# Data Science 1
# SS 2020

Project Presentation - GDPR Fines

Team: NAN 101
Alexandru Bârsan (No.: 7396261)
Johannes-Jona Detjenn (No.: 6836948)

## Introduction

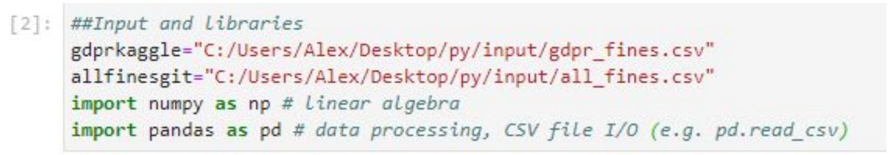
- The General Data Protection Regulation (GDPR) is an EU regulation on data protection and privacy in the European Union and the European Economic Area.
- Our records are about the given fines of the EU members.
- In this project we would like to find out which variables from both dataset relate with each other.
- For example if there are more articles applied, the value of the fine should also be growing. In our hypothesis this relation should also apply for the economic wealth of a country measured in Gross domestic product (GDP) and the total height of fines per country.

## Data acquisition

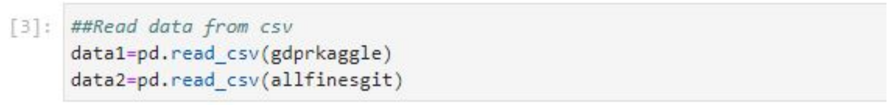
- The datasets which we used for this project are from Kaggle and a Github repository.
- The Kaggle dataset is a sample of given information from the site “www.enforcementtracker.com"
- The sample represents only about 15 to 20% of all given data points from the period 2018-2019, the most recent datapoint being "2019-11-25".
- The second dataset of Github was created by the owner of the repository by web scraping the website “https://www.privacyaffairs.com/gdpr-fines/".
- The relevant information has been extracted in R from the source code of the website.
- The dataset we chose is a resulting comma-separated file (.csv), which is derived from the JSON structure of the raw data.

## Data preprocessing:

- To edit and cleanse the data, we imported both datasets into a Kaggle notebook and edited them using functions of the Python framework pandas. (Screenshots from Jupyter Notebook)
- The two .csv files of the datasets are read in as dataframes named *data1* and *data2* using the parser integrated in

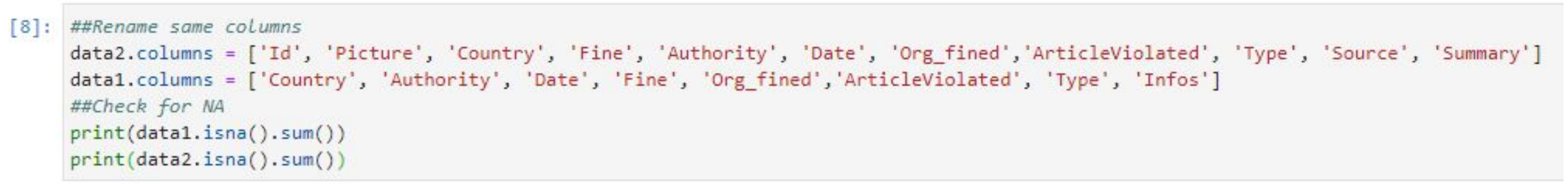


```
[2]: ##Input and libraries
     gdprkaggle="C:/Users/Alex/Desktop/py/input/gdpr_fines.csv"
     allfinesgit="C:/Users/Alex/Desktop/py/input/all_fines.csv"
     import numpy as np # linear algebra
     import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```



```
[3]: ##Read data from csv
     data1=pd.read_csv(gdprkaggle)
     data2=pd.read_csv(allfinesgit)
```

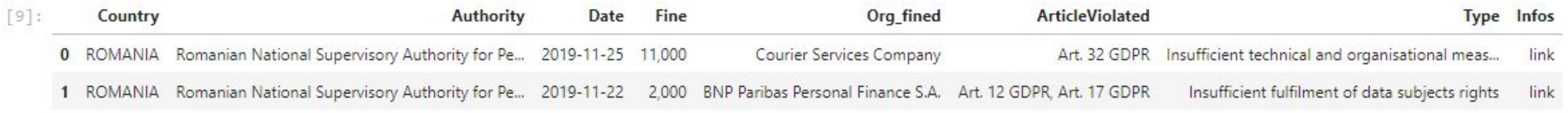
- The column names of *data2* with the same or similar content as in *data1* have been named the same in order to simplify later merging.



