

THE INTERNATIONAL MASTER OF BUSINESS AND ADMINISTRATION

Analysis of birth rates across Europe from 2017 to 2021

Ahmed Fauzia Sabrina

Hamzeh Anahita

Ilie Georgiana Nicole

# Table of Contents

[**Introduction**](#_s7t4pt3x47al) **2**

[**Rationale**](#_vgpud3qphsrc) **3**

[Research Question: To what extent is the number of births determined by the GDP per capita in Europe?](#_djimt9vl4k1n) 3

[**Methods and Data Collection**](#_3u64kumrdz49) **3**

[The first set of data](#_7sejdjnrhzmy) 3

[● Creating graphs](#_3alulricx877) 5

[● Mean birth rates](#_quwjb9q5412z) 10

[The second set of data](#_73vv3a3yixo8) 11

[4. Merging the two datasets](#_tg7o8mtrxccx) 12

[6. Correlation matrix heatmap](#_easy7yy1fbvn) 13

[**Linear Regression**](#_3ptmpp7ol337) **15**

[Step 9: Plotting the residuals](#_uz0jpemol9hg) 20

[**Analysis, Interpretations and Conclusions**](#_psquwb346lax) **21**

# 

# Introduction

From 2017 to 2021, Europe experienced a significant shift in birth rates, and the COVID-19 pandemic played a crucial role in shaping this trend. The pandemic brought about numerous societal changes, including shifts in family planning, economic uncertainties, and disruptions to healthcare systems. This essay delves into the dynamics of the European birth rate from 2017 to 2021, utilizing Python for data-driven insights. Global challenges mark the period under scrutiny, prominently the COVID-19 pandemic, adding a distinctive context to our analysis. Through Python's analytical prowess, we aim to identify trends, highlight countries with the highest and lowest birth rates, and unravel the multifaceted factors shaping demographic patterns.

Economic stability is a crucial influencer of birth rates, and Europe's diverse economic landscape provides a compelling backdrop. Employing Python's data analysis tools, we will quantitatively explore correlations between economic fluctuations and fertility choices during the specified period. This approach allows a nuanced understanding of how financial considerations impact birth rates across European nations.

Cultural dynamics further shape reproductive choices, and Python's data visualization capabilities will aid in uncovering these patterns. We can discern the intricate interplay between societal norms and demographic outcomes by isolating trends within countries where traditional values coexist with progressive social policies.

Python will be instrumental in examining how disruptions in healthcare systems, economic uncertainties, and societal upheavals influenced birth rates during this extraordinary period. We can unravel the complexities of demographic responses to global health crises by quantifying the pandemic's impact on family planning decisions.

In conclusion, this essay employs Python to briefly analyze European birth rates from 2017 to 2021, narrowing the analysis on Romania and its neighbouring countries and employing an independent variable - GDP per capita to identify the degree of relationship between birth rates and GDP per capita.. By utilizing data-driven approaches, we aim to uncover overarching trends and highlight countries with the highest and lowest birth rates. This condensed exploration offers a granular perspective on the intricate forces shaping demographic trajectories in Europe amidst the challenges posed by the pandemic.

## 

# Rationale

This research aims to comprehensively analyze European birth rates from 2017 to 2021, and a correlation between these birth rates and the Gross Domestic Product per capita. The study relies on quantitative secondary data, which was obtained from Eurostat. With the help of Python, we can gather information and process it, displaying it in graphs to further depict and interpret the findings.

To investigate this matter, the correlation between births and the GDP per capita has been studied using a few key input variables, both dependent and independent (i.e. no. of births, GDP per capita, years from 2017-2021, European countries).

Therefore, 3 statistical methods have been performed for analyzing our dataset: Descriptive Statistics, Correlation Heatmap, Plot graph and Simple Linear Regression.

## Research Question: *To what extent is the number of births determined by the GDP per capita in Europe?*

# Methods and Data Collection

## The first set of data

The first set of data, extracted from Eurostat, was evaluated and used as the main dataset. It includes a simple data set of all European countries and the number of births during the years 2017-2021.

* Import data for birth rates into Python for Descriptive Statistics

**Code:**

import pandas as pd

from matplotlib import pyplot as plt

df = pd.read\_csv("/content/drive/MyDrive/birth\_rates.csv")

print(df)

* Analysing raw data

**Code:**

df.columns

Index(['DATAFLOW', 'LAST UPDATE', 'freq', 'indic\_de', 'geo', 'TIME\_PERIOD',

'OBS\_VALUE', 'OBS\_FLAG'],

dtype='object')

**Code:**

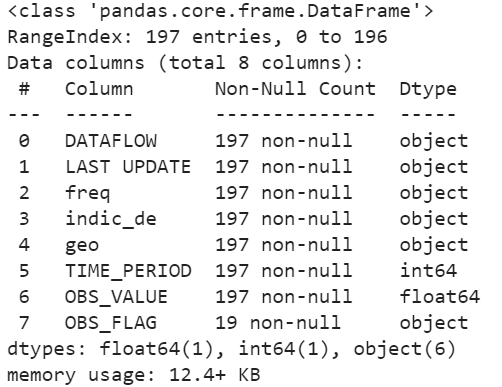
shape = df.shape

shape

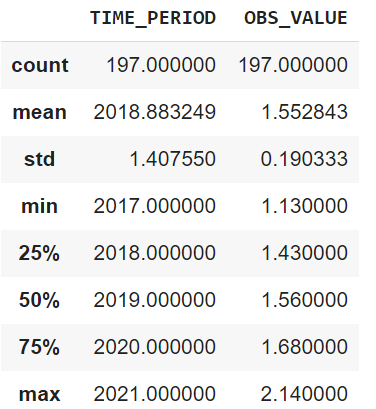
(197, 8) - rows and columns

info = df.info()

info



df.describe()



x = df['OBS\_VALUE'].mean()

print("The mean birth rate in Europe is :", x)

