The development of agricultural land use and permanent crop area across time and comparing regions of Portugal

Analysis and Visualization of Complex Agro-Environmental Data

Authors: Afonso Marques
Diogo Pinto
Julia Anja Zahmow

Deadline: 15/06/2023

Table of Contents

Table of Figures

1 Introduction

The goal of this project is to tell a coherent story from a complex agro-environmental dataset. The country of study is Portugal. An important economic sector of Portugal is agriculture, both of permanent and temporary crops. Many people depend on agriculture for a living.

We wanted to address the following questions/visualization tasks:

- Does the ratio of permanent to temporary crop area change over time in the NUTS2 regions of Portugal?
- Visualization across the years of the relationship between permanent crop area and labor force in agriculture.
- Visualization of the development of labor force in agriculture across the years displayed on a map of Portugal.

Different types of visualization and analysis will be chosen in order to make the relationships and developments easy to understand.

In the following chapters, the database, exploratory data analysis, a conclusion and the SQL and Python code will be presented.

2 Database description

The data for this project was taken from INE database which contains a main table, “region” with the primary key “NUTSID”, working like a secondary key for all the other tables. Other tables include “livestock”, “grassland”, “production”, “labour”, “education”, “temporary_crop”, “permanent_crop”. The INE database thus contains a lot of valuable data related to the agricultural sector. For the purpose of this report, we created two new tables using SQL queries, both containing variables from various tables of the database. The SQL code to create these new tables can be consulted in the ANNEX of this report.

	NutsID	level_ID	region_name	year	pc_name	tc_name	pc_area	tc_area	labour_value	labour_type_id	ratio_pc_tc
1	11	2	Norte	1 989	Total	Total	223 736	534 051	317 855	1	0,4189
2	11	2	Norte	1 999	Total	Total	228 323	316 950	204 053	1	0,7204
3	11	2	Norte	2 009	Total	Total	218 645	206 068	148 088	1	1,0605
4	11	2	Norte	2 019	Total	Total	255 964	151 107	119 432	1	1,6939
5	15	2	Algarve	1 989	Total	Total	59 888	39 606	30 061	1	1,5121
6	15	2	Algarve	1 999	Total	Total	56 309	20 190	16 946	1	2,789
7	15	2	Algarve	2 009	Total	Total	45 007	8 193	11 432	1	5,4933
8	15	2	Algarve	2 019	Total	Total	56 754	12 120	13 720	1	4,6827
9	16	2	Centro	1 989	Total	Total	252 705	487 082	337 724	1	0,5188
10	16	2	Centro	1 999	Total	Total	213 178	345 183	196 066	1	0,6176
11	16	2	Centro	2 009	Total	Total	157 603	212 738	123 809	1	0,7408
12	16	2	Centro	2 019	Total	Total	168 086	171 887	96 184	1	0,9779
13	17	2	Área Metropolitana de Lisboa	1 989	Total	Total	23 694	57 899	29 512	1	0,4092
14	17	2	Área Metropolitana de Lisboa	1 999	Total	Total	14 732	42 353	16 741	1	0,3478
15	17	2	Área Metropolitana de Lisboa	2 009	Total	Total	14 060	34 888	10 273	1	0,403
16	17	2	Área Metropolitana de Lisboa	2 019	Total	Total	16 428	34 812	9 520	1	0,4719
17	18	2	Alentejo	1 989	Total	Total	220 942	751 291	94 853	1	0,2941
18	18	2	Alentejo	1 999	Total	Total	192 689	653 738	63 731	1	0,2947
19	18	2	Alentejo	2 009	Total	Total	251 006	461 650	47 900	1	0,5437
20	18	2	Alentejo	2 019	Total	Total	358 544	473 551	54 381	1	0,7571
21	20	2	Região Autónoma dos Açores	1 989	Total	Total	4 769	20 784	20 344	1	0,2295
22	20	2	Região Autónoma dos Açores	1 999	Total	Total	3 662	17 355	15 452	1	0,211
23	20	2	Região Autónoma dos Açores	2 009	Total	Total	2 021	22 032	11 532	1	0,0917
24	20	2	Região Autónoma dos Açores	2 019	Total	Total	2 574	42 822	10 594	1	0,0601
25	30	2	Região Autónoma da Madeira	1 989	Total	Total	3 679	4 580	19 681	1	0,8033
26	30	2	Região Autónoma da Madeira	1 999	Total	Total	2 735	3 565	13 156	1	0,7672
27	30	2	Região Autónoma da Madeira	2 009	Total	Total	2 482	2 901	14 360	1	0,8556
28	30	2	Região Autónoma da Madeira	2 019	Total	Total	2 322	2 085	10 678	1	1,1137

ratio_labour
NutsID
level_ID
region_name
year
pc_name
tc_name
pc_area
tc_area
labour_value
labour_type_id

