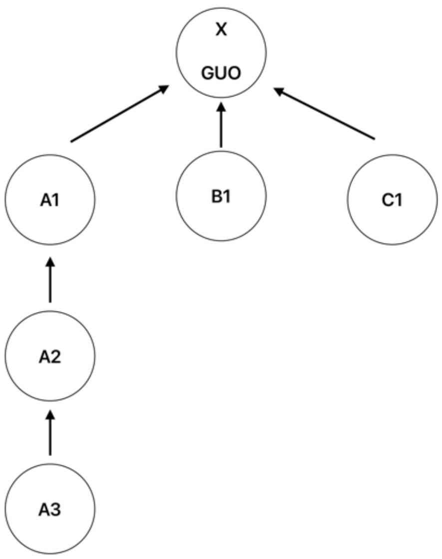
Notes: Bar graphs on the left present the region indicator average in 2021, whereas bar graphs on the right exhibit the region indicator average in 2023.

6. EXTERNAL EXPOSURE THROUGH OWNERSHIP LINKAGES

6.1 Introduction

The External Exposure Indicators through Ownership reflect the sensitivity of Italian regions at the Local Labor Systems (LLS) level to foreign shocks and policies related to circular economy due ownership linkages. Such shocks in foreign countries can heterogeneously affect different sector in Italian regions depending on the presence of Business Groups (BG), parents or affiliates, in these areas.

Consider a German BG "X" with one affiliate located in region r in Italy "A3" and 4 affiliates in foreign countries. The diagram below presents the hierarchy of BG "X".



Affiliate "A3" in Milan is directly owned by affiliate "A2" located in France, indirectly owned by "A1" and "X" located in Germany. Finally, "A3" shares a common GUO as "B1" and "C1" located in Japan.

A shock in any country of the above can affect "A3".

From the BG structure above, we can identify 4 types of ownership links (OLinks) for affiliate "A3" through majority ownership by the parent, direct and indirect majority ownership (other than the parent), and the existence of a common parent.

	Italian Affiliate	Foreign Affiliate
a) Ownership by the parent	A3	X
b) Direct majority ownership (other than parent)	A3	A2
c) Indirect majority ownership (other than parent)	A3	A1

d) Common parent (excluding the above)	A3	B, C1

If we were to imagine a BG where the Italian firm is on the top of the hierarchy as a parent and affiliates are foreign, the same types of links still hold in the opposite direction. We take all these ownership links into consideration, thus, constructing 4 variants of the EEI_O indicators, distinguishing between the two cases where: **(i)** the Italian firm is the parent, and **(ii)** the Italian firm is the affiliate. Direct ownership is defined as ownership of strictly more than 50% share of equity in the target, whereas indirect ownership as ownership of strictly more than 50% share of equity in the target through direct ownership of other firms in the hierarchy.

We construct a battery of 8 EEI_O indicators, illustrating the level of connection of an Italian region *r* with firms located in foreign country *c* through ownership linkages. We consider all sectors *s* in Italy, whether related or not to circular economy (CE) practices. However, we restrict ownership links to the sub-set of links with firms in foreign country *c* that are active in CE activities (depicted by their sector of activity). The idea is that exposure to a foreign policy change related to circular economy is disproportionally higher for Italian LLS that host a higher share of affiliates linked to the foreign country. We undertake a data cleaning procedure to match firms' location to their corresponding LLS based on the reported post codes provided in their addresses.

We construct a sector specific indicator on regional exposure to firms operating in a foreign country c with a CE activity. This would be the share of ownership links between Italian firms in region *r* operating in sector s and country *c* in total links in sector *s* with country *c*, computed as follows:

$$EEI\_O_{rs,c} = \frac{\sum_i OLinks_{irs,c}}{OLinks_{s,c}}$$

Where $\sum_i OLinks_{irs,c}$ is the sum of links for all firms *i* in sector *s* in region *r* and foreign country *c*. Accordingly, $OLinks_{s,c}$ is the total number of links in sector *s* from all over Italy with country *c*. We consider region *r* as the location of the parent if case **(i)** applies, or instead the location of the Italian affiliate in case **(ii)**, as previously detailed.

6.2 Data

We rely on Orbis data provided by Bureau Van Dijk subject to access agreements with Bocconi University, providing raw data on firms' ownership linkages. Namely, for each firm the database provides 2 ownership links with: (1) its direct majority share owner (ISH), and its Global Ultimate Owner (GUO). Using this information, we track the ownership chain of each BG affiliate and construct the BG hierarchical structure as previously presented in section 1.1.

This global database covers firm ownership links for the last available year.

6.3 External Exposure Indicator through Parent Ownership

In case **(i)**, $EEI\_O^{a}_{rs,c}$ reflects the share of ownership links by Italian parents in region *r* operating in sector *s* of their affiliates with circular activities in country *c*.

In case **(ii)**, $EEI\_O^{a}_{rs,c}$ reflects the share of ownership links of Italian affiliates in region *r* operating in sector *s* by their foreign parents with circular activities in country *c*.

$$EEI\_O^{a}_{rs,c} = \frac{\sum_i OLinks^{a}{}_{irs,c}}{OLinks^{a}{}_{s,c}}$$

6.4 External Exposure Indicator through Direct Ownership

We consider here the location of the Italian firm directly owning a foreign firm (in case i) or owned directly by a foreign firm (in case ii), as long as the owner of more that 50% equity is not the parent (GUO) of the BG.

Hence, in case **(i)** the indicator $EEI\_O^{b}_{rs,c}$ reflects the share of direct ownership links by non-GUO Italian affiliates in region *r* operating in sector *s* of affiliates with circular activities in country *c*.

In case **(ii)** instead**,** the indicator $EEI\_O^{b}_{rs,c}$ reflects the share of direct ownership links of Italian affiliates in region *r* operating in sector *s* by foreign firms with circular activities in country *c.*

$$EEI\_O^{b}_{rs,c} = \frac{\sum_i OLinks^{b}{}_{irs,c}}{OLinks^{b}{}_{s,c}}$$

6.5 External Exposure Indicator through Indirect Ownership

We consider here the location of the Italian firm owning indirectly a foreign firm (in case i) or owned indirectly by a foreign firm (in case ii), as long as the indirect owner of more that 50% equity is not the parent (GUO) of the BG.

Hence, in case **(i)** the indicator $EEI\_O^{b}_{rs,c}$ reflects the share of indirect ownership links by non-GUO Italian affiliates in region *r* operating in sector *s* of affiliates with circular activities in country *c.*

In case **(ii)** instead**,** the indicator $EEI\_O^{b}_{rs,c}$ reflects the share of indirect ownership links of Italian affiliates in region *r* operating in sector *s* by foreign firms with circular activities in country *c.*

$$EEI\_O^{c}_{rs,c} = \frac{\sum_i OLinks^{c}{}_{irs,c}}{OLinks^{c}{}_{s,c}}$$

6.6 External Exposure Indicator through Common Parent

We consider here the location of Italian affiliates with a common Parent with a foreign affiliate. Hence, the indicator $EEI\_O^{d}_{rs,c} =$ reflects the share of affiliates operating in sector *s* in region *r* linked to an affiliate operating in a circular economy sector in country *c,* through a common parent:

$$EEI\_O^{d}_{rs,c} = \frac{\sum_i OLinks^{d}{}_{irs,c}}{OLinks^{d}{}_{s,c}}$$

For this indicator, we don't distinguish between case (i) and (ii), since only case (ii) where we observe an Italian affiliate is relevant, by definition.

7. EXTERNAL EXPOSURE THROUGH GVC LINKAGES

7.1 Introduction

The EEI_GVC reflects the integration of Italian regions within the global value chains. It holds a significant potential for facilitating impact assessment studies of various shocks, including a circular economy change in policy. They can effectively serve as disruption risk indicators, capturing the vulnerability of Italian regions to external shocks that disrupt global value chains, thereby impeding the availability of production inputs and the access to global markets for output sales.

7.2 Data

We rely on data from WIOD (2016, last available year) for information on domestic and foreign value added at 2-digits level (ISIC rev. 4) and ISTAT for data on sectoral employment at the Local Labor System (LLS) level.

7.3 External Exposure Indicator through GVC

For a given area (LLS), EEI_GVC is the weighted sum of GVC exposure in circular economy related sectors, where the weight is given by the LLS's or Province's labor share. In particular, GVC is the average of two components. The first component is the share of the Extra-EU Domestic Value Added (DVA) content of Italy's total exports in each of the sectors to total DVA. The second component is the share of Extra-EU Import Value Added (VA) in each sector to total VA, i.e. the foreign content of exports.

Denoting by r the LLS or Province and by j the sector, EEI_GVC is thus given by:

$$\text{EEI\_GVC}_{\,r} = \sum_j \frac{L_{rj}}{L_r} * averageExp_j$$

Where L denotes employment and:

$$averageExp_j = mean(\frac{DVA_j^{extraEU}}{DVA_j}; \frac{VA_{IMP}{}_j^{extraEU}}{VA_{IMP}{}_j})$$

This indicator can be further decomposed by industry-region.

8. GVCs indicators for circular economy at the regional level

8.1 Introduction

Despite the lack of uniform literature in the area, including the standards, definitions and classification of the Circular Economy (CE) (Barrie and Schröder, 2021), it can be seen that the concept and practice of CE has emerged to overcome the drawbacks of the linear economy, which is based on take-use-dispose (Sarkar et al., 2022). On the other hand, CE is based on reuse, reduce and recycle, which is similar to a closed supply chain (Sarkar et al., 2022; Kirchherr et al., 2023).

The European Commission, for example, made the circular economy a key pillar in its European Green Deal with the launch of the Circular Economy Action Plan (CEAP) (Barrie et al., 2022). Italy has been in top positions in terms of CE in Europe with actions also based on its National Recovery and Resilience Plan (NRRP), as part of the Next generation EU program, with the aim of advancing a green and sustainable transition (OECD, 2023).

In addition to the domestic plans, the international trade plays an important role in the success of such actions. The World Economic Forum (2014) argues that supply chains are the key unit of action with regard to CE implementation and success, and it will be the foundation for driving needed change. However, this new dynamic can affect the geography of trade, most probably concentrating power in large buyers and manufactures in the Global North (Hofstetter et al., 2021).

Regardless the importance of this growing phenomenon, few studies have addressed CE on a multi-regional scale (Wiebe et al., 2019; Donati et al., 2020). This report seeks to contribute to this debate by assessing CE in the Italian economy from a regional perspective. The aim is to assess how CE are inserted into Global Value Chains (GVCs) and the heterogeneity of this insertion at regional level in Italy.

Specifically, we measure how CE sectors create value internally and externally through Italian participation in GVCs in an analysis for its NUTS2. To do this, we adapted a methodology to obtain Domestic Value Added (DVA) and Foreign

Value Added (FVA) for CE sectors at Italian level following previous literature (UNCTAD, 2013; Koopman et al., 2014; IMF, 2017) based on global input-output matrices. We also regionalized the DVA, FVA and exports according to the Italian shares of employment at NUTS2 level in each sector and, from these regional measures, we constructed indicators of Italian regional participation in GVCs in terms of CE, considering forward, backward and overall participation.

This report is divided into five parts: in addition to this introduction, section 2 describes the methodology through the input-output matrices; section 3 describes the data used as inpouts. Sections 4, 5 and 6 present, respectively, how we obtain DVA, FVA and exports for the CE sectors at the Italian regional level. Sections 7, 8 and 9 show how we obtain the regional measures of the indicators of participation of CE sectors in GVCs: forward, backward and overall participation, at the Italian regional level. Finally, section 10 presents the R commands.

8.2 Methodology for constructing indicators

To calculate the indicators of participation in GVCs at regional level for Italy, we used a global OECD input-output matrix with a base year of 2018. The choice of OECD matrices stems from the fact that they are widely available over time compared to other matrices in the literature and from the possibility of comparing our indicators with those obtained by the OECD in the Trade in Value-Added (Tiva) database. Although there are more recent years than 2018 in this database, we have carefully followed the accuracy of this year's data in line with the national input-output matrices obtained by the OECD, with the most recent years also estimated based on 2018.

Figure 1 shows a representation of a global input-output matrix, considering m sectors and n countries. Sector x in country s produces a good i that can be used as an intermediate production factor in the production of another good or serve as final demand, which can occur in country s or outside this country, such as in country r. Mathematically, it can be expressed by:

$$x_i^s = \sum_{s=1}^{n} \sum_{j=1}^{m} z_{ij}^{sr} + f_i^s \quad (1)$$

Where x_i^s is the total production, z_{ij}^{sr} represents the interindustry sales by sector i in region s to sector j in region r, and f is the final demand.

Figure 1 – Representation of the OECD interregional input-output matrix

<table>
<tr><th colspan="3" rowspan="3">Output
Input</th><th colspan="7">Intermediate consumption</th><th colspan="13">Final demand</th><th rowspan="3">Total demand</th></tr>
<tr><th colspan="3">Country 1</th><th rowspan="2"></th><th colspan="3">Country n</th><th colspan="7">Country 1</th><th colspan="6">Country n</th></tr>
<tr><th>Activity 1</th><th>...</th><th>Activity m</th><th>Activity 1</th><th>....</th><th>Activity m</th><th>HFCE 1</th><th>NPISH 1</th><th>GGFC 1</th><th>GFCF 1</th><th>INVNT 1</th><th>DPABR 1</th><th>...</th><th>HFCE n</th><th>NPISH n</th><th>GGFC n</th><th>GFCF n</th><th>INVNT n</th><th>DPABR n</th></tr>
<tr><td rowspan="7">Intermediate consumption</td><td rowspan="3">Country 1</td><td>Activity 1</td><td colspan="7" rowspan="7">z_{ij}^{sr}</td><td colspan="13" rowspan="7">f_i^{sr}</td><td rowspan="7">x_i^s</td></tr>
<tr><td>⋮</td></tr>
<tr><td>Activity m</td></tr>
<tr><td colspan="2">⋮</td></tr>
<tr><td rowspan="3">Country n</td><td>Activity 1</td></tr>
<tr><td>⋮</td></tr>
<tr><td>Activity m</td></tr>
<tr><td colspan="3">Value added</td><td colspan="7">w_j^r</td></tr>
<tr><td colspan="3">Total output</td><td colspan="7">x_j^r</td></tr>
</table>

Source: elaborated by the authors. Notes: HFCE = Household Final Consumption; NPISH = Non-Profit Institutions Serving Households; GGFC = General Government Final Consumption; GFCF = Gross Fixed Capital Formation; INVNT = Changes in Inventories and Valuables; DPABR = Direct purchases abroad by residents

The columns of the matrix provide information on production technology, as they indicate the quantities of intermediate goods needed to produce the total product. This information is given in each column from the domestic and foreign shares of intermediates used in the production of one unit of product (UNCTAD, 2013). This technology is considered fixed, at least in the short run, and can be expressed by the following technical coefficient:

$$a_{ij}^{sr} = z_{ij}^{sr} / x_j^r \tag{2}$$

Isolating z_{ij}^{sr} in (2), replacing it in (1) and rewritten the system in matrix format, we have the following expression:

$$X = AX + Y \tag{3}$$

Isolating the total product X:

$$X = (I - A)^{-1} Y \tag{4}$$

In which $(I - A)^{-1}$ is the Leontief inverse, named L, being A the matrix of technical coefficients and I an identity matrix.