The mean birth rate in Europe is : 1.552842639593909

### Creating graphs

1. **Creating a chart displaying the grouped countries' birth rates from 2017 to 2021.**

**Code:**

df.plot(figsize=(16,4),x='geo',y='OBS\_VALUE')

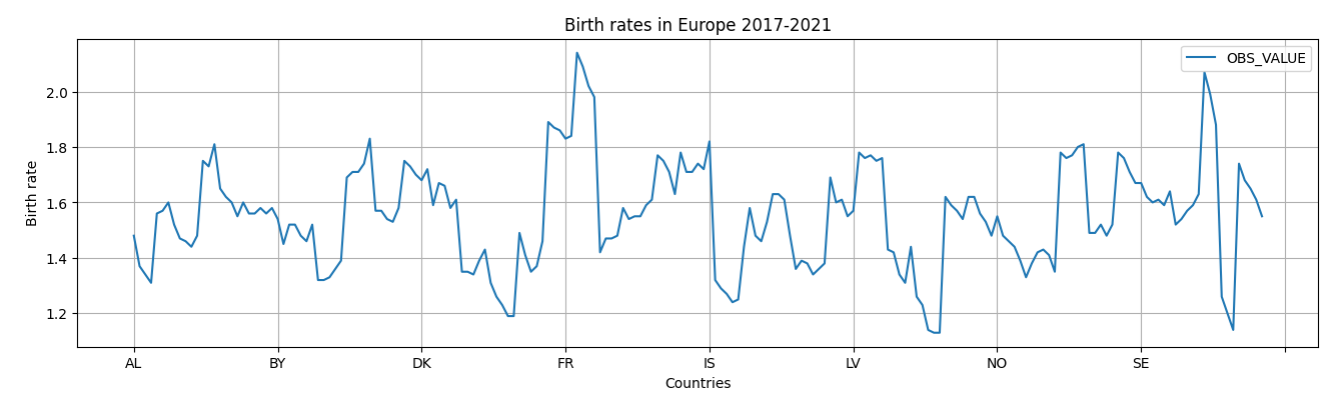
plt.title('Birth rates in Europe 2017-2021')

plt.ylabel('Birth rate')

plt.xlabel('Countries')

plt.grid()

**Displayed in Python:**

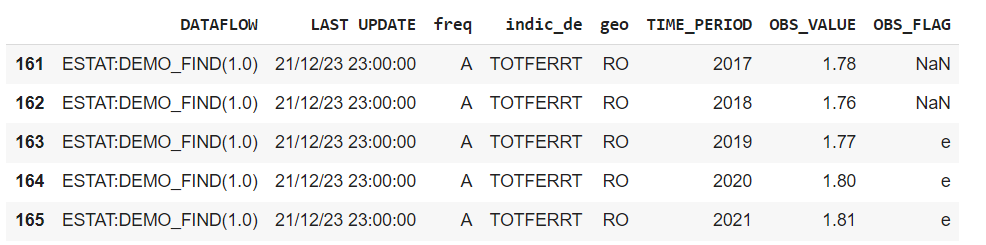


After we analyzed the raw data, the next step was to create a line graph that shows the evolution of the birth rates only in Romania from 2017 to 2021. And we have observed that starting with 2018, there was a high increase in birth rates.

**Code:**

df2 = df.loc[df['geo'] == 'RO']

df2

****

# changing the time period values from floating numbers to strings

df1 = df['TIME\_PERIOD'].astype(str)

# plotting a line graph that shows birth rates only in Romania from 2017 to 2021

plt.plot(df2['TIME\_PERIOD'].astype(str),df2['OBS\_VALUE'])

plt.plot(figsize=(12,6))

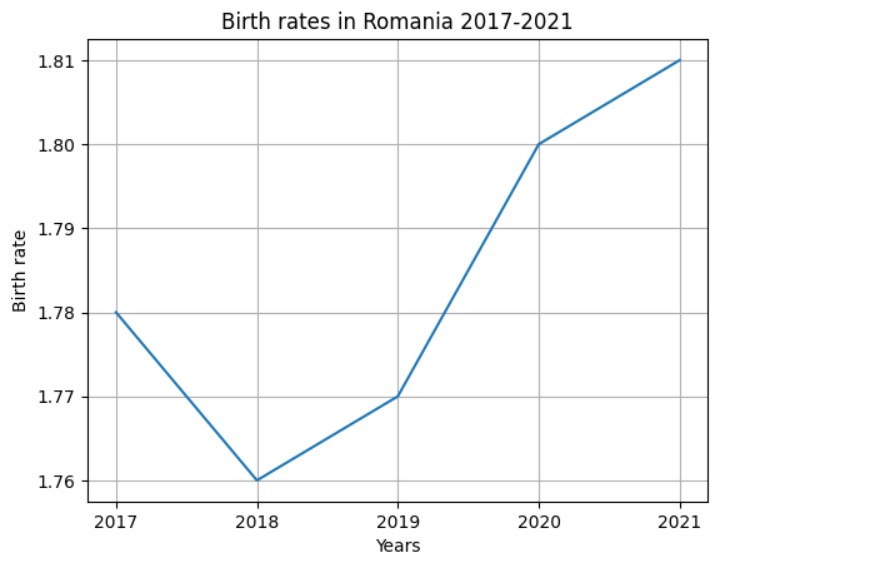
plt.title('Birth rates in Romania 2017-2021')

plt.ylabel('Birth rate')

plt.xlabel('Years')

plt.grid()

**Displayed in Python:**



Furthermore, we have created a new dataset with the countries, year and rates.