Figure 1: Screenshots of the first table created with SQL, for the landuse development (ratio permanent to temporary crop over time)

	municipality	year	area_ha	value
1	Abrantes	1 989	18 587	2 546
2	Abrantes	1 999	15 651	2 402
3	Abrantes	2 009	8 464	1 371
4	Abrantes	2 019	7 905	759
5	Águeda	1 989	1 164	4 702
6	Águeda	1 999	743	2 570
7	Águeda	2 009	337	1 526
8	Águeda	2 019	702	1 060
9	Aguiar da Beira	1 989	622	2 290
10	Aguiar da Beira	1 999	821	1 728
11	Aguiar da Beira	2 009	972	1 124
12	Aguiar da Beira	2 019	1 408	895
13	Alandroal	1 989	6 568	1 261
14	Alandroal	1 999	6 713	1 103
15	Alandroal	2 009	6 739	834
16	Alandroal	2 019	7 484	1 098
17	Albergaria-a-Velha	1 989	194	2 904
18	Albergaria-a-Velha	1 999	165	1 464
19	Albergaria-a-Velha	2 009	70	1 116
20	Albergaria-a-Velha	2 019	86	792
21	Albufeira	1 989	7 622	1 661
22	Albufeira	1 999	8 781	961
23	Albufeira	2 009	5 538	657
24	Albufeira	2 019	5 704	1 017
25	Alcácer do Sal	1 989	4 455	2 159

permcrop_value
municipality
year
area_ha
value

Figure 2: Screenshots of second table created with SQL, for

3 Exploratory data analysis

3.1 Land use development of the NUTS2 regions in Portugal in 1989-2019

This chapter wants to explore how the ratio of permanent to temporary crop area changes over time in the NUTS2 regions of Portugal. The ratio is the dependent variable whereas the year is the independent variable.

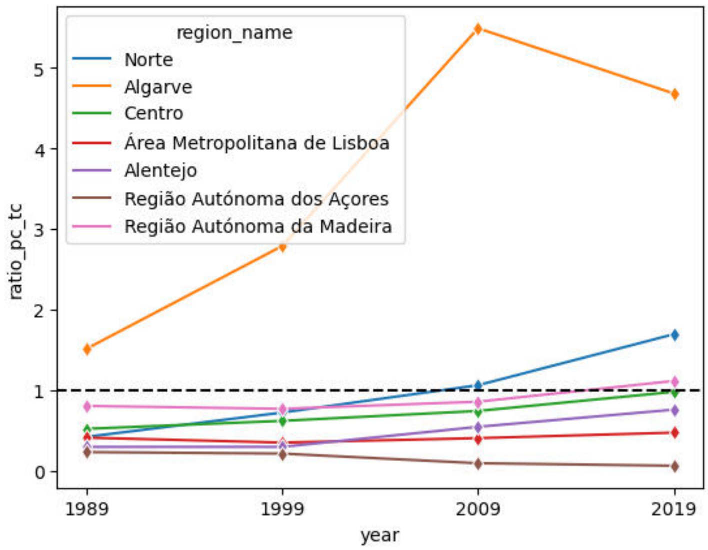


Figure 3: Development of the ratio of permanent crops vs. temporary crops in NUTS2 regions in 1989-2019

This visualization shows well that in most regions of Portugal there is a shift happening towards more permanent cropland vs. temporary cropland. If the ratio is below 1, the region has more temporary cropland; if it is 1, permanent and temporary cropland have the same size; if the ratio is above 1, there is more permanent cropland than temporary cropland in the respective region. As an example, Algarve uses most of its agricultural cropland to produce citrus fruit, almonds, figs and wine, all of which are permanent crops. Also, the North of Portugal seems to be shifting more and more towards permanent crops, such as vineyards, olives, apples and pears.

Find the Python code that was used to create this data visualization in the ANNEX of this report.

3.1.1 OLS Regression and regression diagnostic

The regression analysis is a statistical method that attempt to fit a model to data - quantify the relationship between a continuous dependent (outcome, response) variable and the independent (predictor, covariate) variable(s). Response variable = model + error (part of the response not explained by the model).