```
[8]: ##Rename same columns
     data2.columns = ['Id', 'Picture', 'Country', 'Fine', 'Authority', 'Date', 'Org_fined','ArticleViolated', 'Type', 'Source', 'Summary']
     data1.columns = ['Country', 'Authority', 'Date', 'Fine', 'Org_fined','ArticleViolated', 'Type', 'Infos']
     ##Check for NA
     print(data1.isna().sum())
     print(data2.isna().sum())
```

- Before further processing our data looked as following:

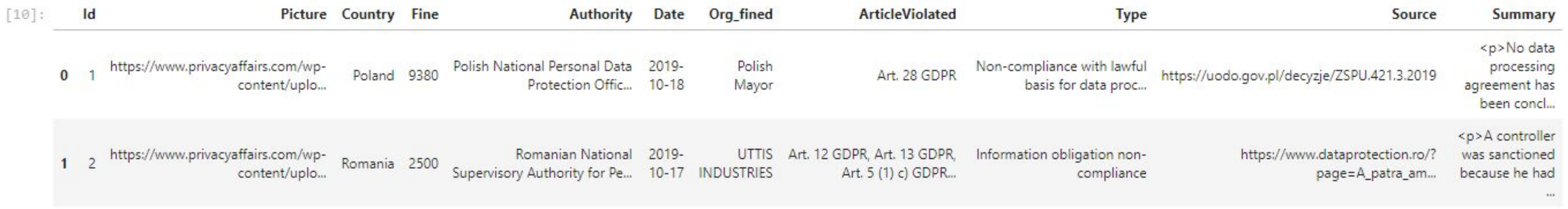
Kaggle dataset:



[9]:

| | Country | Authority | Date | Fine | Org_fined | ArticleViolated | Type | Infos |
|---|---|---|---|---|---|---|---|---|
| 0 | ROMANIA | Romanian National Supervisory Authority for Pe... | 2019-11-25 | 11,000 | Courier Services Company | Art. 32 GDPR | Insufficient technical and organisational meas... | link |
| 1 | ROMANIA | Romanian National Supervisory Authority for Pe... | 2019-11-22 | 2,000 | BNP Paribas Personal Finance S.A. | Art. 12 GDPR, Art. 17 GDPR | Insufficient fulfilment of data subjects rights | link |

Github dataset:



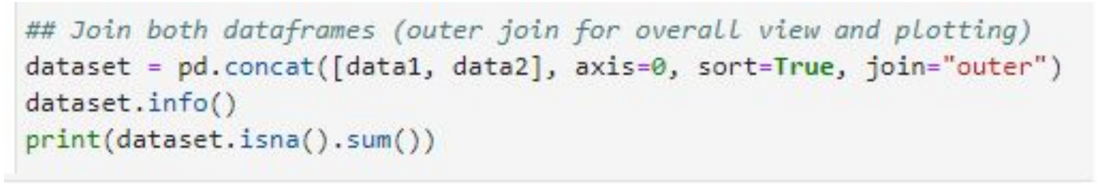
[10]:

| | Id | Picture | Country | Fine | Authority | Date | Org_fined | ArticleViolated | Type | Source | Summary |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | https://www.privacyaffairs.com/wp-content/uplo... | Poland | 9380 | Polish National Personal Data Protection Offic... | 2019-10-18 | Polish Mayor | Art. 28 GDPR | Non-compliance with lawful basis for data proc... | https://uodo.gov.pl/decyzje/ZSPU.421.3.2019 | <p>No data processing agreement has been concl... |
| 1 | 2 | https://www.privacyaffairs.com/wp-content/uplo... | Romania | 2500 | Romanian National Supervisory Authority for Pe... | 2019-10-17 | UTTIS INDUSTRIES | Art. 12 GDPR, Art. 13 GDPR, Art. 5 (1) c) GDPR... | Information obligation non-compliance | https://www.dataprotection.ro/?page=A_patra_am... | <p>A controller was sanctioned because he had ... |

- The first dataset consists of 120 rows and 8 columns: *Country, Authority, Date, Fine, Controller / Processor, Quoted Article, Type, Infos*.
- Consequently the second dataset consists of 250 rows and 11 columns: *Id, Picture, Country, Price, Authority, Date, Org_fined, Article Violated, Type, Source, Summary*,

## Combining/merging the 2 datasets

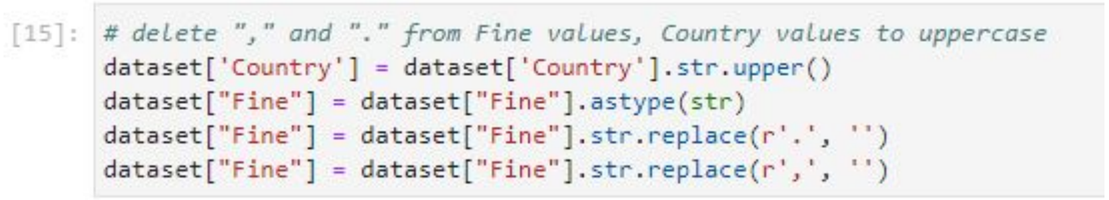
- To get not only the intersection of both datasets, but the total amount of both datasets, we executed the function **concat()** with the parameter *join="outer"* and created a dataset named *dataset*