**Code:**

# creating a new dataset with only the counties, years and birth rates

nd = df[['geo','TIME\_PERIOD','OBS\_VALUE']]

nd

**Displayed in Python:**

We selected only the data relevant for 2019 and 2020:

**Code:**

# selecting only the data relevant for 2019 and 2020

Y2019 = nd.loc[nd['TIME\_PERIOD'] == 2019]

Y2020 = nd.loc[nd['TIME\_PERIOD'] == 2020]

print(Y2019,Y2020)

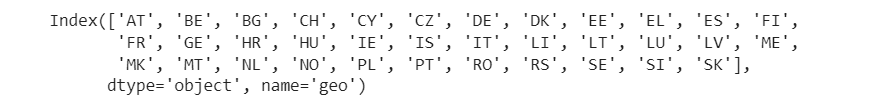
In determining the countries present in both years, we have used the following code:

# determining the countries present in both years

EU\_countries = pd.Index(Y2019.geo).intersection(pd.Index(Y2020.geo))

EU\_countries

**Displayed in Python:**

****

We created a list with all EU countries:

**Code:**

# creating a list with those countires

EU\_countries\_list = list(EU\_countries)

EU\_countries\_list

Moreover we have created a dataset for comparing birth rates in 2019 versus 2020 for the listed countries above:

**Code:**

**# creating a dataset to compare the birth rates of countries in 2019 and 2020**

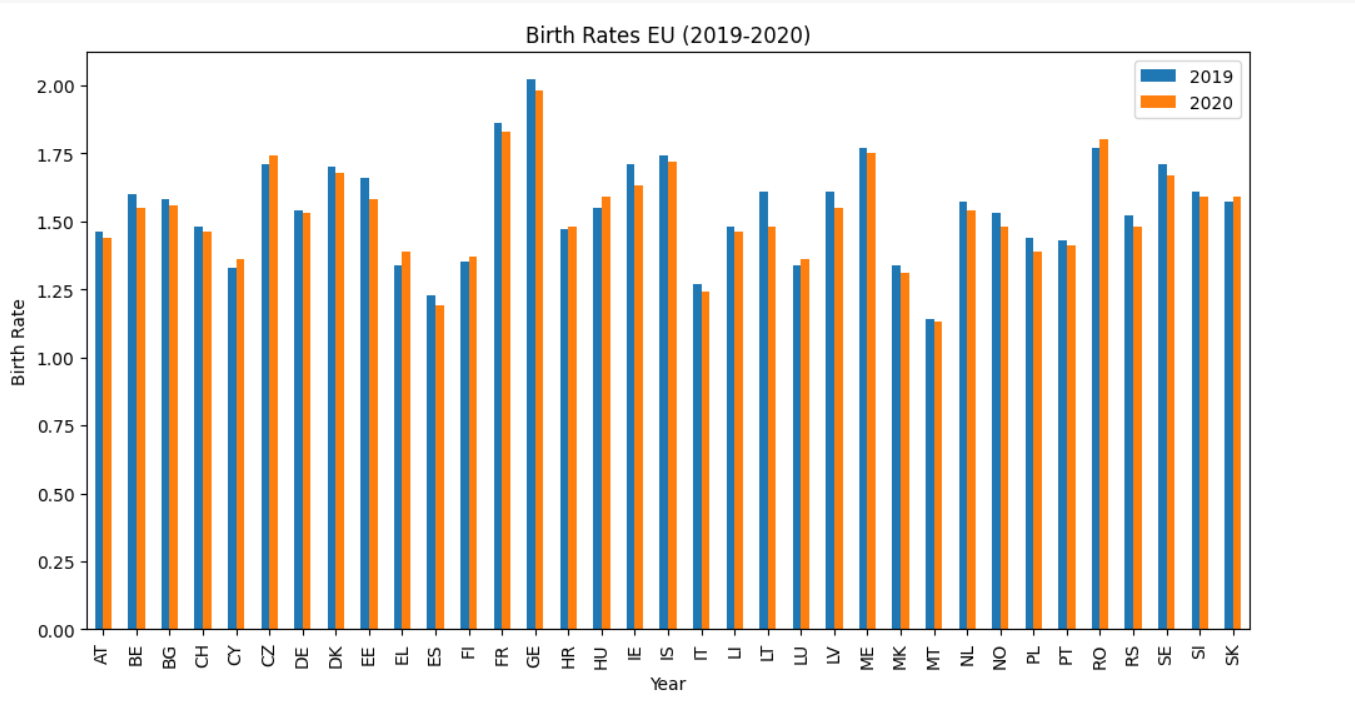
**Data2019 = Y2019.loc[Y2019['geo'].str.contains('|'.join(EU\_countries\_list))]**

