Final Presentation

Analysis of the cassation rate depending on the court of appeal



Goal of the work conducted:

Analyze, through a code written in Python, the rate of cassation of the decisions of courts of appeal according to their jurisdiction, and determine which courts most often try to deviate from, or modify, the constant jurisprudence.

Step 1: Scraping - goals and process



Collect data from Légifrance, i.e. the decisions of the Court of Cassation of the last 3 years (between 1 January 2019 and 31 December 2021)



Sort out, among the targeted decisions, the elements of interest to be extracted (date of the decision, judgment panel, court of appeal whose decision is being challenged, meaning of the decision)



Organize this information in a Dataframe









21092 texte(s) trouvé(s)

Date de décision antéchronologique 💙

Afficher 100 résultats par page >



- Cour de cassation, criminelle, Chambre criminelle, 21 décembre 2021, 21-85.560, Publié au bulletin
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-60.157, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-60.156, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-60.153, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-60.152, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-60.091, Inédit
- > Cour de cassation, civile, Chambre civile 1, 16 décembre 2021, 21-18.941, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 21-70.023, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 20-14.553, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 20-13.748, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 20-11.725, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 20-17.326, Inédit
- > Cour de cassation, civile, Chambre civile 2, 16 décembre 2021, 20-14.233, Inédit



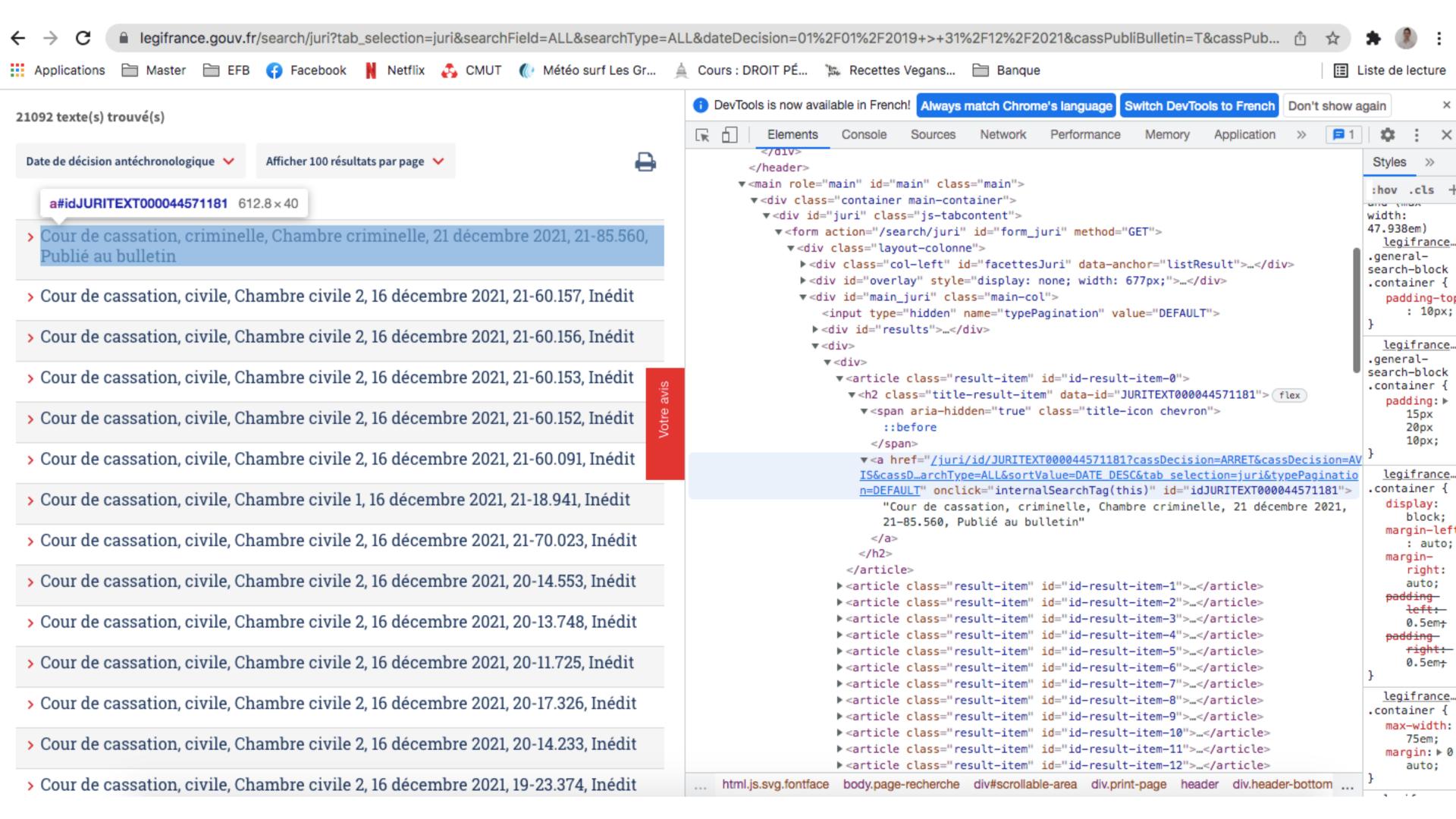
```
⊝import re
                    # Regex, module used to find A 7 A 22 € 8 ^ ∨
       <expressions.</pre>
       import os
                      # Module used to navigate through files
       import pandas as pd  # Module useful for data analysis and ==
         manipulation
       import time
                      # Various time-related functions
       from bs4 import BeautifulSoup # HTML reading module
       from selenium import webdriver
       from selenium.webdriver.common.by import By
      ⊋from collections import Counter, defaultdict
10
11
12
       Data_scrap =[] # Empty list to store future data. This
        will be useful later in the code.
                                                                  url_list =[] # Empty list to store pages with 100
        decisions from Legifrance
15
16
       # We split the URL in 3 parts : a fixed one first, then the
        number of the page (over 211), then the last fixed part of ___
        the URL. We will only change the number of the page to get —
        all the URL.
```

