GLEM

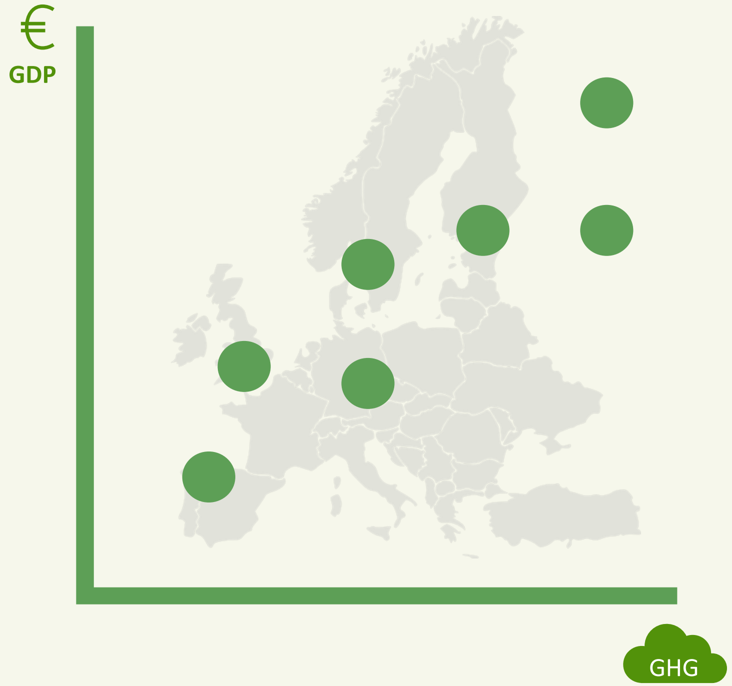
Computer Programming and Data Management

GREENHOUSE GASES EMISSIONS IN EU-27

AN EVOLUTION OVER TIME

OUR RESEARCH QUESTION

What is the relationship between GHG and GDP in the European Union?



INDEX

1. GREENHOUSE GASES: EMISSIONS ANALYSIS
2. GDPs ANALYSIS
3. GHGs VS. GDPs: A CROSS ANALYSIS
4. LINEAR REGRESSION AND CLASSIFICATION
5. CONCLUSION

1. GREENHOUSE GASES: EMISSIONS ANALYSIS

What are greenhouse gases?

Some gases in Earth’s atmosphere absorb energy from heat radiation, keeping the planet warmer than it would otherwise be.

Human activities since the industrial revolution have increased significantly the concentration of such gases, resulting in global warming and changes to Earth’s climate.

1.

Emissions Dataset

All the data on GHG emissions that was analyzed by us comes from a .csv available on the EEA website.
It’s a data frame that contains the reported emissions by European countries, according to a format defined by the IPCC that complies with the United Nations Framework Convention for Climate Change (1992).

```
file_path = "./GLEMfolder/eea_t_national-emissions-reported_p_2023_v01_r01"
emissions_df = pd.read_csv(file_path + "/UNFCCC_v26.csv")
```

European
Environment
Agency

ipcc WMO UNEP

INTERGOVERNMENTAL PANEL ON
climate change

1.

Cleaning the DataFrame

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 671236 entries, 0 to 671235
Data columns (total 13 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Country_code        671236 non-null  object
 1   Country             671236 non-null  object
 2   Format_name         671236 non-null  object
 3   Pollutant_name      671236 non-null  object
 4   Sector_code         671236 non-null  object
 5   Sector_name         671236 non-null  object
 6   Parent_sector_code  614914 non-null  object
 7   Unit                671236 non-null  object
 8   Year                671236 non-null  object
 9   emissions           404440 non-null  float64
 10  Notation            241059 non-null  object
 11  PublicationDate     671236 non-null  int64
 12  DataSource          671236 non-null  object
dtypes: float64(1), int64(1), object(11)
memory usage: 66.6+ MB
```

Each row of the data frame contains the emission value for a given country, each year, by sector, and by pollutant type, as well as some additional information.

The initial cleaning process consisted of selecting only the most relevant information. We made use of the multi-index feature to better organize the data.

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 233742 entries, ('Austria', '
Data columns (total 1 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   emissions  154770 non-null  float64
dtypes: float64(1)
memory usage: 2.7+ MB
```

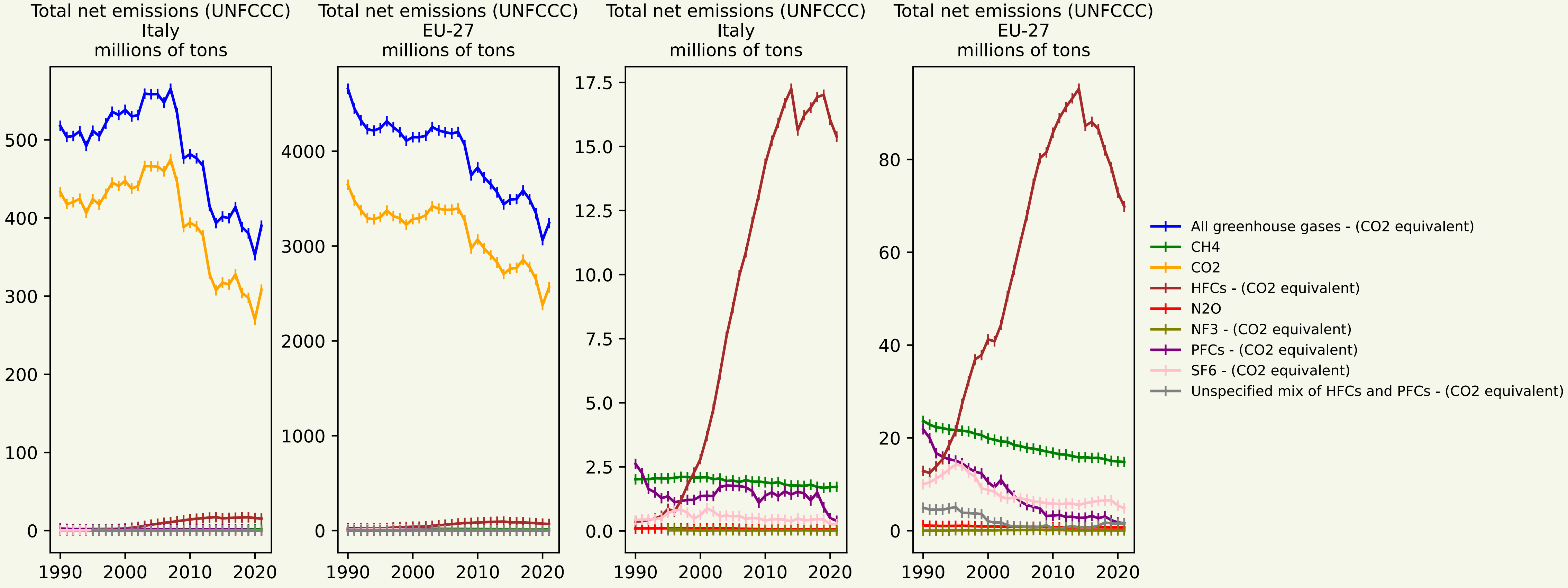
```
df_emissions_multiindex.index.names
```

```
FrozenList(['Country', 'Pollutant_name', 'Sector_name', 'Year'])
```

1. Exploratory analysis

CO_2 and other Greenhouse Gases

Carbon dioxide is by far the gas that is responsible for the highest emissions.

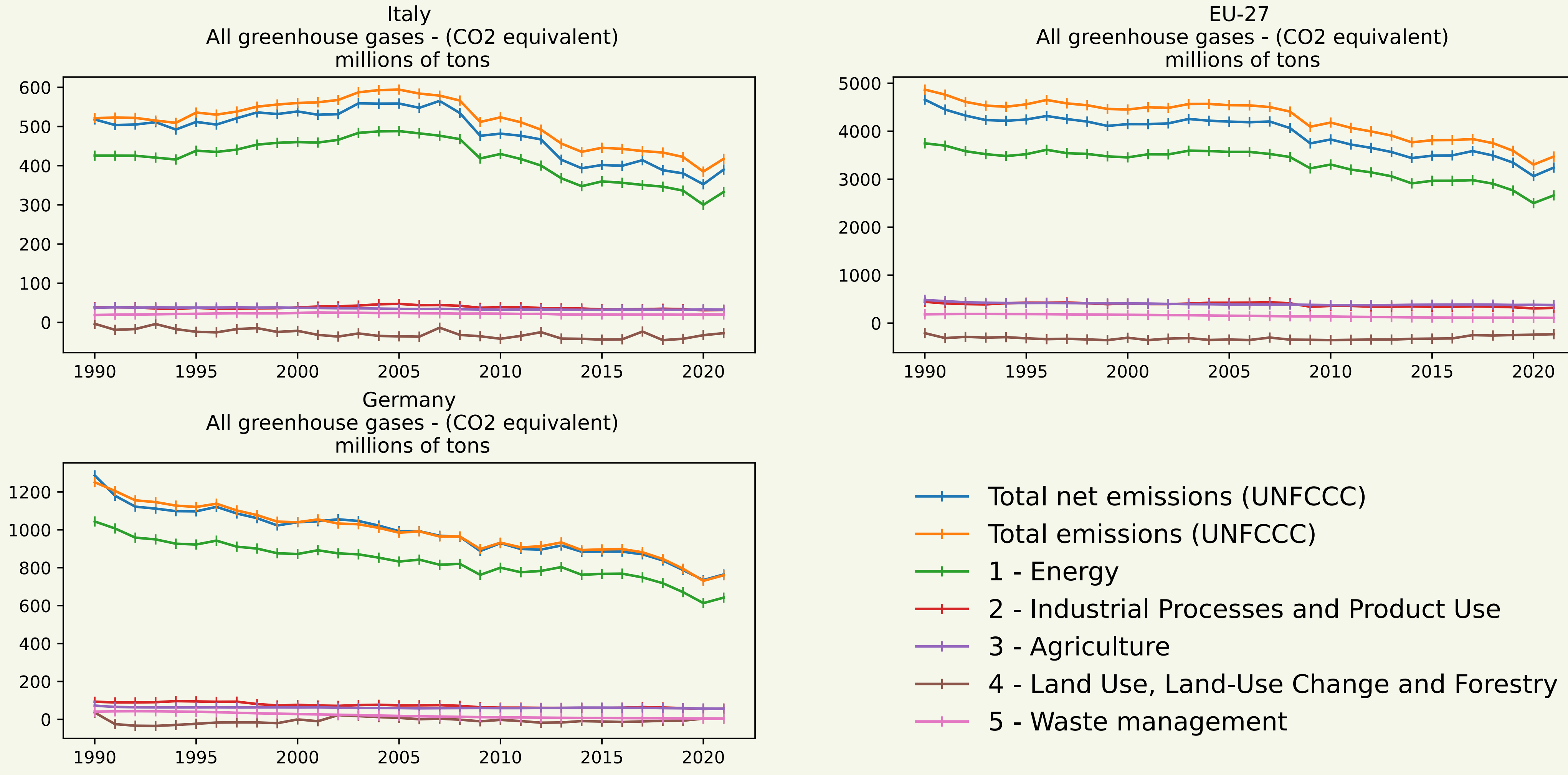


Exploratory analysis

1.

Country by country sector comparison

Emissions from energy production dictate the general trend followed by total and total net emissions.