**Data2020 = Y2020.loc[Y2020['geo'].str.contains('|'.join(EU\_countries\_list))]**

**data = {'Country':EU\_countries\_list,'2019':list(Data2019.OBS\_VALUE),'2020':list(Data2020.OBS\_VALUE)}**

**mean\_values = pd.DataFrame(data)**

**mean\_values**

****

**Code:**

# creating a dataset with Romania and its neighbouring countries

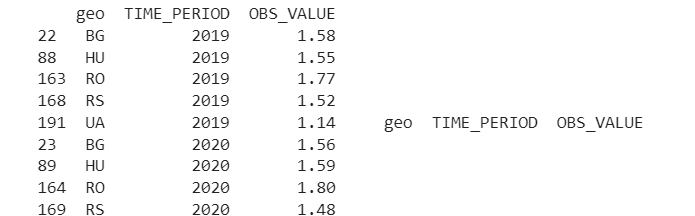
countries\_of\_interest = ['RO','MD','BG','RS','HU','UA']

InterestData2019 = Y2019.loc[Y2019['geo'].str.contains('|'.join(countries\_of\_interest))]

InterestData2020 = Y2020.loc[Y2020['geo'].str.contains('|'.join(countries\_of\_interest))]

print(InterestData2019, InterestData2020)

**Displayed in Python:**



**Code:**

# eliminating countries which do not have data for both years

Neighbouring\_countries = pd.Index(InterestData2019.geo).intersection(pd.Index(InterestData2020.geo))

Neighbouring\_countries

**Displayed in Python:**



Moreover, we aimed to compare Romania’s birth rate with those of the neighbouring countries.

We identified that Romania has a higher birth rate than its neighbours.

**Code:**

# plotting a bar chart for better visualization

mean\_values.plot(figsize=(10,4),x='Country', y=['2019','2020'],kind='bar')

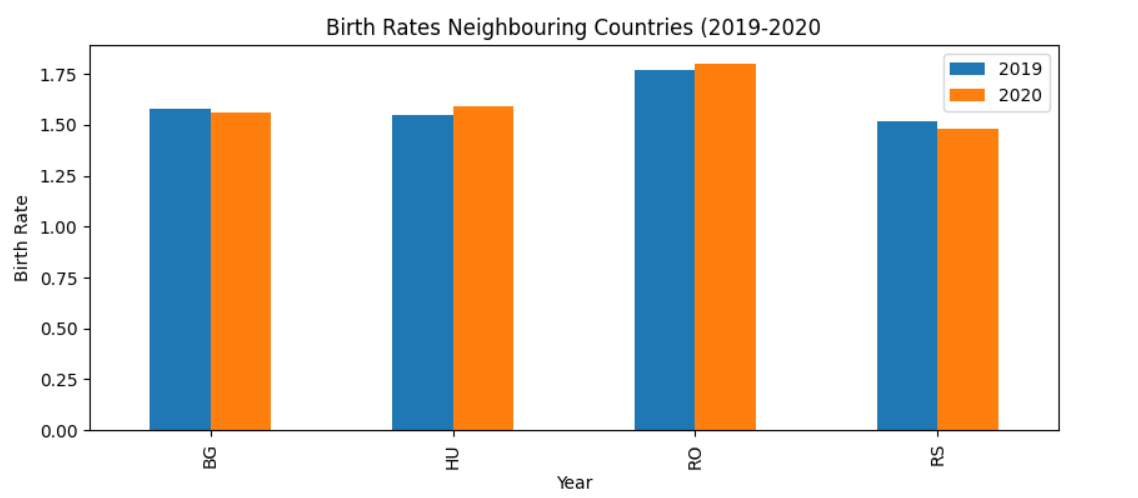
plt.title("Birth Rates Neighbouring Countries (2019-2020")

plt.xlabel("Year")

plt.ylabel("Birth Rate")

plt.show()

**Displayed in Python:**



We observe from the above descriptive statistics, that Romania has a higher birth rate than its neighbouring countries in both years 2019 and 2020. Some potential factors that might influence this performance may be related to: access to better healthcare facilities and services that often leads to lower infant and child mortality rates, which can motivate couples to have more children, higher economic stability and prosperity, families may feel more secure and capable of having and raising more children, leading to an increase in the birth rate, societal and cultural norms promoting large families which could include religious beliefs, traditional family values, or societal norms.

### Mean birth rates

Further, we wanted to find out the mean birth rate in a specific year and plot it on a line graph

**Code:**

def getYearlyData(year):

return df.loc[df['TIME\_PERIOD']==year]['OBS\_VALUE'].mean()

All\_Years = list(set(df.TIME\_PERIOD))

Year\_Mean\_Birth\_Rates = list(map(getYearlyData,All\_Years))

Year\_Mean\_Birth\_Rates

**Result:** [1.5830232558139536,

1.5651162790697677,

1.5505,

1.5180555555555557,

1.5391428571428574]

**Code:**

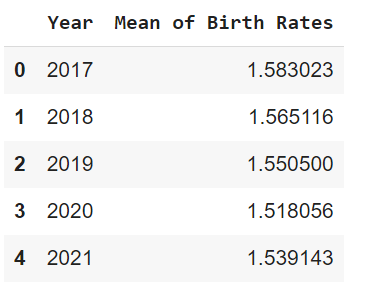
# creating a new dataset that groups the year and mean of birth rates

YearlyMeanDataset = {'Year': All\_Years,'Mean of Birth Rates':Year\_Mean\_Birth\_Rates}

YearlyDataFrame = pd.DataFrame(YearlyMeanDataset)