```
## Join both dataframes (outer join for overall view and plotting)
dataset = pd.concat([data1, data2], axis=0, sort=True, join="outer")
dataset.info()
print(dataset.isna().sum())
```

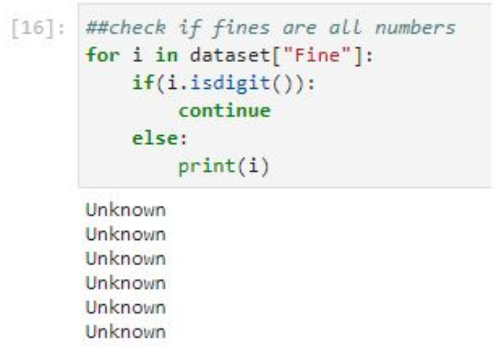
# Further data preprocessing:

- Standardize the country names with the function **upper()** capital letters in example “germany” => "GERMANY".
- To correct the problem of decimal points in currencies we first convert the *Fine* column to a string and then delete the "," and "." from the Fine column, because the dataset does not contain any decimal point values.



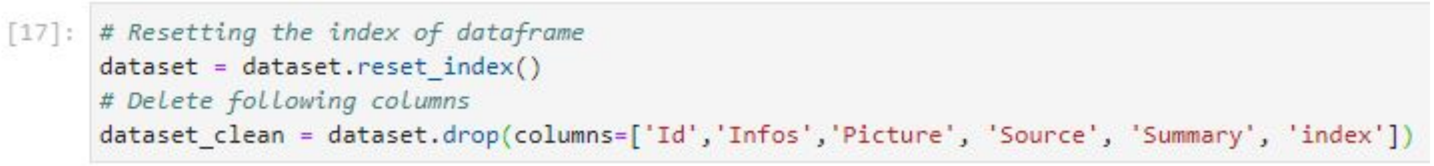
```
[15]: # delete "," and "." from Fine values, Country values to uppercase
      dataset['Country'] = dataset['Country'].str.upper()
      dataset["Fine"] = dataset["Fine"].astype(str)
      dataset["Fine"] = dataset["Fine"].str.replace(r'.', '')
      dataset["Fine"] = dataset["Fine"].str.replace(r',', '')
```

- We check if all entries in the Fine column are numbers and notice that we have 6 "Unknown" entries.



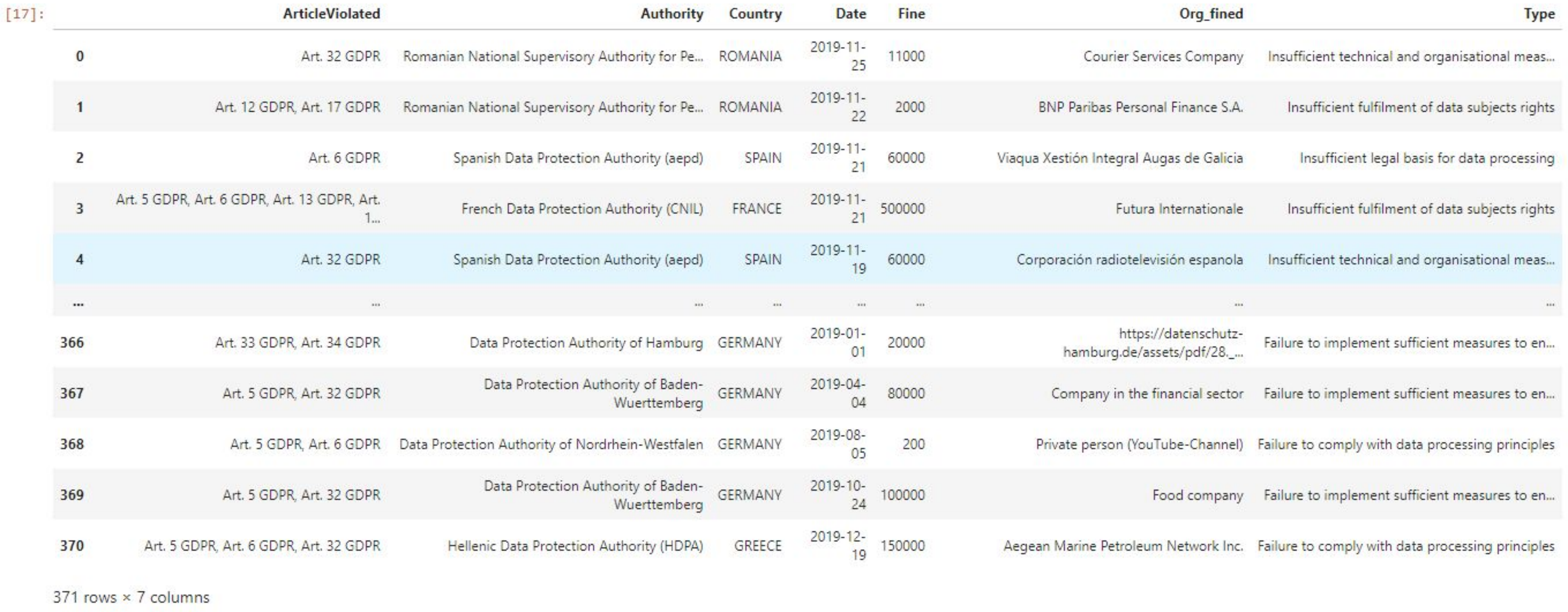
```
[16]: ##check if fines are all numbers
      for i in dataset["Fine"]:
          if(i.isdigit()):
              continue
          else:
              print(i)

      Unknown
      Unknown
      Unknown
      Unknown
      Unknown
      Unknown
```

- Further, because of inconsistency we reset the Index column and drop the following columns from the data frame: *Id, Infos, Picture, Source, Summary, index* while saving it in *dataset_clean*.