However, according to the OLS Regression results when plotting the years along the x-axis and the ratio of permanent to temporary crop area along the y-axis, R-squared is very low (close to 0), see Figure 4. Generally speaking, the values of R-squared can range between 0 and 1, where 0 indicates that none of the variation in the response variable is explained by the predictors, and 1 indicates that all of the variation in the response variable is explained by the predictors. Thus, the shift towards more permanent cropland is not confirmed by statistical analysis.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             ratio_pc_tc   R-squared:                       0.066
Model:                             OLS   Adj. R-squared:                  0.030
Method:                  Least Squares   F-statistic:                     1.824
Date:                 Thu, 15 Jun 2023   Prob (F-statistic):              0.188
Time:                         22:08:33   Log-Likelihood:                -45.190
No. Observations:                   28   AIC:                             94.38
Df Residuals:                       26   BIC:                             97.05
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -56.6698     42.722     -1.326      0.196    -144.486      31.147
year           0.0288      0.021      1.351      0.188      -0.015       0.073
==============================================================================
Omnibus:                        28.357   Durbin-Watson:                   0.572
Prob(Omnibus):                   0.000   Jarque-Bera (JB):               51.038
Skew:                            2.276   Prob(JB):                     8.27e-12
Kurtosis:                        7.799   Cond. No.                     3.59e+05
==============================================================================
```

Figure 4: OLS Regression Results

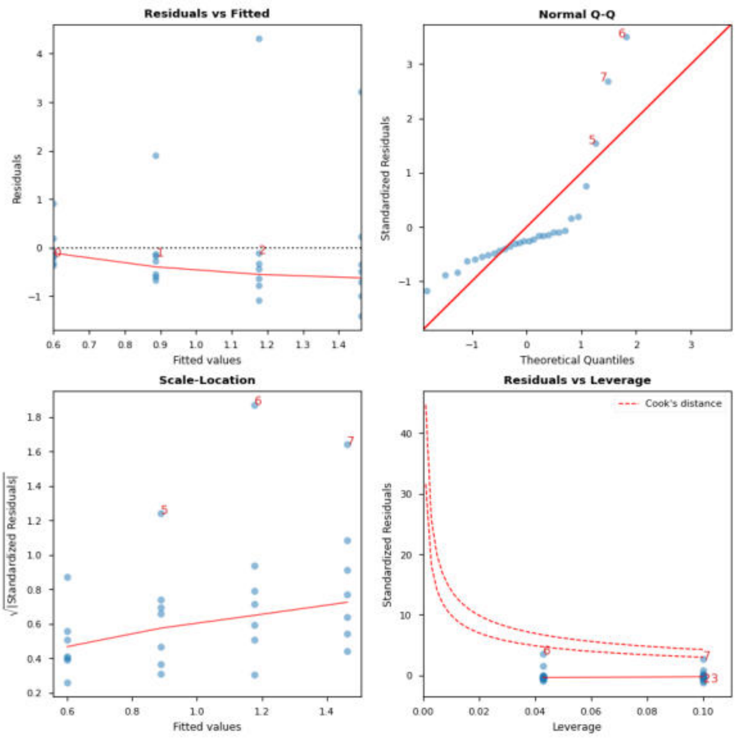


Figure 5: Linear regression diagnostic plots created with Python

In order to detect non-linear patterns in the residuals the standardized residuals are plotted against the fitted values (see Figure, upper left). For linear patterns in the residuals, they have to be equally distributed above and below zero, and the residuals should have no relation to the fitted values. In the case of the here conducted plot, the values do not comply with any of the two rules, and thus, the residuals are non-linear.

For the QQ-plot (see Figure, upper right) of the regression to be normal, values should be arranged along a line. In the case of the here created QQ-plot, the error distribution deviates slightly from a normal distribution.

For analyzing the homogeneity of variance, a scale-location plot (see Figure, lower left) is used with the residuals being standardized by the standard deviation. It evaluates the assumption of constant variance of the residuals over the adjusted values. The here created plot indicates that the variance of the residuals over the adjusted values is not constant.

To detect outliers one can plot the standardized residuals against leverage (see Figure, lower right). Leverage measures the ability of each observation to influence the fit of the model. Very influential observations have high residual values and leverage and thus a high Cook's distance statistic. In the here created plot, a high Cook's distance statistic can be observed, meaning that there are very influential outliers.

Thanks to the regression diagnostic plots, one can reject the assumption that the underlying variables have a linear relationship. These regression diagnostics do not suggest that the ratio of permanent to temporary cropland increases over time. Another reason the shift is not statistically confirmed might be that the y-values are only distributed across four x-values, namely the four years given in the database.

3.2 Development of permanent crop area and labor force in NUTS3 regions of Portugal

What is the relationship between permanent crop area and labor force in agriculture across the years? Two interactive visualizations will be used for understanding the data.

The first one allows to group the data points of the NUTS3 regions into the NUTS2 region by using different colors. By hovering across the data points, more information on the data point appears. Please run the python code to see the interaction version of the visualization.

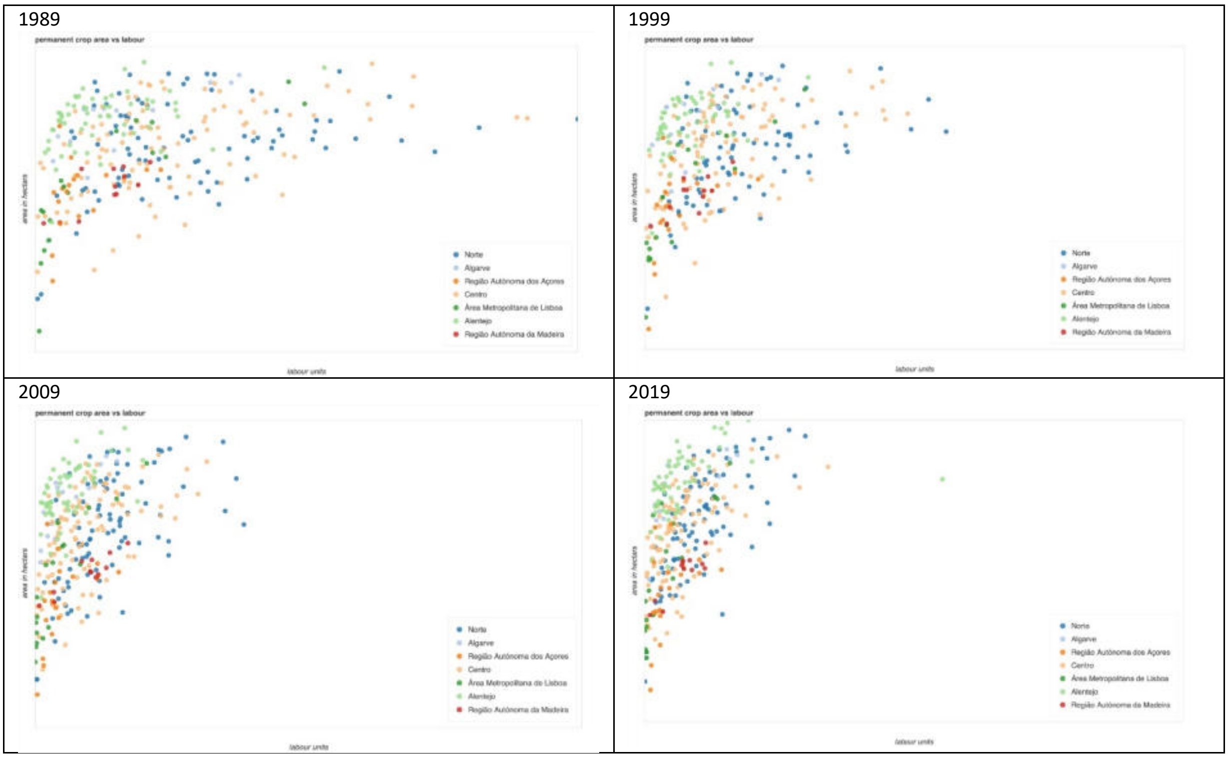


Figure 6: NUTS3 regions of Portugal plotted according to their labor force and permanent crop area in the years 1989, 1999, 2009, 2019; and colored according to NUTS2 regions

In Figure 6, the NUTS3 regions of Portugal are plotted according to their agricultural labor force and permanent crop area in the years 1989, 1999, 2009 and 2019. Moreover, they are grouped by their NUTS2 region, such as Norte, Alentejo or Lisboa, using different colors.

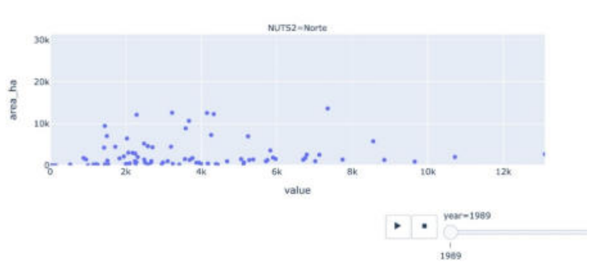
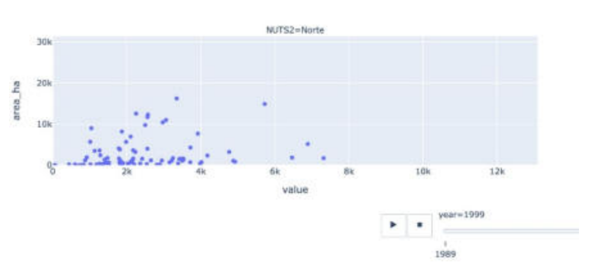