YearlyDataFrame

**Displayed in Python:**

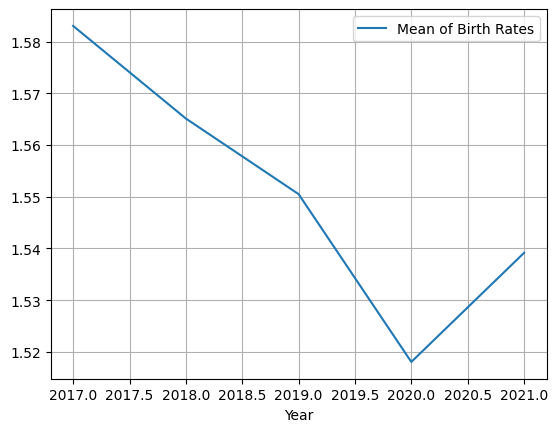
****

**Code:** # plotting a graph for better visualization

print(YearlyDataFrame.plot(x='Year',y='Mean of Birth Rates',kind='line'))

plt.grid()

**Displayed in Python:**



## The second set of data

For the second data set, we have extracted from Eurostat the GDP per capita in Europe. This will help us analyze the data and perform a heatmap correlation matrix and linear regression by merging the two data sets.

1. **Import the GDP per capita data in Europe**

**Code:**

# importing the dataset for GDP per capita in Europe

import pandas as pd

from matplotlib import pyplot as plt

df = pd.read\_csv("/content/drive/MyDrive/gdp\_europe.csv")

print(df)

1. **Plotting a line graph**

# plotting a line graph that shows GDP per capita from 2017 to 2021 grouped by countries

df.plot(figsize=(16,4),x='geo',y='OBS\_VALUE')

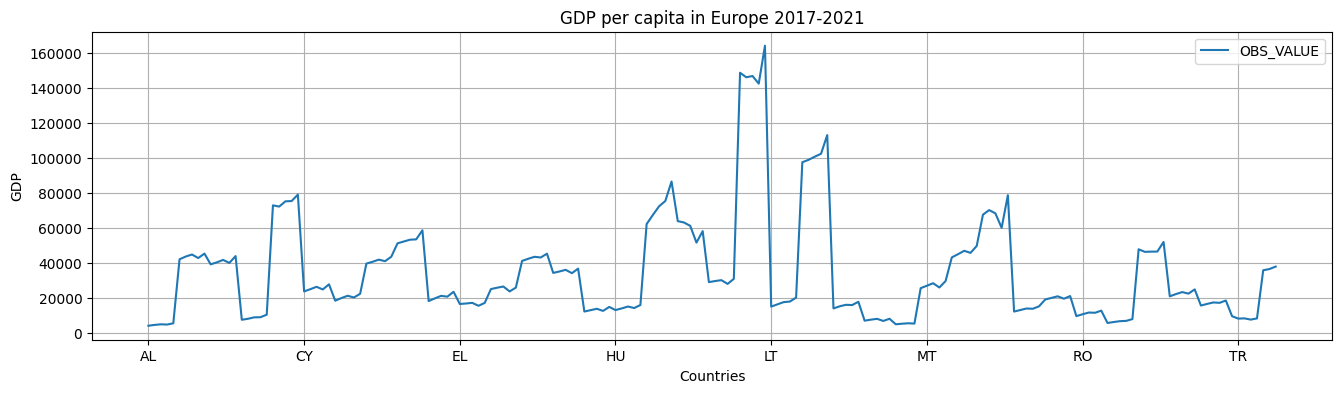
plt.title('GDP per capita in Europe 2017-2021')

plt.ylabel('GDP')

plt.xlabel('Countries')

plt.grid()

**Displayed in Python:**

****

1. **Renaming the column OBS\_VALUE so it does not overlap on the 2 datasets**

**Code:**

GDP\_per\_capita.rename(columns = {'OBS\_VALUE':'GDP'}, inplace = True)

GDP\_per\_capita

### Merging the two datasets

**Code:**

# changing the "OBS\_VALUE" in the birth rates dataset from floating to integer -> needed for merging the 2 datasets

Births['OBS\_VALUE'] = Births['OBS\_VALUE'].multiply(100).astype(int)

Births['TIME\_PERIOD'] = Births['TIME\_PERIOD'].astype(str)

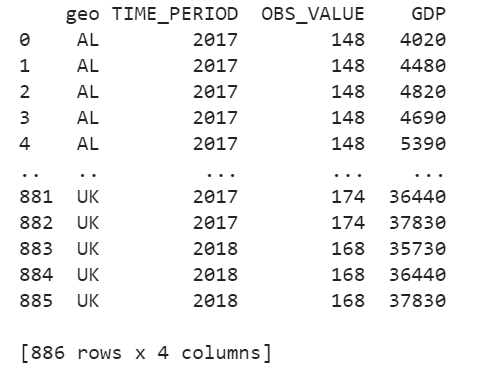
Births

# merging specific columns from the 2 datasets

df\_merged = Births.merge(GDP\_per\_capita[['geo', 'GDP']])

print(df\_merged)

**Displayed in Python:**



1. **Finding the correlation between the no.of births and GDP per capita**

df\_merged['OBS\_VALUE'].corr(df\_merged['GDP'])

Result: **-0.003077507814769421**

We can observe a negative correlation between the number of births and GDP per capita in Europe. This negative correlation means that GDP per capita will decrease when the number of births increases in countries across Europe. Still, the correlation indicator is close to -1. Thus, we can assume that there is a strong correlation between the two variables. However, there is an inverse relationship between them.