```
[17]: # Resetting the index of dataframe
      dataset = dataset.reset_index()
      # Delete following columns
      dataset_clean = dataset.drop(columns=['Id','Infos','Picture', 'Source', 'Summary', 'index'])
```

- For the moment our *dataset_clean* dataframe looks as follows:



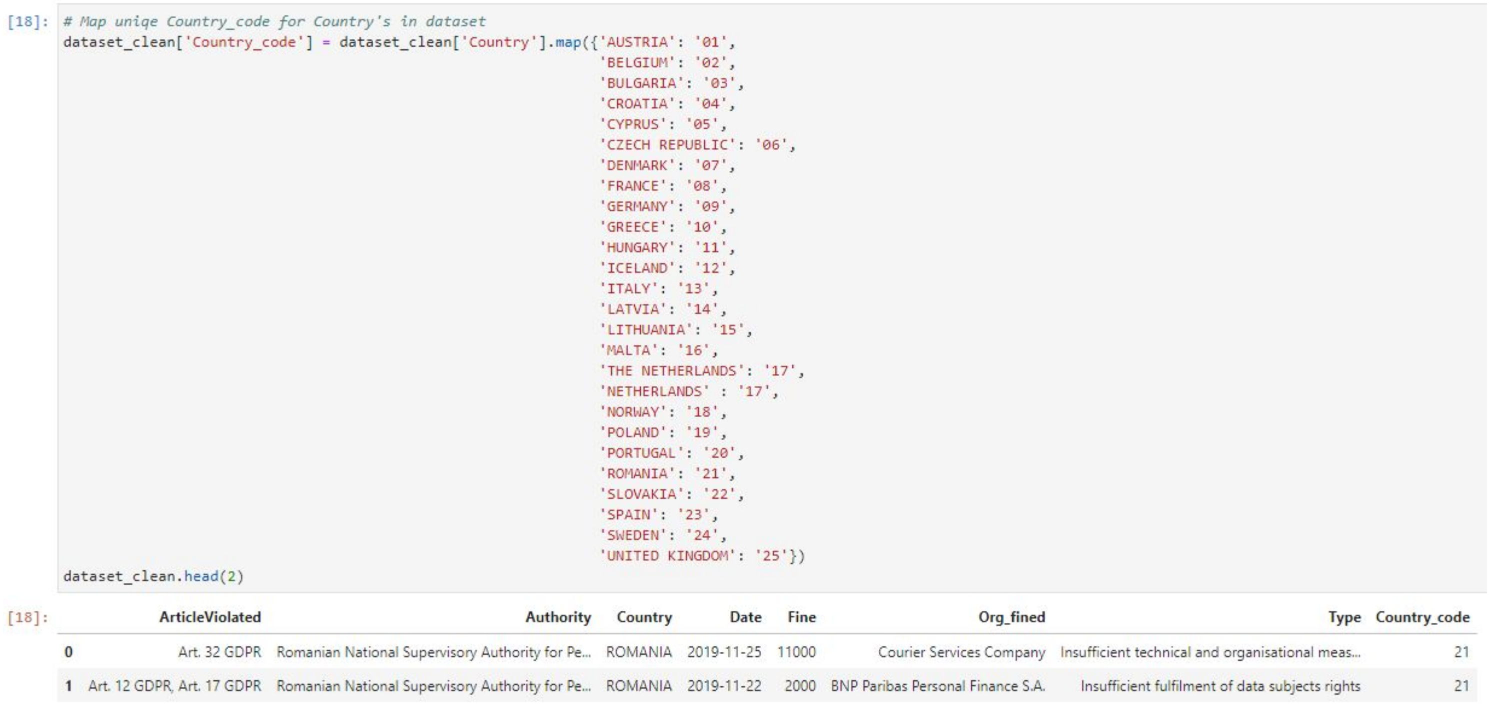
[17]:

| | ArticleViolated | Authority | Country | Date | Fine | Org_fined | Type |
|---|---|---|---|---|---|---|---|
| **0** | Art. 32 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-25 | 11000 | Courier Services Company | Insufficient technical and organisational meas... |
| **1** | Art. 12 GDPR, Art. 17 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-22 | 2000 | BNP Paribas Personal Finance S.A. | Insufficient fulfilment of data subjects rights |
| **2** | Art. 6 GDPR | Spanish Data Protection Authority (aepd) | SPAIN | 2019-11-21 | 60000 | Viaqua Xestión Integral Augas de Galicia | Insufficient legal basis for data processing |
| **3** | Art. 5 GDPR, Art. 6 GDPR, Art. 13 GDPR, Art. 1... | French Data Protection Authority (CNIL) | FRANCE | 2019-11-21 | 500000 | Futura Internationale | Insufficient fulfilment of data subjects rights |
| **4** | Art. 32 GDPR | Spanish Data Protection Authority (aepd) | SPAIN | 2019-11-19 | 60000 | Corporación radiotelevisión espanola | Insufficient technical and organisational meas... |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **366** | Art. 33 GDPR, Art. 34 GDPR | Data Protection Authority of Hamburg | GERMANY | 2019-01-01 | 20000 | https://datenschutz-hamburg.de/assets/pdf/28._... | Failure to implement sufficient measures to en... |
| **367** | Art. 5 GDPR, Art. 32 GDPR | Data Protection Authority of Baden-Wuerttemberg | GERMANY | 2019-04-04 | 80000 | Company in the financial sector | Failure to implement sufficient measures to en... |
| **368** | Art. 5 GDPR, Art. 6 GDPR | Data Protection Authority of Nordrhein-Westfalen | GERMANY | 2019-08-05 | 200 | Private person (YouTube-Channel) | Failure to comply with data processing principles |
| **369** | Art. 5 GDPR, Art. 32 GDPR | Data Protection Authority of Baden-Wuerttemberg | GERMANY | 2019-10-24 | 100000 | Food company | Failure to implement sufficient measures to en... |
| **370** | Art. 5 GDPR, Art. 6 GDPR, Art. 32 GDPR | Hellenic Data Protection Authority (HDPA) | GREECE | 2019-12-19 | 150000 | Aegean Marine Petroleum Network Inc. | Failure to comply with data processing principles |

371 rows × 7 columns

- The dataframe consists of 370 entires(fines) and 7 columns.

- Using a dictionary, we generate a *Country_code* column with the corresponding country code manually. Tricky was the country “Netherlands” which appears in both original datasets differently, we found four time “The Netherlands” instead and had to treat it separately.

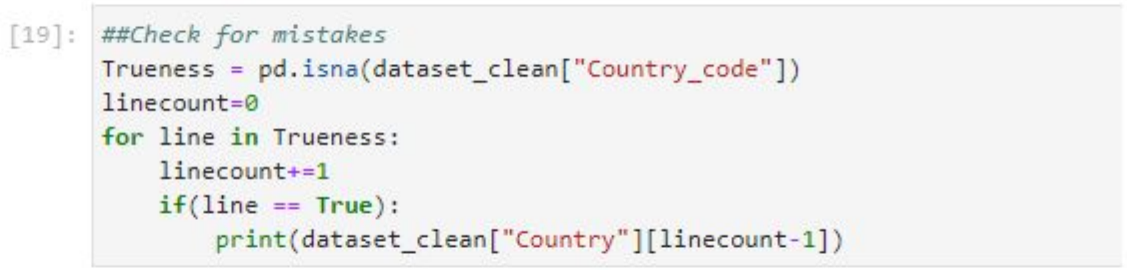