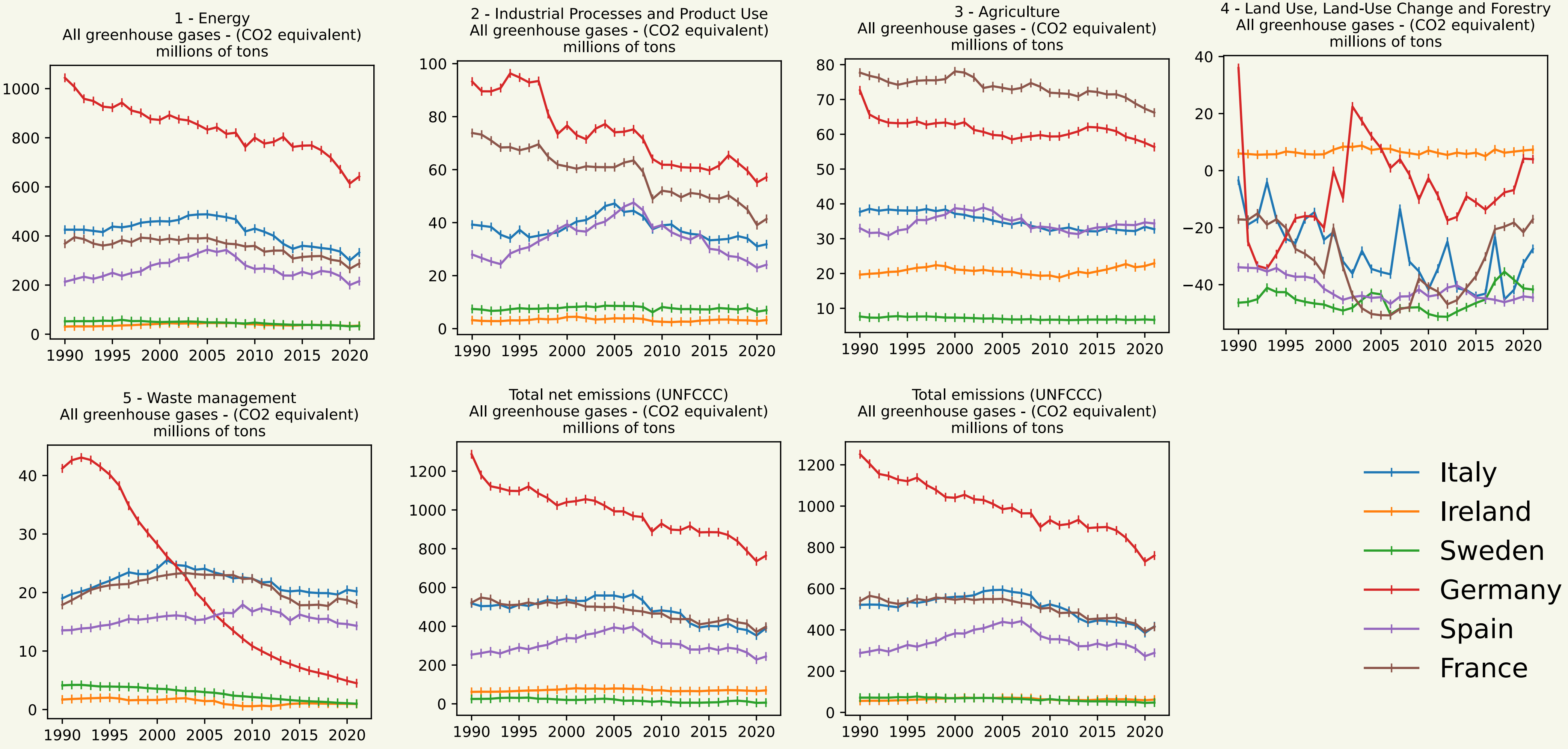


1.

Exploratory analysis

An Overview by Sectors

We plotted different countries on the same graph. It’s clear that population data are needed to have a complete picture.



1.

Merging the Datasets

We downloaded another dataset involving population, and again we performed some data cleaning operations.
Then we did a merge of the datasets concerning emissions and population to obtain per capita data.

```
Data_Pop.drop(Data_Pop[(Data_Pop['SUBJECT'] != 'TOT')].index, inplace = True)
Data_Pop.drop(Data_Pop[(Data_Pop['LOCATION'] == 'OECD')].index, inplace = True)

# Then I eliminated the columns that have uninteresting values
delete = ["INDICATOR","FREQUENCY", "Flag Codes", 'MEASURE', 'SUBJECT']
Data_Pop.drop(columns = delete, inplace = True)

# I've replaced the name of the column and the index to make the dataset more understandable
Data_Pop = Data_Pop.replace('MLN_PER','MLN')

INDEX = [x for x in range(len(Data_Pop["LOCATION"]))]
Data_Pop['INDEX'] = INDEX

Data_Pop = Data_Pop.set_index('INDEX')
```

INDEX	Year	pop_mil	Country
0	1990	7.677850	Austria
1	1991	7.754891	Austria
2	1992	7.840709	Austria
3	1993	7.905632	Austria
4	1994	7.936118	Austria
...	...	...	...
919	2018	446.457200	EU-27
920	2019	447.402405	EU-27
921	2020	447.805543	EU-27
922	2021	447.117504	EU-27
923	2022	447.949164	EU-27

```
emissions_df = emissions_df.merge(Data_Pop, on=("Country","Year"))

# Having merged the two datasets with population and emissions, we can calculate per capita emissions
emissions_df["per_capita_emission"] = emissions_df["emissions"]/(emissions_df["pop_mil"]*1000000)

# Note: the population is in millions, so we multiply the denominator by 1 million
# This way the unit of measure is Gigagram per person

# again we use the multiindex feature
df_emissions_multiindex = emissions_df.set_index(["Country","Pollutant_name", "Sector_name", "Year"]).sort_index()
```

1. Exploratory analysis

Choropleth map

This is a dynamic map showing emission trends for each country in the European Union-27, created by using *plotly* library.

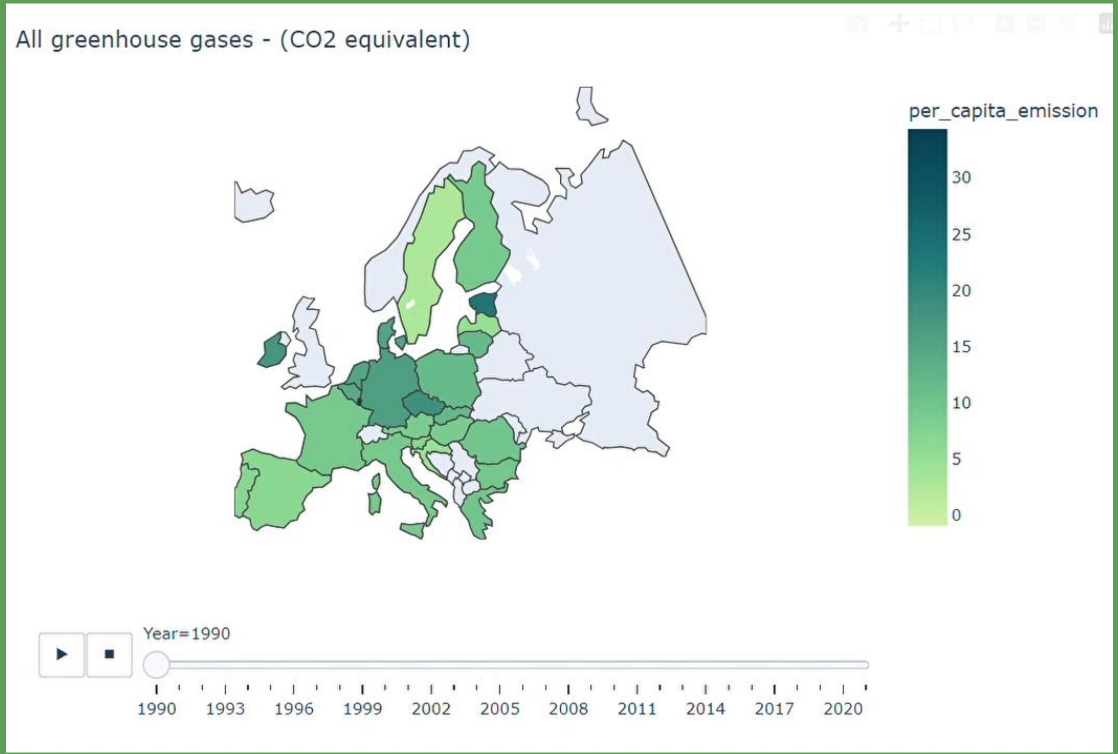
```
for pollutant in df_maps.Pollutant_name.unique():
    pollutant_data = df_maps[df_maps.Pollutant_name == pollutant]
    pollutant_data['per_capita_emission'] = pollutant_data['per_capita_emission'] * 1000

    min = pollutant_data['per_capita_emission'].min()
    max = pollutant_data['per_capita_emission'].max()

    fig = px.choropleth(pollutant_data,
                        locations = "Country_code",
                        range_color = (min, max),
                        color  ="per_capita_emission",
                        color_continuous_scale = "Emrld",
                        scope = "europe",
                        animation_frame = "Year")

    fig.update_geos(fitbounds="locations", visible=True)
    fig.update_layout(title_text = f'{pollutant}', margin={"r":10,"t":10,"l":10,"b":10})
    fig.show()

    fig.write_html(str(pollutant) + '.html')
```



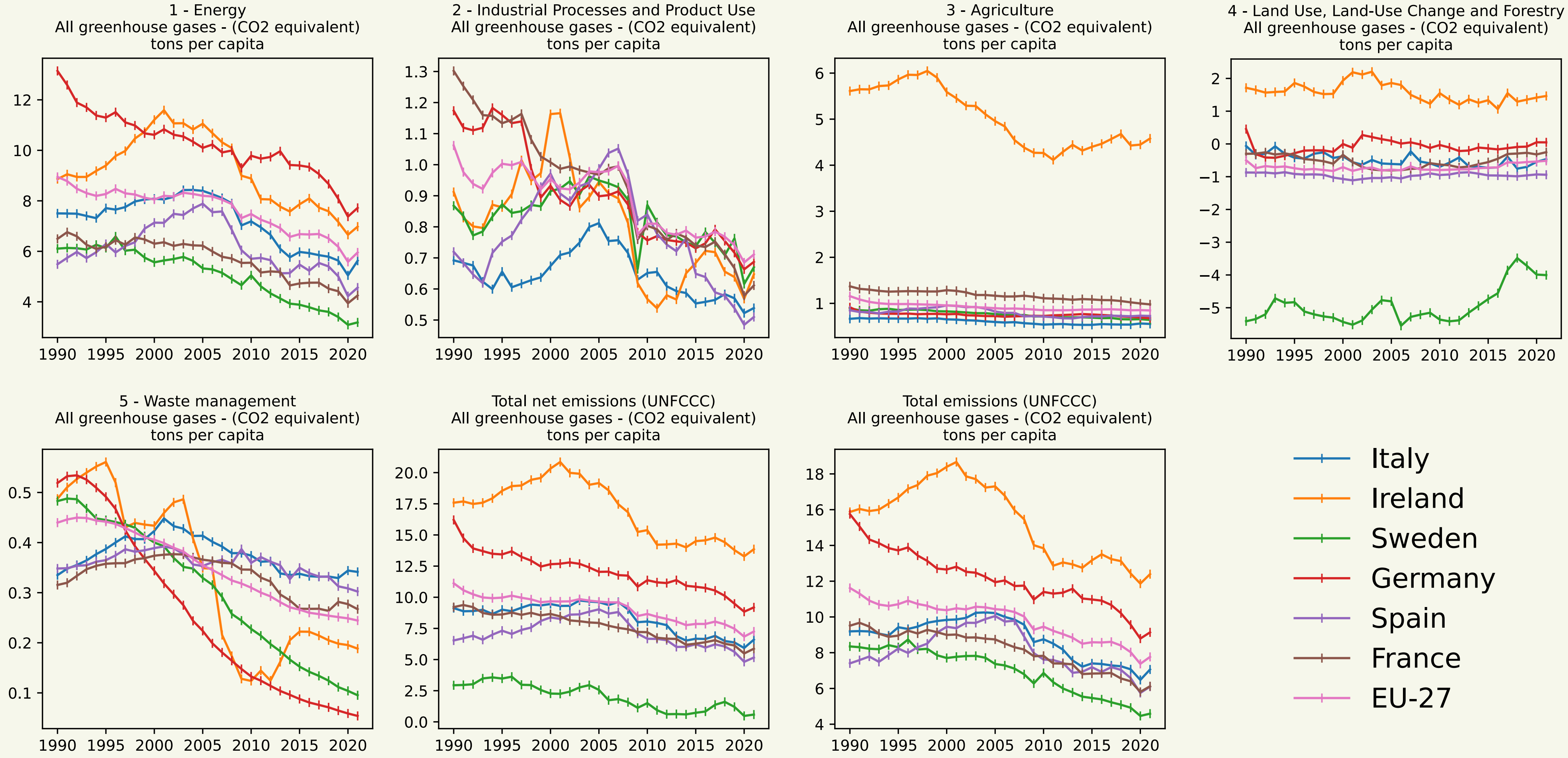
The colours lighten over time, indicating an overall increase in the level of GHGs.