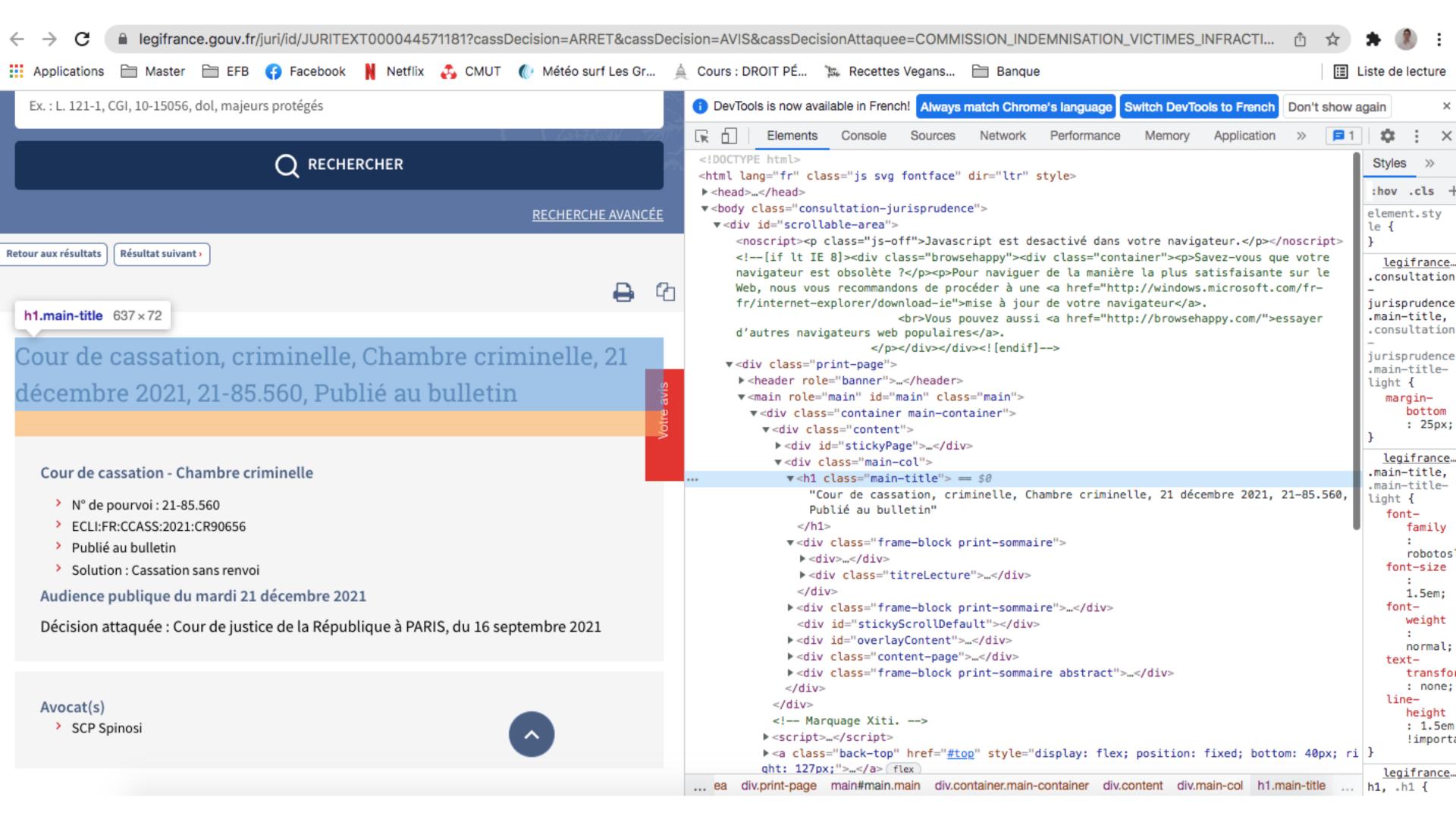
We split the URL in 3 parts : a fixed one first, then the number of the page (over 211), then the last fixed part of the URL. We will only change the number of the page to get all the URL. url_beginning = "https://www.legifrance.gouv.fr/search/juri?tab_selection=juri&searchField=ALL&searchType=ALL&dateDecision=01%2F01%2F2019+%3E+31%2F12 %2F2O21&cassPubliBulletin=T&cassPubliBulletin=F&cassDecision=ARRET&cassDecision=AVIS&cassFormation=ASSEMBLEE_PLENIERE&cassFormation=CHAMBRE_MIXTE &cassFormation=CHAMBRE_CIVILE_1&cassFormation=CHAMBRE_CIVILE_2&cassFormation=CHAMBRE_CIVILE_3&cassFormation=CHAMBRE_COMMERCIALE&cassFormation =CHAMBRE_SOCIALE&cassFormation=CHAMBRE_CRIMINELLE&cassFormation=COMMISSION_REPARATION_DETENTION&cassFormation=AVIS&cassDecisionAttaquee =COMMISSION_INDEMNISATION_VICTIMES_INFRACTIONS&cassDecisionAttaquee=CONSEIL_PRUDHOMME&cassDecisionAttaquee=COUR_APPEL&cassDecisionAttaquee =COUR_ASSISES&cassDecisionAttaquee=COUR_CASSATION&cassDecisionAttaquee=COUR_JUSTICE_REPUBLIQUE&cassDecisionAttaquee =COUR_NATIONAL_INCAPACITE_TARIFICATION&cassDecisionAttaquee=TRIBUNAL_CORRECTIONNEL&cassDecisionAttaquee=TRIBUNAL_COMMERCE&cassDecisionAttaquee <u>=TRIBUNAL_GRANDE_INSTANCE&cassDecisionAttaquee=TRIBUNAL_POLICE&cassDecisionAttaquee=TRIBUNAL_PREMIERE_INSTANCE&cassDecisionAttaquee</u> =TRIBUNAL_AFFAIRES_SECURITE_SOCIALE&cassDecisionAttaquee=TRIBUNAL_FORCES_ARMEES&cassDecisionAttaquee=TRIBUNAL_INSTANCE&cassDecisionAttaquee =TRIBUNAL_CONTENTIEUX_INCAPACITE&cassDecisionAttaquee=TRIBUNAL_MARITIME_COMMERCIAL&cassDecisionAttaquee=TRIBUNAL_PARITAIRE_BAUX_RURAUX_ &cassDecisionAttaquee=TRIBUNAL_SUPERIEURS_APPEL&typePagination=DEFAULT&sortValue=DATE_DESC&pageSize=100&page=" url_ending = "&tab_selection=juri#juri" to_add_url = "https://www.legifrance.gouv.fr" # When we are scrapping, this part of the URL is missing for every decision's URL, so we create a specific part to add at the beginning of every URL. This will be useful later in the code.

