

CONSEIL CONSTITUTIONNEL

QPC ANALYSIS

METHOD

STEP 1 : Creation of the dataframe

the number of the decision,

the title,

the outcome,

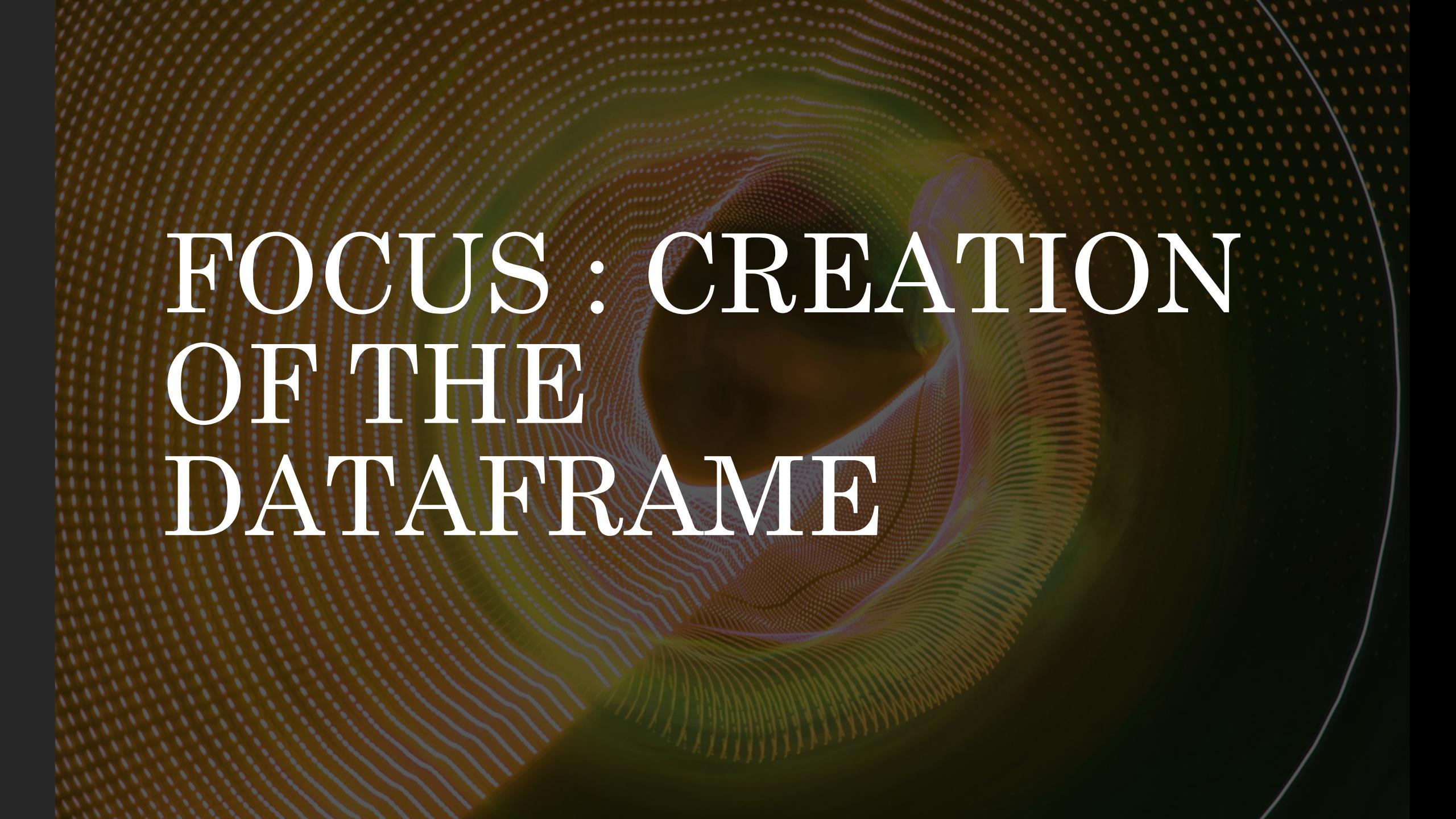
the date,

the legal texts invoked

STEP 2 : Analysis of the dataframe

Identify the sources of law invoked

evaluate the proportion of QPC subjects over time



FOCUS : CREATION
OF THE
DATAFRAME

ON ONE PAGE ...

TRIER PAR Date (antéchronologique) 1 sur 92 >

Décision - 2022-1029 QPC - M. Sami C. [Cause statutaire d'exclusion d'un associé d'une société par actions simplifiée] - du 09 décembre 2022

[Conforme] LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 13 octobre 2022 par la Cour de cassation (chambre commerciale, arrêt n° 699 du 12 octobre 2022), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour le...]

Décision - 2022-1027/1028 QPC - Conseil national de l'ordre des médecins [Dispositif de non-concurrence applicable à certains praticiens exerçant dans un établissement public de santé] - du 09 décembre 2022

[Conforme] LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 28 septembre 2022 par le Conseil d'Etat (décisions nos 462977 et 462978 du même jour), dans les conditions prévues à l'article 61-1 de la Constitution, de deux questions prioritaires de constitutionnalité. Ces questions ont été posées pour le...]