Exploratory analysis

1.

GHG emissions per capita

This allows us to compare countries with different population size

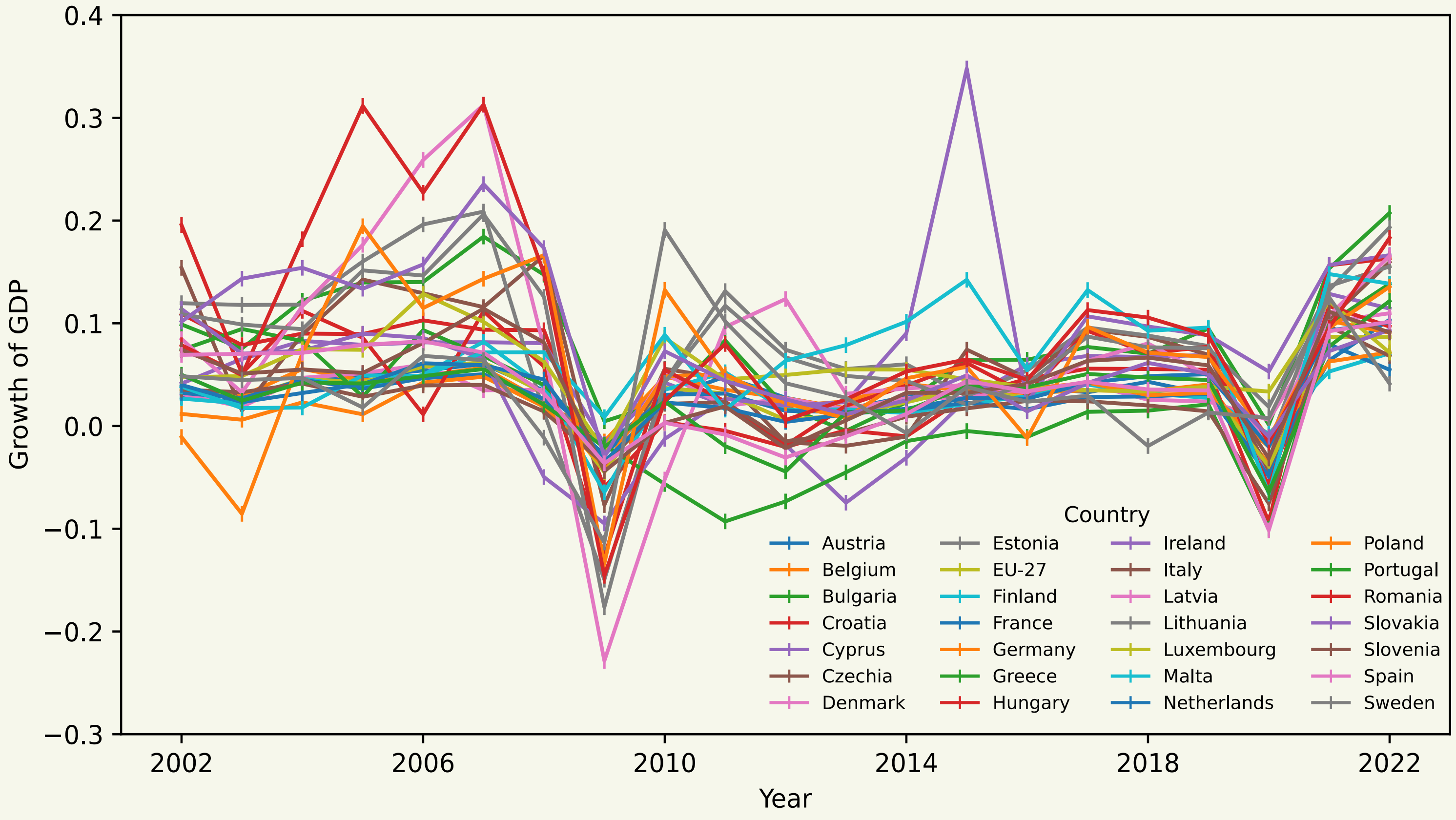


2.GDPs ANALYSIS

Taking 2002 (year of adoption of the Euro in Italy) as a reference, we analysed the development of Gross Domestic Product.

The analysis will give some general data about the evolution of GDP in various countries, starting from general values, and then breaking down the data in singular sectors.

The whole study is functional to cross-analysis of GDP and emissions data.



2.

```
urllib.request.urlretrieve("https://ec.europa.eu/eurostat/api
df_Eurostat = pd.read_csv("./GLEMfolder/df_Eurostat")
df_Eurostat
```

	geo	Country	TIME_PERIOD	OBS_VALUE	unit	freq
0	AUT	Austria	1995	184351.3	CP_MEUR	A
1	AUT	Austria	1996	186968.1	CP_MEUR	A
2	AUT	Austria	1997	187853.7	CP_MEUR	A
3	AUT	Austria	1998	195011.9	CP_MEUR	A
4	AUT	Austria	1999	203850.6	CP_MEUR	A
...	...	...	...	...	...	...
801	SVK	Slovakia	2018	89874.7	CP_MEUR	A
802	SVK	Slovakia	2019	94429.7	CP_MEUR	A
803	SVK	Slovakia	2020	93444.1	CP_MEUR	A
804	SVK	Slovakia	2021	100255.7	CP_MEUR	A
805	SVK	Slovakia	2022	109645.2	CP_MEUR	A

The Eurostat Dataset

For the general analysis we used a .csv file available in the Eurostat website.

The file contains information about the Gross Domestic Products (GDPs) of all 27 countries of the European Union, expressed in millions of euros.

2.

The Eurostat Dataset

Data cleaning

The dataset we are showing is the result of the initial cleaning process. On the right there's the original structure of the dataset.

As you can see, we had a lot of useless columns, and others was missing (e.g. the country name). The cleaning process consisted of a conversion from code to country, a drop of useless columns, a filter of useful data and a general reorder for a better visualization.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   DATAFLOW     1259 non-null   object
 1   LAST UPDATE  1259 non-null   object
 2   freq         1259 non-null   object
 3   unit         1259 non-null   object
 4   na_item      1259 non-null   object
 5   geo          1259 non-null   object
 6   TIME_PERIOD  1259 non-null   int64
 7   OBS_VALUE    1259 non-null   float64
 8   OBS_FLAG     26 non-null     object
dtypes: float64(1), int64(1), object(7)
memory usage: 88.6+ KB
```

```python
# cleaning and ordering data
df_Eurostat = df_Eurostat.drop(columns = ['OBS_FLAG', 'na_item', 'LAST UPDATE', 'DATAFLOW'])
df_Eurostat = df_Eurostat.reindex(['geo', 'Country', 'TIME_PERIOD', 'OBS_VALUE', 'unit', 'freq'] , axis=1)

# dropping some data
mask = df_Eurostat['Country'].isin(lst_of_useless_data)
df_Eurostat = df_Eurostat[~mask]
df_Eurostat = df_Eurostat.reset_index(drop = True)

# rewriting the country code in a 3 lecter format with the previous conversion dictionary
df_Eurostat['geo'] = df_Eurostat['Country'].map(dic)
df_Eurostat
```

	geo	Country	TIME_PERIOD	OBS_VALUE	unit	freq
0	AUT	Austria	1995	184351.3	CP_MEUR	A
1	AUT	Austria	1996	186968.1	CP_MEUR	A
2	AUT	Austria	1997	187853.7	CP_MEUR	A
3	AUT	Austria	1998	195011.9	CP_MEUR	A
4	AUT	Austria	1999	203850.6	CP_MEUR	A
...	...	...	...	...	...	...

2.

The Eurostat Dataset

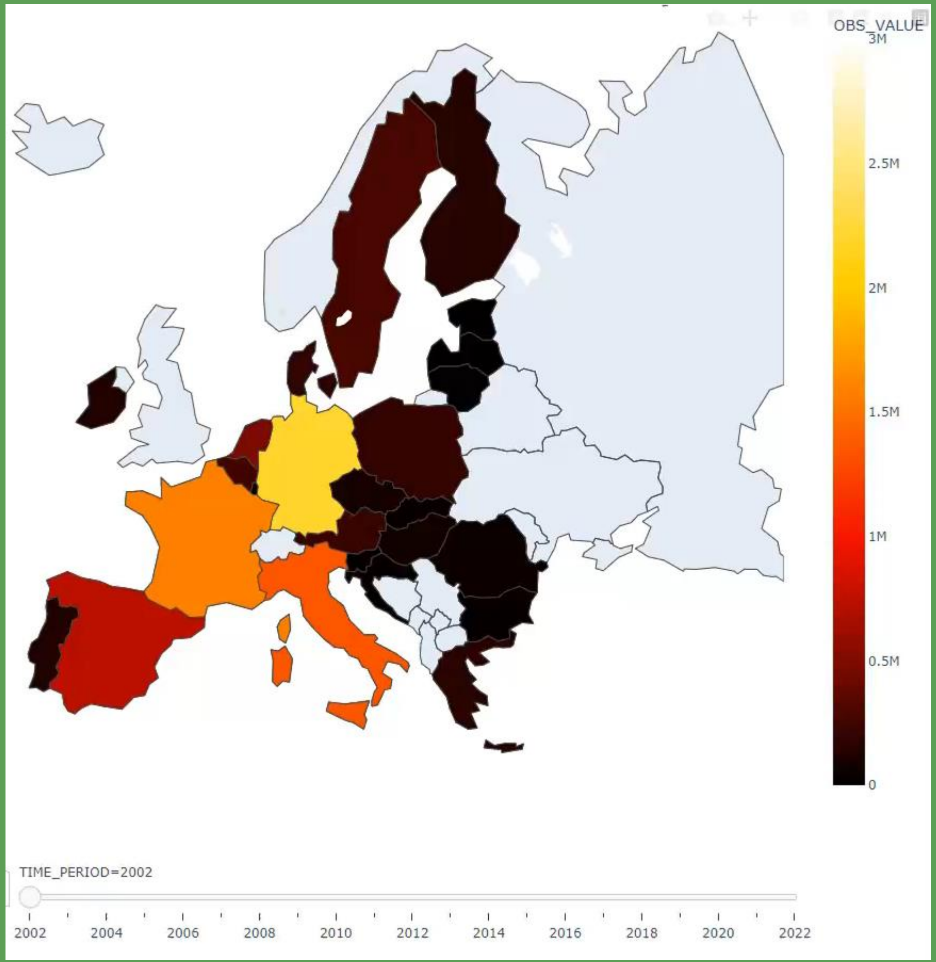
Choropleth map

Using the cleaned data, we created a Choropleth map showing the GDP’s evolution in different countries, as we did in the first paragraph.