To calculate the value-added embodied in gross exports, in addition to the Leontief inverse, L, we need a vector of lines, v, in which each element represents the percentage of value-added per unit of product by country r and sector i:

$$v_i^r = w_i^r / x_i^r \tag{5}$$

An important observation concerns the value-added vector, which was crucial to capture the DVA and FVA in CE sectors in Italy. To do so, we calculate the value-added vector twice: in the first case, using only one measure of value-added for the CE sectors in Italy from CE shares, and in the second, considering all the sectors and countries obtained from the OECD input-output matrix. In the first case, the value-added is equal to zero for the sectors not related to CE in Italy and also for all sectors in the other countries, since we have no assumptions for CE in these countries.

Since the OECD input-output matrix are defined at NACE 2-digit level, we have to isolate the portion of value-added pertaining to CE sectors at this sectoral level.

To get an approximation of the CE shares in each sector in Italy, we divided the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, for the year 2018, based on IstatData database, following the sectoral composition of the CE and applied it to the value-added vector, W, in the first case.

We also have to provide a vector e which represents the exports from one country to the others, named interregional exports. Following Koopman et al. (2014), this vector can be obtained from equation (6):

$$E_{s*} = \sum_{r\neq s}^{G} E_{sr} = \sum_{r\neq s}^{G}(A_{sr}X_r + Y_{sr}) \tag{6}$$

By diagonalizing both vectors v and e, the value-added embodied in gross exports can be calculated from T, which was obtained twice considering the two different value-added vectors, separately:

$$T = \begin{pmatrix} T_{11}^{11} & \cdots & T_{1n}^{mm} \\ \vdots & \ddots & \vdots \\ T_{n1}^{1m} & \cdots & T_{nn}^{mm} \end{pmatrix} = \begin{pmatrix} v^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & v^m \end{pmatrix} \begin{pmatrix} L_{11}^{11} & \cdots & L_{1n}^{mm} \\ \vdots & \ddots & \vdots \\ L_{n1}^{1m} & \cdots & L_{nn}^{mm} \end{pmatrix} \begin{pmatrix} e^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^m \end{pmatrix} \tag{7}$$

Figure 2 shows a representation of an updated UNCTAD (2013) T-matrix for m sectors and n countries.

Figure 2 – T matrix for calculating DVA and FVA for CE in Italy

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="7">Intermediate consumption</td></tr>
<tr><td colspan="3">Country 1</td><td></td><td colspan="3">Country n</td></tr>
<tr><td>Sector 1</td><td></td><td>Sector m</td><td></td><td>Sector 1</td><td></td><td>Sector m</td></tr>
<tr><td rowspan="7">Intermediate consumption</td><td rowspan="3">Country 1</td><td>Sector 1</td><td rowspan="3">DVA in country 1 sector 1</td><td colspan="5" rowspan="7">… …. t_{ij}^{sr}</td><td rowspan="4">FVA in country 1 sector m</td></tr>
<tr><td>⋮</td></tr>
<tr><td>Sector m</td></tr>
<tr><td colspan="2">⋮</td><td rowspan="4">FVA in country 1 sector 1</td></tr>
<tr><td rowspan="3">Country n</td><td>Sector 1</td><td rowspan="3">DVA in country n sector m</td></tr>
<tr><td>⋮</td></tr>
<tr><td>Sector m</td></tr>
</table>

Source: elaborated by the authors.

From the T matrix, we can understand that DVA in gross exports represents the value created in a country by producing goods and services for export. Essentially, it shows how much of the value of the exported product comes from the exporting country itself, focusing on the production that takes place within the country's borders. On the other hand, the FVA in gross exports captures the value of imported intermediate goods and services that are incorporated into the exports of a national industry (OECD, 2021).

As we have no assumptions for the specialization in CE outside Italy in the first case, since the overall value-added vector is zero for the other countries, the FVA obtained for CE is also zero from the "T_CE" matrix in R.

To approximate the FVA of the CE in Italy, we adapted the FVA obtained by considering the vector of value-added for all countries and sectors inside and outside Italy from the T matrix for all sectors, called "T_all" matrix in R. We have applied to the FVA obtained by this matrix the employment shares related to CE in the 2-digit NACE – the same one used to construct the value-added vector for CE presented in the first case.

It should be noted that, when applying these shares, only the sectors directly related to the CE have values other than zero for FVA, while in the case of the DVA we have the direct and indirect value-added generated by the CE exports from Italy.

In this project, the DVA and FVA for CE and the vector of interregional exports, E, for Italy are regionalized for Italy's NUTS2 according to the percentage of employment in each NACE 2-digit sector over the country's total for the year 2018, based on IstatData database.

After regionalizing the DVA and FVA for CE, and interregional exports, E, multiplying their values at the Italian national level by the regional employment shares in each sector, we have the following regional indicators:

Indicator 1: Regional Domestic Value Added of CE sectors embedded in gross regional exports ($\boldsymbol{DVA_{region,CE}}$)

Indicator 2: Foreign Value Added of CE sectors embedded in gross regional exports ($\boldsymbol{FVA_{region,CE}}$)

Indicator 3: gross regional exports ($\boldsymbol{Exports_{region}}$)

We also constructed regional indicators of participation in GVCs, considering forward, backward and overall participation. From DVA, we can obtain a measure of forward participation in GVCs by dividing the DVA in each sector, by region (NUTS2), by the total interregional exports of that region. This measure represents a seller's perspective on participation in GVCs, where an economy produces intermediate goods in order to send them as exports to third countries. We followed the same procedure to obtain backward participation using FVA divided by total interregional exports in each region, which can be considered a buyer perspective in GVCs, where an economy imports intermediate goods to produce its exports. We can also measure the overall participation in GVCs by adding up DVA and FVA in each sector, by region, and then dividing by the total value of gross interregional exports in each region. These indicators are expressed mathematically by the following equations:

$$\textbf{Ind. 4}: \boldsymbol{Regional\ forward\ participation\ of\ CE\ sectors\ in\ GVCs} = DVA_{region,CE}\ /\ Exports_{region}$$

$$\textbf{Ind. 5}: \boldsymbol{Regional\ backward\ participation\ of\ CE\ sectors\ in\ GVCs} = FVA_{region,CE}\ /\ Exports_{region}$$

$$\textbf{Ind. 6}: \boldsymbol{Regional\ overall\ participation\ of\ CE\ sectors\ in\ GVCs} = \frac{(DVA_{region,CE} + FVA_{region,CE})}{Exports_{region}}$$

Where the region dimension represents each of the 21 Italian NUTS2, and CE the circular economy sectors.

8.3 Input data for the construction of the indicators

As discussed in section 1, indicators 1-6 were calculated for the year 2018. The calculation of the indicators requires some input data that, at the moment in which the present document is prepared, are not available for more recent years. To replicate the calculation for the next years, it is therefore necessary to download (when it will be available) the more recent version of the data described below. Notice that this part cannot be authomatized, as it requires a manual search on the websites of the statistical institutes providing the needed information and, once the data are downloaded, a minimum reshaping of the format of the data, so to obtain excel tables comparable to those provided together with this report. The input data are the following ones:

- Data on trade: OECD Inter-Country Input-Output (ICIO) tables for Italy, with 45 NACE Rev.2 sectors. These data can be downloaded here: https://www.oecd.org/sti/ind/inter-country-input-output-tables.htm . They are saved into .xls format in the sheet "IOM" in the file Database_CE.xls
- Definition of the CE sectors: the definition of the CE sectors is based on the results from the EU project whose results are reported here: https://www.prognos.com/en/project/economic-aspects-circular-economy
- Data on the specialization of the Italian national economy in the CE sectors: ISTAT provide data on the specialization of the country in the 4-digit NACE Rev.2 sectors that are listed among those involved in the CE (see point above). Data are available at dati.istat.it. Select the table "Occupazione regolare e irregolare per

branca di attività e popolazione” → “Occupati, unità di lavoro, posizioni lavorative e ore lavorate”. . They are saved into.xls format in the sheet “Labor_4.0” in the file Database_CE.xls.

- Data on the specialization of the NUTS2 Italian regions in the CE sectors: ISTAT provide data on the specialization of the NUTS2 Italian regions in the 2-digit NACE Rev.2 sectors. Data are available at dati.istat.it. Select the table “Occupazione regolare e irregolare per branca di attività e popolazione” → “Occupazione regionale” . They are saved into .xls format in the sheets “Industries and services” and “General employment” in the file Database_CE.xls

8.4 Indicator 1: guide for the calculation of Regional Domestic Value Added of CE sectors embedded in gross regional exports

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.
5) Diagonalize the value-added in CE sectors.
6) Subtract the intraregional flows of the total sales to obtain the interregional exports.
7) Diagonalize the interregional export vector.
8) Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.
9) Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional DVA of CE sectors embodied in exports by NUTS2 and sector.

8.5 Indicator 2: guide for the calculation of Regional Foreign Value Added of CE sectors embedded in gross regional exports

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by gross output, X, and diagonalize it.
5) Subtract the intraregional flows of the total sales to obtain the interregional exports.
6) Diagonalize the interregional export vector.
7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-added, for all countries and sectors, in order to obtain the T matrix.
8) Sum the columns relating to Italy's transaction with other countries to obtain the FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA), that is, the flows from Italy with iself, and subtract this value from the total of column.
9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional FVA of CE sectors embodied in exports by NUTS2 and sector.

8.6 Indicator 3: guide for the calculation of regional gross exports

1) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

2) Subtract the intraregional flows of the total sales to obtain the interregional exports and split it just for Italy.
3) Multiply the regional shares by the regional exports vector for Italy.

8.7 Indicator 4: guide for the calculation of the regional forward participation of CE sectors in GVCs

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.
5) Diagonalize the value-added in CE sectors.
6) Subtract the intraregional flows of the total sales to obtain the interregional exports.
7) Diagonalize the interregional export vector.
8) Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.
9) Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional DVA of CE sectors embodied in exports by NUTS2 and sector.
12) Split the interregional export vector in (7) only for Italy and multiply it by the regional shares obtained in (10) to obtain the regional exports.
13) Obtain a measure of forward participation in GVCs by dividing the regional DVA in each sector, by region (NUTS2), by the total regional exports in all sectors of that region.

8.8 Indicator 5: guide for the calculation of the regional backward participation of CE sectors in GVCs

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by gross output, X, and diagonalize it.
5) Subtract the intraregional flows of the total sales to obtain the interregional exports.
6) Diagonalize the interregional export vector.
7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-added, for all countries and sectors, in order to obtain the T matrix.
8) Sum the columns relating to Italy's transaction with other countries to obtain the FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA), that is, the flows from Italy with iself, and subtract this value from the total of column.
9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional FVA of CE sectors embodied in exports by NUTS2 and sector.
12) Split the interregional export vector in (7) only for Italy and multiply it by the regional shares obtained in (10) to obtain the regional exports.
13) Obtain a measure of backward participation in GVCs by dividing the regional FVA in each sector, by region (NUTS2), by the total regional exports in all sectors of that region.

8.9 Indicator 6: guide for the calculation of the regional overall participation of CE sectors in GVCs

8.9.1 Regional DVA in CE

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.
5) Diagonalize the value-added in CE sectors.
6) Subtract the intraregional flows of the total sales to obtain the interregional exports.
7) Diagonalize the interregional export vector.
8) Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.
9) Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional DVA of CE sectors embodied in exports by NUTS2 and sector.

8.9.2 Regional FVA in CE

1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.
3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by gross output, X, and diagonalize it.

5) Subtract the intraregional flows of the total sales to obtain the interregional exports.
6) Diagonalize the interregional export vector.
7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-added, for all countries and sectors, in order to obtain the T matrix.
8) Sum the columns relating to Italy's transaction with other countries to obtain the FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA), that is, the flows from Italy with iself, and subtract this value from the total of column.
9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.
10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
11) Multiply the regional shares of employment to obtain the regional FVA of CE sectors embodied in exports by NUTS2 and sector.

8.9.3 Regional exports

1) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.
2) Subtract the intraregional flows of the total sales to obtain the interregional exports and split it just for Italy.
3) Multiply the regional shares by the regional exports vector for Italy.

8.9.4 Overall regional participation in CE

1) Sum the regional DVA and FVA and divide it by the sum of regional exports in all sectors within each region.