Décision - 2022-1026 QPC - Association France horizon [Assujettissement de certaines associations à la taxe pour la création de locaux à usage de bureaux, de commerce ou de stockage en Ile-de-France] - du 25 novembre 2022

[Conforme] LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 23 septembre 2022 par le Conseil d'Etat (décision n° 462956 du même jour), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour l'association...]

Décision - 2022-1021 QPC - Mme Marie P. [Requête en nullité d'un acte d'investigation déposée par un journaliste n'ayant ni la qualité de partie à la procédure ni celle de témoin assisté] - du 28 octobre 2022

[Conforme] LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 28 juillet 2022 par la Cour de cassation (chambre criminelle, arrêt n° 1025 du 27 juillet 2022), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour le...]

Décision - 2022-1020 QPC - Mme Célia C. [Accès des tiers au dossier de la procédure d'instruction dans le cadre d'une demande de restitution d'un bien saisi] - du 28 octobre 2022

[Conforme] LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 28 juillet 2022 par la Cour de cassation (chambre criminelle, arrêt n° 1022 du 27 juillet 2022), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour le...]

Décision - 2022-1019 QPC - M. Bruno M. [Composition des instances disciplinaires de l'ordre des experts-comptables] - du 27 octobre 2022

```

url = "https://recherche.conseil-constitutionnel.fr/?expert=2&filtres1%5B0%5D=sous_type_decision:QPC"
casdf=pd.DataFrame([{}],columns=[ "Numéro-décision", "titre", "type", "date", "visa"]) #création du dataframe

def construct_sublist(a): #a est un paramètre de la fonction construite, il renvoie la décision
    sublist = 5*[[]] #la sublist ne peut contenir que 5 éléments : numéro décision, titre, type, date, sujet
    solution = a.find("span", class_="type").get_text() #on extrait le texte lié à la balise "span de classe type"
    solution = re.split("\n|\n", solution)[1] #grâce à une regex, on retire les crochets, on garde le deuxième élément
    date = a.find("span", class_="date").get_text() #on extrait le texte lié à la balise "span de classe date"
    sujetQPC = a.find("a").get_text() #on extrait le texte lié à la balise "a" de la décision
    tab = re.split('(\|\|)', sujetQPC) # "\|" enlève le caractère spécial. "||" ou, on split la chaîne
    sublist[0]=tab[1] #a la position zero on a tab 1, numéro de décision
    sublist[1]=tab[4] #a la position 2 on a tab 4, rajouter sujet de la décision, entre crochets
    sublist[2]=date #rajouter la date
    sublist[3]=solution #rajouter la solution
    return sublist

i=0 # i est le compteur qui augmentera à chaque texte ligne 49
for x in range_(92): #boucle 92 fois pour les 92 pages
    webpage = requests.get(url) # We connect on the first page that lists 10 decisions
    soup = BeautifulSoup(webpage.content, "html.parser") # We get the html of that page
    aas = soup.find_all("article", class_="type-decision") # And we find the elements representing these decisions
    for a in aas: # We start a second loop, going now decision by decision, using the elements we just found
        href = a.find("a").get("href") # Each of that element, in the attribute href, has the url to the decision
        # Looping through that list, we create a sublist in which we add info about each case, starting with the decision itself
        sublist=construct_sublist(a) #sublist avec 4 éléments à ce moment. Subliste est la liste des données pour cette décision
        webpage2=requests.get(href) # we connect on the page of the décision en fonction du href (URL)
        soup2=BeautifulSoup(webpage2.content, "html.parser") # We get the html of that page

```

conseil-constitutionnel.fr/decision/2022/20221026QPC.htm

Décision n° 2022-1026 QPC du 25 novembre 2022

Association France horizon [Assujettissement de certaines associations à la taxe pour la création de locaux à usage de bureaux, de commerce ou de stockage en Ile-de-France]

[Conforme]

LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 23 septembre 2022 par le Conseil d'Etat (décision n° 452256 du même jour), dans les conditions prévues à l'article 61 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée par l'association France horizon par la SCP Hélène Didier et François Pinet, avocat au Conseil d'Etat et à la Cour de cassation. Elle a été enregistrée au secrétariat général du Conseil constitutionnel sous le n° 2022-1026 QPC. Elle est relative à la conformité aux droits et libertés que la Constitution garantit des articles L. 520-1 et L. 520-6 du code de l'urbanisme, dans leur rédaction résultant de la loi n° 2015-1786 du 29 décembre 2015 de finances rectificative pour 2015.

542,55 x 217,69

* la Constitution ;
* l'ordonnance n° 58-1067 du 7 novembre 1958 portant loi organique sur le Conseil constitutionnel ;
* le code général des impôts ;
* la loi n° 2015-1786 du 29 décembre 2015 de finances rectificative pour 2015 ;
* le règlement du 4 février 2010 sur la procédure suivre devant le Conseil constitutionnel pour les questions prioritaires de constitutionnalité ;

Au vu des pièces suivantes :

- les observations présentées par l'association requérante par la SCP Hélène Didier et François Pinet, enregistrées le 12 octobre 2022 ;
- les observations présentées par la Première ministre, enregistrées le même jour ;
- les autres pièces produites et jointes au dossier ;

DevTools is now available in French!

