AMF’s Sanction Commission

- Context
  - Origine of AMF’s Sanction Commission
  - Role and sanctions imposed
- Problematic
  - *Is it possible to estimate the differences between fines and/or sanctions inflicted by the AMF's sanction commission depending on the facts of the case and their date ?*
  - *Is it possible to establish a pattern from the AMF’s sanction commission ?*
- Code
  - Explanation of the code
  - Difficulties encountered
- Graphs

T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy

Context

- ***What is the AMF : Autorité des Marchés Financiers ?***

  - An independent public authority responsible for protecting **savings** invested in **financial products**, providing **investors** with **information** and **ensuring** that markets are **operational**.

  - The AMF monitors the markets and has investigative and supervisory powers that can lead to sanction proceedings **in the event of non-compliance** → AMF's sanction commission

Context

- ***What is the AMF's sanction commission*** :
  - Made up of **judges and professionnals**
  - Has *complete* **decision-making autonomy**
  - Can sanction **any person** or **company** whose practices violate the laws and regulations that **fall withing** the AMF's jurisdiction
  - Its sanctions are used as **deterrent** and **precedent** for companies

- 2 **types** of sanctions can be inflicted :
  - **Monetary fines**
  - **Disciplinary measures** : warning, reprimand, or temporary or permanent ban on the exercise of all or part of the services provided

T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy

Interest of the analysis

- Determining which practice is most severely sanctioned can be useful for:

Companies

- Possibility of carriyng out a risk map
- Possibility of drawing the attention of operational staff and training them on particularly risky practices
- Ability to assess risks, particularly during the due diligence

AMF

- Allows tracking of sanction decisions
- Allows the consistency of these decisions to be assessed on a regular basis
- Allows for possible adjustments of future sanctions in case of inconsistency

Explanation of the code

- 1st step: to go on AMF website to parse sanction decisions

```
driver = webdriver.Chrome()

# Navigate to the web page
url = 'https://www.amf-france.org/fr/sanction-transaction/Decisions-de-la-commission-des-sanctions'
driver.get(url)
```

Difficulties encountered:

1) The content did not have time to load

2) Impossible to move forward because of the cookie authorization button

Explanation of the code

- 1st step: to go on AMF website to parse sanction decisions

Solutions

1) To wait until the downloading of the content:

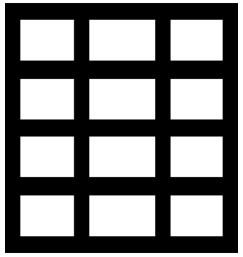
```
# Wait for the dynamic content to load
time.sleep(10)
```

2) To disable cookie consent:

```
# Click the "refuser tout" button to disable cookie consent
button = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.ID, "tarteaucitronAllDenied2")))
button.click()
```

Explanation of the code

- 2nd step: to transcribe the data in a csv file



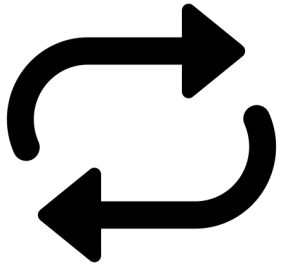
1) Creation of an output file with 3 columns

```
with open('output1.csv', mode='w') as output_file:
    output_writer = csv.writer(output_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
    output_writer.writerow(['date', 'theme', 'montant'])
```

Manquement d'initié	700000	2019
Manquement d'initié	830000	2019
Manquement d'initié	530000	2019
Manquement d'initié	350000	2019
CIF, CIP ou autres prestataires	50000	2019