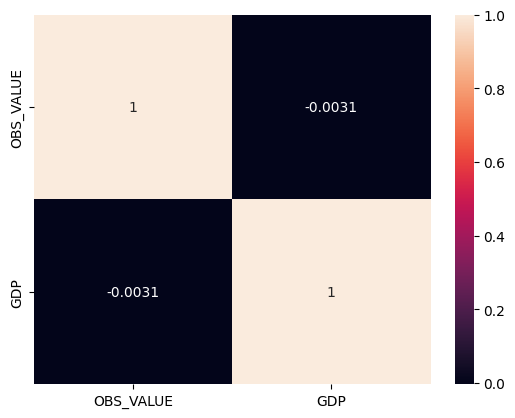
While it may seem logical to assume that higher-income countries would have lower fertility rates, the existing empirical research does not consistently support this assumption. It's possible that higher income could contribute to a decrease in the demand for children directly, but it's not accurate to affirm that a relationship between high income and low fertility rates is solely responsible for a country's fertility decline.

## Correlation matrix heatmap

import seaborn as sns

sns.heatmap(df\_merged.corr(),annot = True)

**Displayed in Python:**

****

1. **Plotting a data distribution for Births**

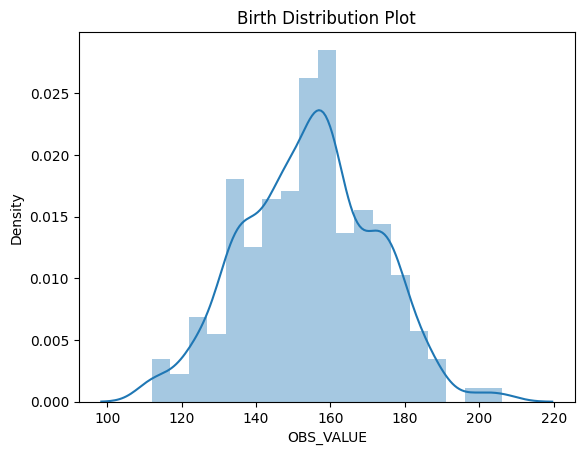
**Code:**

plt.title('Birth Distribution Plot')

sns.distplot(df\_merged['OBS\_VALUE'])

plt.show()

**Displayed in Python:**

****

1. **Plotting a scatter graph to see the relationship between Births and GDP per capita**

plt.scatter(df\_merged['OBS\_VALUE'], df\_merged['GDP'], color = 'lightcoral')

plt.title('Births vs GDP per capita')

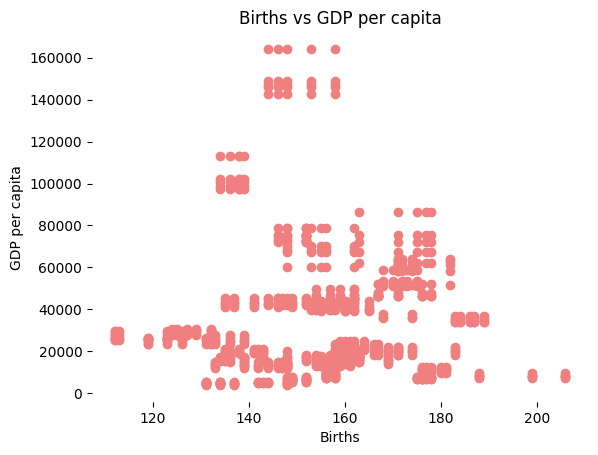
plt.xlabel('Births')

plt.ylabel('GDP per capita')

plt.box(False)

plt.show()

**Displayed in Python:**



# 

# Linear Regression

The primary method used in this analysis is linear regression, a basic yet powerful statistical and machine-learning tool. This paper centres its regression analysis around European birth rates, examining them as distinct observations.

The overarching goal of this study is to establish a relationship between specific attributes, hypothesizing that these elements are interconnected. It considers birth rates as an independent variable, while the associated GDP per capita is the dependent variable.

**Step 1: Generate table with Births and GDP per capita and define the X and y variables**

**Code:**

# making a table only with the birth rates and GDP per capita from the merged dataset

linear = df\_merged[['OBS\_VALUE', 'GDP']]

linear

# creating x and y variables

x = df\_merged['GDP'] #independent

y = df\_merged['OBS\_VALUE'] #dependent

linear\_new = pd.DataFrame({'X': x, 'Y': y})

linear\_new.head()