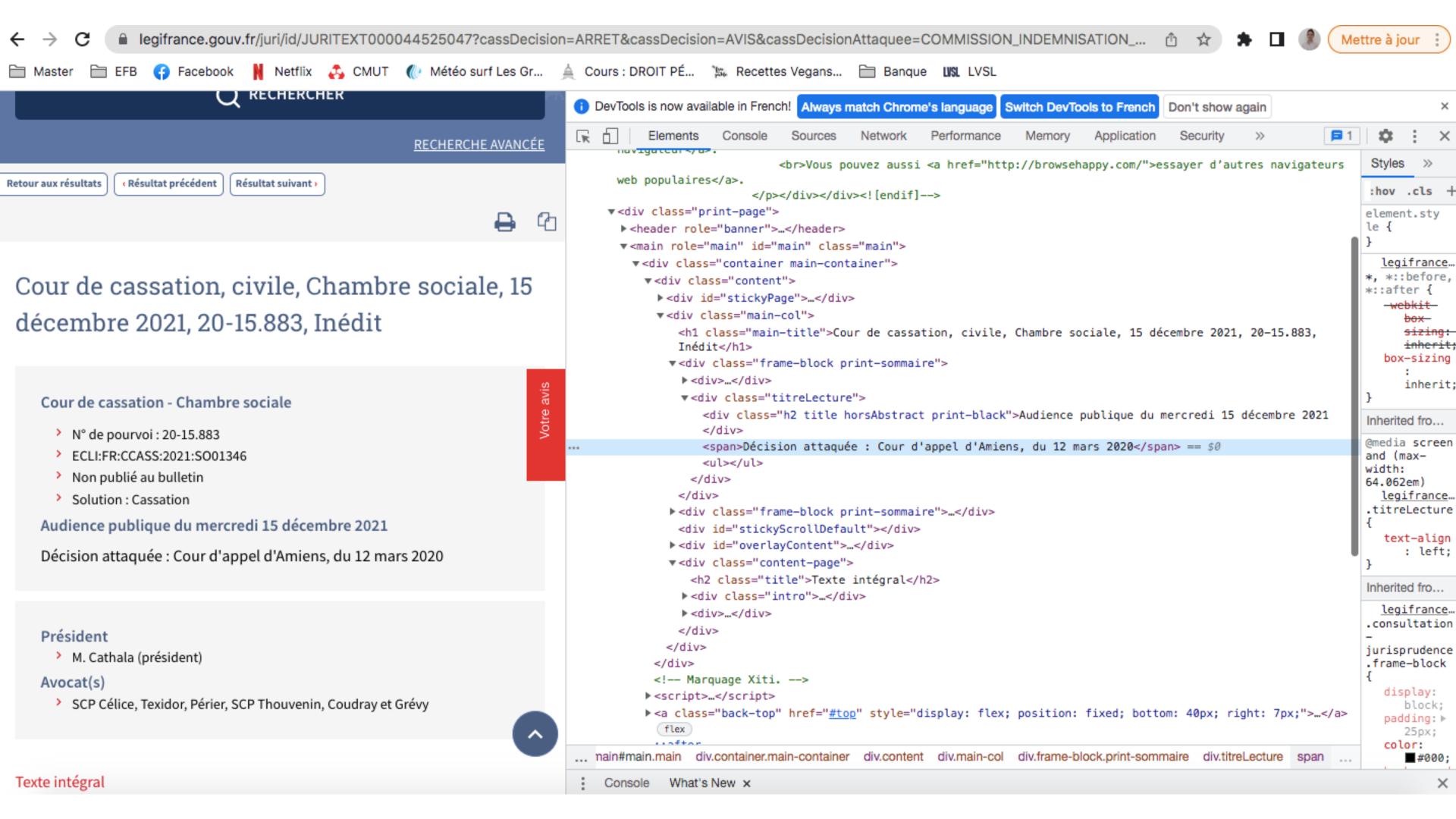
21

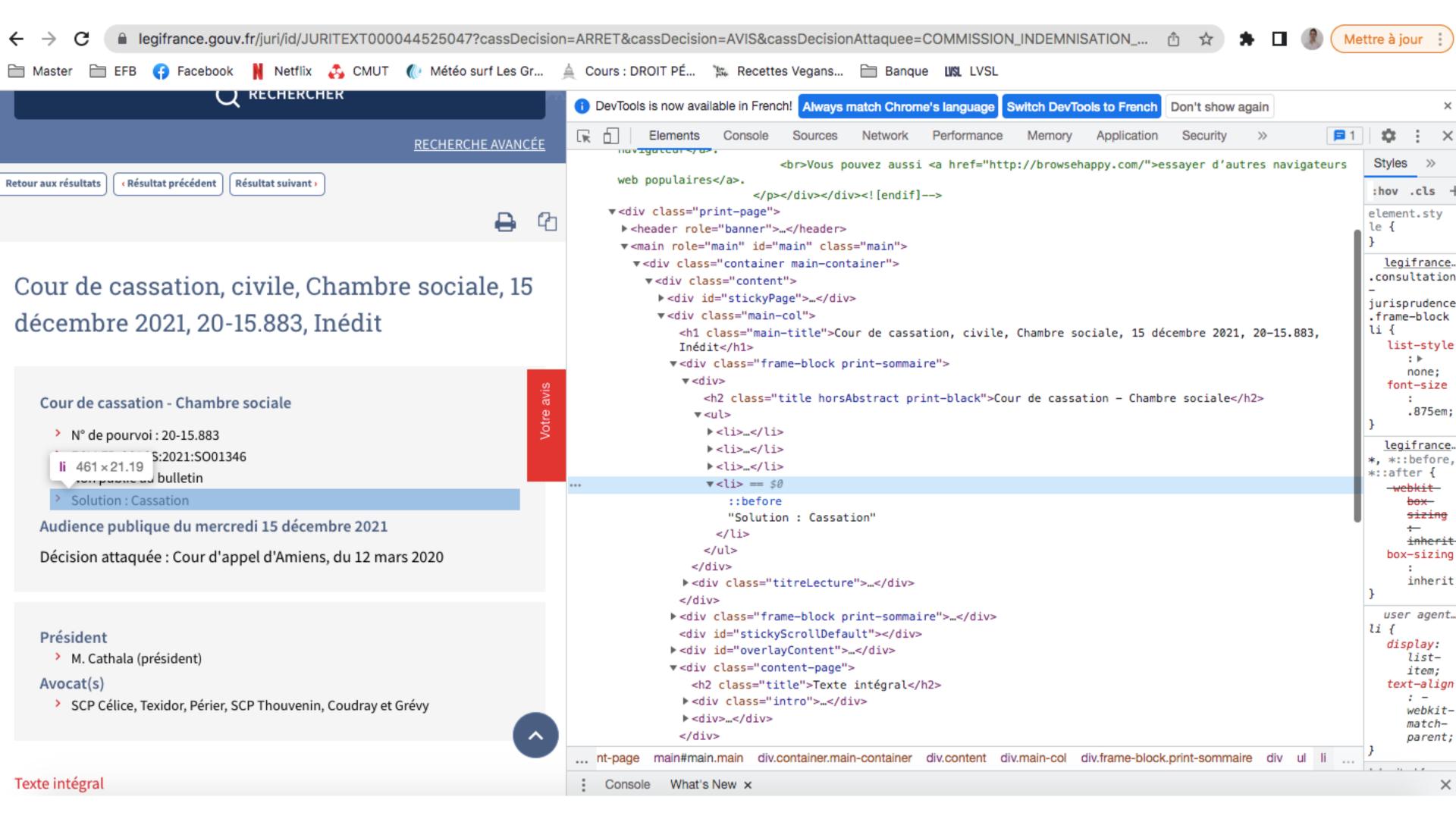
22

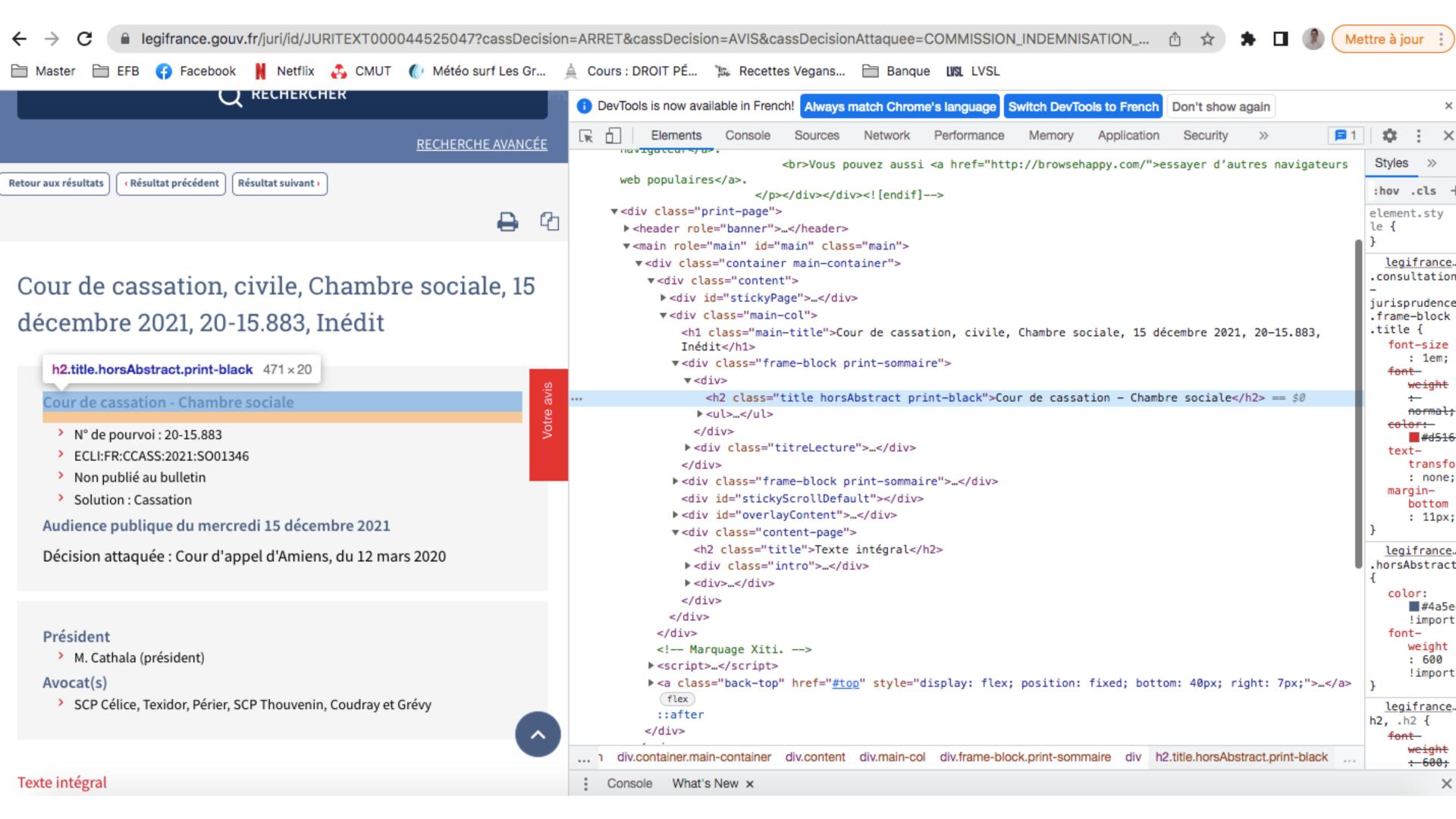
```
r = range(211) # Enter number of pages (211)
driver = webdriver.Chrome(executable_path=_r"/Users/fannycaunes/Downloads/chromedriver") # Open browser
for i in r:
    a = i+1 # Give the real page number
    url_tot = url_beginning + str(a) + url_ending_# Recompose the URL with the 3 different parts
    driver.get(url_tot)
    soup = BeautifulSoup(driver.page_source, features="html.parser") # Putting the first page result in soup, to analyse the HTML
    set_articles = soup.find_all("article") # Find all "articles" in the code source, by looking for all links whose text starts with "article" (cf.
     Screen 1)
    for article in set_articles:
        article_url = article.find(href=True) # Verify that there is a href, because the URL of every decision is behind an element called href in the
         code source
        url_list.append(to_add_url + article_url['href']) # Extract the URL from article_url, and adding, first, the beginning of the URL. We store all
          the URL of the decisions in a list of URLs created at the beginning of the code (cf. Screen 2)
driver.close()
```