Explanation of the code

- 2nd step: to transcribe the data in a csv file



2) Creation of a loop to transcribe the wished information

```
# Keep scraping the page until the "next page" button is disabled
count = 0
while count < 41:
    # Get all <p> tags in the page that contain the Euro symbol
    p_tags = [p for p in soup.find_all('p') if '€' in p.get_text() or 'euros' in p.get_text()]

    # Get all <span> tags with class="tag"
    s_tags = [s for s in soup.find_all('span', {'class': 'tag'})]

    # Get all <span> tags with class="date"
    d_tags = [d for d in soup.find_all('span', {'class': 'date'})]

    # Write the data to the CSV file
    for p, s, d in zip(p_tags, s_tags, d_tags):
        year = d.get_text().split()[-1]  # Extract the last word of the date string
        output_writer.writerow([year, s.get_text(), p.get_text()])
    # Click the "next page" button if it's clickable, otherwise exit the loop
    try:
        button = WebDriverWait(driver, 2).until(EC.element_to_be_clickable((By.ID, "DataTables_Table_0_next")), message="Button
        button.click()
        time.sleep(2)
        # Get the updated HTML content and parse it using BeautifulSoup
        html = driver.page_source
        soup = BeautifulSoup(html, 'html.parser')
    except TimeoutException:
        print("Button not found or not clickable within 10 seconds.")
        break
    count += 1
```

Explanation of the code

- 2nd step: to transcribe the data in a csv file

FOCUS ON SOME DIFFICULTIES:

➢ **Non-uniformity of the the amount of the sanction on the AMF website**

17 juillet 2014	MANQUEMENT D'INITIÉ	**SAN-2014-14** MM. C et D	62 000 euros
30 juin 2014	OBLIGATIONS PROFESSIONNELLES	**SAN-2014-13** Société CLARESCO GESTION et M. A	90 000 €

Consequence: sanction of June 30, 2014 not taken into account in this code

```
while count < 41:
    # Get all <p> tags in the page that contain the Euro symbol
    p_tags = [p for p in soup.find_all('p') if '€' in p.get_text()
```

Explanation of the code

- 2nd step: to transcribe the data in a csv file

SOLUTION:

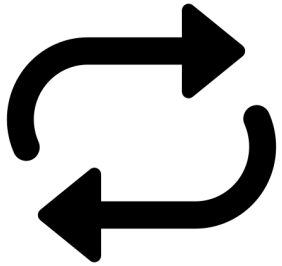
➢ **To add an alternative condition that includes the term euros written in letters**

```
while count < 41:
    # Get all <p> tags in the page that contain the Euro symbol
    p_tags = [p for p in soup.find_all('p') if '€' in p.get_text() or 'euros' in p.get_text()]
```

Consequence: all sanctions having "€" or "euros" in their text are taken into account

Explication of the code

- 2nd step: to transcribe the data in a csv file

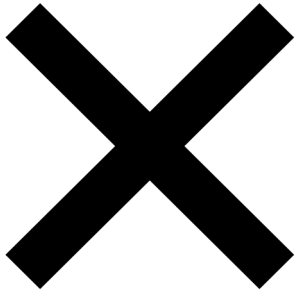


2) Creation of a loop to transcribe the wished information

```
# Keep scraping the page until the "next page" button is disabled
count = 0
while count < 41:
    # Get all <p> tags in the page that contain the Euro symbol
    p_tags = [p for p in soup.find_all('p') if '€' in p.get_text() or 'euros' in p.get_text()]

    # Get all <span> tags with class="tag"
    s_tags = [s for s in soup.find_all('span', {'class': 'tag'})]

    # Get all <span> tags with class="date"
    d_tags = [d for d in soup.find_all('span', {'class': 'date'})]

    # Write the data to the CSV file
    for p, s, d in zip(p_tags, s_tags, d_tags):
        year = d.get_text().split()[-1]  # Extract the last word of the date string
        output_writer.writerow([year, s.get_text(), p.get_text()])
    # Click the "next page" button if it's clickable, otherwise exit the loop
    try:
        button = WebDriverWait(driver, 2).until(EC.element_to_be_clickable((By.ID, "DataTables_Table_0_next")), message="Button
        button.click()
        time.sleep(2)
        # Get the updated HTML content and parse it using BeautifulSoup
        html = driver.page_source
        soup = BeautifulSoup(html, 'html.parser')
    except TimeoutException:
        print("Button not found or not clickable within 10 seconds.")
        break
    count += 1
```