```
mask = df_Eurostat['TIME_PERIOD'].isin(range(2002,2023))
df_to_plot = df_Eurostat[mask]

fig = px.choropleth(df_to_plot.sort_values('TIME_PERIOD'),
                    locations = "geo",
                    range_color = (0, 3000000),
                    color  ="OBS_VALUE",
                    color_continuous_scale = "Hot",
                    scope = "europe",
                    animation_frame = "TIME_PERIOD")

fig.update_geos(fitbounds="locations", visible=True)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
fig.write_html("cheroplet_GDP.html")
```



The colours lighten over time, indicating an overall increase in the level of GDPs.

The OECD Dataset

A Division by Sectors

```
# Tell pdmx we want OECD data
oecd = pdmx.Request("OECD")

# Set out everything about the request in the format specified by the OECD API
data = oecd.data(
    resource_id="SNA_TABLE1",
    key="AUT+BEL+CZE+DNK+EST+FIN+FRA+DEU+GRC+HUN+IRL+ITA+LVA+LTU+LUX+NLD+POL+PRT+SVK+SVN+ESP+SWE+EU27_2020+BGR+HRV+CYP
).to_pandas()

df_OECD = pd.DataFrame(data).reset_index()
```

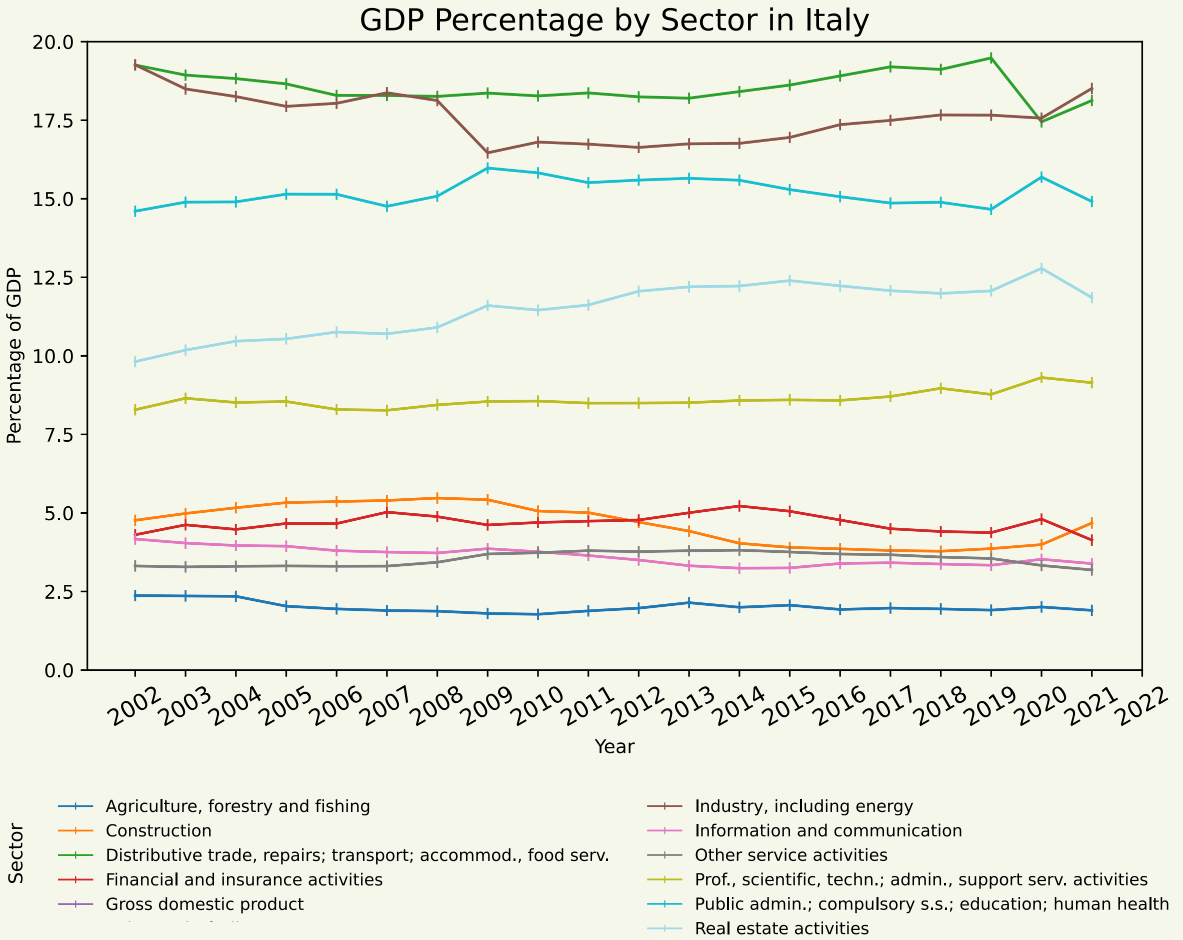
Country	Year (Sectors)	Agriculture, forestry and fishing	Construction	Distributive trade, repairs; transport; accommod., food serv.	Financial and insurance activities	Gross domestic product	Industry, including energy	Information and communication	Other service activities	Prof., scientific, techn.; admin., support serv. activities	Public admin.; compulsory s.s.; education; human health	Real estate activities
Austria	2002	3543.36	14131.37	46363.35	10225.13	226735.22	47967.05	7737.20	5569.00	15074.06	35065.61	16677.23
	2003	3459.23	15085.13	47043.26	10146.91	231862.46	48673.29	7934.08	5726.13	15706.52	36126.77	17346.36
	2004	3578.21	15602.28	48916.81	10541.95	242348.26	50737.57	7756.07	6098.38	16569.11	37329.60	18968.34
	2005	3199.82	15833.70	51439.88	11005.26	254075.03	52955.19	8134.18	6420.25	18008.69	38288.82	20602.33
	2006	3490.26	16134.83	54489.52	11912.31	267824.45	56872.43	8378.78	6604.81	19576.53	40003.16	21613.38
...	...	...	...	...	...	...	...	...	...	...	...	...
Sweden	2017	66765.00	277343.00	733143.00	168311.00	4625094.00	746115.00	293666.00	119919.00	476047.00	877225.00	342112.00
	2018	67248.00	287200.00	744416.00	170110.00	4828306.00	775145.00	324299.00	124348.00	503101.00	913718.00	369333.00
	2019	70297.00	290094.00	781180.00	172044.00	5049619.00	815960.00	365031.00	131259.00	518286.00	940782.00	399682.00
	2020	66980.00	306998.00	745228.00	196029.00	5038538.00	780857.00	376653.00	123556.00	517394.00	958946.00	399754.00
	2021	70672.00	313556.00	829879.00	213675.00	5486558.00	931485.00	418941.00	127833.00	551629.00	1013264.00	401960.00

Once we have looked at the general GDP growth for each country, we wonder which is the leading sector for GDP in each EU member.
To do this, we have to look at a dataset coming from OECD website by *pandasdmx* library.

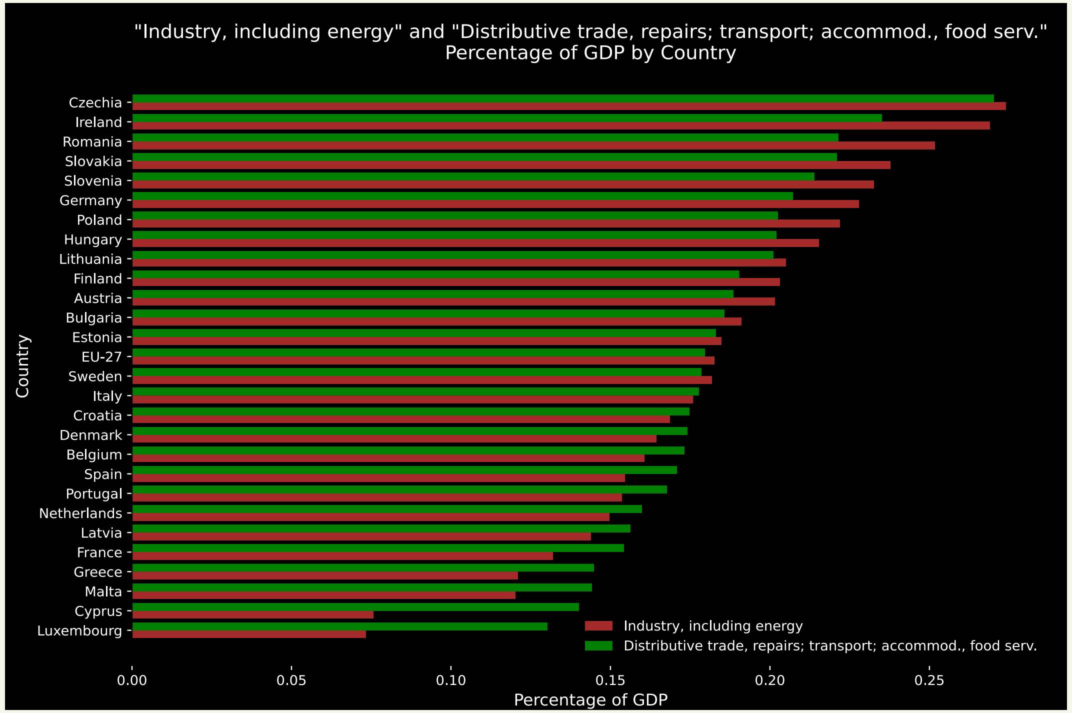
2.

The OECD Dataset

A Division by Sectors



Looking in which other countries the "*Industry, including energy*" and "*Distributive trade, repairs; transport; accommod., food serv.*" are the leading sectors over years




```
Percent_GDP.groupby("Country")["Sector_name"].mean() /
Percent_GDP.groupby("Country")["Gross domestic product"].mean()
```

GDP growth and GHG emissions

2.

```
emissions_small
# This data frame only has 1 pollutant: All greenhouse gases
# There are only the main sectors 1-5 plus total and net emissions

emissions_small["per_capita_emiss_variation"] = emissions_small.per_capita_emission.pct_change()
# we add a column that has the percentage variation in emissions from the previous year
```

```
# creating a pass-by dataframe to merge with the one above
gdp_tomerge = DataFrame(GDP_Euro.stack().swaplevel().sort_index())

gdp_tomerge.rename(columns = {0:"GDP"},inplace = True)

# merge
emissions_gdp_var = emissions_small.merge(gdp_tomerge, left_index = True, right_index = True)
emissions_gdp_var
```

```
# computing per_capita_GDP
emissions_gdp_var = emissions_gdp_var.swaplevel("Sector_name", "Year").copy()
emissions_gdp_var.sort_index(inplace = True)
emissions_gdp_var["per_capita_GDP"] = emissions_gdp_var["GDP"]/(emissions_gdp_var["pop_mil"]*1000000)

# computing the variation percentage
emissions_gdp_var["per_capita_GDP_var"] = emissions_gdp_var["per_capita_GDP"].pct_change()
```

Once computed our variables, we calculated the correlation between *per_capita_GDP_var* and *per_capita_emiss_var* for Italy and EU-27.