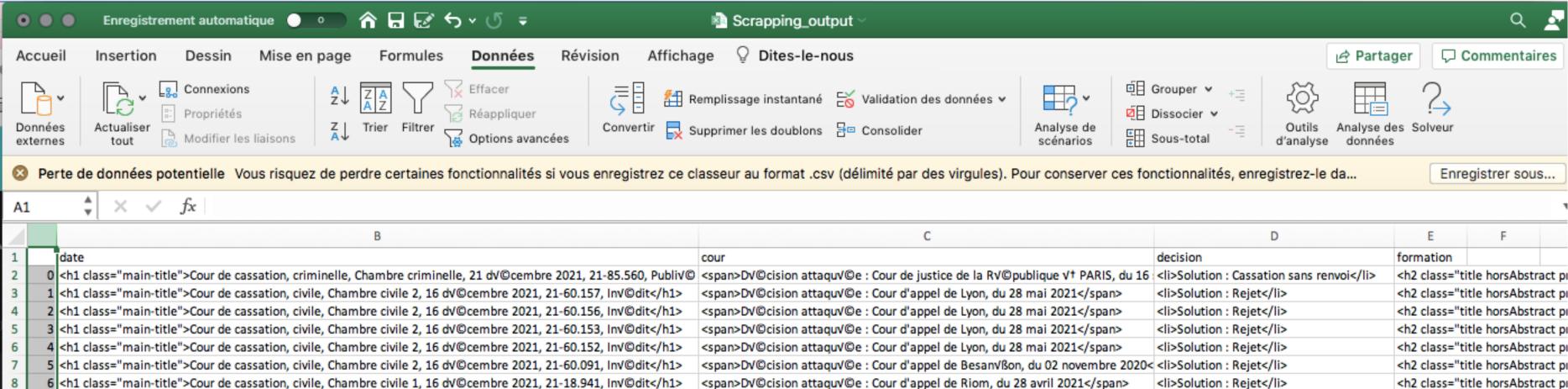








```
⊖# Now we go through all the URLs
  driver = webdriver.Chrome(executable_path=_r"/Users/fannycaunes/Downloads/chromedriver") # Open browser
 driver.get(url)
     soup = BeautifulSoup(driver.page_source, features="html.parser") # Read HTML, pass it to a soup object
      date = soup.find(class_="main-title")_# Here we can find the date of the decision of the Cour de cassation (cf. Screen 3)
     data2 = soup.find_all("span") # The name of the court is stored in a "span" object (cf. Screen 4)
     pattern = "Décision attaquée" # We search this pattern to get the court
     for j in data2:
if re.search(pattern, str(j)):
             cour = j
             break
 # Now we can find the solution of the decision (cf. Screen 5)
     data3 = soup.find_all("li")
     pattern2 = "Solution :"
for k in data3:
        if re.search(pattern2, str(k)):
             decision = k
             break
 # Now we find the division of the Cour de <u>cassation</u> that issued the decision
     data4 = soup.find_all("h2")
     pattern3 = "Cour de cassation -"
    for <u>l</u> in data4:
        if re.search(pattern3, str(l)):
            formation = l
            break
     vector = [date_cour_decision_formation]
 # We add this to the DataFrame
    Data_scrap.append(vector)
 driver.close()
 #print(Data_scrap)
 df = pd.DataFrame(Data_scrap_columns=_["date"_t"cour"_t"decision"_t"formation"])
 df.to_csv("Scrapping_output.csv", encoding="utf8")
```





Step 2: Data treatment - goals and process



Clear the Dataframe that week-endek-end created while scrapping



Organize the data in order to exploit them



```
df['date'] = df['date'].fillna("")_# Replace the empty cells
# We remove the text coming from scrapping manually for it to be clearer
# We use this code because of issue using regex with pandas
df['date'] = df['date'].str.replace('<h1 class="main-title">', '').str.strip()
df['date'] = df['date'].str.replace('</h1>', '').str.strip()
df['cour'] = df['cour'].str.replace('<span>', '').str.strip()
df['cour'] = df['cour'].str.replace('</span>', '').str.strip()
df['decision'] = df['decision'].str.replace('', '').str.strip()
df['decision'] = df['decision'].str.replace('', '').str.strip()
df['formation'] = df['formation'].str.replace('<h2 class="title horsAbstract print-black">', '').str.strip()
df['formation'] = df['formation'].str.replace('</h2>', '').str.strip()
# For the decision column, we remove the string "Solution: "
df['decision'] = df['decision'].str.replace('Solution : ', '').str.strip()
# In a same way, for the formation column, we remove the string "Cour de cassation - "
df['formation'] = df['formation'].str.replace('Cour de cassation - ', '').str.strip()
# Creation of a function to extract the court concerned from phrase
def cour_extraction(phrase):
    a = re.split(': ',phrase)[1]
    return re.split(',',a)[0]
df['cour'] = df['cour'].apply(cour_extraction)
 df['date'] = df['date'].apply(date_extraction)
 # For missing date, we simplify by applying the date of the previous decision
```