Explanation of the code

- 3rd step: Clean the data - *Split columns*



- Difficulty : in the 2nd step (slide 7), the Excel file created has only 3 columns : *date*, *theme* and *montant*
  - The "montant" column does not differenciate between the monetary fines and the disciplinary sanctions
  - Impossibility to determine pattern from the commission

Obligations professionnelles	50 000 €␣2 blâmes␣1 interdiction d'exercer ␣l'activité de gestionnaire d'actifs pour le compte de tiers et de ges
Manipulation de marché	5 000 000 €
Manipulation de marché	20 000 000 €
Obligations professionnelles	680 000 €␣1 avertissement
CIF, CIP ou autres prestataires	100 000 €
Obligation d'information	120 000 €
PSI	3 000 000 €␣1 mise hors de cause
Obligations professionnelles	200 000 €␣2 avertissements
PSI	220 000 €␣1 blâme
CIF, CIP ou autres prestataires	150 000 €␣1 blâme
Obligations professionnelles	70 000 €
CIF, CIP ou autres prestataires	20 000 €
CIF, CIP ou autres prestataires	50 000 €␣1 avertissement
Manquement d'initié	700 000 €␣4 mises hors de cause
Manquement d'initié	830 000 €
Manquement d'initié	530 000 €
Manquement d'initié	350 000 €␣1 avertissement
CIF, CIP ou autres prestataires	50 000 €
Obligations professionnelles	170 000 €␣2 interdictions d'exercer l'activité de CIF (10 ans)
CIF, CIP ou autres prestataires	1 666 000 €
CIF, CIP ou autres prestataires	450 000 €
Obligation d'information	50 000 €␣2 interdictions d'exercer l'activité d'intermédiaires en biens divers (10 ans)
Manipulation de marché	920 000 €
Manquement d'initié	345 000 €

Explanation of the code

- 3rd step: Clean the data - *Split columns*

SOLUTION:

- **To add an additionnal column every time a "comma" is present in the "montant" column**

```
for row in reader:
    # Split the "montant" column by comma and write each element as a separate column
    montant = re.split('\n(?=\d)', row[2])
    writer.writerow([row[0], row[1]] + montant)
```

Consequence: the different sanctions, and thus the favor of the AMF for disciplinary sanctions, monetary fines or both can be analysed. Decisions without any monetary fine are taken into account

Explanation of the code

- 3rd step: Clean the data – *Change monetary sanction in numbers*
  - Difficulty : to create graphs with numbers, the Excel file must only contain numeric symbol
    - €, spaces or "*euros*" must be removed from the column "amount of the sanction"

SOLUTION:

➢ **New lines in the code :**

- **For** : to iterate on each line of on the Csv file
- Access to the 3rd column thanks to `row[2]`
- Substitution : elimination of all *non numerical elements* in this column with : `re.sub(r'\D', '', row[2]`

```
for row in reader:
    row[2] = re.sub(r'\D', '', row[2])
    writer.writerow(row)
```

Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

→ Pandas module

```
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('ggplot')
plt.rcParams["figure.figsize"] = (15,8)
df = pd.read_excel("/content/sanctions_amf[71] (1).xlsx", header = 1)
df = df.drop(columns=['Unnamed: 3', 'Unnamed: 4', 'Unnamed: 5', 'Unnamed: 6'])
df = df.rename(columns={"Unnamed: 2": "montant"})
```

```
df.head()
```

	date	theme	montant
0	2023	Manquement d'initié	700000
1	2023	CIF, CIP ou autres prestataires	120000
2	2023	Manquement d'initié	3160000
3	2023	Obligations professionnelles	93000000
4	2022	Obligations professionnelles	150000

1) We used **pandas module** *read_excel()* function to read the excel sheet generated from the previous scrapper code into a Dataframe object.