8.10 R Commands

```
# Step by step to build the regional DVA indicator for CE sectors in Italy

# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function
```

```
# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)

### Step by step to build the regional DVA indicator for CE sectors in Italy

# 1)   Divide the intermediate transactions matrix, Z, by gross output, X, in order to get
technical coefficient, A.

# Read the elements of the input-output matrix

# Read the "IOM" sheet from the Excel file
data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[1:3465, 1] # first column
col_names <- data[1, -1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)

Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
```

```
num_rows <- nrow(data)

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Transpose W to get a single column matrix
W_column <- t(W)

# Display the result
print(X_column)

# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient

# 2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.

n = length(X) # Number of sectors by countries

I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse

# 3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.
```

```
# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
```

```
 "35" = "35",
 "36T39" = c("36", "37", "38", "39"),
 "41T43" = c("41", "42", "43"),
 "45T47" = c("45", "46", "47"),
 "49" = "49",
 "50" = "50",
 "51" = "51",
 "52" = "52",
 "53" = "53",
 "55T56" = c("55", "56"),
 "58T60" = c("58", "59", "60"),
 "61" = "61",
 "62T63" = c("62", "63"),
 "64T66" = c("64", "65", "66"),
 "68" = "68",
 "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
 "77T82" = c("77", "78", "79", "80", "81", "82"),
 "84" = "84",
 "85" = "85",
 "86T88" = c("86", "87", "88"),
 "90T93" = c("90", "91", "92", "93"),
 "94T96" = c("94", "95", "96"),
 "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
 sectors <- group_mapping[[new_name]]
 filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
 summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
 summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

 df <- agg_sectors %>%
  mutate(
   shares_CE = case_when(
```

```
    name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
    name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
    name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
    name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) / sum(value[name == "20"]),
    name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name == "22"]),
    name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name == "23"]),
    name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) / sum(value[name == "24"]),
    name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
    name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
    name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891", "2892", "2896", "2899")]) / sum(value[name == "28"]),
    name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
    name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name == "30"]),
    name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316", "3317", "3319", "3320")]) / sum(value[name == "31T33"]),
    name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) / sum(value[name == "36T39"]),
    name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name == "41T43"]),
    name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) / sum(value[name == "45T47"]),
    name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name == "69T75"]),
    name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731", "7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
    name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524", "9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
   )
  )

# 4)   Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.
```

```
# column vector of W
w_matrix <- matrix(W_column)

# Combine row_names_matrix with W
value_added_names <- cbind(row_names, w_matrix)

# Rename the columns
colnames(value_added_names) <- c("matriz_names", "W")

# Visualize
print(value_added_names)

# Data frame
value_added_names <- as.data.frame(value_added_names)

# Splitting the first column from the first underscore _
split_names <- strsplit(as.character(value_added_names$matriz_names), "_")

# Creating the two new columns
value_added_names$column1 <- sapply(split_names, "[[", 1)
value_added_names$column2 <- sapply(split_names, "[[", 2)

# Create a data frame with the provided data
new_names <- data.frame(
  x1 = c("A01", "A03", "B05", "B07", "B09", "C10T12", "C13T15", "C16", "C17", "C19", "C20", "C21", "C22",
"C23", "C24", "C25", "C26", "C27", "C28", "C29", "C30", "C31T33", "D", "E", "F", "G", "H49", "H50", "H51",
"H52", "H53", "I", "J58T60", "J61", "J62", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T"),
  x2 = c("01T02", "03", "05T06", "07T08", "09", "10T12", "13T15", "16", "17T18", "19", "20", "21", "22", "23",
"24", "25", "26", "27", "28", "29", "30", "31T33", "35", "36T39", "41T43", "45T47", "49", "50", "51", "52", "53",
"55T56", "58T60", "61", "62T63", "64T66", "68", "69T75", "77T82", "84", "85", "86T88", "90T93",
"94T96", "97T98")
  )

# Assuming that the base value_added_names is already loaded and has columns
column1 and column2
# Replace column2 in value_added_names with the new names
value_added_names$column2 <- ifelse(value_added_names$column2 %in%
new_names$x1,
                    new_names$x2[match(value_added_names$column2,
new_names$x1)],
                    value_added_names$column2)
```

```
# Rename column 1 to "country" and column 2 to "code_sector"
names(value_added_names)[3] <- "country"
names(value_added_names)[4] <- "code_sector"

value_added_names <- data.frame(value_added_names)

# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Merge the filtered dataframes
merged_data_new <- merge(value_added_names, df_na, by.x = "code_sector", by.y =
"name", all.x = TRUE)

# Filter by country and sector
merged_data_new <- merged_data_new[order(merged_data_new$country,
merged_data_new$code_sector), ]

# Replace NA with 0 in the shares_CE column
merged_data_new$shares_CE[is.na(merged_data_new$shares_CE)] <- 0

# Visualize the result
print(merged_data_new)

# Convert shares_CE and X2 columns to numeric
merged_data_new$shares_CE <- as.numeric(merged_data_new$shares_CE)
merged_data_new$W <- as.numeric(merged_data_new$W)

# Multiply shares_CE by W only if country is "ITA", otherwise assign 0
merged_data_new$W_CE <- ifelse(merged_data_new$country == "ITA",
merged_data_new$shares_CE * merged_data_new$W, 0)

# Viewing the updated data frame
print(merged_data_new)

# Extrair a coluna W_CE de merged_data
W_CE_column <- merged_data_new$W_CE

# Converte W_CE into a matrix
W_CE_matrix <- as.matrix(W_CE_column)

V_CE = W_CE_matrix / X_column # share of value-added in the total of production
```

```
# 5) Diagonalize the value-added in CE sectors.

# Convert V_CE in matrix
V_CE = data.matrix(V_CE)

diag_V_CE = matrix(0, 3465, 3465)
diag(diag_V_CE) = V_CE # diagonal of the share of value-added in CE sectors

# 6) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of
that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
            "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
            "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
            "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
            "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
            "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
            "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
            "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
```

```
        "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

# 7)    Diagonalize the interregional export vector.

diag_inter_exp= matrix(0, 3465, 3465)
diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports

# 8)    Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.
```

```
T_CE = diag_V_CE %*% L %*% diag_inter_exp # T matrix for CE in Italy

T_CE_ = data.frame(T_CE)
write_xlsx (T_CE_, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_CE_.xlsx")

# 9)  Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.

# Define the indices of the rows and columns you want to sum
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# Extract the submatrix containing only the desired rows and columns
submatrix <- T_CE[start_index:end_index, start_index:end_index]

# Sum of intraregional flows for columns, DVA for CE in Italy
intra_CE_ITA <- colSums(submatrix)

# 10)  Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <-  industries_and_services %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))
```

```
# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
  "9" = "09",
  "10" = "10T12",
  "11" = "10T12",
  "12" = "10T12",
  "13" = "13T15",
  "14" = "13T15",
  "15" = "13T15",
  "16" = "16",
  "17" = "17T18",
  "18" = "17T18",
  "19" = "19",
  "20" = "20",
  "21" = "21",
```

```
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
"59" = "58T60",
"60" = "58T60",
"61" = "61",
"62" = "62T63",
"63" = "62T63",
"64" = "64T66",
"65" = "64T66",
"66" = "64T66",
"68" = "68",
"69" = "69T75",
"70" = "69T75",
```

```
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
    name %in% c("3") ~ "03",
    name %in% c("5", "6") ~ "05T06",
    name %in% c("7", "8") ~ "07T08",
    name %in% c("9") ~ "09",
    name %in% c("10", "11", "12") ~ "10T12",
    name %in% c("13", "14", "15") ~ "13T15",
    name %in% c("16") ~ "16",
    name %in% c("17", "18") ~ "17T18",
    name %in% c("19") ~ "19",
    name %in% c("20") ~ "20",
    name %in% c("21") ~ "21",
```

```
    name %in% c("22") ~ "22",
    name %in% c("23") ~ "23",
    name %in% c("24") ~ "24",
    name %in% c("25") ~ "25",
    name %in% c("26") ~ "26",
    name %in% c("27") ~ "27",
    name %in% c("28") ~ "28",
    name %in% c("29") ~ "29",
    name %in% c("30") ~ "30",
    name %in% c("31", "32", "33") ~ "31T33",
    name %in% c("35") ~ "35",
    name %in% c("36", "37", "38", "39") ~ "36T39",
    name %in% c("41", "42", "43") ~ "41T43",
    name %in% c("45", "46", "47") ~ "45T47",
    name %in% c("49") ~ "49",
    name %in% c("50") ~ "50",
    name %in% c("51") ~ "51",
    name %in% c("52") ~ "52",
    name %in% c("53") ~ "53",
    name %in% c("55", "56") ~ "55T56",
    name %in% c("58", "59", "60") ~ "58T60",
    name %in% c("61") ~ "61",
    name %in% c("62", "63") ~ "62T63",
    name %in% c("64", "65", "66") ~ "64T66",
    name %in% c("68") ~ "68",
    name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
    name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
    name %in% c("84") ~ "84",
    name %in% c("85") ~ "85",
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
```

```
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)
```

```
# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11)  Multiply the regional shares of employment to obtain the regional DVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
intra_CE_ITA_vector <- as.numeric(intra_CE_ITA)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(intra_CE_ITA_vector) == nrow(combined_shares))

# Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from
`intra_CE_ITA`
combined_DVA_multiplied <- combined_shares
combined_DVA_multiplied[, 2:22] <- combined_shares[, 2:22] * intra_CE_ITA_vector

write_xlsx (combined_DVA_multiplied, "C:/Users/damar/OneDrive/Documentos/Polimi
Project 3/combined_DVA_multiplied.xlsx")

# Sum the rows of columns 2 to 22 in `combined_DVA_multiplied`
row_sums <- rowSums(combined_DVA_multiplied[, 2:22])

# Sum the columns of columns 2 to 22 in `combined_DVA_multiplied`
```

```
# Calculate the sum for each of the 21 columns to get a total value per region
inter_DVA_sum <- sapply(combined_DVA_multiplied[, 2:22], sum)

# Convert to a data frame
inter_DVA_sum <- as.data.frame(t(inter_DVA_sum))

# Write to Excel file
write_xlsx(inter_DVA_sum, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/inter_DVA_sum.xlsx")
# Step by step to build the regional FVA indicator for CE sectors in Italy

# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function

# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)


### Step by step to build the regional FVA indicator for CE sectors in Italy

# 1)    Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.

data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[,1] # first column
```

```
col_names <- data[1,-1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)

Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
num_rows <- nrow(data)

X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Display the result
print(X_column)

# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient

# 2)    Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief
inverse matrix.

n = length(X) # Number of sectors by countries
```

```
I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse

# 3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

# Path to the Excel file
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
   "03" = "03",
   "05T06" = c("05", "06"),
   "07T08" = c("07", "08"),
   "09" = "09",
   "10T12" = c("10", "11", "12"),
   "13T15" = c("13", "14", "15"),
   "16" = "16",
   "17T18" = c("17", "18"),
   "19" = "19",
   "20" = "20",
   "21" = "21",
   "22" = "22",
   "23" = "23",
```

```
    "24" = "24",
    "25" = "25",
    "26" = "26",
    "27" = "27",
    "28" = "28",
    "29" = "29",
    "30" = "30",
    "31T33" = c("31", "32", "33"),
    "35" = "35",
    "36T39" = c("36", "37", "38", "39"),
    "41T43" = c("41", "42", "43"),
    "45T47" = c("45", "46", "47"),
    "49" = "49",
    "50" = "50",
    "51" = "51",
    "52" = "52",
    "53" = "53",
    "55T56" = c("55", "56"),
    "58T60" = c("58", "59", "60"),
    "61" = "61",
    "62T63" = c("62", "63"),
    "64T66" = c("64", "65", "66"),
    "68" = "68",
    "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
    "77T82" = c("77", "78", "79", "80", "81", "82"),
    "84" = "84",
    "85" = "85",
    "86T88" = c("86", "87", "88"),
    "90T93" = c("90", "91", "92", "93"),
    "94T96" = c("94", "95", "96"),
    "97T98" = c("97", "98")
  )

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
    filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
    summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
    summarised_data
  }))
```

```
# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
  mutate(
    shares_CE = case_when(
      name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
      name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
      name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
      name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) / sum(value[name == "20"]),
      name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name == "22"]),
      name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name == "23"]),
      name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) / sum(value[name == "24"]),
      name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
      name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
      name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891", "2892", "2896", "2899")]) / sum(value[name == "28"]),
      name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
      name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name == "30"]),
      name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316", "3317", "3319", "3320")]) / sum(value[name == "31T33"]),
      name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) / sum(value[name == "36T39"]),
      name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name == "41T43"]),
      name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) / sum(value[name == "45T47"]),
      name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name == "69T75"]),
      name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731", "7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
      name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
```

```
      name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524", "9525", "9529", "9601")]) / sum(value[name == "94T96"]),
      TRUE ~ 0
    )
  )

# 4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by gross output, X, and diagonalize it.

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)

# Transpose W to get a single column matrix
W_column <- t(W)

V_X = W_column / X_column # share of value-added in the total of production

# Convert V_X in matrix
V_X = data.matrix(V_X)

# Diagonalize V_X
diag_V_X = matrix(0, 3465, 3465)
diag(diag_V_X) = V_X # diagonal of the share of value-added in CE sectors

# 5) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
```

```
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
               "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
               "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
               "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
               "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
               "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
               "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
               "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
               "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
```

```
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

print(inter_exp)

# 6) Diagonalize the interregional export vector.

diag_inter_exp= matrix(0, 3465, 3465)
diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports

# 7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-
added, for all countries and sectors, in order to obtain the T matrix.

T_all = diag_V_X %*% L %*% diag_inter_exp # T matrix for all sectors

T_all = data.frame(T_all)
write_xlsx (T_all, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_all.xlsx")

# 8)    Sum the columns relating to Italy's transaction with other countries to obtain the
FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA),
that is, the flows from Italy with iself, and subtract this value from the total of column.

# Define the indices of the rows and columns you want to sum
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# FVA

# Calculate the sum of the columns within the specified range, ignoring NA values
sum_subset <- colSums(T_all[, start_index:end_index], na.rm = TRUE)

intra_all_ITA <- (colSums(T_all[start_index:end_index, start_index:end_index]))

print(intra_all_ITA)

# Subtract intra-regional flows from ITA from the sum
FVA_all_ITA <- sum_subset - intra_all_ITA
```

```
# Visualize the results
print(FVA_all_ITA)

# 9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.

# Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
  "19" = "19",
  "20" = "20",
  "21" = "21",
```

```
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
  summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
  summarised_data
```

```
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
  mutate(
    shares_CE = case_when(
      name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
      name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
      name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
      name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) /
sum(value[name == "20"]),
      name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name ==
"22"]),
      name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name ==
"23"]),
      name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) /
sum(value[name == "24"]),
      name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
      name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
      name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891",
"2892", "2896", "2899")]) / sum(value[name == "28"]),
      name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
      name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name ==
"30"]),
      name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316",
"3317", "3319", "3320")]) / sum(value[name == "31T33"]),
      name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) /
sum(value[name == "36T39"]),
      name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name
== "41T43"]),
      name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) /
sum(value[name == "45T47"]),
      name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name ==
"69T75"]),
      name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731",
"7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
      name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
```

```
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524",
"9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
  )
 )


# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Convert shares_CE to numeric
shares_CE_ita <- as.numeric(df_na$shares_CE)

# Multiply shares_CE by the vector FVA_all_ITA
FVA_CE <- df_na$shares_CE * FVA_all_ITA

print(FVA_CE)

# 10) Calculate the percentage of employment in each NACE 2-digit sector in the
country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and
services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <-  industries_and_services %>%
 separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
 mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
 mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all colums
labor_2_new_two <- labor_2_new_two %>%
 mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
```

```
 mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
 mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
 mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
 mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
 mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
 "1" = "01T02",
 "2" = "01T02",
 "3" = "03",
 "5" = "05T06",
 "6" = "05T06",
 "7" = "07T08",
 "8" = "07T08",
 "9" = "09",
 "10" = "10T12",
 "11" = "10T12",
 "12" = "10T12",
 "13" = "13T15",
 "14" = "13T15",
 "15" = "13T15",
 "16" = "16",
 "17" = "17T18",
 "18" = "17T18",
 "19" = "19",
 "20" = "20",
 "21" = "21",
 "22" = "22",
 "23" = "23",
```

```
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
"59" = "58T60",
"60" = "58T60",
"61" = "61",
"62" = "62T63",
"63" = "62T63",
"64" = "64T66",
"65" = "64T66",
"66" = "64T66",
"68" = "68",
"69" = "69T75",
"70" = "69T75",
"71" = "69T75",
"72" = "69T75",
```

```
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
    name %in% c("3") ~ "03",
    name %in% c("5", "6") ~ "05T06",
    name %in% c("7", "8") ~ "07T08",
    name %in% c("9") ~ "09",
    name %in% c("10", "11", "12") ~ "10T12",
    name %in% c("13", "14", "15") ~ "13T15",
    name %in% c("16") ~ "16",
    name %in% c("17", "18") ~ "17T18",
    name %in% c("19") ~ "19",
    name %in% c("20") ~ "20",
    name %in% c("21") ~ "21",
    name %in% c("22") ~ "22",
    name %in% c("23") ~ "23",
```

```
    name %in% c("24") ~ "24",
    name %in% c("25") ~ "25",
    name %in% c("26") ~ "26",
    name %in% c("27") ~ "27",
    name %in% c("28") ~ "28",
    name %in% c("29") ~ "29",
    name %in% c("30") ~ "30",
    name %in% c("31", "32", "33") ~ "31T33",
    name %in% c("35") ~ "35",
    name %in% c("36", "37", "38", "39") ~ "36T39",
    name %in% c("41", "42", "43") ~ "41T43",
    name %in% c("45", "46", "47") ~ "45T47",
    name %in% c("49") ~ "49",
    name %in% c("50") ~ "50",
    name %in% c("51") ~ "51",
    name %in% c("52") ~ "52",
    name %in% c("53") ~ "53",
    name %in% c("55", "56") ~ "55T56",
    name %in% c("58", "59", "60") ~ "58T60",
    name %in% c("61") ~ "61",
    name %in% c("62", "63") ~ "62T63",
    name %in% c("64", "65", "66") ~ "64T66",
    name %in% c("68") ~ "68",
    name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
    name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
    name %in% c("84") ~ "84",
    name %in% c("85") ~ "85",
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")
```

```
# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
```

```
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11)  Multiply the regional shares of employment to obtain the regional FVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
FVA_CE <- as.numeric(FVA_CE)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(FVA_CE) == nrow(combined_shares))

# Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from
`intra_CE_ITA`
combined_FVA_multiplied <- combined_shares
combined_FVA_multiplied[, 2:22] <- combined_shares[, 2:22] * FVA_CE

write_xlsx (combined_FVA_multiplied, "C:/Users/damar/OneDrive/Documentos/Polimi
Project 3/combined_FVA_multiplied.xlsx")

# Sum the rows of columns 2 to 22 in `combined_FVA_multiplied`
row_sums <- rowSums(combined_FVA_multiplied[, 2:22])

# Calculate the sum for each of the 21 columns to get a total value per region
inter_FVA_sum <- sapply(combined_FVA_multiplied[, 2:22], sum)

# Convert to a data frame
```

```
inter_FVA_sum <- as.data.frame(t(inter_FVA_sum))

# Write to Excel file
write_xlsx(inter_FVA_sum, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/inter_FVA_sum.xlsx")
```