```
[18]: # Map uniqe Country_code for Country's in dataset
      dataset_clean['Country_code'] = dataset_clean['Country'].map({'AUSTRIA': '01',
                                                                'BELGIUM': '02',
                                                                'BULGARIA': '03',
                                                                'CROATIA': '04',
                                                                'CYPRUS': '05',
                                                                'CZECH REPUBLIC': '06',
                                                                'DENMARK': '07',
                                                                'FRANCE': '08',
                                                                'GERMANY': '09',
                                                                'GREECE': '10',
                                                                'HUNGARY': '11',
                                                                'ICELAND': '12',
                                                                'ITALY': '13',
                                                                'LATVIA': '14',
                                                                'LITHUANIA': '15',
                                                                'MALTA': '16',
                                                                'THE NETHERLANDS': '17',
                                                                'NETHERLANDS' : '17',
                                                                'NORWAY': '18',
                                                                'POLAND': '19',
                                                                'PORTUGAL': '20',
                                                                'ROMANIA': '21',
                                                                'SLOVAKIA': '22',
                                                                'SPAIN': '23',
                                                                'SWEDEN': '24',
                                                                'UNITED KINGDOM': '25'})
      dataset_clean.head(2)
```

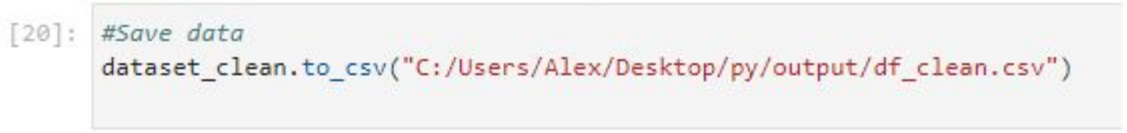
[18]:

| | ArticleViolated | Authority | Country | Date | Fine | Org_fined | Type | Country_code |
|---|---|---|---|---|---|---|---|---|
| 0 | Art. 32 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-25 | 11000 | Courier Services Company | Insufficient technical and organisational meas... | 21 |
| 1 | Art. 12 GDPR, Art. 17 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-22 | 2000 | BNP Paribas Personal Finance S.A. | Insufficient fulfilment of data subjects rights | 21 |

- We check for mistakes in the new *Country_code* column and export the *dataset_clean* to a new CSV named *df_clean.csv* since we continue to clean the dataset in OpenRefine.



```
[19]: ##Check for mistakes
      Trueness = pd.isna(dataset_clean["Country_code"])
      linecount=0
      for line in Trueness:
          linecount+=1
          if(line == True):
              print(dataset_clean["Country"][linecount-1])
```

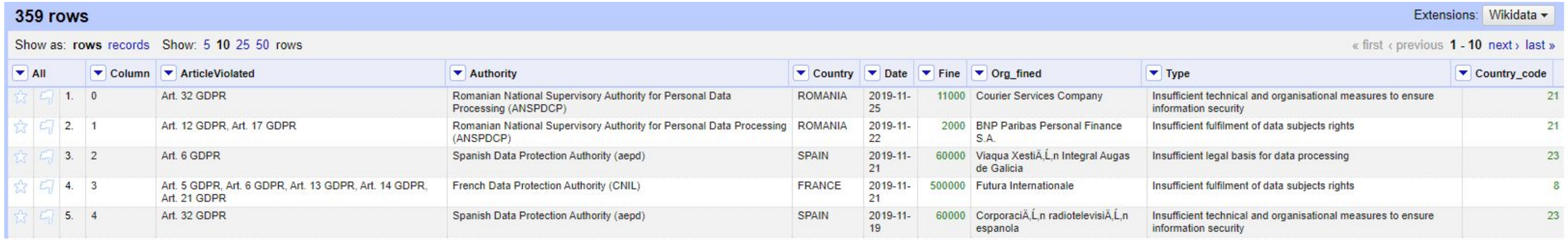