**Displayed in Python:**

****

**Step 2: Plot a scatter plot on it with our data**

**Code:**

# plotting the data and getting a current axis of the scatter graph

import numpy as np

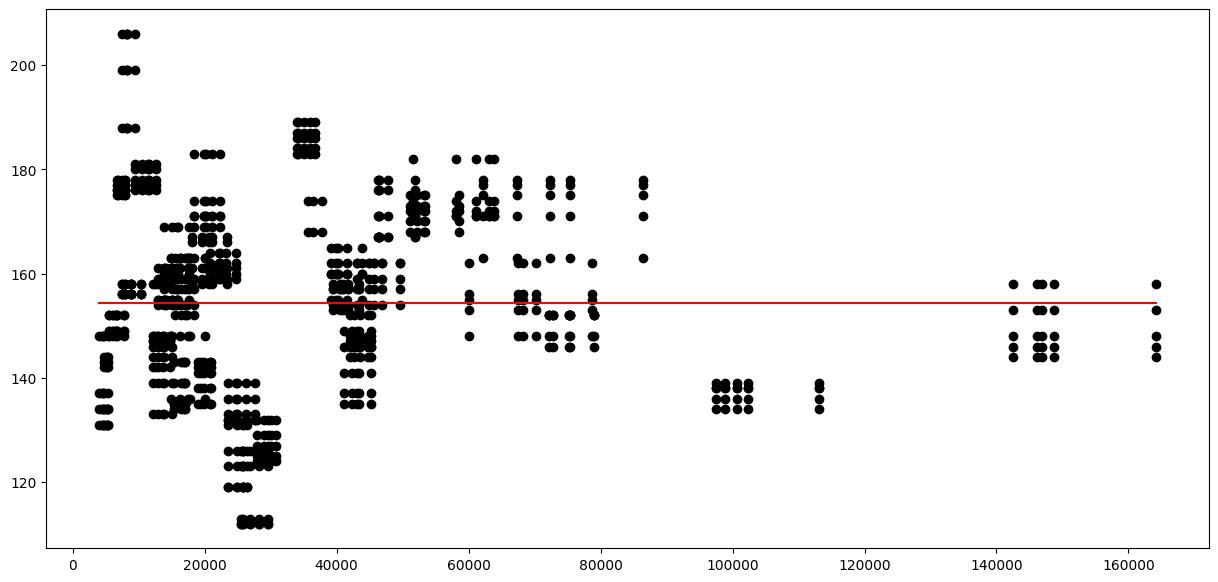
fig = plt.figure(figsize=(15,7))

ax = plt.gca()

ax.scatter(x, y, c ='k')

ax.plot((linear\_new['X'].min(), linear\_new['X'].max()),(np.mean(linear\_new['Y']), np.mean(linear\_new['Y'])), color='r');

**Displayed in Python:**

****

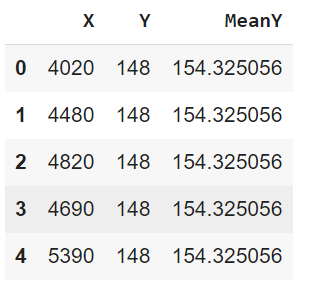
**Step 3: Finding the Y-intercept**

# finding the value that interceps with the y axis

linear\_new['MeanY'] = linear\_new['Y'].mean()

linear\_new.head()

**Displayed in Python:**

****

**Step 4: Calculating the SSE to see how much variation in y is left unexplained by the model**

**Code:**

np.sum(np.square(linear\_new['Y'] - linear\_new['MeanY']))

Result: 273590.3837471783

**Step 5: Calculating the MSE - how close a regression line is to a set of data points**

**Code:**

np.mean(np.square(linear\_new['Y'] - linear\_new['MeanY']))

Result: 308.7927581796595

**Step 6: Calculating the RMSE - the average difference between values predicted by a model and the actual values**

**Code:**

(np.mean(np.square(linear\_new['Y'] - linear\_new['MeanY']))) \*\* 0.5

Result: 17.57250005490566

**Step 7: Calculating the beta coefficients by hand**

**Code:**

y\_bar = linear\_new['Y'].mean()

x\_bar = linear\_new['X'].mean()

std\_y = np.std(linear\_new['Y'], ddof = 1)

std\_x = np.std(linear\_new['X'], ddof = 1)

r\_xy = linear\_new.corr().loc['X','Y']

beta\_1 = r\_xy\*(std\_y/std\_x)

beta\_0 = y\_bar - beta\_1\*x\_bar

linear\_new['Linear\_Y'] = beta\_0 + beta\_1 \* linear\_new['X']

np.square(linear\_new['Y'] - linear\_new['Linear\_Y']).mean()

Result: 308.7898335866639

**Step 8: Re- plotting the regression line**

**Code:**

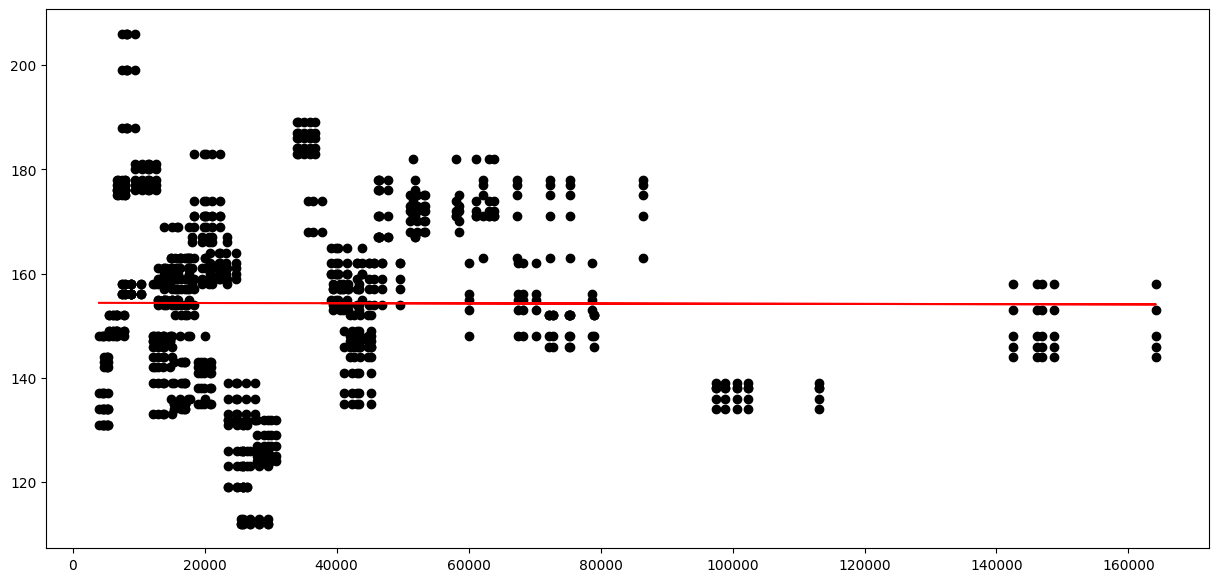
fig = plt.figure(figsize=(15,7))

ax = plt.gca()

ax.scatter(x, y, c='k')

ax.plot(linear\_new['X'], linear\_new['Linear\_Y'], color='r');