➢ A dataframe is an object that can be manipulated to make calculations on datatable.
- For instance, the head function helpeds us get the different rows from excel
- **Therefore the datafame looks like the tabular view of the excel sheet.**

Difficulties encountered:

- **Excel** generated by Thomas and Tali was not easily manipulated : delete first line of the excel, left the header
- Text processing necessary to reform the dataframe

Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

```
df.describe()
```

	date	montant
count	353.000000	3.530000e+02
mean	2012.504249	1.321590e+06
std	5.208779	6.053432e+06
min	2004.000000	1.000000e+00
25%	2008.000000	5.000000e+04
50%	2012.000000	2.000000e+05
75%	2017.000000	6.250000e+05
max	2023.000000	9.300000e+07

2) **Vizualisation** step thanks to pandas

By using dataframe. describe () function with pandas we also generated some **descriptive statistics** all dates combined**:** the average amounts, the minimum and maximum amounts of the fines.

- The average amount of fine imposed between 2004 and 2023 is 1.321590 power 6.

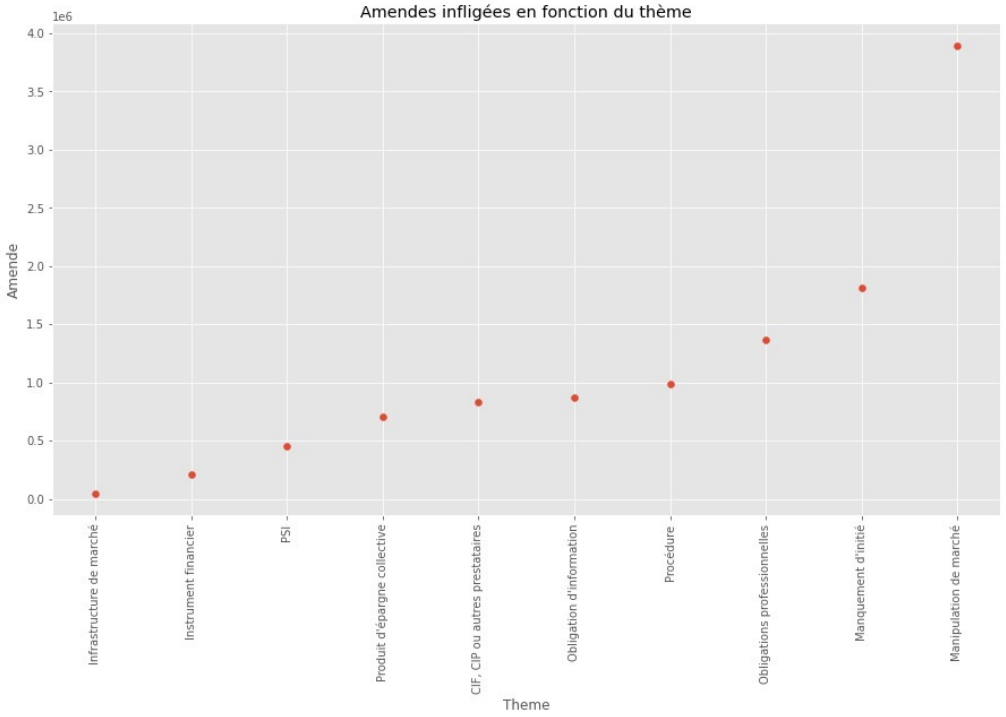
Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

→ Matplotlib module

1st graph:

- **Visualization of the amounts of fines by theme:** for each theme, we have the averages of the fines.
  - E.g.: all dates combined, the largest fines inflicted are for **market manipulation** (average fine of 3.9M€) and **insider trading** (average fine of 1.8M€) as opposed to market infrastructure, for which the amount of the fine imposed is very low (close to 0.1).
- Depending on the theme, the average amount of fine is different : large differences between amounts imposed