The main observation that becomes apparent is the reduced dispersion of the data points along the x-Axis, being the labor force. With the advance in time, there are always less people working in agriculture. This development might be due to mechanization and industrialization of permanent crop agriculture. Labor force is being replaced by mechanical cultural practices and smaller agricultural land units are being united and managed by bigger corporations. However, no big change concerning the permanent crop area in hectares can be observed over time. The land use distribution seems to not to have gone through big changes during the past 30 years. The only exception might be some regions in Alentejo, which added permanent crop hectares from 1989 to 2019 (see light green data points with the highest permanent crop area in 2019).

Moreover, the data visualization shows that Alentejo’s data points are grouped together in the upper left corner (especially in the more recent years), being defined by a large permanent crop area and comparatively little labor force working in agriculture, whereas Norte’s NUTS3 regions vary more in permanent crop area and have comparatively more labor force working in agriculture. Having a closer look at Norte’s data points, one can make out another relationship: the higher the permanent crop area of the NUTS2 region, the higher the number of people working in agriculture, which is probably due to the higher demand for working the permanent cropland. Similar relationships can also be found checking the other NUTS2 regions (colors).

Another observation that can be made is the development of labor force in the NUTS2 region Lisboa. Whereas in 1989, every NUTS2 region of Lisboa had people employed in agriculture, in 2019, there are several parts of Lisboa that have labor force numbers close to 0.

In order to see the visualizations in detail with the scales of the axis and more info popping up when the cursor touches the data points (e.g. which municipality the value represents), please run the respective python code in the ANNEX.

In the second visualization, we present labor force working in agriculture in the seven different NUTS2 regions of Portugal across the years 1989-2019, using an interactive slider.

1989	1999
	

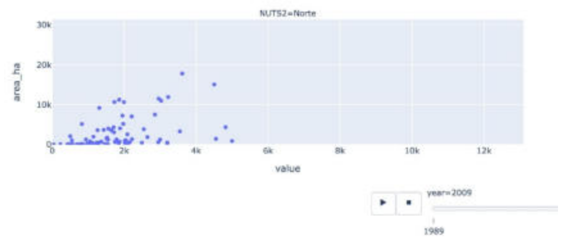
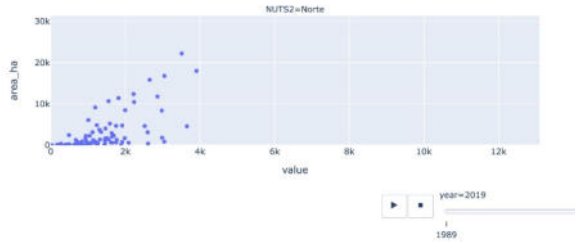
2009	2019
	

Figure 7: Screenshots of an interactive visualization across the years 1989-2019 (slider) of labor force working in agriculture in the NUTS2 Norte region of Portugal

Unfortunately, we were not able to export the visualizations done in Python into an HTML link. Therefore, we kindly ask you to run the code to use the interactive visualization with slider simulating the development across the years.

In Figure 7, one can notice the reduction in labor force working in agriculture in the Norte region in Portugal. Moreover, one can see an increase in permanent crop area.

This is a similar conclusion as in the first interaction visualization but presented in different ways. Moreover, the distribution of the data of the first interactive visualization appears differently due to the variable axis, in comparison to the non-variable axis of the second interactive visualization.

3.3 Geographical visualization of labor force development across the years in the NUTS3 regions of Portugal

In the following, for the better understanding of the development of labor force working in agriculture in the different regions of Portugal across the years, the data will be displayed geographically.

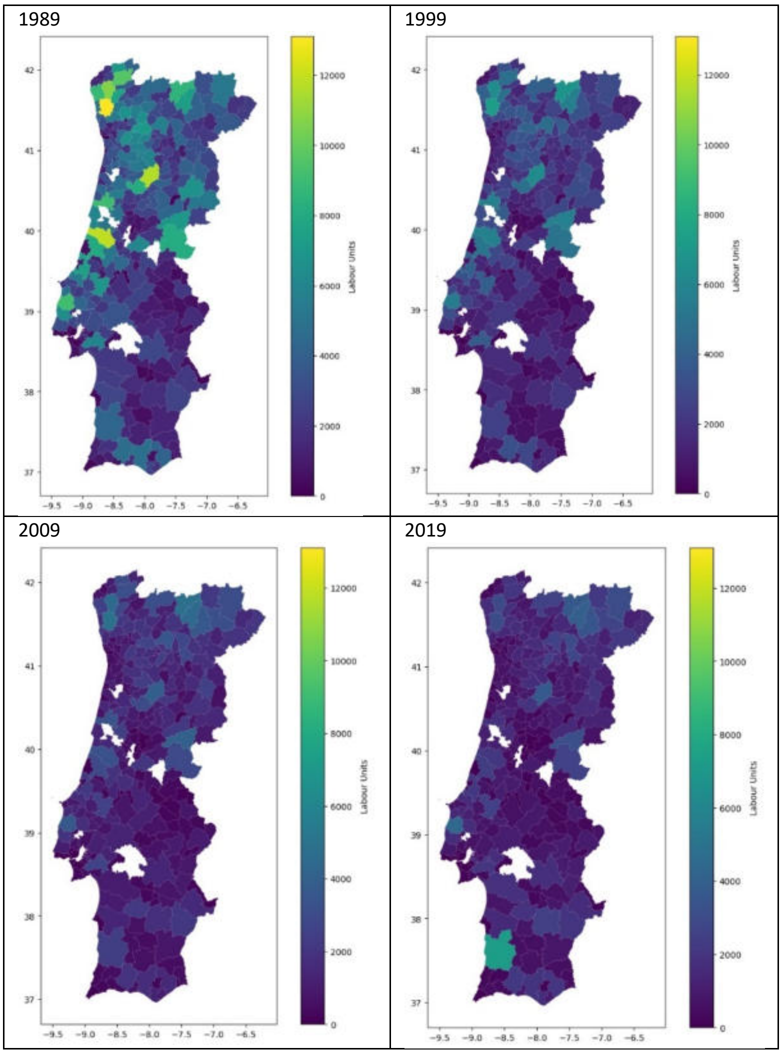


Figure 8: Development of labor force working in agriculture in the NUTS3 regions of Portugal across the years 1989-2019

The visualizations show the decrease in labor force working in agriculture across the years. Regions that used to employ many people in agriculture like in the North-West of Portugal gradually lost labor force with time. This change is probably due to the mechanization and automation of agriculture in the past decades. In 2019, Odemira had a sudden jump in agricultural labor force. Unfortunately, we were not able to show the data for all NUTS3 regions of Portugal since there were some which had discrepancies in the naming when comparing the poligons and the INE database.

4 Discussion/Conclusion

For this analysis we made two different studies: In the first one we observed in four different years (1989/1999/2009/2019), the development of the permanent and temporary crop area in Portugal, based on the ratio between them. In the second study, we observed the labor force working in agriculture changing in NUTS2 and 3 regions across the four years.

This data analysis including visualizations turned out to be a valuable tool to better understand relationships between collected data points. The provided database offers many possibilities for further analysis into agricultural relationships.

5 ANNEX

5.1 SQL code

5.1.1 Ratio of permanent to temporary cropland: table