```
[20]: #Save data
      dataset_clean.to_csv("C:/Users/Alex/Desktop/py/output/df_clean.csv")
```

## Cleaning in OpenRefine

- We have edited the following in OpenRefine:
  - Deleted 6 rows that had Fine and Date as "Unknown" values
  - Deleted 6 rows that had Fine value 0 and Date 1970-01-01
  - 4 lines had 2.018 as year specification, so we rewrote them to 2018-01-01
  - 8 lines had 2.019 as year specification, so we rewrote them to 2019-01-01
  - 9 lines had "Unknown" as year, so we changed it to 2021-01-01
  - 9 lines had the year 1970-01-01, so we rewrote them to 2021-01-01
  - Furthermore we have converted the *Fine* and *Country_code* columns to numbers with the "Common Transform" function

- After further processing of the data in OpenRefine, our dataset looks like this:



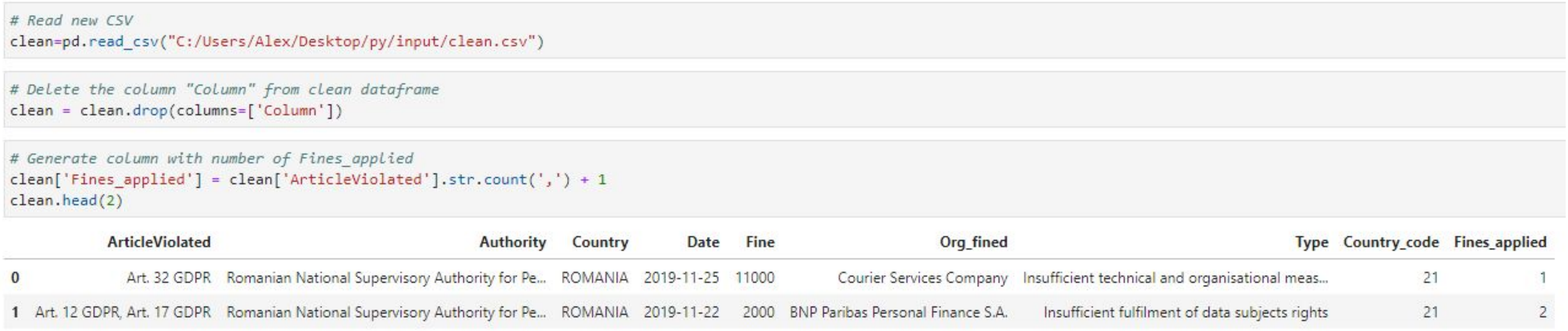
359 rows

Extensions: Wikidata

Show as: rows records Show: 5 10 25 50 rows « first ‹ previous 1 - 10 next › last »

| All | | | Column | ArticleViolated | Authority | Country | Date | Fine | Org_fined | Type | Country_code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ☆ | | 1. | 0 | Art. 32 GDPR | Romanian National Supervisory Authority for Personal Data Processing (ANSPDCP) | ROMANIA | 2019-11-25 | 11000 | Courier Services Company | Insufficient technical and organisational measures to ensure information security | 21 |
| ☆ | | 2. | 1 | Art. 12 GDPR, Art. 17 GDPR | Romanian National Supervisory Authority for Personal Data Processing (ANSPDCP) | ROMANIA | 2019-11-22 | 2000 | BNP Paribas Personal Finance S.A. | Insufficient fulfilment of data subjects rights | 21 |
| ☆ | | 3. | 2 | Art. 6 GDPR | Spanish Data Protection Authority (aepd) | SPAIN | 2019-11-21 | 60000 | Viaqua XestiĂ‚Ĺ‚n Integral Augas de Galicia | Insufficient legal basis for data processing | 23 |
| ☆ | | 4. | 3 | Art. 5 GDPR, Art. 6 GDPR, Art. 13 GDPR, Art. 14 GDPR, Art. 21 GDPR | French Data Protection Authority (CNIL) | FRANCE | 2019-11-21 | 500000 | Futura Internationale | Insufficient fulfilment of data subjects rights | 8 |
| ☆ | | 5. | 4 | Art. 32 GDPR | Spanish Data Protection Authority (aepd) | SPAIN | 2019-11-19 | 60000 | CorporaciĂ‚Ĺ‚n radiotelevisiĂ‚Ĺ‚n espanola | Insufficient technical and organisational measures to ensure information security | 23 |

- We are left with 358 entries and import a new CSV in python to process our data further
- The new CSV is read in as a new dataframe in *clean* variable



```
# Read new CSV
clean=pd.read_csv("C:/Users/Alex/Desktop/py/input/clean.csv")
```

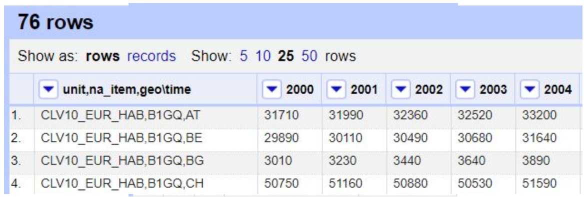
```
# Delete the column "Column" from clean dataframe
clean = clean.drop(columns=['Column'])
```