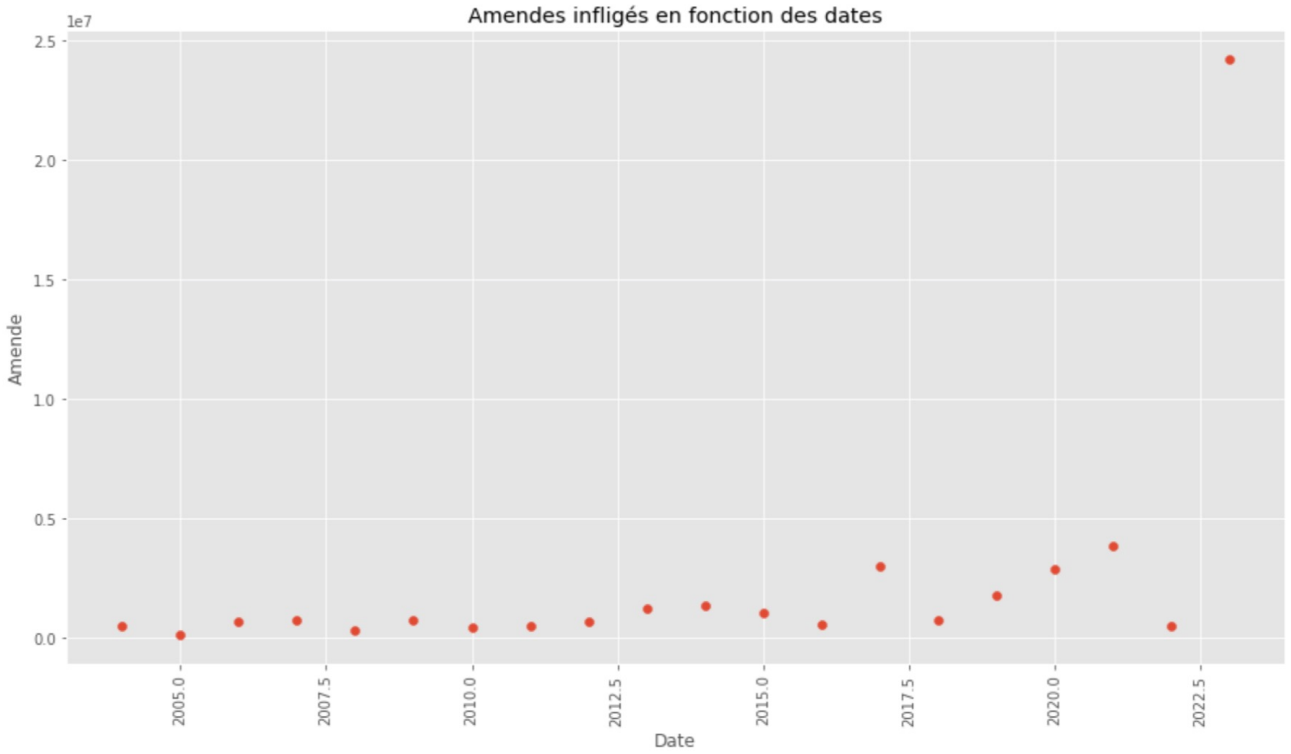


Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

2nd graph:

- **Visualization of the amount of fine per year**: so I filtered on the years and calculated the average of the amounts



Difficulties encountered:

- Because of year 2023 : the amounts of the fines are very big compared to before => the display does not allow to compare the years before 2023