5. Step by step to build the regional exports indicator

```
# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function

# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)

# Step by step to build the regional exports indicator

# 1) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <- industries_and_services %>%
```

```
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
  "9" = "09",
  "10" = "10T12",
```

```
"11" = "10T12",
"12" = "10T12",
"13" = "13T15",
"14" = "13T15",
"15" = "13T15",
"16" = "16",
"17" = "17T18",
"18" = "17T18",
"19" = "19",
"20" = "20",
"21" = "21",
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
```

```
  "59" = "58T60",
  "60" = "58T60",
  "61" = "61",
  "62" = "62T63",
  "63" = "62T63",
  "64" = "64T66",
  "65" = "64T66",
  "66" = "64T66",
  "68" = "68",
  "69" = "69T75",
  "70" = "69T75",
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
```

```
name %in% c("3") ~ "03",
name %in% c("5", "6") ~ "05T06",
name %in% c("7", "8") ~ "07T08",
name %in% c("9") ~ "09",
name %in% c("10", "11", "12") ~ "10T12",
name %in% c("13", "14", "15") ~ "13T15",
name %in% c("16") ~ "16",
name %in% c("17", "18") ~ "17T18",
name %in% c("19") ~ "19",
name %in% c("20") ~ "20",
name %in% c("21") ~ "21",
name %in% c("22") ~ "22",
name %in% c("23") ~ "23",
name %in% c("24") ~ "24",
name %in% c("25") ~ "25",
name %in% c("26") ~ "26",
name %in% c("27") ~ "27",
name %in% c("28") ~ "28",
name %in% c("29") ~ "29",
name %in% c("30") ~ "30",
name %in% c("31", "32", "33") ~ "31T33",
name %in% c("35") ~ "35",
name %in% c("36", "37", "38", "39") ~ "36T39",
name %in% c("41", "42", "43") ~ "41T43",
name %in% c("45", "46", "47") ~ "45T47",
name %in% c("49") ~ "49",
name %in% c("50") ~ "50",
name %in% c("51") ~ "51",
name %in% c("52") ~ "52",
name %in% c("53") ~ "53",
name %in% c("55", "56") ~ "55T56",
name %in% c("58", "59", "60") ~ "58T60",
name %in% c("61") ~ "61",
name %in% c("62", "63") ~ "62T63",
name %in% c("64", "65", "66") ~ "64T66",
name %in% c("68") ~ "68",
name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
name %in% c("84") ~ "84",
name %in% c("85") ~ "85",
name %in% c("86", "87", "88") ~ "86T88",
name %in% c("90", "91", "92", "93") ~ "90T93",
```

```
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")
```

```
# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 2)   Subtract the intraregional flows of the total sales to obtain the interregional
exports and split it just for Italy.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of
that country
```

```
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
             "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
             "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
             "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
             "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
             "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
             "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
             "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
             "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)
```

```
# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

print(inter_exp)

# Exports for Italy

inter_exp <- as.matrix(inter_exp)
exp_subset <- inter_exp[1621:1665,1]

# 3)	Multiply the regional shares by the regional exports vector for Italy.

combined_shares_1 <- combined_shares

inter_exports <- combined_shares_1
inter_exports[, 2:22] <- combined_shares[, 2:22] * exp_subset

write_xlsx (inter_exports, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/inter_exports.xlsx")

inter_exports_sum <- colSums(inter_exports[, 2:22])

inter_exports_sum <- as.numeric(inter_exports_sum)

# Sum the rows of columns 2 to 22 in `inter_exports`
row_sums <- rowSums(inter_exports[, 2:22])

# Calculate the sum for each of the 21 columns to get a total value per region
exports_sum_c <- sapply(inter_exports[, 2:22], sum)

# Convert to a data frame
```

```
exports_sum_c <- as.data.frame(t(exports_sum_c ))

# Write to Excel file
write_xlsx(exports_sum_c, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/exports_sum_c.xlsx")

# Step by step to build the regional forward participation of CE sectors in GVCs

# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function

# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)

###   Step by step to build the regional forward participation of CE sectors in GVCs


# 1)   Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.

# Read the elements of the input-output matrix

# Read the "IOM" sheet from the Excel file
data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[1:3465, 1] # first column
```

```
col_names <- data[1, -1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)

Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
num_rows <- nrow(data)

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Transpose W to get a single column matrix
W_column <- t(W)

# Display the result
print(X_column)

# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient
```

2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.

```
n = length(X) # Number of sectors by countries

I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse
```

3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

```
# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
```

```
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
```

```
 summarised_data <- filtered_data %>% summarise(name = new_name, value = sum(value))
 summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
 mutate(
  shares_CE = case_when(
   name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
   name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
   name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
   name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) / sum(value[name == "20"]),
   name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name == "22"]),
   name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name == "23"]),
   name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) / sum(value[name == "24"]),
   name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
   name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
   name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891", "2892", "2896", "2899")]) / sum(value[name == "28"]),
   name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
   name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name == "30"]),
   name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316", "3317", "3319", "3320")]) / sum(value[name == "31T33"]),
   name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) / sum(value[name == "36T39"]),
   name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name == "41T43"]),
   name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) / sum(value[name == "45T47"]),
   name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name == "69T75"]),
```

```
    name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731",
"7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
    name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524",
"9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
   )
  )

# 4)   Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.

# column vector of W
w_matrix <- matrix(W_column)

# Combine row_names_matrix with W
value_added_names <- cbind(row_names, w_matrix)

# Rename the columns
colnames(value_added_names) <- c("matriz_names", "W")

# Visualize
print(value_added_names)

# Data frame
value_added_names <- as.data.frame(value_added_names)

# Splitting the first column from the first underscore _
split_names <- strsplit(as.character(value_added_names$matriz_names), "_")

# Creating the two new columns
value_added_names$column1 <- sapply(split_names, "[[", 1)
value_added_names$column2 <- sapply(split_names, "[[", 2)

# Create a data frame with the provided data
new_names <- data.frame(
  x1 = c("A01", "A03", "B05", "B07", "B09", "C10T12", "C13T15", "C16", "C17", "C19", "C20", "C21", "C22",
"C23", "C24", "C25", "C26", "C27", "C28", "C29", "C30", "C31T33", "D", "E", "F", "G", "H49", "H50", "H51",
"H52", "H53", "I", "J58T60", "J61", "J62", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T"),
  x2 = c("01T02", "03", "05T06", "07T08", "09", "10T12", "13T15", "16", "17T18", "19", "20", "21", "22", "23",
"24", "25", "26", "27", "28", "29", "30", "31T33", "35", "36T39", "41T43", "45T47", "49", "50", "51", "52", "53",
```

```
"55T56", "58T60", "61", "62T63", "64T66", "68", "69T75", "77T82", "84", "85", "86T88", "90T93",
"94T96", "97T98")
)

# Assuming that the base value_added_names is already loaded and has columns
column1 and column2
# Replace column2 in value_added_names with the new names
value_added_names$column2 <- ifelse(value_added_names$column2 %in%
new_names$x1,
                         new_names$x2[match(value_added_names$column2,
new_names$x1)],
                         value_added_names$column2)

# Rename column 1 to "country" and column 2 to "code_sector"
names(value_added_names)[3] <- "country"
names(value_added_names)[4] <- "code_sector"

value_added_names <- data.frame(value_added_names)

# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Merge the filtered dataframes
merged_data_new <- merge(value_added_names, df_na, by.x = "code_sector", by.y =
"name", all.x = TRUE)

# Filter by country and sector
merged_data_new <- merged_data_new[order(merged_data_new$country,
merged_data_new$code_sector), ]

# Replace NA with 0 in the shares_CE column
merged_data_new$shares_CE[is.na(merged_data_new$shares_CE)] <- 0

# Visualize the result
print(merged_data_new)

# Convert shares_CE and X2 columns to numeric
merged_data_new$shares_CE <- as.numeric(merged_data_new$shares_CE)
merged_data_new$W <- as.numeric(merged_data_new$W)

# Multiply shares_CE by W only if country is "ITA", otherwise assign 0
```

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
GRINS FOUNDATION

```
merged_data_new$W_CE <- ifelse(merged_data_new$country == "ITA", merged_data_new$shares_CE * merged_data_new$W, 0)

# Viewing the updated data frame
print(merged_data_new)

# Extrair a coluna W_CE de merged_data
W_CE_column <- merged_data_new$W_CE

# Converte W_CE into a matrix
W_CE_matrix <- as.matrix(W_CE_column)

V_CE = W_CE_matrix / X_column # share of value-added in the total of production

# 5) Diagonalize the value-added in CE sectors.

# Convert V_CE in matrix
V_CE = data.matrix(V_CE)

diag_V_CE = matrix(0, 3465, 3465)
diag(diag_V_CE) = V_CE # diagonal of the share of value-added in CE sectors

# 6) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}
```

```
# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
              "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
              "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
              "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
              "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
              "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
              "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
              "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
              "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)
```

```
# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector
```

\# 7) Diagonalize the interregional export vector.

```
diag_inter_exp= matrix(0, 3465, 3465)
diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports
```