```
# Generate column with number of Fines_applied
clean['Fines_applied'] = clean['ArticleViolated'].str.count(',') + 1
clean.head(2)
```

| | ArticleViolated | Authority | Country | Date | Fine | Org_fined | Type | Country_code | Fines_applied |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Art. 32 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-25 | 11000 | Courier Services Company | Insufficient technical and organisational meas... | 21 | 1 |
| 1 | Art. 12 GDPR, Art. 17 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-22 | 2000 | BNP Paribas Personal Finance S.A. | Insufficient fulfilment of data subjects rights | 21 | 2 |

- The column “Column” is deleted (added while working in OpenRefine) and a new column *Fines_applied* is generated in which the number of articles applied to a fine is retained.

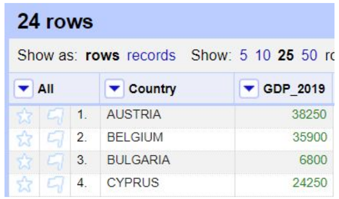
- We added to our existing data the GDP per citizen and GDP per country after processing it in OpenRefine, together with some string manipulation in Python.
- Before we processed the GDP datasets in OpenRefine they looked like this:



76 rows

Show as: **rows** records Show: 5 10 **25** 50 rows

| | unit,na_item,geo\time | 2000 | 2001 | 2002 | 2003 | 2004 |
|---|---|---|---|---|---|---|
| 1. | CLV10_EUR_HAB,B1GQ,AT | 31710 | 31990 | 32360 | 32520 | 33200 |
| 2. | CLV10_EUR_HAB,B1GQ,BE | 29890 | 30110 | 30490 | 30680 | 31640 |
| 3. | CLV10_EUR_HAB,B1GQ,BG | 3010 | 3230 | 3440 | 3640 | 3890 |
| 4. | CLV10_EUR_HAB,B1GQ,CH | 50750 | 51160 | 50880 | 50530 | 51590 |

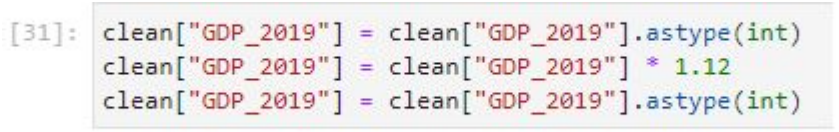


24 rows

Show as: **rows** records Show: 5 10 **25** 50 rc

| All | | Country | GDP_2019 |
|---|---|---|---|
| | 1. | AUSTRIA | 38250 |
| | 2. | BELGIUM | 35900 |
| | 3. | BULGARIA | 6800 |
| | 4. | CYPRUS | 24250 |

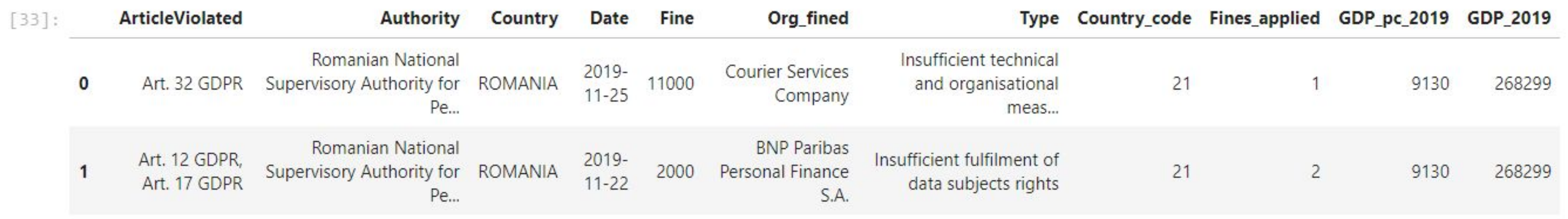
- One dataset contains all GDPs per citizen from the year 2000 to 2019 while the other one contains the GDP of each country for the year of 2019. The GDPs for each country are given in $, which we convert in € by by multiplying the column by 1.12.