for i in range(len(df)):

if df.at[i,'date'] == "":

df.at[i,'date']=df.at[i-1,'date']

```
def day_extraction(phrase):
        return re.split(' '_phrase)[1]
def month_extraction(phrase):
    try:
        return re.split(' '_phrase)[2]
    except:
        return ""
def year_extraction(phrase):
    try:
        return re.split(' '__phrase)[3]
    except:
        return ""
df['day'] = df['date'].apply(day_extraction)
df['month'] = df['date'].apply(month_extraction)
df['year'] = df['date'].apply(year_extraction)
df.drop(df[df['month'] == ""].index, inplace_=_True)
df.to_csv("Data_cleaned.csv", encoding="utf8")
```

72

81

82

83

87

date	cour	decision	formation	day	month	year
21 dV@cembre 2021	Cour de justice de la République à PARIS	Cassation sans renvoi	Chambre criminelle	21	décembre	2021
16 dV@cembre 2021	Cour d'appel de Lyon	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Lyon	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Lyon	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Lyon	Rejet	Chambre civile 2	16	dV@cembre	2021
16 dV©cembre 2021	Cour d'appel de Besanvßon	Rejet	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Riom	Rejet	Chambre civile 1	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Bordeaux	Irrecevabilité de la demande d'avis	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de NVÆmes	Cassation	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel d'Aix-en-Provence	Cassation	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Nancy	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel d'Amiens	Cassation	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Nancy	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Papeete	Rejet	Chambre civile 2	16	dV@cembre	2021
16 dV©cembre 2021	Cour d'appel de Bastia	Rejet	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Tribunal d'instance de Bonneville	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Grenoble	Cassation partielle	Chambre civile 2	16	dV@cembre	2021
16 dV@cembre 2021	Cour d'appel de Paris	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Montpellier	Rejet	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Fort-de-France	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Colmar	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Dijon	Rejet	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Nancy	Cassation	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Versailles	Cassation partielle	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Nouméa	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel d'Aix-en-Provence	Cassation	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel d'Aix-en-Provence	Rejet	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Paris	Rejet	Chambre civile 2	16	décembre	2021
16 dV©cembre 2021	Cour d'appel de Caen	Annulation	Chambre civile 2	16	décembre	2021
16 dV@cembre 2021	Cour d'appel de Reims	Rejet	Chambre civile 2	16	dV@cembre	2021
16 dV@cembre 2021	Cour d'appel de Montpellier	Cassation partielle	Chambre civile 2	16	dV@cembre	2021

```
#Pourcentage de rejet par formation
A=data_interet['formation'].unique()
B=list(A)
rejet=[0 for i in range(0,len(B))]
decision=[0 for i in range(0,len(B))]
for i in range(0,data_interet.shape[0]):
    indice=B.index(data_interet.loc[i,'formation'])
    decision[indice] = decision[indice] + 1
    if data_interet.loc[i,'decision']=='Rejet':
        rejet[indice]=rejet[indice]+1
pourcent_rejet=[0 for i in range(0,len(B))]
for i in range(0,len(rejet)):
    pourcent_rejet[i]=rejet[i]/decision[i]
data_sortie_0=pd.DataFrame(columns=B)
data_sortie_0.loc[0]=pourcent_rejet
data_sortie_0
```

```
#Pourcentage de cassation sans renvoi par mois
 C=data_interet['month'].unique()
 D=list(C)
rejet=[0 for i in range(0,len(D))]
 decision=[0 for i in range(0,len(D))]
for i in range(0,data_interet.shape[0]):
     indice=D.index(data_interet.loc[i,'month'])
     decision[indice] = decision[indice] + 1
     if data_interet.loc[i,'decision']=='Cassation sans renvoi':
         rejet[indice]=rejet[indice]+1
pourcent_rejet=[0 for i in range(0,len(D))]
for i in range(0,len(rejet)):
```

pourcent_rejet[i]=rejet[i]/decision[i]

```
print(data_sortie_1)
print(data_sortie_0)
plt.bar(data_sortie_0.columns, data_sortie_0.values[0])
plt.xticks(rotation = 45)
plt.title('rejet par formation')
plt.show()
plt.plot(data_sortie_1.columns, data_sortie_1.values[0])
plt.xticks(rotation = 45)
plt.title('cassation sans renvoi par mois')
plt.show()
```