\# 8) Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.

```
T_CE = diag_V_CE %*% L %*% diag_inter_exp # T matrix for CE in Italy

T_CE_ = data.frame(T_CE)
write_xlsx (T_CE_, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_CE_.xlsx")
```

\# 9) Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.

```
# Define the indices of the rows and columns you want to sum
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# Extract the submatrix containing only the desired rows and columns
submatrix <- T_CE[start_index:end_index, start_index:end_index]

# Sum of intraregional flows for columns, DVA for CE in Italy
intra_CE_ITA <- colSums(submatrix)
```

\# 10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

```
# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")
```

```
# Separate the column containing names and numeric codes
labor_2_new_two <-  industries_and_services %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
```

"9" = "09",
"10" = "10T12",
"11" = "10T12",
"12" = "10T12",
"13" = "13T15",
"14" = "13T15",
"15" = "13T15",
"16" = "16",
"17" = "17T18",
"18" = "17T18",
"19" = "19",
"20" = "20",
"21" = "21",
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",

```
  "56" = "55T56",
  "58" = "58T60",
  "59" = "58T60",
  "60" = "58T60",
  "61" = "61",
  "62" = "62T63",
  "63" = "62T63",
  "64" = "64T66",
  "65" = "64T66",
  "66" = "64T66",
  "68" = "68",
  "69" = "69T75",
  "70" = "69T75",
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
```

```
mutate(code = case_when(
  name %in% c("1", "2") ~ "01T02",
  name %in% c("3") ~ "03",
  name %in% c("5", "6") ~ "05T06",
  name %in% c("7", "8") ~ "07T08",
  name %in% c("9") ~ "09",
  name %in% c("10", "11", "12") ~ "10T12",
  name %in% c("13", "14", "15") ~ "13T15",
  name %in% c("16") ~ "16",
  name %in% c("17", "18") ~ "17T18",
  name %in% c("19") ~ "19",
  name %in% c("20") ~ "20",
  name %in% c("21") ~ "21",
  name %in% c("22") ~ "22",
  name %in% c("23") ~ "23",
  name %in% c("24") ~ "24",
  name %in% c("25") ~ "25",
  name %in% c("26") ~ "26",
  name %in% c("27") ~ "27",
  name %in% c("28") ~ "28",
  name %in% c("29") ~ "29",
  name %in% c("30") ~ "30",
  name %in% c("31", "32", "33") ~ "31T33",
  name %in% c("35") ~ "35",
  name %in% c("36", "37", "38", "39") ~ "36T39",
  name %in% c("41", "42", "43") ~ "41T43",
  name %in% c("45", "46", "47") ~ "45T47",
  name %in% c("49") ~ "49",
  name %in% c("50") ~ "50",
  name %in% c("51") ~ "51",
  name %in% c("52") ~ "52",
  name %in% c("53") ~ "53",
  name %in% c("55", "56") ~ "55T56",
  name %in% c("58", "59", "60") ~ "58T60",
  name %in% c("61") ~ "61",
  name %in% c("62", "63") ~ "62T63",
  name %in% c("64", "65", "66") ~ "64T66",
  name %in% c("68") ~ "68",
  name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
  name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
  name %in% c("84") ~ "84",
  name %in% c("85") ~ "85",
```

```
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
```

```
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11)   Multiply the regional shares of employment to obtain the regional DVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
intra_CE_ITA_vector <- as.numeric(intra_CE_ITA)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(intra_CE_ITA_vector) == nrow(combined_shares))
```

```
  # Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from
`intra_CE_ITA`
  combined_DVA_multiplied <- combined_shares
combined_DVA_multiplied[, 2:22] <- combined_shares[, 2:22] * intra_CE_ITA_vector

# Sum the rows of columns 2 to 22 in `combined_DVA_multiplied`
row_sums <- rowSums(combined_DVA_multiplied[, 2:22])

# Sum the columns of columns 2 to 22 in `combined_DVA_multiplied`

# Calculate the sum for each of the 21 columns to get a total value per region
inter_DVA_sum <- sapply(combined_DVA_multiplied[, 2:22], sum)

# Convert to a data frame
inter_DVA_sum <- as.data.frame(t(inter_DVA_sum))

# 12) Split the interregional export vector in (7) only for Italy and multiply it by the regional
shares obtained in (10) to obtain the regional exports.

inter_exp <- as.matrix(inter_exp)
exp_subset <- inter_exp[1621:1665,1]

combined_shares_1 <- combined_shares

inter_exports <- combined_shares_1
inter_exports[, 2:22] <- combined_shares[, 2:22] * exp_subset

inter_exports_sum <- colSums(inter_exports[, 2:22])

# 13)   Obtain a measure of forward participation in GVCs by dividing the regional DVA in
each sector, by region (NUTS2), by the total regional exports in all sectors of that region.

combined_shares_2 <- combined_shares

forward <- combined_shares_2

# Iterate over the columns of forward and perform element-wise division
for (i in 2:22) {
  forward[, i] <- combined_DVA_multiplied[, i] / inter_exports_sum[i - 1]
}

# Write the modified forward to an Excel file
```

```
write_xlsx(forward, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/forward.xlsx")

# Iterate over the total of columns of forward and exports to get a total value per region

for (i in 2:22) {
  forward_total_region <- inter_DVA_sum / inter_exports_sum
}

forward_total_region <- as.data.frame(forward_total_region)

write_xlsx (forward_total_region, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/forward_total_region.xlsx")


# Step by step to build the regional backward participation of CE sectors in GVCs

# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function

# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)

### Step by step to build the regional backward participation of CE sectors in GVCs

# 1) Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.
```

```
data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[,1] # first column
col_names <- data[1,-1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)

Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
num_rows <- nrow(data)

X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Display the result
print(X_column)

# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient
```

2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.

n = length(X) # Number of sectors by countries

I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse

3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

```
# Path to the Excel file
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
```

```
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
```

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
GRINS FOUNDATION

```
 summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
 summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
 mutate(
  shares_CE = case_when(
   name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
   name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
   name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
   name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) /
sum(value[name == "20"]),
   name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name ==
"22"]),
   name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name ==
"23"]),
   name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) /
sum(value[name == "24"]),
   name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
   name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
   name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891",
"2892", "2896", "2899")]) / sum(value[name == "28"]),
   name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
   name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name ==
"30"]),
   name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316",
"3317", "3319", "3320")]) / sum(value[name == "31T33"]),
   name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) /
sum(value[name == "36T39"]),
   name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name
== "41T43"]),
   name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) /
sum(value[name == "45T47"]),
   name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name ==
"69T75"]),
```

```
    name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731",
"7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
    name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524",
"9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
   )
  )

# 4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by
gross output, X, and diagonalize it.

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)

# Transpose W to get a single column matrix
W_column <- t(W)

V_X = W_column / X_column # share of value-added in the total of production

# Convert V_X in matrix
V_X = data.matrix(V_X)

# Diagonalize V_X
diag_V_X = matrix(0, 3465, 3465)
diag(diag_V_X) = V_X # diagonal of the share of value-added in CE sectors

# 5) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of
that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
 # Filter rows and columns based on the country code
 rows <- grep(paste0("^", country_codes), rownames(matrix))
 cols <- grep(paste0("^", country_codes), colnames(matrix))
```

```
  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
               "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
               "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
               "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
               "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
               "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
               "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
               "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
               "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
```

```
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

print(inter_exp)

# 6) Diagonalize the interregional export vector.

diag_inter_exp= matrix(0, 3465, 3465)
diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports

# 7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-added, for all countries and sectors, in order to obtain the T matrix.

T_all = diag_V_X %*% L %*% diag_inter_exp # T matrix for all sectors

T_all = data.frame(T_all)
write_xlsx (T_all, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_all.xlsx")

# 8)   Sum the columns relating to Italy's transaction with other countries to obtain the FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA), that is, the flows from Italy with iself, and subtract this value from the total of column.

# Define the indices of the rows and columns you want to sum
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# FVA

# Calculate the sum of the columns within the specified range, ignoring NA values
sum_subset <- colSums(T_all[, start_index:end_index], na.rm = TRUE)

intra_all_ITA <- (colSums(T_all[start_index:end_index, start_index:end_index]))

print(intra_all_ITA)
```

```
# Subtract intra-regional flows from ITA from the sum
FVA_all_ITA <- sum_subset - intra_all_ITA

# Visualize the results
print(FVA_all_ITA)

# 9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.

# Calculate the CE sectoral shares by dividing the sectoral employment related to the
CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE,
following the sectoral composition of the CE sectors.

# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
```

```
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
```

```
 summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
 summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
 mutate(
  shares_CE = case_when(
   name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
   name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
   name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
   name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) /
sum(value[name == "20"]),
   name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name ==
"22"]),
   name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name ==
"23"]),
   name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) /
sum(value[name == "24"]),
   name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
   name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
   name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891",
"2892", "2896", "2899")]) / sum(value[name == "28"]),
   name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
   name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name ==
"30"]),
   name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316",
"3317", "3319", "3320")]) / sum(value[name == "31T33"]),
   name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) /
sum(value[name == "36T39"]),
   name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name
== "41T43"]),
   name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) /
sum(value[name == "45T47"]),
   name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name ==
"69T75"]),
```

```
    name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731", "7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
    name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524", "9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
   )
  )


# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Convert shares_CE to numeric
shares_CE_ita <- as.numeric(df_na$shares_CE)

# Multiply shares_CE by the vector FVA_all_ITA
FVA_CE <- df_na$shares_CE * FVA_all_ITA

print(FVA_CE)

# 10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <- industries_and_services %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all colums
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))
```

```
# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
  "9" = "09",
  "10" = "10T12",
  "11" = "10T12",
  "12" = "10T12",
  "13" = "13T15",
  "14" = "13T15",
  "15" = "13T15",
  "16" = "16",
  "17" = "17T18",
  "18" = "17T18",
  "19" = "19",
  "20" = "20",
```

"21" = "21",
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
"59" = "58T60",
"60" = "58T60",
"61" = "61",
"62" = "62T63",
"63" = "62T63",
"64" = "64T66",
"65" = "64T66",
"66" = "64T66",
"68" = "68",
"69" = "69T75",

```
  "70" = "69T75",
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
    name %in% c("3") ~ "03",
    name %in% c("5", "6") ~ "05T06",
    name %in% c("7", "8") ~ "07T08",
    name %in% c("9") ~ "09",
    name %in% c("10", "11", "12") ~ "10T12",
    name %in% c("13", "14", "15") ~ "13T15",
    name %in% c("16") ~ "16",
    name %in% c("17", "18") ~ "17T18",
    name %in% c("19") ~ "19",
    name %in% c("20") ~ "20",
```

```
    name %in% c("21") ~ "21",
    name %in% c("22") ~ "22",
    name %in% c("23") ~ "23",
    name %in% c("24") ~ "24",
    name %in% c("25") ~ "25",
    name %in% c("26") ~ "26",
    name %in% c("27") ~ "27",
    name %in% c("28") ~ "28",
    name %in% c("29") ~ "29",
    name %in% c("30") ~ "30",
    name %in% c("31", "32", "33") ~ "31T33",
    name %in% c("35") ~ "35",
    name %in% c("36", "37", "38", "39") ~ "36T39",
    name %in% c("41", "42", "43") ~ "41T43",
    name %in% c("45", "46", "47") ~ "45T47",
    name %in% c("49") ~ "49",
    name %in% c("50") ~ "50",
    name %in% c("51") ~ "51",
    name %in% c("52") ~ "52",
    name %in% c("53") ~ "53",
    name %in% c("55", "56") ~ "55T56",
    name %in% c("58", "59", "60") ~ "58T60",
    name %in% c("61") ~ "61",
    name %in% c("62", "63") ~ "62T63",
    name %in% c("64", "65", "66") ~ "64T66",
    name %in% c("68") ~ "68",
    name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
    name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
    name %in% c("84") ~ "84",
    name %in% c("85") ~ "85",
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"
```

```
# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)
```

```
# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11) Multiply the regional shares of employment to obtain the regional FVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
FVA_CE <- as.numeric(FVA_CE)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(FVA_CE) == nrow(combined_shares))

  # Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from
`intra_CE_ITA`
  combined_FVA_multiplied <- combined_shares
combined_FVA_multiplied[, 2:22] <- combined_shares[, 2:22] * FVA_CE

# Sum the rows of columns 2 to 22 in `combined_FVA_multiplied`
row_sums <- rowSums(combined_FVA_multiplied[, 2:22])

# Calculate the sum for each of the 21 columns to get a total value per region
inter_FVA_sum <- sapply(combined_FVA_multiplied[, 2:22], sum)

# Convert to a data frame
```

```
inter_FVA_sum <- as.data.frame(t(inter_FVA_sum))
```

12) Split the interregional export vector in (7) only for Italy and multiply it by the regional shares obtained in (10) to obtain the regional exports.

```
inter_exp <- as.matrix(inter_exp)
exp_subset <- inter_exp[1621:1665,1]

combined_shares_3 <- combined_shares

inter_exports <- combined_shares_3
inter_exports[, 2:22] <- combined_shares[, 2:22] * exp_subset

inter_exports_sum <- colSums(inter_exports[, 2:22])
```

13) Obtain a measure of backward participation in GVCs by dividing the regional FVA in each sector, by region (NUTS2), by the total regional exports in all sectors of that region.

```
combined_shares_3 <- combined_shares

backward <- combined_shares_3

# Iterate over the columns of forward and perform element-wise division
for (i in 2:22) {
  backward[, i] <- combined_FVA_multiplied[, i] / inter_exports_sum[i - 1]
}

write_xlsx (backward, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/backward.xlsx")

# Iterate over the total of columns of backward and exports to get a total value per region

for (i in 2:22) {
  backward_total_region <- inter_FVA_sum / inter_exports_sum
}

backward_total_region <- as.data.frame(backward_total_region)

write_xlsx (backward_total_region, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/backward_total_region.xlsx")
```