**Displayed in Python:**

****

### Step 9: Plotting the residuals

**Code:**

fig = plt.figure(figsize=(15,7))

fig.set\_figheight(8)

fig.set\_figwidth(15)

ax = fig.gca()

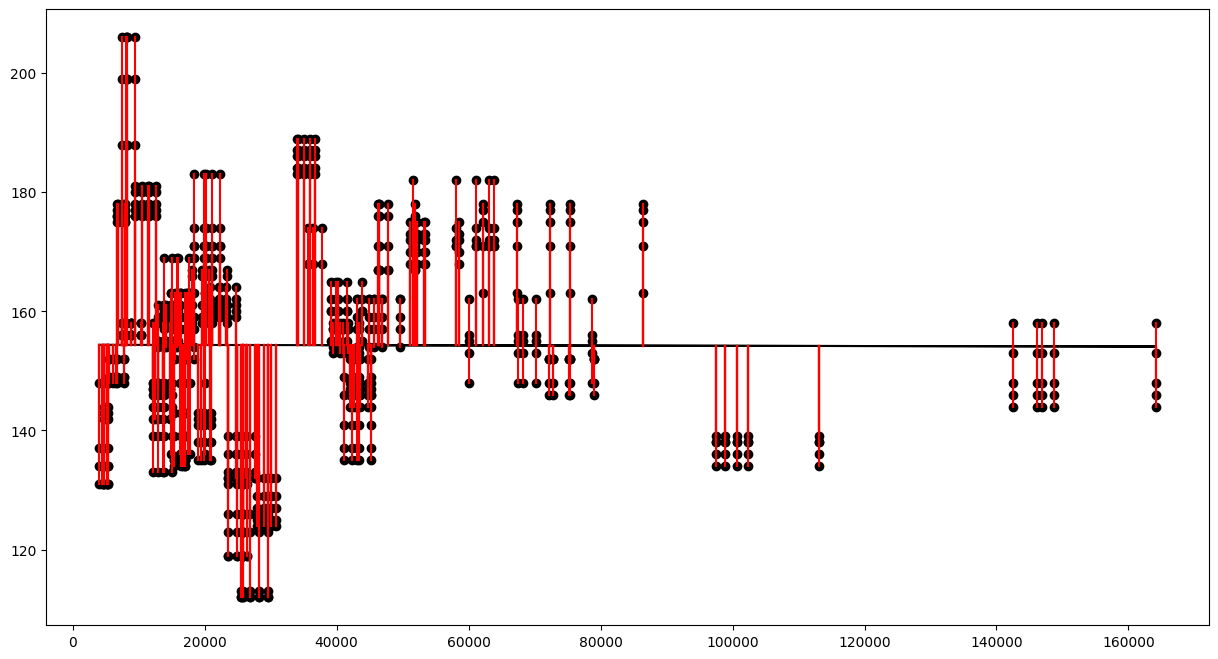
ax.scatter(x=linear\_new['X'], y=linear\_new['Y'], c='k')

ax.plot(linear\_new['X'], linear\_new['Linear\_Y'], color='k');

for \_, row in linear\_new.iterrows():

plt.plot((row['X'], row['X']), (row['Y'], row['Linear\_Y']), 'r-')

**Displayed in Python:**

****

# Analysis, Interpretations and Conclusions

How does our new MSE compare to our old MSE? Small difference between the two, hence the linear regression graph did not have any significant change. After calculating beta coefficients by hand, MSE slightly decreased, but not significantly. Some explanations could be related to the fact that the original model was already correctly specified, and your hand calculation yielded the same estimates for the beta coefficients. In this case, the model's predictions and the associated errors would remain the same, resulting in the same MSE.

Based on the results provided including SSE, MSE, and RMSE but not including any coefficients or other specifics about the model, we know that the linear regression model predicting GDP per capita based on birth rates has a certain degree of error. This error is particularly encapsulated in the RMSE value of 17.57 units. This tells us that, on average, the model's predictions of GDP per capita are approximately 17.57 units away from the actual observed values.

However, it's important to note that the appropriateness of this model and what it reveals about the relationship between GDP per capita and birth rates would require more information. Some of the information required includes the coefficients of the regression model (i.e., the intercept and slope), the p-values of these coefficients to determine their statistical significance, and other indicators of model fit such as R-squared, adjusted R-squared, or AIC/BIC values for model comparison.

Therefore, based solely on the results provided so far, aside from providing a broad measure of the model's accuracy, we can't yet make any specific claims about the relationship between GDP per capita and birth rates. A more thorough analysis of your regression output would be necessary.

In addition, interpreting such results should always be done in conjunction with a residuals analysis, cross-validation, or out-of-sample testing, all of which provide important context and validity checks for your model.