```
use avcd2_INE

#create table: ratio_labour

create table ratio_labour as

select

r.NutsID, r.level_ID , r.region_name, pc.`year`,

pcn.crop_name as pc_name,

tcn.crop_name as tc_name,

pc.area as pc_area,

tc.area as tc_area,

l.value as labour_value,

tl.type_labour_ID as labour_type_id

from permanent_crop  as pc

inner join region as r on

pc.NutsID = r.NutsID

inner join permanent_crop_name as pcn on

pc.pc_name_ID = pcn.pc_name_ID

inner join temporary_crop as tc on

r.NutsID = tc.NutsID and pc.`year` = tc.`year`

inner join temporary_crop_name as tcn on

tc.tc_name_ID = tcn.tc_name_ID

inner join labour as l on

r.NutsID = l.NutsID and pc.`year` = l.`year`

inner join type_labour as tl on

l.type_labour_ID = tl.type_labour_ID ;
```

5.1.2 Permanent crop area – labor force: table

Total labor force related to the total of the permanent crop area in each municipality (NUTS4)

```
#create table:permcrop_interactive

create table permcrop_value

select

r.region_name as municipality,

pc.`year`,

pc.area as area_ha,

l.value

from

permanent_crop pc

inner join permanent_crop_name pcn on
```

```
pc.pc_name_ID = pcn.pc_name_ID
inner JOIN region r on
pc.NutsID = r.NutsID
inner join region_level rl on
r.level_ID = rl.level_ID
inner join labour l on
l.NutsID = r.NutsID
WHERE
rl.region_level = 'municipality'
And
pc.pc_name_id= '1'
l.type_labour_ID = '1'
AND
pc.`year` = l.`year`
GROUP BY
r.region_name,
pc.`year`,
l.value;

#analises
SELECT *, pc_area / tc_area AS ratio_pc_tc
FROM ratio_labour
WHERE pc_name = 'total' AND tc_name = 'total' and level_ID = 3 and labour_type_id=1 and `year`=2019;