Step by step to build the regional overall participation of CE sectors in GVCs

```
# Path on your computer
setwd("C:/Users/damar/OneDrive/Documentos/Polimi Project 3")

rm(list = ls()) # cleaning function

# Install and read the necessary packages

#install.packages("writexl")
#install.packages("openxlsx")
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("officer")
#install.packages("dplyr")

library(writexl)
library(openxlsx)
library(readxl)
library(tidyverse)
library(officer)
library(dplyr)

###   Step by step to build the regional overall participation of CE sectors in GVCs

# 9.1 Regional DVA in CE


# 1)    Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.

# Read the elements of the input-output matrix

# Read the "IOM" sheet from the Excel file
data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[1:3465, 1] # first column
col_names <- data[1, -1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)
```

```
Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
num_rows <- nrow(data)

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Transpose W to get a single column matrix
W_column <- t(W)

# Display the result
print(X_column)

# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient

# 2) Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.

n = length(X) # Number of sectors by countries
```

```
I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse

# 3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)

# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
```

```
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
  summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
  summarised_data
}))
```

```
# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
 mutate(
   shares_CE = case_when(
     name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
     name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
     name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
     name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) /
sum(value[name == "20"]),
     name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name ==
"22"]),
     name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name ==
"23"]),
     name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) /
sum(value[name == "24"]),
     name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
     name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
     name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891",
"2892", "2896", "2899")]) / sum(value[name == "28"]),
     name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
     name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name ==
"30"]),
     name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316",
"3317", "3319", "3320")]) / sum(value[name == "31T33"]),
     name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) /
sum(value[name == "36T39"]),
     name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name
== "41T43"]),
     name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) /
sum(value[name == "45T47"]),
     name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name ==
"69T75"]),
     name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731",
"7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
     name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
     name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524",
"9525", "9529", "9601")]) / sum(value[name == "94T96"]),
     TRUE ~ 0
```

```
   )
  )
```

4) Multiply the weights of the CE sectors by the total value-added, obtaining value-added for the CE, V_CE, and divide by gross output, X.

```
# column vector of W
w_matrix <- matrix(W_column)

# Combine row_names_matrix with W
value_added_names <- cbind(row_names, w_matrix)

# Rename the columns
colnames(value_added_names) <- c("matriz_names", "W")

# Visualize
print(value_added_names)

# Data frame
value_added_names <- as.data.frame(value_added_names)

# Splitting the first column from the first underscore _
split_names <- strsplit(as.character(value_added_names$matriz_names), "_")

# Creating the two new columns
value_added_names$column1 <- sapply(split_names, "[[", 1)
value_added_names$column2 <- sapply(split_names, "[[", 2)

# Create a data frame with the provided data
new_names <- data.frame(
  x1 = c("A01", "A03", "B05", "B07", "B09", "C10T12", "C13T15", "C16", "C17", "C19", "C20", "C21", "C22", "C23", "C24", "C25", "C26", "C27", "C28", "C29", "C30", "C31T33", "D", "E", "F", "G", "H49", "H50", "H51", "H52", "H53", "I", "J58T60", "J61", "J62", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T"),
  x2 = c("01T02", "03", "05T06", "07T08", "09", "10T12", "13T15", "16", "17T18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31T33", "35", "36T39", "41T43", "45T47", "49", "50", "51", "52", "53", "55T56", "58T60", "61", "62T63", "64T66", "68", "69T75", "77T82", "84", "85", "86T88", "90T93", "94T96", "97T98")
)

# Assuming that the base value_added_names is already loaded and has columns column1 and column2
# Replace column2 in value_added_names with the new names
```

```
value_added_names$column2 <- ifelse(value_added_names$column2 %in%
new_names$x1,
                              new_names$x2[match(value_added_names$column2,
new_names$x1)],
                              value_added_names$column2)

# Rename column 1 to "country" and column 2 to "code_sector"
names(value_added_names)[3] <- "country"
names(value_added_names)[4] <- "code_sector"

value_added_names <- data.frame(value_added_names)

# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Merge the filtered dataframes
merged_data_new <- merge(value_added_names, df_na, by.x = "code_sector", by.y =
"name", all.x = TRUE)

# Filter by country and sector
merged_data_new <- merged_data_new[order(merged_data_new$country,
merged_data_new$code_sector), ]

# Replace NA with 0 in the shares_CE column
merged_data_new$shares_CE[is.na(merged_data_new$shares_CE)] <- 0

# Visualize the result
print(merged_data_new)

# Convert shares_CE and X2 columns to numeric
merged_data_new$shares_CE <- as.numeric(merged_data_new$shares_CE)
merged_data_new$W <- as.numeric(merged_data_new$W)

# Multiply shares_CE by W only if country is "ITA", otherwise assign 0
merged_data_new$W_CE <- ifelse(merged_data_new$country == "ITA",
merged_data_new$shares_CE * merged_data_new$W, 0)

# Viewing the updated data frame
print(merged_data_new)

# Extrair a coluna W_CE de merged_data
W_CE_column <- merged_data_new$W_CE
```

```
# Converte W_CE into a matrix
W_CE_matrix <- as.matrix(W_CE_column)

V_CE = W_CE_matrix / X_column # share of value-added in the total of production

# 5) Diagonalize the value-added in CE sectors.

# Convert V_CE in matrix
V_CE = data.matrix(V_CE)

diag_V_CE = matrix(0, 3465, 3465)
diag(diag_V_CE) = V_CE # diagonal of the share of value-added in CE sectors

# 6) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
           "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
           "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
```

```
        "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
        "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
        "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
        "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
        "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
        "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

# 7)    Diagonalize the interregional export vector.

diag_inter_exp= matrix(0, 3465, 3465)
```

diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports

8) Multiply the diagonalized interregional export matrix by the Leontief matrix and the diagonalized value-added in CE in order to obtain the T matrix for CE.

T_CE = diag_V_CE %*% L %*% diag_inter_exp # T matrix for CE in Italy

T_CE_ = data.frame(T_CE)
write_xlsx (T_CE_, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_CE_.xlsx")

9) Sum the columns relating to Italy's transaction with itself to obtain the DVA from the T matrix for CE.

Define the indices of the rows and columns you want to sum
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# Extract the submatrix containing only the desired rows and columns
submatrix <- T_CE[start_index:end_index, start_index:end_index]