```
[31]: clean["GDP_2019"] = clean["GDP_2019"].astype(int)
      clean["GDP_2019"] = clean["GDP_2019"] * 1.12
      clean["GDP_2019"] = clean["GDP_2019"].astype(int)
```

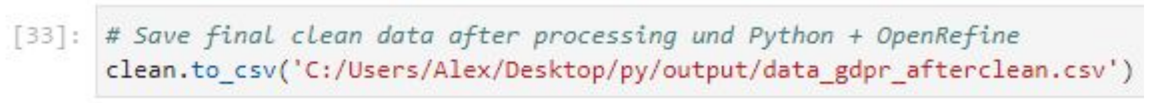
- GDP per capital dataset: : https://ec.europa.eu/eurostat/web/products-datasets/-/sdg_08_10
- GDP per country dataset: https://datacatalog.worldbank.org/dataset/gdp-ranking

- These were added by mapping the values of the GDP per capital and GDP per country to the each country. (same as Country_code)
- All values which should be numbers, are converted to a numerical value in OpenRefine. That includes the columns: “Fine”, “Country_code”, “GDP_pc_2019”, “GDP_2019”,  “ Fines_applied” .
- The final cleaned data is exported to data_gdpr_afterclean.csv for further application of the 2 algorithms of the same type



[33]:

| | ArticleViolated | Authority | Country | Date | Fine | Org_fined | Type | Country_code | Fines_applied | GDP_pc_2019 | GDP_2019 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Art. 32 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-25 | 11000 | Courier Services Company | Insufficient technical and organisational meas... | 21 | 1 | 9130 | 268299 |
| **1** | Art. 12 GDPR, Art. 17 GDPR | Romanian National Supervisory Authority for Pe... | ROMANIA | 2019-11-22 | 2000 | BNP Paribas Personal Finance S.A. | Insufficient fulfilment of data subjects rights | 21 | 2 | 9130 | 268299 |

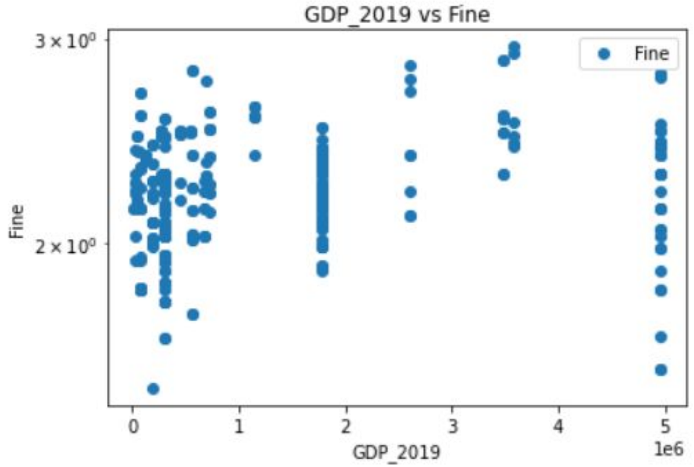


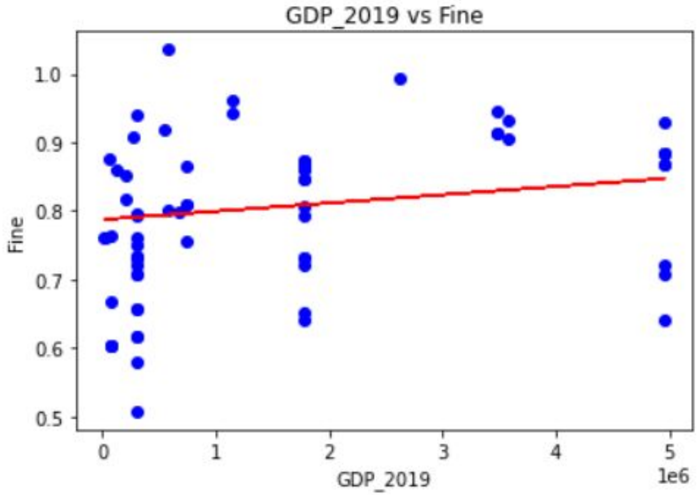
```
[33]: # Save final clean data after processing und Python + OpenRefine
      clean.to_csv('C:/Users/Alex/Desktop/py/output/data_gdpr_afterclean.csv')
```

# Linear Regression

- With linear regression we want to model the relationship between dependent variables and one or more independent variables. The case of one explanatory variable is called simple linear regression.

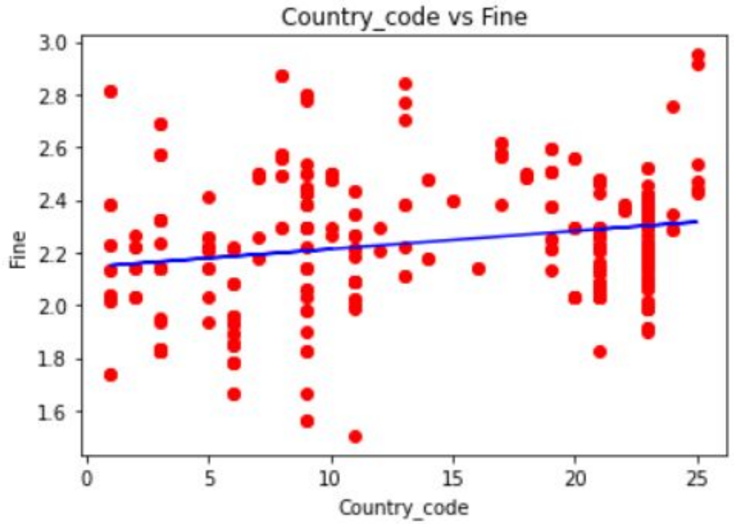
- We saw a relation between GDP and Fines in a scatterplot and try to fit a regression function :

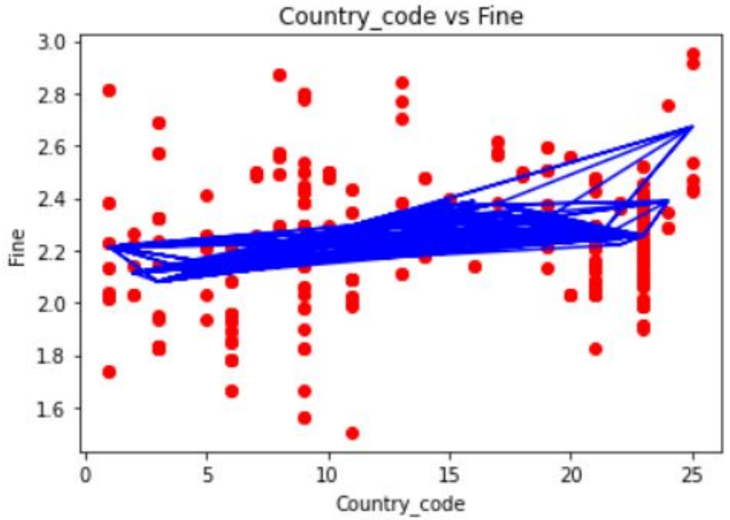




# Polynomial Regression

- We thought we could try to fit a polynom on the data.
    - `from sklearn.preprocessing import PolynomialFeatures`
    - `poly_reg = PolynomialFeatures(degree=6)`





## End

Thank you for listening!

Question?