#landuse development
SELECT *, pc_area / tc_area AS ratio_pc_tc
FROM ratio_labour
WHERE pc_name = 'total' AND tc_name = 'total' and level_ID = 2 and labour_type_id=1;
```

Relation between NUTS2, NUTS3 and municipalities

```
use dms_2022;
select
        n.Name as NUTS2,
        n2.Name as NUTS3,
        n3.Name as municipality
from
        NUTS2 n
inner join NUTS3 n2 on
        n.NutsID = n2.ParentCodeID
inner join NUTS4 n3 on
```

```
        n2.NutsID = n3.ParentCodeID;
```

5.2 Python code

5.2.1 Ratio permanent to temporary cropland: visualization and regression

```
import numpy as np
import pandas as pd
import scipy.stats as sts
import statsmodels.stats as stm
import scikit_posthocs as sp
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.formula.api import ols
import seaborn as sns # for plotting
from scipy import stats # to compute statistics
df = pd.read_csv('pc_tc_labour_2.csv')
print(df)
sns.lmplot(x="year",
           y="ratio_pc_tc",
           hue="region_name",
           data=df,
           height=10)
plt.xlabel("Ratio permanent crop:temporary crop")
plt.ylabel("Year")
sns.lineplot(x='year', y='ratio_pc_tc', data=df, hue='region_name', marker='d')
plt.axhline(y=1, color='black', linestyle='--')
plt.xticks([1989,1999,2009,2019])
landuse2 = pd.read_csv('pc_tc_labour_2.csv')
print(landuse2)
y=landuse2["ratio_pc_tc"]
x=landuse2[["year"]]

x = sm.add_constant(x) # adding a constant (Intercept)

model = sm.OLS(y, x).fit()
predictions = model.predict(x)

print_model = model.summary()
print(print_model)
# import formula api as alias smf
import statsmodels.formula.api as smf

# formula: response ~ predictor1 + predictor2 + ...
model = smf.ols(formula='ratio_pc_tc ~ year', data=landuse3).fit()
print_model = model.summary()
print(print_model)
# formula: response ~ predictor
model2 = smf.ols(formula='ratio_pc_tc ~ year', data=landuse3).fit()
fig = sm.graphics.plot_partregress_grid(model2)
fig.tight_layout(pad=1.0)
fig = sm.graphics.plot_fit(model, "year")
fig.tight_layout(pad=1.0)
# Code to produce functions to run diagnostic plots
# https://www.statsmodels.org/dev/examples/notebooks/generated/linear_regression_diagnostics_plots.html

# base code
import numpy as np
import seaborn as sns
from statsmodels.tools.tools import maybe_unwrap_results
from statsmodels.graphics.gofplots import ProbPlot
from statsmodels.stats.outliers_influence import variance_inflation_factor
import matplotlib.pyplot as plt
from typing import Type
import statsmodels

style_talk = 'seaborn-talk'    #refer to plt.style.available

class Linear_Reg_Diagnostic():
    """
    Diagnostic plots to identify potential problems in a linear regression fit.
    Mainly,
        a. non-linearity of data
        b. Correlation of error terms
        c. non-constant variance
        d. outliers
        e. high-leverage points
        f. collinearity
```

```
    Author:
        Prajwal Kafle (p33ajkafle@gmail.com, where 3 = r)
        Does not come with any sort of warranty.
        Please test the code one your end before using.
    """

    def __init__(self,
                 results: Type[statsmodels.regression.linear_model.RegressionResultsWrapper]) -> None:
        """
        For a linear regression model, generates following diagnostic plots:

        a. residual
        b. qq
        c. scale location and
        d. leverage

        and a table

        e. vif

        Args:
            results (Type[statsmodels.regression.linear_model.RegressionResultsWrapper]):
                must be instance of statsmodels.regression.linear_model object

        Raises:
            TypeError: if instance does not belong to above object

        Example:
        >>> import numpy as np
        >>> import pandas as pd
        >>> import statsmodels.formula.api as smf
        >>> x = np.linspace(-np.pi, np.pi, 100)
        >>> y = 3*x + 8 + np.random.normal(0,1, 100)
        >>> df = pd.DataFrame({'x':x, 'y':y})
        >>> res = smf.ols(formula= "y ~ x", data=df).fit()
        >>> cls = Linear_Reg_Diagnostic(res)
        >>> cls(plot_context="seaborn-paper")

        In case you do not need all plots you can also independently make an individual plot/table
        in following ways

        >>> cls = Linear_Reg_Diagnostic(res)
        >>> cls.residual_plot()
        >>> cls.qq_plot()
        >>> cls.scale_location_plot()
        >>> cls.leverage_plot()
        >>> cls.vif_table()
        """

        if isinstance(results, statsmodels.regression.linear_model.RegressionResultsWrapper) is False:
            raise TypeError("result must be instance of
statsmodels.regression.linear_model.RegressionResultsWrapper object")

        self.results = maybe_unwrap_results(results)

        self.y_true = self.results.model.endog
        self.y_predict = self.results.fittedvalues
        self.xvar = self.results.model.exog
        self.xvar_names = self.results.model.exog_names

        self.residual = np.array(self.results.resid)
        influence = self.results.get_influence()
        self.residual_norm = influence.resid_studentized_internal
        self.leverage = influence.hat_matrix_diag
        self.cooks_distance = influence.cooks_distance[0]
        self.nparams = len(self.results.params)


    def __call__(self, plot_context='seaborn-paper'):
        # print(plt.style.available)
        with plt.style.context(plot_context):
            fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(10,10))
            self.residual_plot(ax=ax[0,0])
            self.qq_plot(ax=ax[0,1])
            self.scale_location_plot(ax=ax[1,0])
            self.leverage_plot(ax=ax[1,1])
            plt.show()

        self.vif_table()
        return fig, ax
```

```
def residual_plot(self, ax=None):
    """
    Residual vs Fitted Plot

    Graphical tool to identify non-linearity.
    (Roughly) Horizontal red line is an indicator that the residual has a linear pattern
    """
    if ax is None:
        fig, ax = plt.subplots()

    sns.residplot(
        x=self.y_predict,
        y=self.residual,
        lowess=True,
        scatter_kws={'alpha': 0.5},
        line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8},
        ax=ax)