- Not enough data to establish comparisons since the year 2023 is not finished

T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy

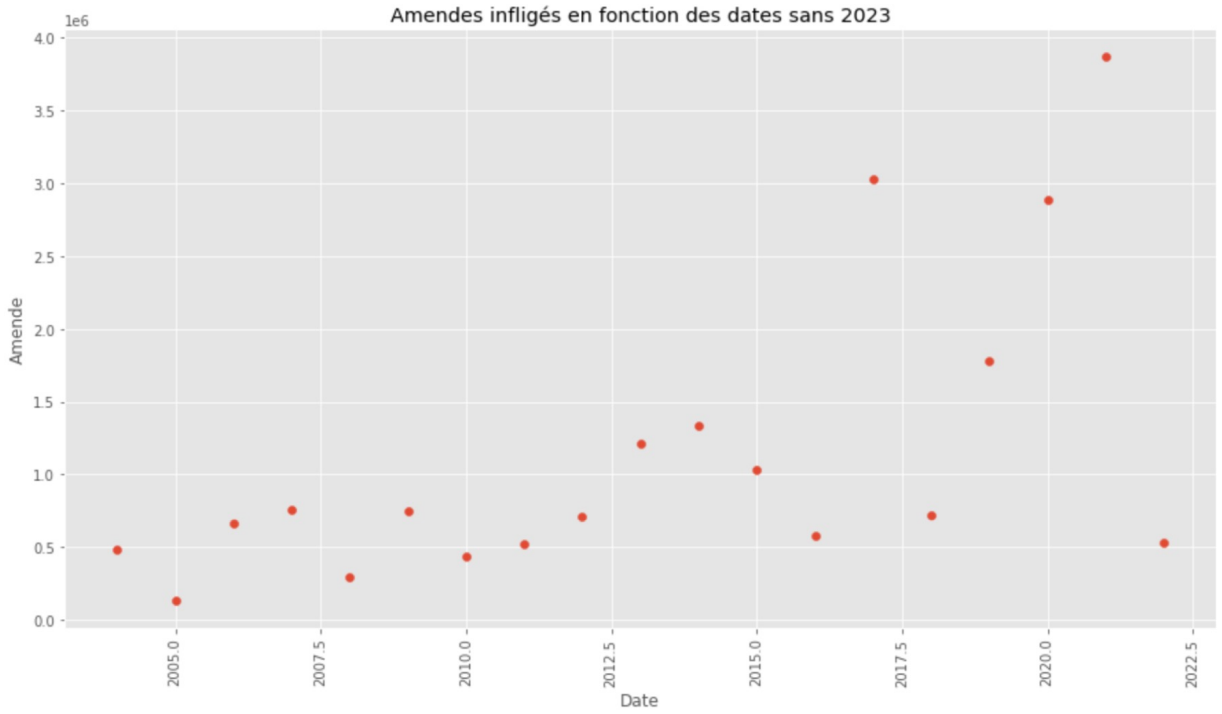
Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

2nd graph:

- So we made the same graph but **without year 2023** which can be seen on graph number 3

- The average amount of fines imposed is relatively homogeneous over the period 2004-2012 but from 2013 onwards, the AMF starts to impose more severe penalties (average fine is 1,7 million euros).