```
Italy_corr: 0.6010900209766104
EU-27_corr: 0.8015322339555422
```

We can finally look at GDP vs emissions.

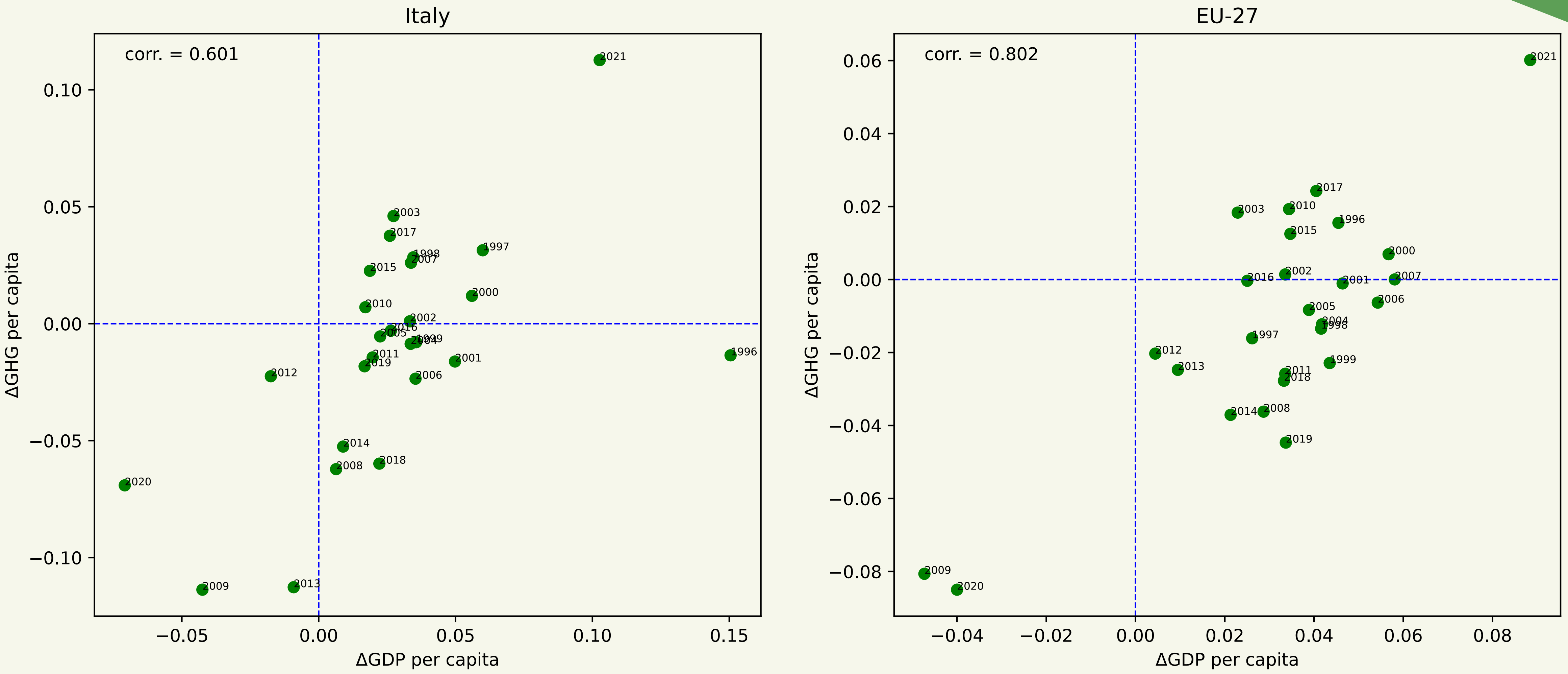
```
emissions_gdp_var.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 5096 entries, ('1 - Energy', 'Austria', 1996)
Data columns (total 7 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   emissions                   5096 non-null   float64
 1   pop_mil                     5096 non-null   float64
 2   per_capita_emission         5096 non-null   float64
 3   per_capita_emiss_variation  5096 non-null   float64
 4   GDP                         5096 non-null   float64
 5   per_capita_GDP              5096 non-null   float64
 6   per_capita_GDP_var          5096 non-null   float64
dtypes: float64(7)
memory usage: 296.6+ KB
```

2.

GDP and Net Emissions per capita

Many years fall into the fourth quadrant and some outliers are present.



3.GHG vs. GDP: A CROSS ANALYSIS

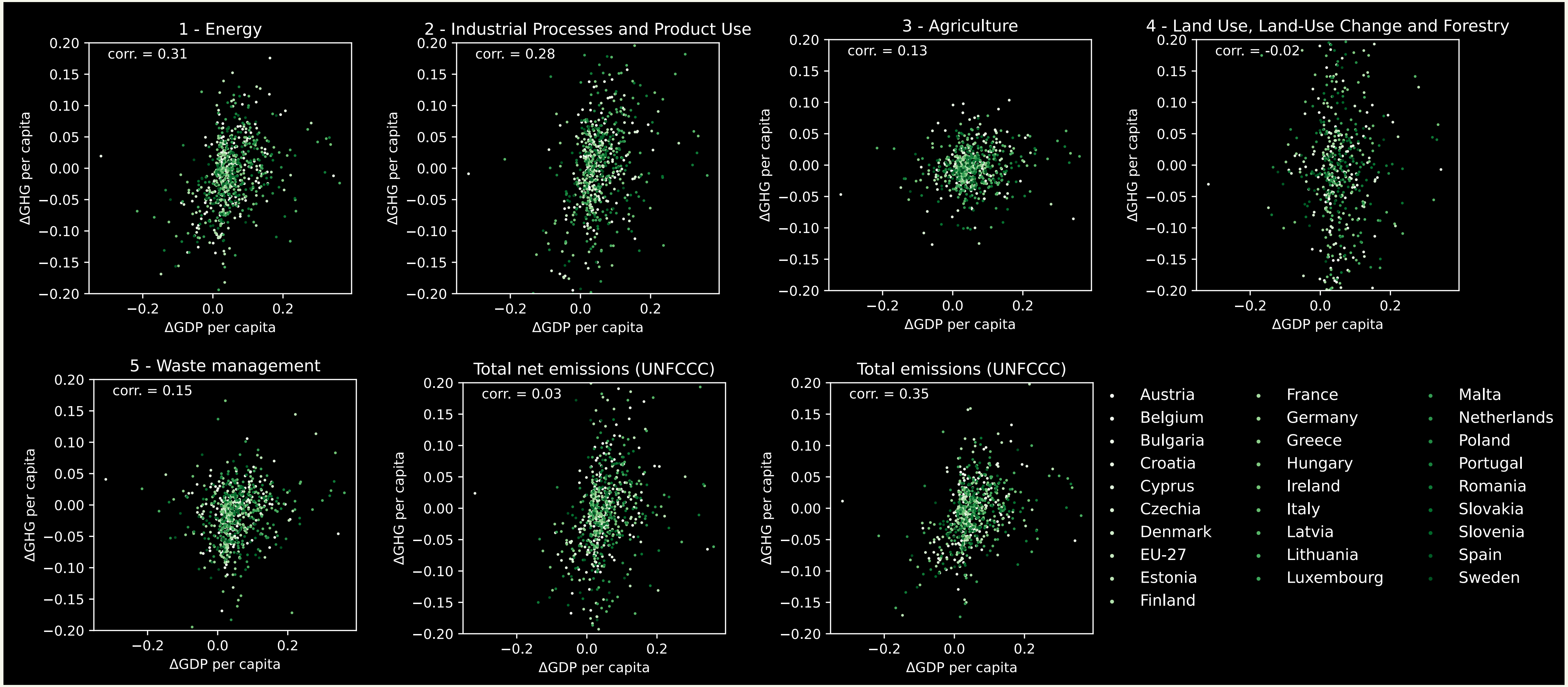
In this third section, our aim is to identify the sector with the highest Greenhouse Gas Emissions and assess its impact on the total GDP of each European Union country. To achieve this, we recall the previous merged dataset *emissions_gdp_var*.

We introduce a new package *seaborn* too, in order to implement other features in graphs.

3.

ΔGHG per capita vs. ΔGDP per capita

A Division by IPCC Sectors



ΔGHG per capita vs. ΔGDP per capita

An Overview by Countries

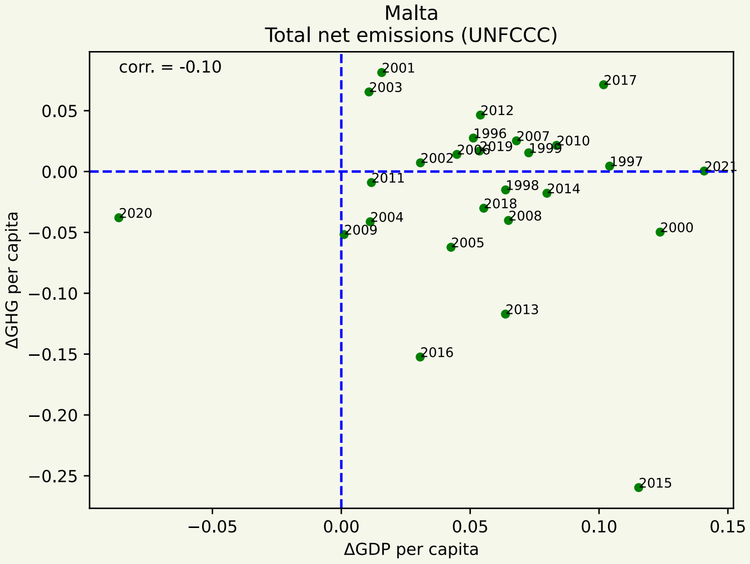
The *ΔTotal Net Emissions'* correlation with the *ΔGDP* is tending to zero.

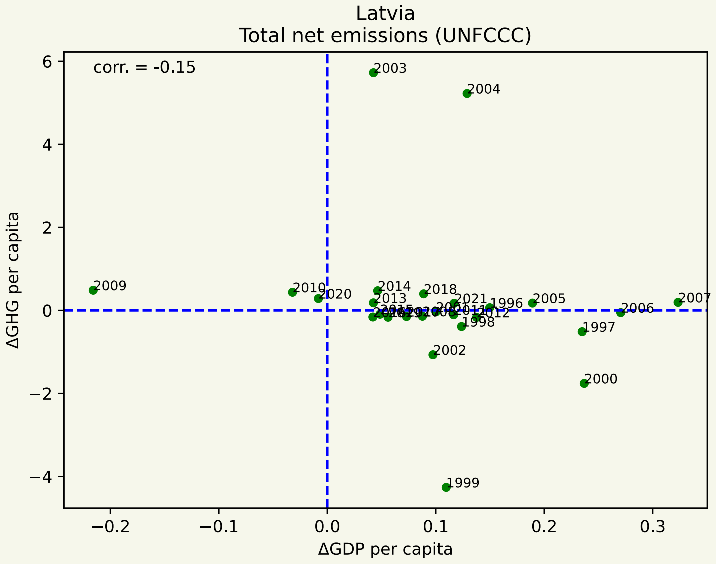
For this reason, we want to understand which is the correlation between the two variables in each country.

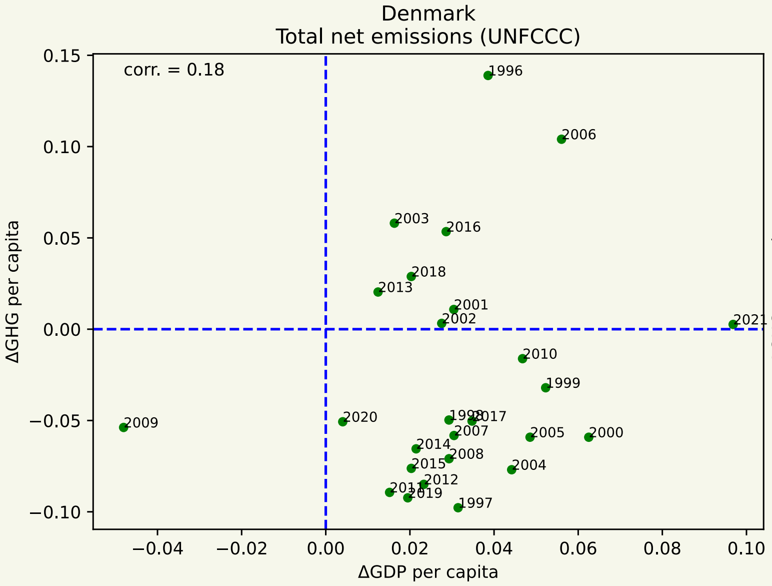
If the outlier 'year 2020', characterized by the COVID-19 pandemic lockdown, is excluded from the correlation calculation, distinct correlation values are obtained, as displayed.

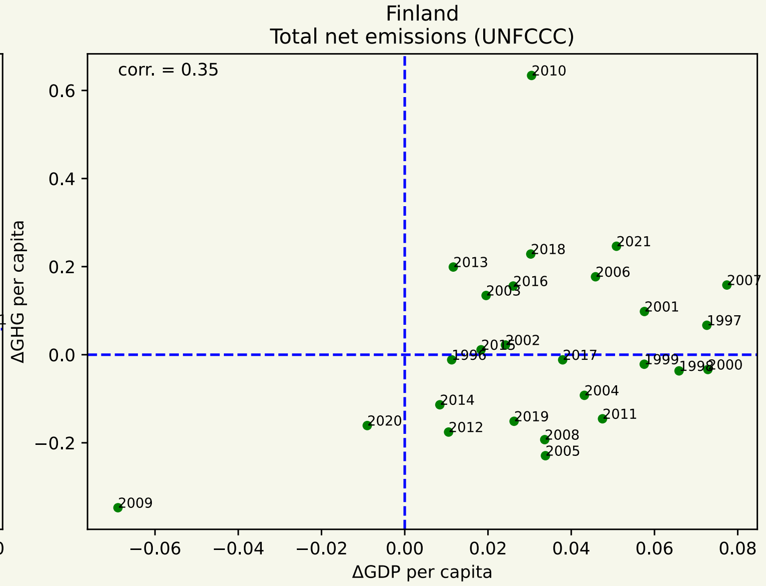
Country	Total emissions	Total NET emissions	Total w/o 2020	Total NET w/o 2020
EU-27	0.805063	0.801532	0.646364	0.648302
Spain	0.725030	0.714336	0.590334	0.563141
Italy	0.686701	0.601090	0.518116	0.457335
Croatia	0.680157	0.558770	0.649344	0.548899
France	0.659436	0.629927	0.315734	0.208473
Greece	0.649733	0.633318	0.517454	0.496991
Cyprus	0.623044	0.629803	0.604573	0.600880
Ireland	0.607729	0.609142	0.590502	0.589690
Sweden	0.587396	0.228215	0.559296	0.150883
Slovenia	0.577011	0.154197	0.522499	0.099648
Belgium	0.528215	0.537905	0.342512	0.363858
Czechia	0.503892	0.462035	0.388031	0.376661
Portugal	0.493503	0.205853	0.421760	0.162235
Germany	0.487155	0.452845	0.287696	0.274712
Lithuania	0.471325	0.284682	0.473382	0.272365
Romania	0.453958	0.428538	0.459708	0.425851
Hungary	0.427244	0.319999	0.388202	0.217244
Estonia	0.424490	0.320747	0.353110	0.281618
Luxembourg	0.391530	0.379115	0.352985	0.360459
Austria	0.342034	0.282500	0.007719	0.119851
Slovakia	0.246729	0.262978	0.243684	0.206424
Netherlands	0.229200	0.230243	-0.053516	-0.059919
Bulgaria	0.206854	0.181675	0.129179	0.108387
Denmark	0.200041	0.183644	0.130731	0.146024
Poland	0.166636	0.236687	0.106859	0.189859
Finland	0.117287	0.350453	0.058239	0.300506
Latvia	0.084291	-0.154992	0.026703	-0.156113
Malta	-0.099161	-0.099908	-0.223790	-0.216350

3.









Country	Total emissions	Total NET emissions
EU-27	0.805063	0.801532
Spain	0.725030	0.714336
Italy	0.686701	0.601090
Croatia	0.680157	0.558770
France	0.659436	0.629927
Greece	0.649733	0.633318
Cyprus	0.623044	0.629803
Ireland	0.607729	0.609142
Sweden	0.587396	0.228215
Slovenia	0.577011	0.154197
Belgium	0.528215	0.537905
Czechia	0.503892	0.462035
Portugal	0.493503	0.205853
Germany	0.487155	0.452845
Lithuania	0.471325	0.284682
Romania	0.453958	0.428538
Hungary	0.427244	0.319999
Estonia	0.424490	0.320747
Luxembourg	0.391530	0.379115
Austria	0.342034	0.282500
Slovakia	0.246729	0.262978
Netherlands	0.229200	0.230243
Bulgaria	0.206854	0.181675
Denmark	0.200041	0.183644
Poland	0.166636	0.236687
Finland	0.117287	0.350453
Latvia	0.084291	-0.154992
Malta	-0.099161	-0.099908

3.

A Detailed Examination Through Data

Our objective is to analyze the classification of the sectors identified in the first part of this presentation, within the GDP sectors.

The aim is to assess the magnitude to which emissions contribute to the GDPs within each sector.