    # annotations
    residual_abs = np.abs(self.residual)
    abs_resid = np.flip(np.sort(residual_abs))
    abs_resid_top_3 = abs_resid[:3]
    for i, _ in enumerate(abs_resid_top_3):
        ax.annotate(
            i,
            xy=(self.y_predict[i], self.residual[i]),
            color='C3')

    ax.set_title('Residuals vs Fitted', fontweight="bold")
    ax.set_xlabel('Fitted values')
    ax.set_ylabel('Residuals')
    return ax


def qq_plot(self, ax=None):
    """
    Standarized Residual vs Theoretical Quantile plot

    Used to visually check if residuals are normally distributed.
    Points spread along the diagonal line will suggest so.
    """
    if ax is None:
        fig, ax = plt.subplots()

    QQ = ProbPlot(self.residual_norm)
    QQ.qqplot(line='45', alpha=0.5, lw=1, ax=ax)

    # annotations
    abs_norm_resid = np.flip(np.argsort(np.abs(self.residual_norm)), 0)
    abs_norm_resid_top_3 = abs_norm_resid[:3]
    for r, i in enumerate(abs_norm_resid_top_3):
        ax.annotate(
            i,
            xy=(np.flip(QQ.theoretical_quantiles, 0)[r], self.residual_norm[i]),
            ha='right', color='C3')

    ax.set_title('Normal Q-Q', fontweight="bold")
    ax.set_xlabel('Theoretical Quantiles')
    ax.set_ylabel('Standardized Residuals')
    return ax

def scale_location_plot(self, ax=None):
    """
    Sqrt(Standarized Residual) vs Fitted values plot

    Used to check homoscedasticity of the residuals.
    Horizontal line will suggest so.
    """
    if ax is None:
        fig, ax = plt.subplots()

    residual_norm_abs_sqrt = np.sqrt(np.abs(self.residual_norm))

    ax.scatter(self.y_predict, residual_norm_abs_sqrt, alpha=0.5);
    sns.regplot(
        x=self.y_predict,
        y=residual_norm_abs_sqrt,
        scatter=False, ci=False,
        lowess=True,
        line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8},
        ax=ax)
```

```
    # annotations
    abs_sq_norm_resid = np.flip(np.argsort(residual_norm_abs_sqrt), 0)
    abs_sq_norm_resid_top_3 = abs_sq_norm_resid[:3]
    for i in abs_sq_norm_resid_top_3:
        ax.annotate(
            i,
            xy=(self.y_predict[i], residual_norm_abs_sqrt[i]),
            color='C3')
    ax.set_title('Scale-Location', fontweight="bold")
    ax.set_xlabel('Fitted values')
    ax.set_ylabel(r'$\sqrt{|\mathrm{Standardized\ Residuals}|}$');
    return ax

def leverage_plot(self, ax=None):
    """
    Residual vs Leverage plot

    Points falling outside Cook's distance curves are considered observation that can sway the fit
    aka are influential.
    Good to have none outside the curves.
    """
    if ax is None:
        fig, ax = plt.subplots()

    ax.scatter(
        self.leverage,
        self.residual_norm,
        alpha=0.5);

    sns.regplot(
        x=self.leverage,
        y=self.residual_norm,
        scatter=False,
        ci=False,
        lowess=True,
        line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8},
        ax=ax)

    # annotations
    leverage_top_3 = np.flip(np.argsort(self.cooks_distance), 0)[:3]
    for i in leverage_top_3:
        ax.annotate(
            i,
            xy=(self.leverage[i], self.residual_norm[i]),
            color = 'C3')

    xtemp, ytemp = self.__cooks_dist_line(0.5) # 0.5 line
    ax.plot(xtemp, ytemp, label="Cook's distance", lw=1, ls='--', color='red')
    xtemp, ytemp = self.__cooks_dist_line(1) # 1 line
    ax.plot(xtemp, ytemp, lw=1, ls='--', color='red')

    ax.set_xlim(0, max(self.leverage)+0.01)
    ax.set_title('Residuals vs Leverage', fontweight="bold")
    ax.set_xlabel('Leverage')
    ax.set_ylabel('Standardized Residuals')
    ax.legend(loc='upper right')
    return ax

def vif_table(self):
    """
    VIF table

    VIF, the variance inflation factor, is a measure of multicollinearity.
    VIF > 5 for a variable indicates that it is highly collinear with the
    other input variables.
    """
    vif_df = pd.DataFrame()
    vif_df["Features"] = self.xvar_names
    vif_df["VIF Factor"] = [variance_inflation_factor(self.xvar, i) for i in range(self.xvar.shape[1])]

    print(vif_df
            .sort_values("VIF Factor")
            .round(2))


def __cooks_dist_line(self, factor):
    """
    Helper function for plotting Cook's distance curves
    """
    p = self.nparams
    formula = lambda x: np.sqrt((factor * p * (1 - x)) / x)
    x = np.linspace(0.001, max(self.leverage), 50)
```