# Sum of intraregional flows for columns, DVA for CE in Italy
intra_CE_ITA <- colSums(submatrix)

10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <- industries_and_services %>%
 separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
 mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
 mutate_if(is.character, ~na_if(., ".."))

```
# Substituir ".." por 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
  "9" = "09",
  "10" = "10T12",
  "11" = "10T12",
  "12" = "10T12",
  "13" = "13T15",
  "14" = "13T15",
  "15" = "13T15",
  "16" = "16",
```

```
"17" = "17T18",
"18" = "17T18",
"19" = "19",
"20" = "20",
"21" = "21",
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
"59" = "58T60",
"60" = "58T60",
"61" = "61",
"62" = "62T63",
"63" = "62T63",
"64" = "64T66",
```

```
  "65" = "64T66",
  "66" = "64T66",
  "68" = "68",
  "69" = "69T75",
  "70" = "69T75",
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
    name %in% c("3") ~ "03",
    name %in% c("5", "6") ~ "05T06",
    name %in% c("7", "8") ~ "07T08",
    name %in% c("9") ~ "09",
    name %in% c("10", "11", "12") ~ "10T12",
    name %in% c("13", "14", "15") ~ "13T15",
```

```
  name %in% c("16") ~ "16",
  name %in% c("17", "18") ~ "17T18",
  name %in% c("19") ~ "19",
  name %in% c("20") ~ "20",
  name %in% c("21") ~ "21",
  name %in% c("22") ~ "22",
  name %in% c("23") ~ "23",
  name %in% c("24") ~ "24",
  name %in% c("25") ~ "25",
  name %in% c("26") ~ "26",
  name %in% c("27") ~ "27",
  name %in% c("28") ~ "28",
  name %in% c("29") ~ "29",
  name %in% c("30") ~ "30",
  name %in% c("31", "32", "33") ~ "31T33",
  name %in% c("35") ~ "35",
  name %in% c("36", "37", "38", "39") ~ "36T39",
  name %in% c("41", "42", "43") ~ "41T43",
  name %in% c("45", "46", "47") ~ "45T47",
  name %in% c("49") ~ "49",
  name %in% c("50") ~ "50",
  name %in% c("51") ~ "51",
  name %in% c("52") ~ "52",
  name %in% c("53") ~ "53",
  name %in% c("55", "56") ~ "55T56",
  name %in% c("58", "59", "60") ~ "58T60",
  name %in% c("61") ~ "61",
  name %in% c("62", "63") ~ "62T63",
  name %in% c("64", "65", "66") ~ "64T66",
  name %in% c("68") ~ "68",
  name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
  name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
  name %in% c("84") ~ "84",
  name %in% c("85") ~ "85",
  name %in% c("86", "87", "88") ~ "86T88",
  name %in% c("90", "91", "92", "93") ~ "90T93",
  name %in% c("94", "95", "96") ~ "94T96",
  name %in% c("97", "98") ~ "97T98",
  TRUE ~ as.character(name)
)) %>%
group_by(code) %>%
summarize(across(where(is.numeric), sum, na.rm = TRUE))
```

```
# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
```

```
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11)  Multiply the regional shares of employment to obtain the regional DVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
intra_CE_ITA_vector <- as.numeric(intra_CE_ITA)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(intra_CE_ITA_vector) == nrow(combined_shares))

  # Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from
`intra_CE_ITA`
  combined_DVA_multiplied <- combined_shares
combined_DVA_multiplied[, 2:22] <- combined_shares[, 2:22] * intra_CE_ITA_vector

# Sum the rows of columns 2 to 22 in `combined_DVA_multiplied`
row_sums <- rowSums(combined_DVA_multiplied[, 2:22])
```

```
# Sum the columns of columns 2 to 22 in `combined_DVA_multiplied`

# Calculate the sum for each of the 21 columns to get a total value per region
inter_DVA_sum <- sapply(combined_DVA_multiplied[, 2:22], sum)

# Convert to a data frame
inter_DVA_sum <- as.data.frame(t(inter_DVA_sum))

# 9.2   Regional FVA in CE

# 1)    Divide the intermediate transactions matrix, Z, by gross output, X, in order to get technical coefficient, A.

data <- read_excel("Database_CE.xlsx", sheet = "IOM")

# Extract row and column names
row_names <- data[,1] # first column
col_names <- data[1,-1] # first row, excluding the first element

# Extract the data matrix

data.matrix <- as.matrix(data)

Z <- as.matrix(data[1:3465, 2:3466])

Y <- as.matrix(data[2:3466, 3467:3928])

num_rows <- nrow(data)

# Extract the number of rows in the data
num_rows <- nrow(data)

X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Suppose X is a matrix with a single row
X <- matrix(as.numeric(data[num_rows, -1])[1:3465], nrow = 1)

# Transpose X to get a single column matrix
X_column <- t(X)

# Display the result
print(X_column)
```

```
# Getting the inverse of Leontief

X[X == 0] = 0.000001 # enter a small number to avoid division by zero

X_d = 1 / X

diag_X = matrix(0, 3465,3465)
diag(diag_X) = X_d # diagonal of total production

A = Z %*% diag_X # Technical coefficient

# 2)   Subtract the A matrix from the identity matrix, I, and invert it to get the Leontief inverse matrix.

n = length(X) # Number of sectors by countries

I = diag(n) # Create an identity matrix

L = solve(I - A) # Leontief inverse

# 3) Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

# Path to the Excel file
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)
```

```
# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
```

```
 "84" = "84",
 "85" = "85",
 "86T88" = c("86", "87", "88"),
 "90T93" = c("90", "91", "92", "93"),
 "94T96" = c("94", "95", "96"),
 "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
 sectors <- group_mapping[[new_name]]
 filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
 summarised_data <- filtered_data %>% summarise(name = new_name, value = sum(value))
 summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
 mutate(
  shares_CE = case_when(
   name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
   name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
   name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
   name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) / sum(value[name == "20"]),
   name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name == "22"]),
   name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name == "23"]),
   name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) / sum(value[name == "24"]),
   name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
   name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
   name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891", "2892", "2896", "2899")]) / sum(value[name == "28"]),
   name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
```

```
      name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name == "30"]),
      name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316", "3317", "3319", "3320")]) / sum(value[name == "31T33"]),
      name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) / sum(value[name == "36T39"]),
      name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name == "41T43"]),
      name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) / sum(value[name == "45T47"]),
      name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name == "69T75"]),
      name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731", "7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
      name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
      name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524", "9525", "9529", "9601")]) / sum(value[name == "94T96"]),
      TRUE ~ 0
    )
  )

# 4) Obtain the value-added vector, V_X, in all sectors and countries by dividing the by gross output, X, and diagonalize it.

# Extract vectors W and X as matrices with the first 3465 elements
W <- matrix(as.numeric(data[num_rows - 1, -1])[1:3465], nrow = 1)

# Transpose W to get a single column matrix
W_column <- t(W)

V_X = W_column / X_column # share of value-added in the total of production

# Convert V_X in matrix
V_X = data.matrix(V_X)

# Diagonalize V_X
diag_V_X = matrix(0, 3465, 3465)
diag(diag_V_X) = V_X # diagonal of the share of value-added in CE sectors

# 5) Subtract the intraregional flows of the total sales to obtain the interregional exports.

# Load the data from the xlsx file
```

```
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of
that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
               "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
               "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
               "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
               "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
               "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
               "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
               "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
               "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)
```

```
# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

print(inter_exp)

# 6) Diagonalize the interregional export vector.

diag_inter_exp= matrix(0, 3465, 3465)
diag(diag_inter_exp) = inter_exp # diagonal of the interregional exports

# 7) Multiply the diagonalized matrix E by the Leontief matrix and the diagonalized value-added, for all countries and sectors, in order to obtain the T matrix.

T_all = diag_V_X %*% L %*% diag_inter_exp # T matrix for all sectors

T_all = data.frame(T_all)
write_xlsx (T_all, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/T_all.xlsx")

# 8)   Sum the columns relating to Italy's transaction with other countries to obtain the FVA from the T matrix for all sectors. First, we obtain the intraregional transactions (DVA), that is, the flows from Italy with iself, and subtract this value from the total of column.

# Define the indices of the rows and columns you want to sum
```

```
#(Italy with Italy start in row 1621 and finish in row 1665 - always check it - very important)
start_index <- 1621
end_index <- 1665

# FVA

# Calculate the sum of the columns within the specified range, ignoring NA values
sum_subset <- colSums(T_all[, start_index:end_index], na.rm = TRUE)

intra_all_ITA <- (colSums(T_all[start_index:end_index, start_index:end_index]))

print(intra_all_ITA)

# Subtract intra-regional flows from ITA from the sum
FVA_all_ITA <- sum_subset - intra_all_ITA

# Visualize the results
print(FVA_all_ITA)

# 9) Multiply the FVA for the share vector of CE to obtain the FVA in CE sectors.

# Calculate the CE sectoral shares by dividing the sectoral employment related to the CE sectors in the 4-digit NACE in relation to the sectoral employment in the 2-digit NACE, following the sectoral composition of the CE sectors.

# Read the spreadsheet
labor_4 <- read_excel("Database_CE.xlsx", sheet = "Labor_4.0")

# Print the data
print(labor_4)

# Separate the column containing names and numeric codes
labor_4_new <- labor_4 %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge")

# Remove the "[" character from the first column
labor_4_new_two <- labor_4_new %>%
  mutate(name = gsub("\\[", "", name))

# Rename the third column as "value"
labor_4_new_two <- labor_4_new_two %>%
  rename(value = 3)
```

```
# Define a list of groupings for aggregation
group_mapping <- list(
  "01T02" = c("01", "02"),
  "03" = "03",
  "05T06" = c("05", "06"),
  "07T08" = c("07", "08"),
  "09" = "09",
  "10T12" = c("10", "11", "12"),
  "13T15" = c("13", "14", "15"),
  "16" = "16",
  "17T18" = c("17", "18"),
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
  "26" = "26",
  "27" = "27",
  "28" = "28",
  "29" = "29",
  "30" = "30",
  "31T33" = c("31", "32", "33"),
  "35" = "35",
  "36T39" = c("36", "37", "38", "39"),
  "41T43" = c("41", "42", "43"),
  "45T47" = c("45", "46", "47"),
  "49" = "49",
  "50" = "50",
  "51" = "51",
  "52" = "52",
  "53" = "53",
  "55T56" = c("55", "56"),
  "58T60" = c("58", "59", "60"),
  "61" = "61",
  "62T63" = c("62", "63"),
  "64T66" = c("64", "65", "66"),
  "68" = "68",
  "69T75" = c("69", "70", "71", "72", "73", "74", "75"),
  "77T82" = c("77", "78", "79", "80", "81", "82"),
```

```
  "84" = "84",
  "85" = "85",
  "86T88" = c("86", "87", "88"),
  "90T93" = c("90", "91", "92", "93"),
  "94T96" = c("94", "95", "96"),
  "97T98" = c("97", "98")
)

# Create a new data frame with the aggregated values
agg_sectors <- do.call(rbind, lapply(names(group_mapping), function(new_name) {
  sectors <- group_mapping[[new_name]]
  filtered_data <- labor_4_new_two %>% filter(name %in% sectors)
  summarised_data <- filtered_data %>% summarise(name = new_name, value =
sum(value))
  summarised_data
}))

# Combine with original data if needed
agg_sectors <- bind_rows(agg_sectors, labor_4_new_two)

# Shares_CE

df <- agg_sectors %>%
  mutate(
    shares_CE = case_when(
      name == "07T08" ~ sum(value[name %in% c("0812")]) / sum(value[name == "07T08"]),
      name == "09" ~ sum(value[name %in% c("09")]) / sum(value[name == "09"]),
      name == "17T18" ~ sum(value[name %in% c("171")]) / sum(value[name == "17T18"]),
      name == "20" ~ sum(value[name %in% c("2011", "2013", "2016", "2059")]) /
sum(value[name == "20"]),
      name == "22" ~ sum(value[name %in% c("2211", "2219", "2229")]) / sum(value[name ==
"22"]),
      name == "23" ~ sum(value[name %in% c("2313", "2352", "2361")]) / sum(value[name ==
"23"]),
      name == "24" ~ sum(value[name %in% c("2410", "2442", "2444", "2445", "2451")]) /
sum(value[name == "24"]),
      name == "25" ~ sum(value[name %in% c("2599")]) / sum(value[name == "25"]),
      name == "26" ~ sum(value[name %in% c("2651")]) / sum(value[name == "26"]),
      name == "28" ~ sum(value[name %in% c("2813", "2822", "2829", "2841", "2849", "2891",
"2892", "2896", "2899")]) / sum(value[name == "28"]),
      name == "29" ~ sum(value[name %in% c("2910", "2920")]) / sum(value[name == "29"]),
```

```
    name == "30" ~ sum(value[name %in% c("3011", "3020", "3030")]) / sum(value[name == "30"]),
    name == "31T33" ~ sum(value[name %in% c("3311", "3312", "3313", "3314", "3315", "3316", "3317", "3319", "3320")]) / sum(value[name == "31T33"]),
    name == "36T39" ~ sum(value[name %in% c("37", "381", "382", "383", "39")]) / sum(value[name == "36T39"]),
    name == "41T43" ~ sum(value[name %in% c("4221", "4299", "4322")]) / sum(value[name == "41T43"]),
    name == "45T47" ~ sum(value[name %in% c("4520", "4677", "4779")]) / sum(value[name == "45T47"]),
    name == "69T75" ~ sum(value[name %in% c("7111", "7112")]) / sum(value[name == "69T75"]),
    name == "77T82" ~ sum(value[name %in% c("7711", "7712", "7721", "7722", "7729", "7731", "7732", "7733", "7734", "7735", "7739")]) / sum(value[name == "77T82"]),
    name == "90T93" ~ sum(value[name %in% c("9101")]) / sum(value[name == "90T93"]),
    name == "94T96" ~ sum(value[name %in% c("9511", "9512", "9521", "9522", "9523", "9524", "9525", "9529", "9601")]) / sum(value[name == "94T96"]),
    TRUE ~ 0
   )
  )


# Filter the lines where the "code" column is NA in df
df_na <- df[is.na(df$code), ]

# Convert shares_CE to numeric
shares_CE_ita <- as.numeric(df_na$shares_CE)

# Multiply shares_CE by the vector FVA_all_ITA
FVA_CE <- df_na$shares_CE * FVA_all_ITA

print(FVA_CE)

# 10) Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")
```

```
# Separate the column containing names and numeric codes
labor_2_new_two <-  industries_and_services %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all colums
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
```

```
"9" = "09",
"10" = "10T12",
"11" = "10T12",
"12" = "10T12",
"13" = "13T15",
"14" = "13T15",
"15" = "13T15",
"16" = "16",
"17" = "17T18",
"18" = "17T18",
"19" = "19",
"20" = "20",
"21" = "21",
"22" = "22",
"23" = "23",
"24" = "24",
"25" = "25",
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
```

```
  "56" = "55T56",
  "58" = "58T60",
  "59" = "58T60",
  "60" = "58T60",
  "61" = "61",
  "62" = "62T63",
  "63" = "62T63",
  "64" = "64T66",
  "65" = "64T66",
  "66" = "64T66",
  "68" = "68",
  "69" = "69T75",
  "70" = "69T75",
  "71" = "69T75",
  "72" = "69T75",
  "73" = "69T75",
  "74" = "69T75",
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
```

```
mutate(code = case_when(
  name %in% c("1", "2") ~ "01T02",
  name %in% c("3") ~ "03",
  name %in% c("5", "6") ~ "05T06",
  name %in% c("7", "8") ~ "07T08",
  name %in% c("9") ~ "09",
  name %in% c("10", "11", "12") ~ "10T12",
  name %in% c("13", "14", "15") ~ "13T15",
  name %in% c("16") ~ "16",
  name %in% c("17", "18") ~ "17T18",
  name %in% c("19") ~ "19",
  name %in% c("20") ~ "20",
  name %in% c("21") ~ "21",
  name %in% c("22") ~ "22",
  name %in% c("23") ~ "23",
  name %in% c("24") ~ "24",
  name %in% c("25") ~ "25",
  name %in% c("26") ~ "26",
  name %in% c("27") ~ "27",
  name %in% c("28") ~ "28",
  name %in% c("29") ~ "29",
  name %in% c("30") ~ "30",
  name %in% c("31", "32", "33") ~ "31T33",
  name %in% c("35") ~ "35",
  name %in% c("36", "37", "38", "39") ~ "36T39",
  name %in% c("41", "42", "43") ~ "41T43",
  name %in% c("45", "46", "47") ~ "45T47",
  name %in% c("49") ~ "49",
  name %in% c("50") ~ "50",
  name %in% c("51") ~ "51",
  name %in% c("52") ~ "52",
  name %in% c("53") ~ "53",
  name %in% c("55", "56") ~ "55T56",
  name %in% c("58", "59", "60") ~ "58T60",
  name %in% c("61") ~ "61",
  name %in% c("62", "63") ~ "62T63",
  name %in% c("64", "65", "66") ~ "64T66",
  name %in% c("68") ~ "68",
  name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
  name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
  name %in% c("84") ~ "84",
  name %in% c("85") ~ "85",
```

```
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
```

```
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))

# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 11)  Multiply the regional shares of employment to obtain the regional FVA of CE
sectors embodied in exports by NUTS2 and sector.

# Ensure `intra_CE_ITA` is a numeric vector
FVA_CE <- as.numeric(FVA_CE)

# Check if the length of `intra_CE_ITA` matches the number of rows in
`combined_shares_final`
if (length(FVA_CE) == nrow(combined_shares))
```

```
  # Multiply columns 2 to 22 of `combined_shares_final` by the corresponding value from `intra_CE_ITA`
  combined_FVA_multiplied <- combined_shares
combined_FVA_multiplied[, 2:22] <- combined_shares[, 2:22] * FVA_CE

# Sum the rows of columns 2 to 22 in `combined_FVA_multiplied`
row_sums <- rowSums(combined_FVA_multiplied[, 2:22])

# Calculate the sum for each of the 21 columns to get a total value per region
inter_FVA_sum <- sapply(combined_FVA_multiplied[, 2:22], sum)

# Convert to a data frame
inter_FVA_sum <- as.data.frame(t(inter_FVA_sum))

# 9.3   Regional exports

# 1)    Calculate the percentage of employment in each NACE 2-digit sector in the country's total for the year 2018 to obtain the regional shares.

# Path to the Excel file
industries_and_services <- read_excel("Database_CE.xlsx", sheet = "Industries and services")

general <- read_excel("Database_CE.xlsx", sheet = "General employment")

# Separate the column containing names and numeric codes
labor_2_new_two <-  industries_and_services %>%
  separate(col = 1, into = c("name", "code"), sep = "\\]", extra = "merge") %>%
  mutate(name = gsub("\\[", "", name))

# Replace ".." with NA in character columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_if(is.character, ~na_if(., ".."))

# Substituir ".." por 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Converter all columns to numeric
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)
```

```
# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Replace ".." with 0 in all columns
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(~ifelse(. == "..", 0, .))

# Convert all columns to numeric type
labor_2_new_two <- labor_2_new_two %>%
  mutate_all(as.numeric)

# Define the mapping of sectors to aggregated sectors
sector_aggregation_mapping <- c(
  "1" = "01T02",
  "2" = "01T02",
  "3" = "03",
  "5" = "05T06",
  "6" = "05T06",
  "7" = "07T08",
  "8" = "07T08",
  "9" = "09",
  "10" = "10T12",
  "11" = "10T12",
  "12" = "10T12",
  "13" = "13T15",
  "14" = "13T15",
  "15" = "13T15",
  "16" = "16",
  "17" = "17T18",
  "18" = "17T18",
  "19" = "19",
  "20" = "20",
  "21" = "21",
  "22" = "22",
  "23" = "23",
  "24" = "24",
  "25" = "25",
```

```
"26" = "26",
"27" = "27",
"28" = "28",
"29" = "29",
"30" = "30",
"31" = "31T33",
"32" = "31T33",
"33" = "31T33",
"35" = "35",
"36" = "36T39",
"37" = "36T39",
"38" = "36T39",
"39" = "36T39",
"41" = "41T43",
"42" = "41T43",
"43" = "41T43",
"45" = "45T47",
"46" = "45T47",
"47" = "45T47",
"49" = "49",
"50" = "50",
"51" = "51",
"52" = "52",
"53" = "53",
"55" = "55T56",
"56" = "55T56",
"58" = "58T60",
"59" = "58T60",
"60" = "58T60",
"61" = "61",
"62" = "62T63",
"63" = "62T63",
"64" = "64T66",
"65" = "64T66",
"66" = "64T66",
"68" = "68",
"69" = "69T75",
"70" = "69T75",
"71" = "69T75",
"72" = "69T75",
"73" = "69T75",
"74" = "69T75",
```

```
  "75" = "69T75",
  "77" = "77T82",
  "78" = "77T82",
  "79" = "77T82",
  "80" = "77T82",
  "81" = "77T82",
  "82" = "77T82",
  "84" = "84",
  "85" = "85",
  "86" = "86T88",
  "87" = "86T88",
  "88" = "86T88",
  "90" = "90T93",
  "91" = "90T93",
  "92" = "90T93",
  "93" = "90T93",
  "94" = "94T96",
  "95" = "94T96",
  "96" = "94T96",
  "97" = "97T98",
  "98" = "97T98"
)

# Perform sectoral aggregation for all NUTS2 regions
agg_sectors_regional <- labor_2_new_two %>%
  mutate(code = case_when(
    name %in% c("1", "2") ~ "01T02",
    name %in% c("3") ~ "03",
    name %in% c("5", "6") ~ "05T06",
    name %in% c("7", "8") ~ "07T08",
    name %in% c("9") ~ "09",
    name %in% c("10", "11", "12") ~ "10T12",
    name %in% c("13", "14", "15") ~ "13T15",
    name %in% c("16") ~ "16",
    name %in% c("17", "18") ~ "17T18",
    name %in% c("19") ~ "19",
    name %in% c("20") ~ "20",
    name %in% c("21") ~ "21",
    name %in% c("22") ~ "22",
    name %in% c("23") ~ "23",
    name %in% c("24") ~ "24",
    name %in% c("25") ~ "25",
```

```
    name %in% c("26") ~ "26",
    name %in% c("27") ~ "27",
    name %in% c("28") ~ "28",
    name %in% c("29") ~ "29",
    name %in% c("30") ~ "30",
    name %in% c("31", "32", "33") ~ "31T33",
    name %in% c("35") ~ "35",
    name %in% c("36", "37", "38", "39") ~ "36T39",
    name %in% c("41", "42", "43") ~ "41T43",
    name %in% c("45", "46", "47") ~ "45T47",
    name %in% c("49") ~ "49",
    name %in% c("50") ~ "50",
    name %in% c("51") ~ "51",
    name %in% c("52") ~ "52",
    name %in% c("53") ~ "53",
    name %in% c("55", "56") ~ "55T56",
    name %in% c("58", "59", "60") ~ "58T60",
    name %in% c("61") ~ "61",
    name %in% c("62", "63") ~ "62T63",
    name %in% c("64", "65", "66") ~ "64T66",
    name %in% c("68") ~ "68",
    name %in% c("69", "70", "71", "72", "73", "74", "75") ~ "69T75",
    name %in% c("77", "78", "79", "80", "81", "82") ~ "77T82",
    name %in% c("84") ~ "84",
    name %in% c("85") ~ "85",
    name %in% c("86", "87", "88") ~ "86T88",
    name %in% c("90", "91", "92", "93") ~ "90T93",
    name %in% c("94", "95", "96") ~ "94T96",
    name %in% c("97", "98") ~ "97T98",
    TRUE ~ as.character(name)
  )) %>%
  group_by(code) %>%
  summarize(across(where(is.numeric), sum, na.rm = TRUE))

# Rename the first column as "first_column"
colnames(general)[1] <- "first_column"

# Split the first column by ":"
general <- general %>%
  separate(col = first_column, into = c("name", "code"), sep = ":", extra = "merge")

# Delete columns 2 and 3 from general
```

```
general <- general %>%
  select(-2, -3)

# Rename the first column of general to "code"
general <- general %>%
  rename(code = 1)

# Create a line and rename the code to "01T02"
new_row1 <- general %>%
  filter(row_number() == 3) %>%
  mutate(code = "01T02")

# Create a line and rename the code to "03"
new_row2 <- general %>%
  filter(row_number() == 4) %>%
  mutate(code = "03")

# Create a line and rename the code to "84"
new_row40 <- general %>%
  filter(row_number() == 36) %>%
  mutate(code = "84")

# Create a line and rename the code to "97T98"
new_row45 <- general %>%
  filter(row_number() == 42) %>%
  mutate(code = "97T98")

# Delete columns 2, 3, and 4 from agg_sectors_regional
agg_sectors_regional <- agg_sectors_regional %>%
  select(-2, -3, -4)

# Insert the new rows
combined <- agg_sectors_regional %>%
  add_row(!!!new_row1, .before = 1) %>%
  add_row(!!!new_row2, .before = 2) %>%
  add_row(!!!new_row40, .before = 40) %>%
  add_row(!!!new_row45, .after = 44)

# Filter out rows where code is NA
combined <- combined %>%
  filter(!is.na(code))
```

```
# Calculate total for each row starting from the 2nd column
total_row <- rowSums(combined[, 2:ncol(combined)], na.rm = TRUE)

print(total_row)

total_row_df <- as.data.frame(total_row)
print(total_row_df)

# Divide each cell in combined starting from the 2nd column by total_row
combined_shares <- combined %>%
  mutate(across(2:ncol(combined), ~ . / total_row_df))

total_rows_check <- rowSums(combined_shares[, 2:ncol(combined_shares)], na.rm =
TRUE)

# 2)  Subtract the intraregional flows of the total sales to obtain the interregional
exports and split it just for Italy.

# Load the data from the xlsx file
dataraw <- read.xlsx("Database_CE.xlsx")

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Define a function to calculate row sums by country, considering only the columns of
that country
calculate_row_sums_by_country <- function(matrix, country_codes) {
  # Filter rows and columns based on the country code
  rows <- grep(paste0("^", country_codes), rownames(matrix))
  cols <- grep(paste0("^", country_codes), colnames(matrix))

  # Calculate row sums for the filtered subset of rows and columns
  row_sums <- rowSums(matrix[rows, cols, drop = FALSE])
  return(row_sums)
}

# List to store the row sums
row_sums_list <- list()

# Countries
countries <- c("ARG", "AUS", "AUT", "BEL", "BGD", "BGR", "BLR", "BRA", "BRN",
```

```
        "CAN", "CHE", "CHL", "CHN", "CIV", "CMR", "COL", "CRI", "CYP",
        "CZE", "DEU", "DNK", "EGY", "ESP", "EST", "FIN", "FRA", "GBR",
        "GRC", "HKG", "HRV", "HUN", "IDN", "IND", "IRL", "ISL", "ISR",
        "ITA", "JOR", "JPN", "KAZ", "KHM", "KOR", "LAO", "LTU", "LUX",
        "LVA", "MAR", "MEX", "MLT", "MMR", "MYS", "NGA", "NLD", "NOR",
        "NZL", "PAK", "PER", "PHL", "POL", "PRT", "ROU", "RUS", "SAU",
        "SEN", "SGP", "SVK", "SVN", "SWE", "THA", "TUN", "TUR", "TWN",
        "UKR", "USA", "VNM", "ZAF", "ROW")

# Calculate and store the row sums for each country
for (country in countries) {
  row_sums_list[[country]] <- calculate_row_sums_by_country(data, country)
}

# Combine the row sums into a single vector
result_vector <- unlist(row_sums_list)

# Convert the vector into a matrix of dimension 3465 by 1
result_matrix <- matrix(result_vector, ncol = 1)

# Display the resulting matrix
print(result_matrix)

# Change the row names
rownames(dataraw) <- dataraw[,1] # first column
data <- subset(dataraw, select = -V1)

# Convert data to a matrix excluding the last column and last three rows
data_matrix <- as.matrix(data[1:(nrow(data)-3), -ncol(data), drop = FALSE])

# Sum horizontally
total_sales <- rowSums(data_matrix)

# Display the resulting row sums
print(total_sales)

# Interregional exports

# Subtrair total_sales de result_vector
inter_exp <- total_sales - result_vector

print(inter_exp)
```

```
# Exports for Italy

inter_exp <- as.matrix(inter_exp)
exp_subset <- inter_exp[1621:1665,1]

# 3)   Multiply the regional shares by the regional exports vector for Italy.

combined_shares_1 <- combined_shares

inter_exports <- combined_shares_1
inter_exports[, 2:22] <- combined_shares[, 2:22] * exp_subset

inter_exports_sum <- colSums(inter_exports[, 2:22])

inter_exports_sum <- as.numeric(inter_exports_sum)

# Sum the rows of columns 2 to 22 in `inter_exports`
row_sums <- rowSums(inter_exports[, 2:22])

# Calculate the sum for each of the 21 columns to get a total value per region
exports_sum_c <- sapply(inter_exports[, 2:22], sum)

# Convert to a data frame
exports_sum_c <- as.data.frame(t(exports_sum_c ))

# 9.4  Overall regional participation in CE

# 1)   Sum the regional DVA and FVA and divide it by the sum of regional exports in all
sectors within each region.

combined_shares_4 <- combined_shares

overall <- combined_shares_4
# Iterate over the columns of forward and perform element-wise division
for (i in 2:22) {
 overall[, i] <- (combined_DVA_multiplied[, i] + + combined_FVA_multiplied [, i])/
inter_exports_sum[i - 1]
}

write_xlsx (overall, "C:/Users/damar/OneDrive/Documentos/Polimi Project
3/overall.xlsx")
```

```
# Iterate over the total of columns of DVA and FVA and exports to get a total value per region
overall_total_region <- (inter_DVA_sum  + inter_FVA_sum ) / exports_sum_c

# Convert to a data frame
overall_total_region <- as.data.frame(overall_total_region)

# Write to Excel file
write_xlsx(overall_total_region, "C:/Users/damar/OneDrive/Documentos/Polimi Project 3/overall_total_region.xlsx")
```