```
correct_list = ['1 - Energy', '1.A.1 - Energy Industries','1.A.2 - Manufacturing Industries and Construction',
            '1.A.3 - Transport', '1.B - Fugitive Emissions from Fuels', '1.C - CO2 Transport and Storage',
            '2 - Industrial Processes and Product Use', '2.A - Mineral Industry','2.B - Chemical Industry',
            '2.C - Metal Industry', '2.E - Electronics Industry', '3 - Agriculture', '3.1 - Livestock',
            '3.B - Manure Management', '3.C - Rice Cultivation', '3.D - Agricultural Soils', '4.A - Forest Land',
            '4.B - Cropland', '4.C - Grassland', '4.D - Wetlands', '4.F - Other Land', '4.G - Harvested Wood Products',
            'Total emissions (UNFCCC)', 'Total net emissions (UNFCCC)']

ita_eu27_emi = df_emissions_multiindex.swaplevel("Country","Pollutant_name").loc["All greenhouse gases - (CO2 equivalent)"]

ita_eu27_emi["per_capita_emiss_variation"] = ita_eu27_emi.per_capita_emission.pct_change()
# We add a column that has the percentage variation in emissions from the previous year

# We need to set each row that has year = 1990 to NA
ita_eu27_emi.reset_index(inplace = True)
ita_eu27_emi.set_index("Year", inplace = True)
ita_eu27_emi.loc[1990,"per_capita_emiss_variation"] = np.nan

# After setting the pct variation for 1990 to NA we go back to the original multiindex
ita_eu27_emi.reset_index(inplace = True)
ita_eu27_emi.set_index(["Sector_name","Country","Year"], inplace = True)

# Slicing
ita_eu27_emi = ita_eu27_emi.swaplevel(0,1).loc[["Italy", "EU-27"]]
```

3.

A Detailed Examination Through Data

```python
# Creating a function to add a new agreggate sector to ita_eu27_emi dataset
def agg_sector(df, country, sector, prefix, new_sector):
  country_sector = df.loc[(country, sector), 'emissions'].groupby('Year').sum()
  country_prefix = df.loc[(country, df.index.get_level_values('Sector_name').str.startswith(prefix)), 'emissions'].groupby(
  new_data = country_sector - country_prefix

  df = df.reset_index()

  new_row_country = pd.DataFrame({
      'Country': country,
      'Sector_name': new_sector,
      'Year': new_data.index,
      'emissions': new_data.values
    })

  df = pd.concat([df, new_row_country], ignore_index=True, sort=False)
  df = df.sort_values(by=['Sector_name', 'Country', 'Year'])
  df.set_index(['Country','Sector_name','Year'], inplace=True)

  pop = df.loc[(country, sector), 'pop_mil']
  df.loc[(country, new_sector), 'pop_mil'] = pop.values