```
        y = formula(x)
        return x, y
#Now we can run the diagnostic plots to our model:
# Run diagnostic plots
cls = Linear_Reg_Diagnostic(model)
fig, ax = cls()
```

5.2.2 Permanent crop area – labor force: Interactive analysis

DATA VISUALISATION

This notebook allowed for a better analitic comprehension of the data

---

The vatrious plots presented are based in the class scripts provided by the teacher but were addapted to the subdject matter

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from bokeh.io import curdoc, output_notebook
from bokeh.plotting import figure, show
from bokeh.models import HoverTool, ColumnDataSource, CategoricalColorMapper, Slider
from bokeh.palettes import Category20
from bokeh.layouts import column, row
```

Load the table area_labour.csv and nuts2_to_mun.csv

```
rg = '/Users/afonsomarques/mestrado/2_semestre/AVCA-DE/greends-
avcd/people/AfonsoMarques/final_proj/data/nuts2_to_mun.csv'
rg_pd = pd.read_csv(rg)
perm_labour = '/Users/afonsomarques/mestrado/2_semestre/AVCA-DE/greends-
avcd/people/AfonsoMarques/final_proj/data/area_labour.csv'
perm_labour_pd = pd.read_csv(perm_labour)
rg_pd
perm_labour_pd
```

Merge them together to easen the comprehension

```
data = pd.merge(rg_pd, perm_labour_pd, on='municipality')
data = data.dropna()
data.head()
output_notebook()
```

Get a palette with enough colors

```
# Get a palette with 20 colors
palette = Category20[20]
```

Assign a different color to each NUTS2 entry

```
# create list of regions - to color the datapoints based on the region
NUTS2_LIST = data.NUTS2.unique().tolist()
# assign colors to each region
color_mapper = CategoricalColorMapper(factors=NUTS2_LIST, palette=palette)
```

Crete the source dataset for the table

```
# make a data source for the plot
#for the year either choose
#####  1989, 1999, 2009, 2019
year = 1989

source = ColumnDataSource(data={
    'x': data.value[data['year'] == year],
    'y': data.area_ha[data['year'] == year],
    'municipality': data.municipality[data['year'] == year],
    'NUTS2': data.NUTS2[data['year'] == year],
```

```
})
# Save the minimum and maximum values of the gdp column: xmin, xmax
xmin, xmax = min(data.value), max(data.value)
print(xmax )

# Save the minimum and maximum values of the co2 column: ymin, ymax
ymin, ymax = min(data.area_ha), max(data.area_ha)
print(ymax)
# Create the figure: plot
plot = figure(title='permanent crop area vs labour',
              height=600, width=1000,
              x_range=(xmin, xmax),
              y_range=(ymin, ymax), y_axis_type='log')
import numpy as np
from scipy.stats import linregress

from bokeh.plotting import figure, show, output_file, save
# Add circle glyphs to the plot
plot.circle(x='x', y='y', fill_alpha=0.8, source=source, legend_field='NUTS2',
            color=dict(field='NUTS2', transform=color_mapper),
            size=7)
```

GlyphRenderer(

id = 'p16564', …)

This the set of code that makes the actual graph saves and show it

One of the best particularities of this graph is that the axis are variable which allows the points to be more disperse, This helps a lot with the visual analysis

```
# Set the legend.location attribute of the plot
plot.legend.location = 'bottom_right'

# Set the x-axis label
plot.yaxis.axis_label = 'area in hectars'

# Set the y-axis label
plot.xaxis.axis_label = 'labour units'

# Create a HoverTool - will allow the user to hover above a datapoint to see the name of the country, CO2
emissions nd GDP
hover = HoverTool(tooltips=[('Municipality', '@municipality'), ('labour', '@x'), ('area', '@y')])

# Add the HoverTool to the plot
plot.add_tools(hover)

output_file(filename="custom_filename.html", title="Static HTML file")

save(plot, "bokeh_plot_1989.html") # save html plot

show(plot) # show plot in the web browser
```

Here we are usinga different visaualization technic with plotly In this one each NUTS2 entry is descriminated in order to see the evalution of each one through time

```
# Save the minimum and maximum values of the gdp column: xmin, xmax
xmin, xmax = min(data.value), max(data.value)
# Save the minimum and maximum values of the co2 column: ymin, ymax
ymin, ymax = min(data.area_ha), max(data.area_ha)
import pandas as pd
import plotly.express as px

fig = px.scatter(data, x="value", y="area_ha", animation_frame="year", animation_group="municipality",
           color="NUTS2", hover_name="municipality", facet_col="NUTS2", width=6000, height=400,
           log_x=False, size_max=20, range_x=[xmin,xmax], range_y=[ymin,ymax])


fig.show()
```

In this one is just a diffeent format of the same graph that was showed in first but now with a slider to tool, a box plot in the left and a rug plot on top

This one is really good beacuse of the time slider despite the points being all clusterd becuse its a set scale in comparison with bokeh one

```
fig = px.scatter(data, x="value", y="area_ha", animation_frame="year",
        color="NUTS2", hover_name="municipality", width=1000, height=600,
        size_max=45, range_x=[xmin,xmax], range_y=[ymin,ymax],
        marginal_y = 'box', marginal_x = 'rug')
```

```
fig.show()
```

5.2.3 Permanent crop area – labor force: Geographic visualization