9. CIRCULAR ECONOMY INNOVATION IN SEMICONDUCTOR EQUIPMENT MANUFACTURING

9.1 Introduction

We first develop a methodology to identify innovations that refer to circular practices and processes. The unit of analysis is first the firm level, which is then aggregated at the regional level. Innovations are proxied with patent data, and the empirical analysis refers to the semiconductor industry.

9.2 Data

The central focus of this study revolves around investigating the CE innovations within the Semiconductor Equipment manufacturers located in Europe, and extracted from the Orbis database (Bureau Van Dijk). Innovation activities are proxied by patents granted to these companies within the period 2014-2023 and extracted from Orbis IP database (accessed through the University of Manchester, as the co-supervisor of Priyanshu Pathak, Professor Silvia Massini, is based there).

9.3 Indicator 1: Firm level circular innovation

Our database refers to 895 Semiconductor Equipment manufacturers located in Europe which are identified using the respective NACE/NAICS codes combined with specific keywords and extracted from the Orbis database (Bureau Van Dijk).

Type	Criteria
NAICS 2017 (All codes)	333242 - Semiconductor Machinery Manufacturing
NACE Rev. 2 (All codes)	2651 - Manufacture of instruments and appliances for measuring, testing and navigation

NACE Rev. 2 (All codes)	2670 - Manufacture of optical instruments and photographic equipment
NACE Rev. 2 (All codes)	2790 - Manufacture of other electrical equipment
NACE Rev. 2 (All codes)	2899 - Manufacture of other special-purpose machinery
NAICS 2017 (All codes)	333318 - Other Commercial and Service Industry Machinery Manufacturing
NAICS 2017 (All codes)	334515 - Instrument Manufacturing for Measuring and Testing Electricity and Electrical Signals
US SIC (Primary codes only)	3699 - Electrical machinery, equipment, and supplies, not elsewhere specified
NACE Rev. 2 (All codes)	2611 - Manufacture of electronic components
US SIC (All codes)	3674 - Semiconductors and related devices

Once identified the companies, we gathered all the information about their patents. Only 275 of the companies considered have patenting activities in the period considered. For Italian companies, 1245 patents have been identified.

We developed a methodology for identifying those patents that refer to circular innovation. Specifically, we relied on previous literature (e.g. Giglio et al., 2021; Portillo-Tarragona et al., 2022) and we used both the specific IPC and CPC codes (see table below). According to this methodology, about 30% of the patents can be defined as circular innovation patents.

Circular innovations have been further distinguished in different categories, namely "Remanufacturing", "Recovery of Resource and Energy", "Reuse of Energy and Resource", "Recycling", "Regenerating", "Repairing and Refurbish", "Refuse Management", "Resource and Energy Optimisation", and "Waste reduction and Sustainable Production".

CE Technology Classification	Related IPC/CPC codes (4 digits)
Remanufacturing	C04B; Y02W
Recovery of Resource and Energy	C02F; D01F; D21F
Reuse of Energy and Resource	B29C; B29C; C04B
Recycling	Y02W; C03B
Regenerating	B01J; H01J
Repairing and Refurbish	H01J; H01K
Refuse Management	C04B; B03B; B65F; Y02W
Resource and Energy Optimisation	Y02B
Waste reduction and Sustainable Production	Y02P; G01R; G05B

9.4 Indicator 2: Circular innovation and the regional level

Firm level patent data have been aggregated at the regional level (based on the location of the first inventor).

As far as Italian regions, 9 out of the 20 regions in Italy contribute to circular innovation in the semiconductor equipment industry, where Veneto emerges as the region with the highest number of circular patents (208), followed by Lombardia (77 patents) and Emilia-Romagna (72 patents). The table below also reports the distinct technological focus of the circular innovations across these regions.

Region In Italy	Reuse of Energy and Materials	Waste reduction and Sustainable Production	Resource and Energy Optimization	Repairing and Refurbish	Recovery of Resource and Energy	Grand Total
Veneto	172	29	4	3		208
Lombardia	2	20	37	4	4	77
Emilia-Romagna	2	70				72
Sardegna		2	17			19
Piemonte		6	7		4	17
Umbria				4		4
Trentino-Alto Adige		2				2

Puglia					1	1
Toscana					1	1
Grand Total	**176**	**129**	**65**	**11**	**10**	**394**

The analysis also allows a comparative study across European regions (with the aim of relating circular innovation with specific European and regional industrial policies).

10. ROLE OF FOREIGN COMPANIES AND FOREIGN KNOWLEDGE IN CIRCULAR INNOVATIONS

10.1 Introduction

We analyse the role of different actors involved in the circular innovation processes (e.g. companies, research centers and universities, start-ups, local and national governmental institutions), and we disentangle the role of foreign actors and foreign knowledge in fostering circular innovation (both at the firm and regional level).

10.2 Indicator 3: Identification of multiple actors in circular innovation patents

Once identified circular innovation patents, the analysis of inventors in each patent provides evidence about collaborative innovation processes. Therefore, it is possible to distinguish between different types of inventors:

- Companies
- Research centers
- Universities
- Independent inventors

Therefore circular innovation patents can be classified as:

- Intra firm: all the inventors belong to the focal company or companies from the same group (by merging the information with data from Orbis)
- Inter firms: the inventors belong to the focal company and other companies (by merging the information with data from Orbis)
- Multiactor: inventors come from at least two different typologies of different actors (companies, research centers, universities, independent)

10.3 Indicator 4: Identification of foreign companies and foreign knowledge in circular innovation patents

Among the inventors identified in the previous section, it is possible to distinguish those that are located in the same country of the focal company and those that are located in other (foreign) countries in order to assess the role of foreign companies and/or foreign knowledge in the development of circular innovation in Italian regions.

Specifically, we identify the following six categories of circular innovation patents:

	National	Cross border
Intra-company		
Inter-company		
Multi-actor		