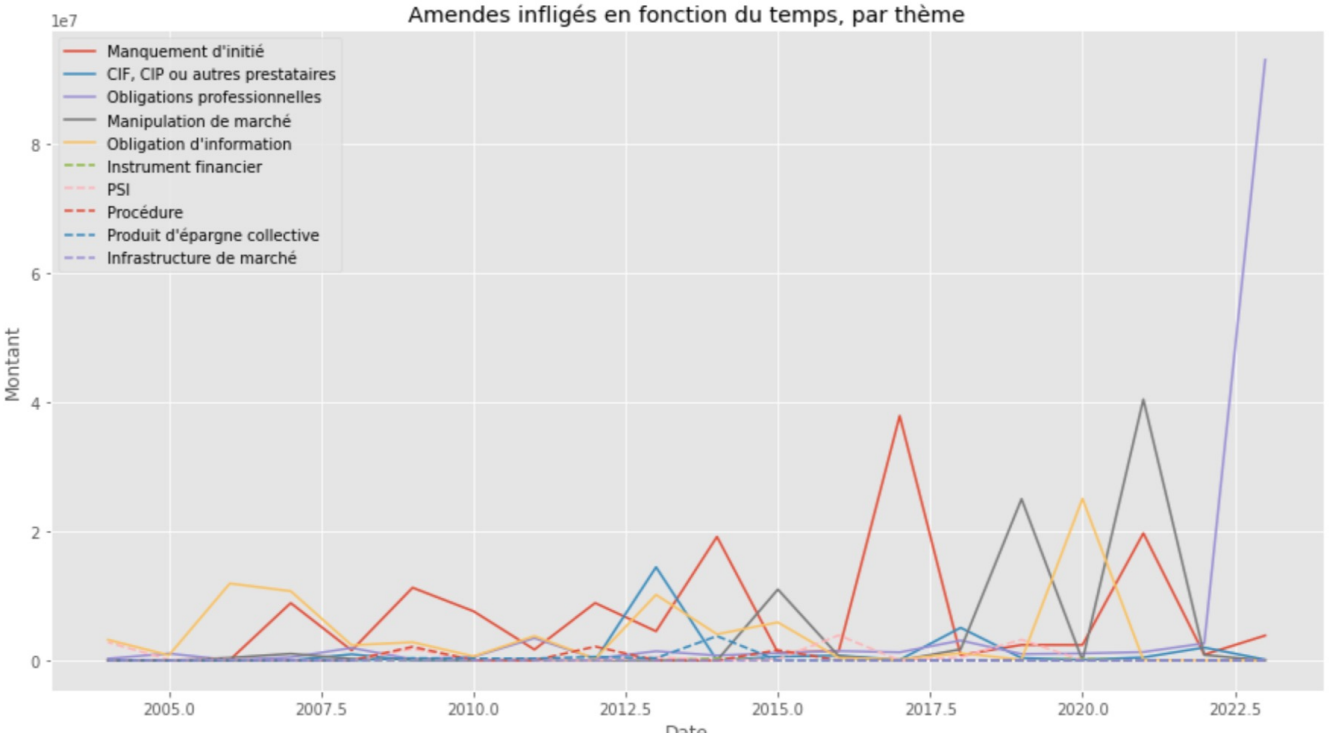
T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy

Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

3rd graph:

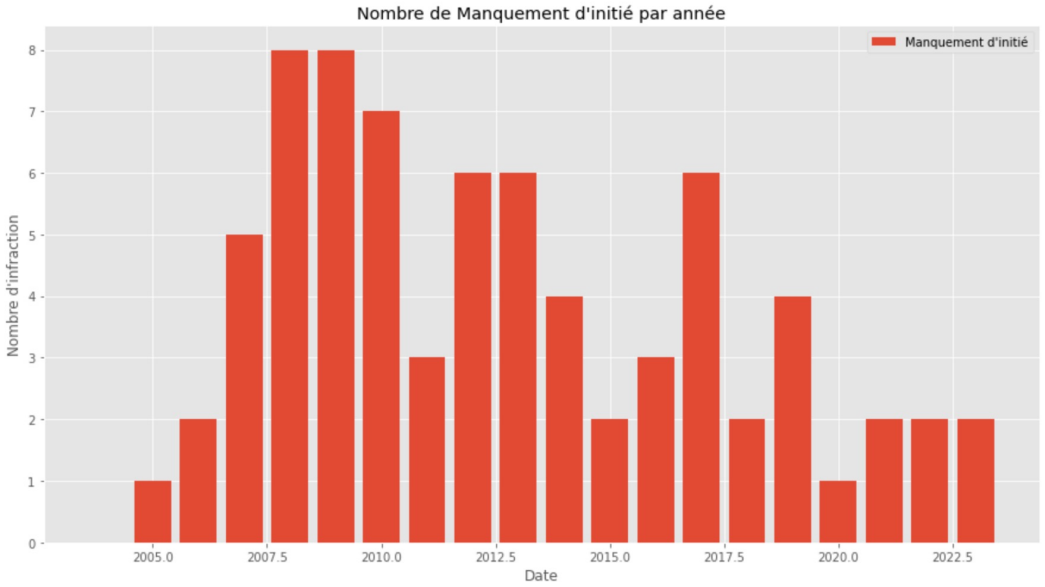
- **Visualization to compare each theme by year by doing a double filter**
  - E.g.: if we focus at themes insider trading (orange curve) and market manipulations (grey curve), the average amount of of the fines for insider trading is more important in 2017 (with an average fine two times higher than in 2021).
  - Unlike the theme market manipulation which has a bigger average amount of fines in 2021