  df['per_capita_emission'] = df['emissions'] / (df['pop_mil'] * 1000000)
```

Since not all 172 labels in the first part were chosen due to their marginal significance, a **custom function** is utilized to compute the difference between the main label and each corresponding sub-label.

3.

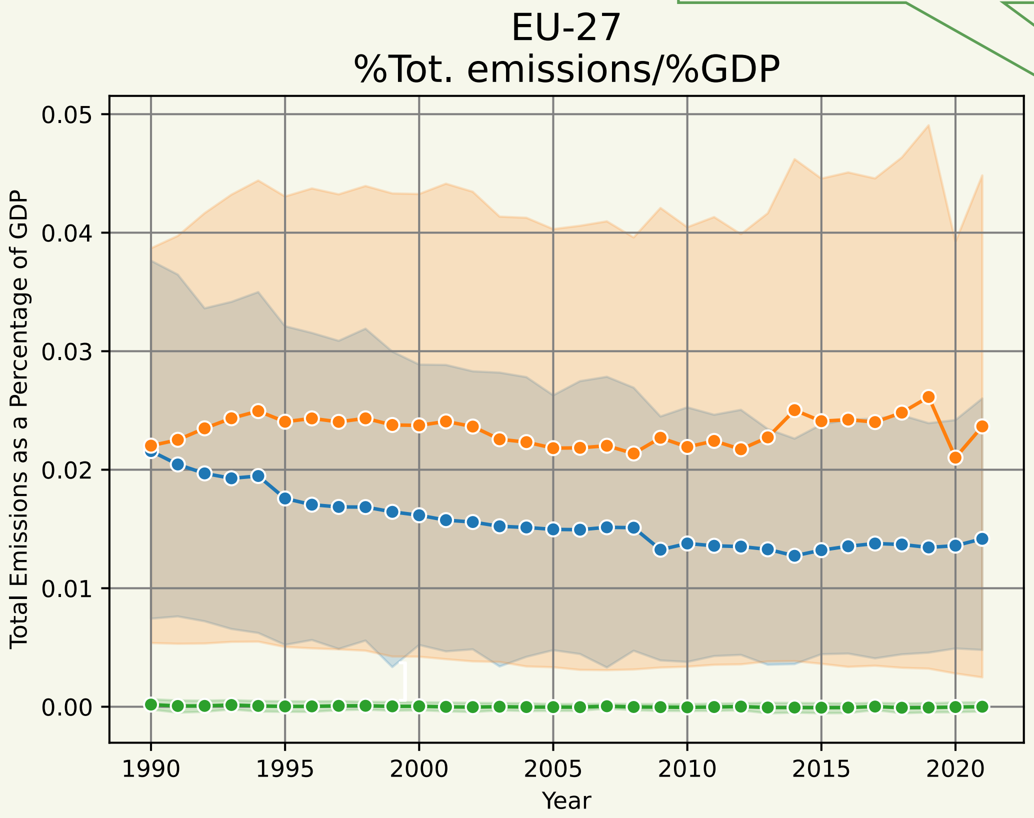
```
# A brand new DataFrame in which we compute the indicators below, useful  for our analysis
ita_eu27_emi
```

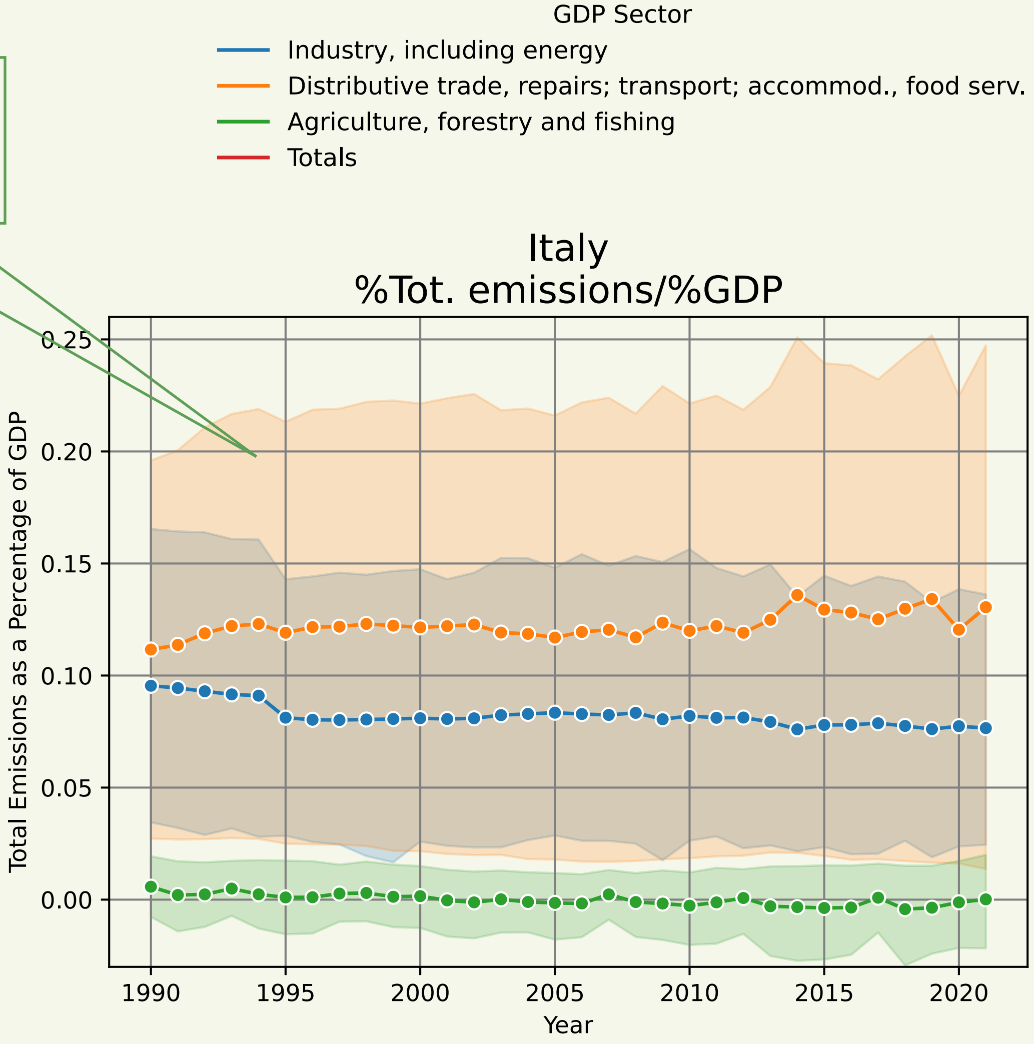
Country	Sector_name	Year	emissions	pop_mil	per_capita_emission	per_capita_emiss_variation	gdp_sector	tot_emiss_%	net_emiss_%	gdp_%	tot_emiss_%/gdp_%	net_emiss_%/gdp_%
EU-27	1.A.1 - Energy Industries	1995	1.318747e+06	425.685108	0.003098	-0.001349	Industry, including energy	0.289189	0.310736	21.206657	0.061327	0.065897
		1996	1.347416e+06	426.364400	0.003160	0.020112	Industry, including energy	0.289706	0.312135	20.819486	0.060315	0.064985
		1997	1.312623e+06	426.941575	0.003074	-0.027139	Industry, including energy	0.286619	0.308631	20.776907	0.059551	0.064124
		1998	1.315705e+06	427.424910	0.003078	0.001214	Industry, including energy	0.289773	0.313218	20.705117	0.059998	0.064852
		1999	1.278436e+06	428.042160	0.002987	-0.029728	Industry, including energy	0.286376	0.311023	20.215601	0.057893	0.062875
...	...	...	...	...	...	...	...	...	...	...	...	...
Italy	4.G - Harvested Wood Products	2017	-9.744199e+02	60.002254	-0.000016	158.888122	Agriculture, forestry and fishing	-0.002228	-0.002353	1.972685	-0.000044	-0.000046
		2018	-7.776014e+02	59.877216	-0.000013	-0.200319	Agriculture, forestry and fishing	-0.001793	-0.002002	1.945420	-0.000035	-0.000039
		2019	-3.480525e+03	59.729077	-0.000058	3.487077	Agriculture, forestry and fishing	-0.008242	-0.009149	1.907296	-0.000157	-0.000174
		2020	-2.240009e+03	59.438845	-0.000038	-0.353274	Agriculture, forestry and fishing	-0.005819	-0.006356	2.008187	-0.000117	-0.000128
		2021	-2.035387e+03	59.133173	-0.000034	-0.086652	Agriculture, forestry and fishing	-0.004874	-0.005217	1.900184	-0.000093	-0.000099

3. ita_eu27_emi

Some plots

The shallow represent the I.C. given by the data for the same year



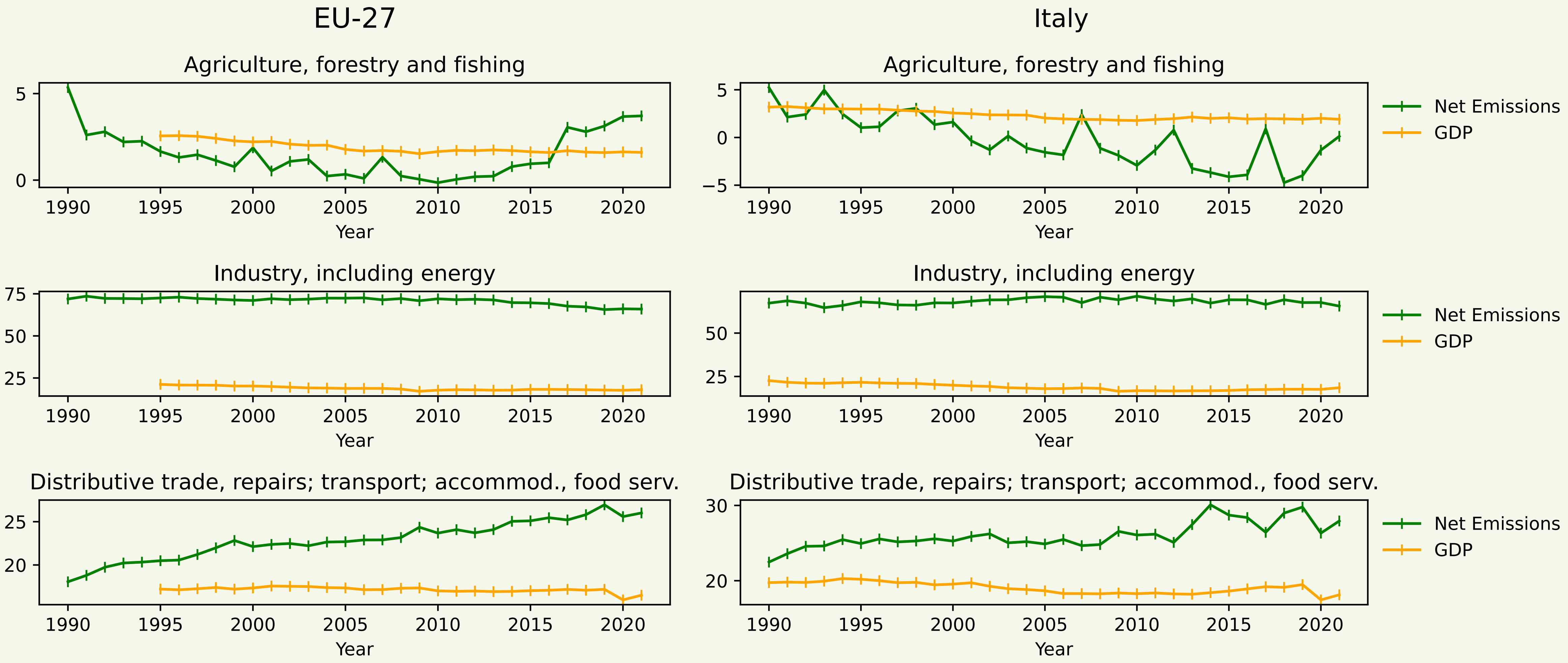


3.

ita_eu27_emi

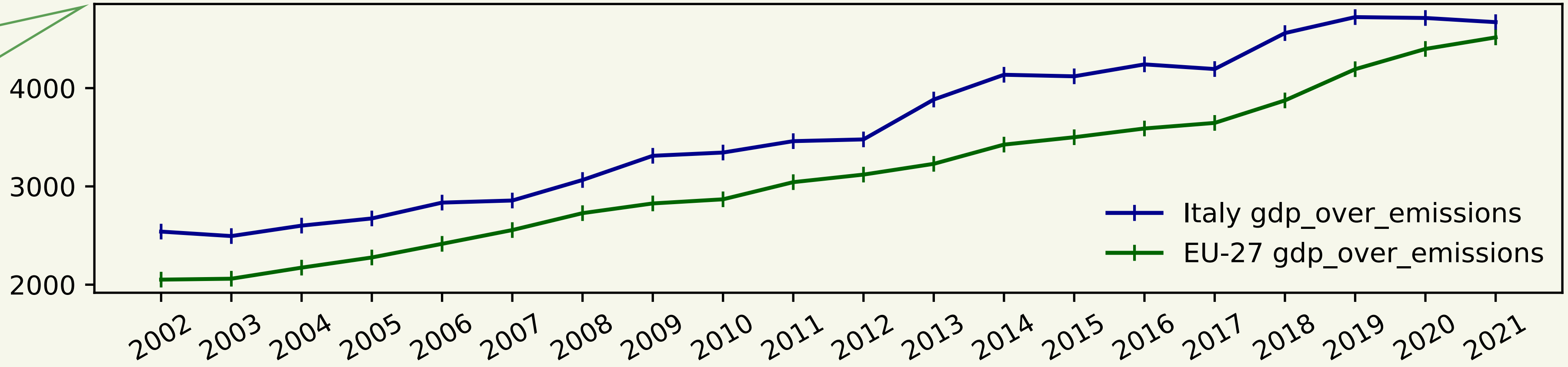
Some plots

To understand better the previous graphs, we plot the paths of *net_emission_%* and *gdp_%* over years of EU-27 and Italy both.



3. GDP over Emissions: An Inverse Overview of Data

How much money is generated for every ton of emissions in euros?



Country	gdp_sector	Year	emissions	gdp	gdp_over_emissions
EU-27	Agriculture, forestry and fishing	2002	4.503798e+04	177354.529	3937.887745
	Distributive trade, repairs; transport; accommod., food serv.	2002	9.358913e+05	1497124.067	1599.677287
	Industry, including energy	2002	2.976846e+06	1668550.504	560.509505
	Agriculture, forestry and fishing	2003	5.078995e+04	176102.491	3467.270636
	Distributive trade, repairs; transport; accommod., food serv.	2003	9.454558e+05	1535268.612	1623.839649
...	...	...	...	...	...
Italy	Total net emissions (UNFCCC)	2017	4.140434e+05	1736592.800	4194.228809
		2018	3.884603e+05	1771391.200	4560.032038
		2019	3.804392e+05	1796648.500	4722.564607
		2020	3.524252e+05	1661239.800	4713.737876
		2021	3.901183e+05	1822344.500	4671.261554

Country	gdp_sector	gdp_emi_corr
EU-27	Agriculture, forestry and fishing	0.138122
	Distributive trade, repairs; transport; accommod., food serv.	0.923520
	Industry, including energy	0.773890
	Total emissions (UNFCCC)	0.816252
	Total net emissions (UNFCCC)	0.811233
Italy	Agriculture, forestry and fishing	-0.410376
	Distributive trade, repairs; transport; accommod., food serv.	0.902344
	Industry, including energy	0.696729
	Total emissions (UNFCCC)	0.870134
	Total net emissions (UNFCCC)	0.767843

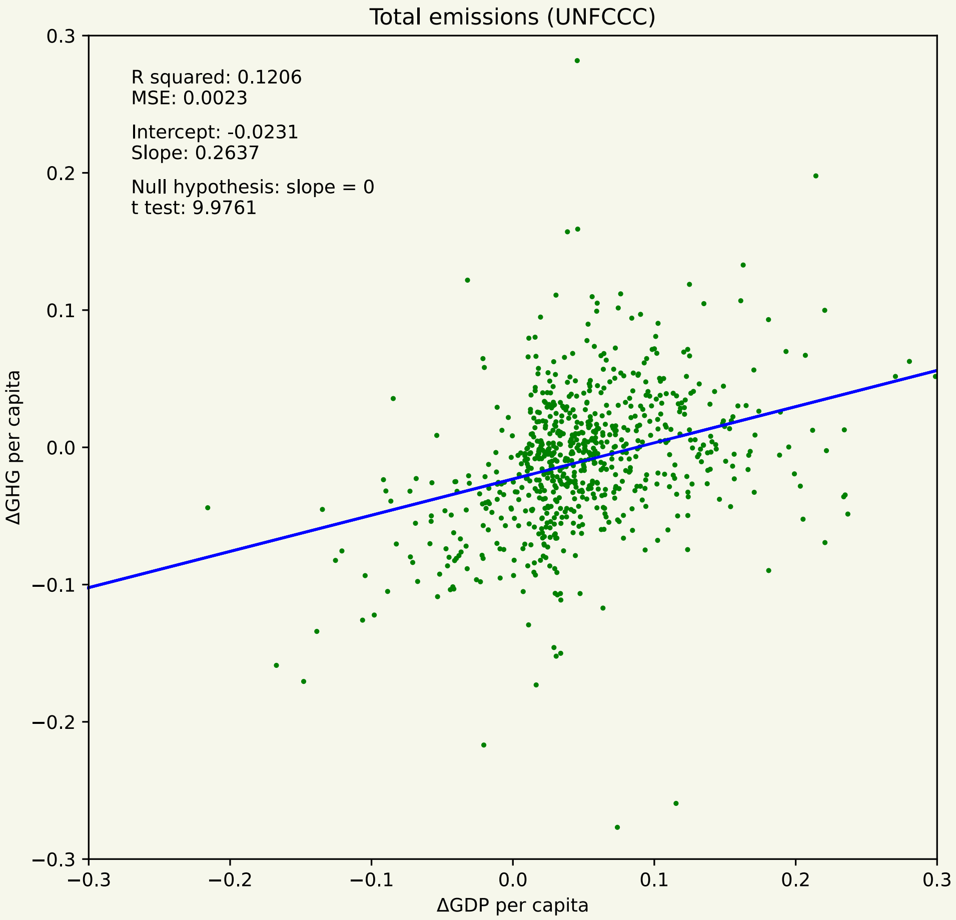
4. LINEAR REGRESSION AND CLASSIFICATION

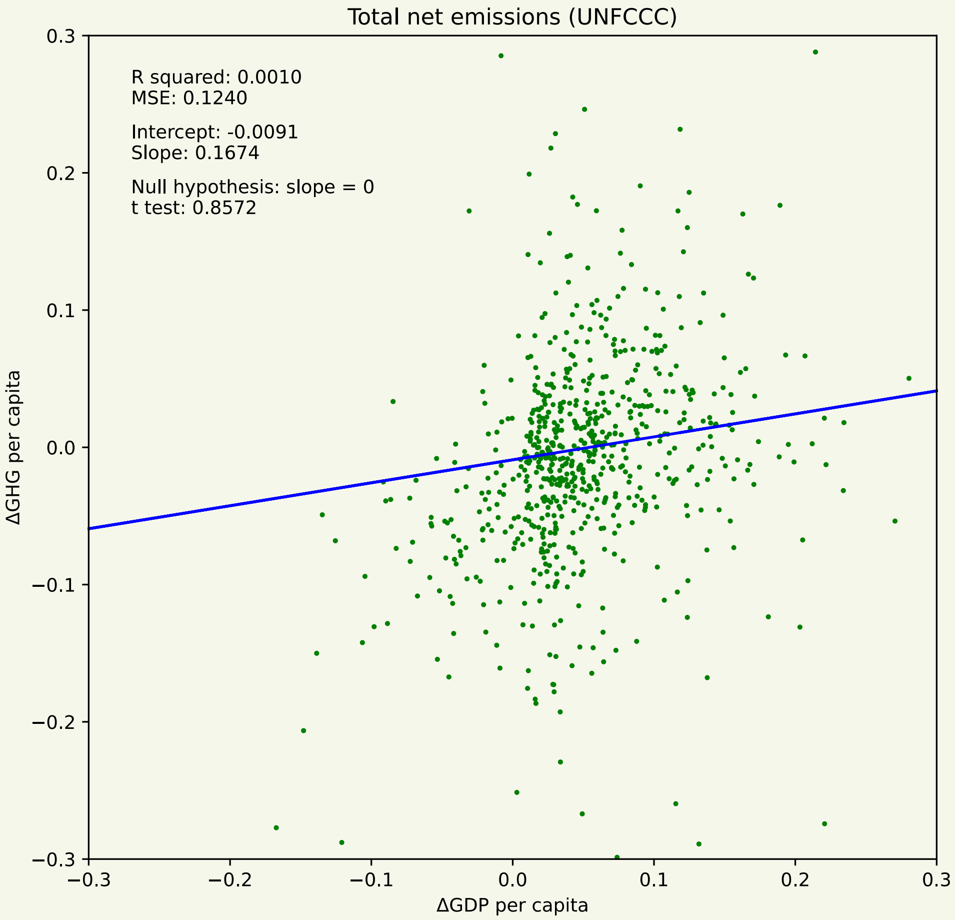
In this fourth section, we fit some statistical models to better gauge the relationship between the data we are going to analyse.

We introduce a new package *scikit-learn* too, to compute Logistic Regression and Classification.

4. Linear Regression

The y variable is per capita emissions change from the previous year.
The x variable is per capita GDP growth. Only the regression with total emissions is significant.





4. Classification with logistic regression

We want to see if logistic regression can predict if, for a given GDP per capita growth, emissions will fall or not. The Y variable is 1 if emissions decreased from the previous year, 0 otherwise. The X variable is per capita GDP growth.

If there is no relation between the two variables, we expect that the classifier will perform poorly.

$$Y = I(\Delta\ per\ capita\ emissions \leq 0) \qquad \Pr\left(y = 1 \mid x = \Delta\ GDP\right) = \frac{e^{(\beta_0+\beta_1 x)}}{1+e^{(\beta_0+\beta_1 x)}}$$