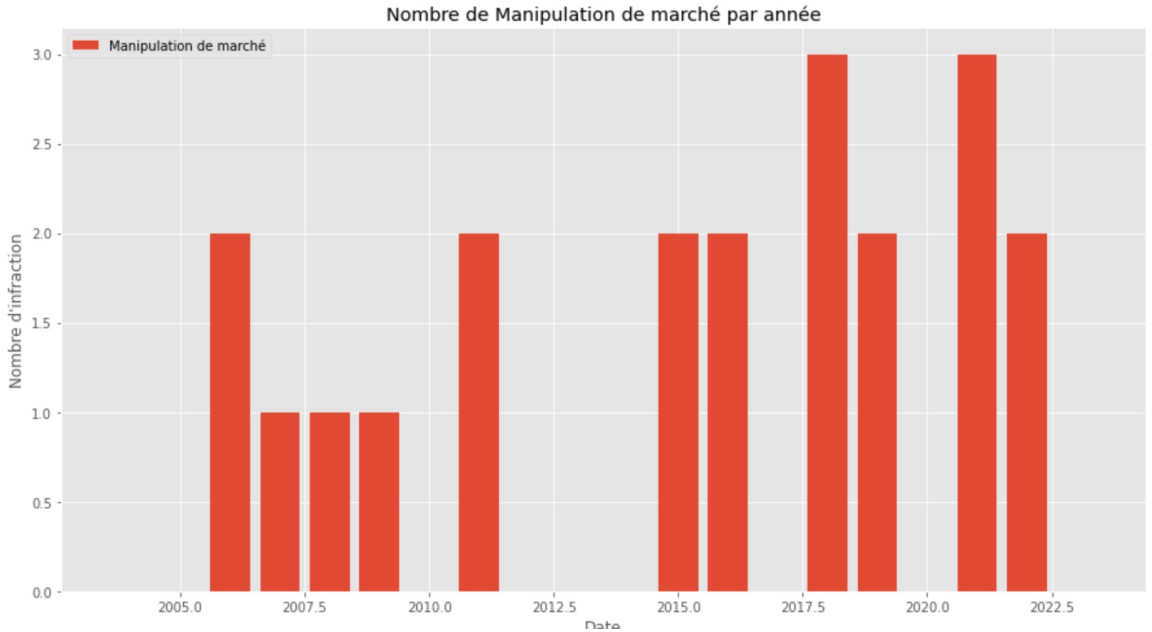


Explanation of the graph

- 4th step: Analyzing a dataset with pandas and creating graphs with matplotlib

4th graph:





Difficulties encountered:

We didn't know the number of lawsuits concerning insider trading offences there has been each year. Maybe one ? compared to market manipulation.

- **Histogram by calculating the number of offences per year per theme:** there is the number of offences committed in relation to the theme by dates

T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy

Conclusion

- To conclude, we were able to identify certain trends in the penalties imposed by the AMF according to the facts and the date.

- Nevertheless, other algorithms could be used from this data, in particular to **predict in the future** (with even more precision) the amount of the fine imposed by the AMF.

Thank you for listening !

Questions ?

T.Chevreteau – E.Poupet--Cauli – L.Zynger – T.Levy