```
y = np.array(emissions_gdp_var.loc['Total emissions (UNFCCC)'].per_capita_emiss_variation <= 0)
X = emissions_gdp_var.loc['Total emissions (UNFCCC)'][["per_capita_GDP_var","per_capita_GDP"]]

logreg= LogisticRegression()
logreg.fit(X, y)

y_prediction = logreg.predict(X)
```

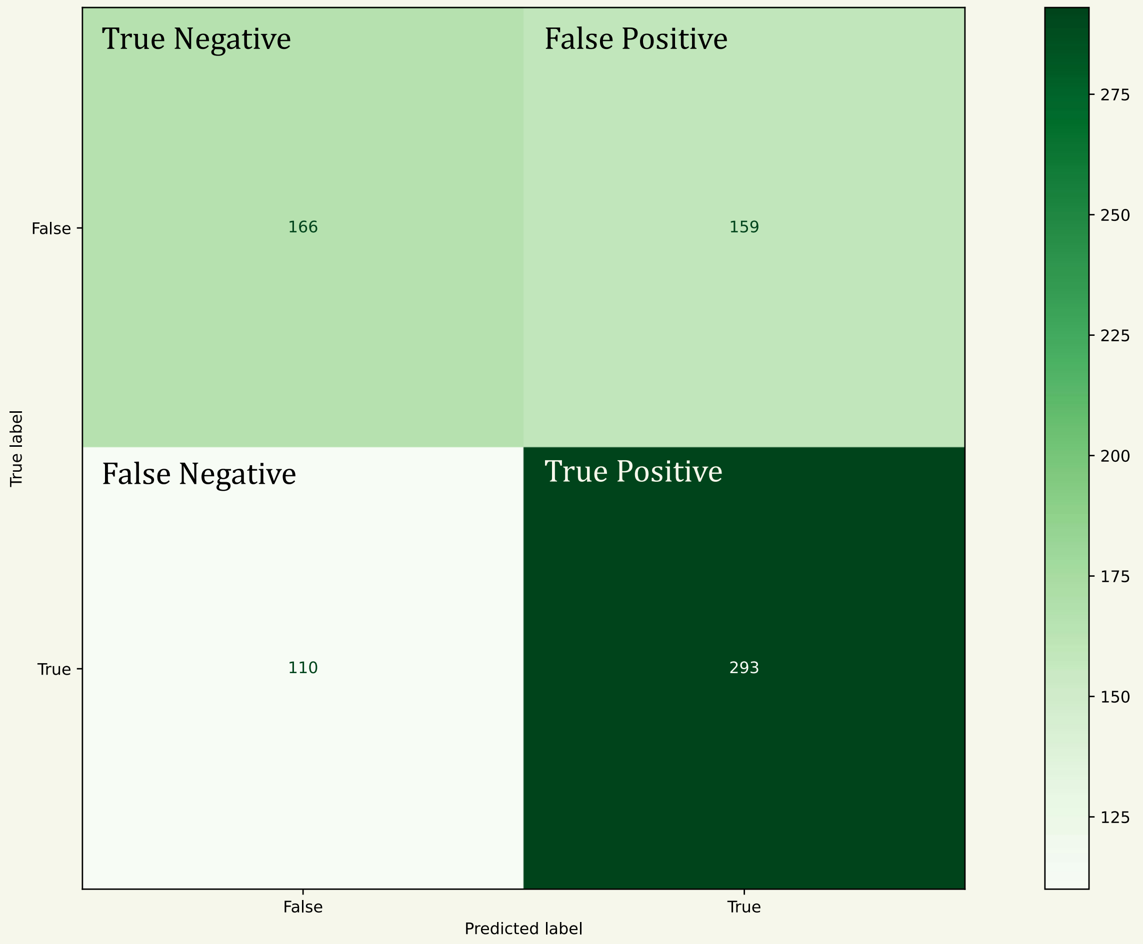
$$Accuracy = 0.5947$$

The accuracy is higher than the accuracy of the null classifier, confirming that GDP growth has some impact on whether emission increase or decrease. However, it's not a good classifier and we can conclude that the relation is weak.

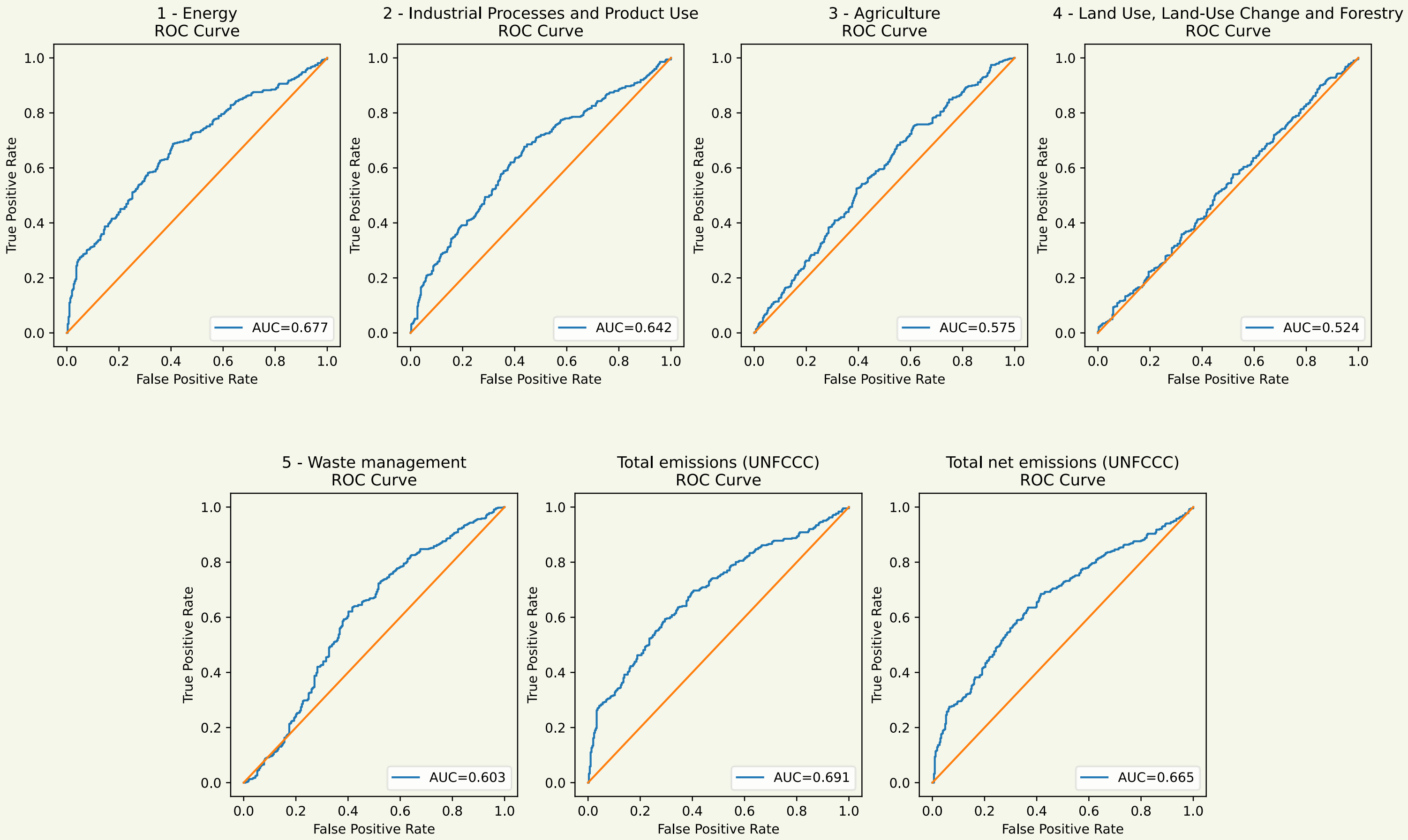
4. Classification with logistic regression

Confusion matrix

This confusion matrix is generated based on the changes in per capita emissions (Δ per capita emissions) and the changes in Gross Domestic Product (Δ GDP). Upon examining this matrix, it becomes evident that the number of False-Positive instances equals the number of True-Negative instances. This implies that making predictions based on our data is not possible.



4. Classification with logistic regression



We fit the same model, once per sector. We can compare the performances looking at the area under curve of the ROC graph.
In sectors 3 and 4 it is not possible to predict if emissions will decrease knowing GDP growth.
Emissions from energy production have a stronger link with growth than total net emissions.

5.CONCLUSION

- In the last 30 years there is a slight decrease in GHG emissions, both total and per capita.

- During the same time frame GDP experienced some growth.

- However, there is still an undeniable positive correlation persists between the two factors: GDP growth is consistently associated with increased emissions over the years.

- In the EU-27 context, the Industrial and Distributive sectors exhibit the highest correlation with greenhouse gas (GHG) emissions. However, upon closer examination of various sectors or countries, instances emerge where the correlation approaches zero.

- Recent years indicate a positive trend: producing an equivalent GDP now requires fewer GHG emissions, signifying an improvement in environmental efficiency.

Thanks for your attention!

GLEM

Giuseppe Massidda, 901029
Eleonora Marcassa, 900571
Lorenzo Muscillo, 1000916
Manuel Trevisan, 886937