Always match Chrome's language Switch DevTools to French Don't show again

```

for a in das: # We start a second loop, going now decision by decision. Using the elements we just found, on loop s
    href = a.find("a").get("href") # Each of that element, in the attribute href, has the url to the decision itself
    # Looping through that list, we create a sublist in which we add info about each case, starting with the url
    sublist=construct_sublist(a) #sublist avec 4 éléments à ce moment. Subliste est la liste des données extraites p
    webpage2=requests.get(href) # we connect on the page of the decision en fonction du href (URL de redirection de
    soup2=BeautifulSoup(webpage2.content, "html.parser") # We get the html of that page

    aas2=[] #tableau de textes de visa
    if (x<4):
        aas2=soup2.find_all("div",class_="wrapper-content")[0].find_all("ul") # jusqu'à la page 44.0
    else:
        temp=soup2.find_all("div",class_="wrapper-content")[0].find_all("p") # on récupère toutes les balises paragra
        for elt in temp: #on parcourt le tableau temp. Deux choses ne nous intéressent pas : le paragraphe introducti
            if re.match("^Vu le règlement du 4 février",elt.get_text()): #que l'élément match la regex, à partir de
                break #on quitte le for et on passe aux deuxième for
            elif temp.index(elt)==0 or temp.index(elt)==1: # si premier ou deuxième élément on continue, on les next
                continue
            else:
                aas2.append(elt) #si aucun des 2 précédent, il append
    for obj in aas2: #on parcourt les textes dans aas2
        sublist[4]=re.split('\xa0',obj.get_text())[0] # pour chaque texte on le nettoie, et on l'ajoute à la positio
        cassdf.loc[i]=sublist #.loc est une fonction qui ajoute la donnée à la location index "i" dans le data fram
        i+=1 # pour mettre une ligne un visa, on augmente i à chaque fois qu'on a ajouté un texte

```

The screenshot shows a decision document from the Constitutional Council of France. The main title is "Décision n° 2010-3 QPC du 28 mai 2010". The document discusses the constitutionality of Article 61-1 of the Constitution regarding the protection of family rights. It includes several sections of text and legal citations.

The screenshot shows the browser's developer tools with the "Elements" tab selected. It highlights the HTML structure of the decision document, specifically focusing on the main article content and its sub-sections.

```

Users / doabiddine / Desktop / version finale.py

main.py x try.py x version finale.py x VERSION FINALE x QPC.py x
27 tor a in das: # We start a second loop, doing now decision by decision, using the elements we just found, on loop s
28 href = a.find("a").get("href") # Each of that element, in the attribute href, has the url to the decision itself
29
30 # Looping through that list, we create a sublist in which we add info about each case, starting with the url
31 sublist=construct_sublist(a) #sublist avec 4 éléments à ce moment. Subliste est la liste des données extraites p
32 webpage2=requests.get(href) # we connect on the page of the décision en fonction du href (URL de redirection de
33 soup2=BeautifulSoup(webpage2.content, "html.parser") # We get the html of that page
34
35 aas2=[] #tableau de textes de visa
36 if (x<4):
37     aas2=soup2.findAll("div",{"class":"wrapper-content"})[0].findAll("ul")[0].findAll("li") # jusqu'à la page 44 o
38 else:
39     temp=soup2.findAll("div",{"class":"wrapper-content"})[0].findAll(["p"])
40     for elt in temp:#on parcourt le tableau temp. Deux choses ne nous intéressent pas, le paragraphe introducti
41         if re.match("Vu le règlement du 4 février 2010",elt.getText()): #que l'élément match la regex, a partir de
42             break #on quitte le for et on passe aux deuxième for
43         elif temp.index(elt)==0 or temp.index(elt)==1:# si premier ou deuxième élément on continue, on les next
44             continue
45         else:
46             aas2.append(elt) #si aucun des 2 précédent, il append
47
48 for obj in aas2:#on parcours les textes dans aas2
49     sublist[4]=re.split('|\xa0;|\n',obj.getText())[0] # pour chaque texte on le nettoie, et on l'ajoute à la position
      cassdf.loc[i]=sublist # loc est une fonction qui ajoute la donnée à la location index "i" dans le data frame
      i+=1 # pour mettre une ligne un visa, on augmente i à chaque fois qu'on a ajouté un texte
    
```

PAGE BY PAGE ...

recherche.conseil-constitutionnel.fr/?id=a35262a4dccb2f69a36693ec74e69d26&filtres[]&type_doc%3AdossierComplet&filtres[]&sous... En pause

Il y a 914 résultats correspondant à votre recherche

FILTRER VOTRE RECHERCHE

TRIER PAR Date (antéchronologique) 1 sur 92 16.78

Décision - 2022-1029 QPC - M. Sami C. [Clause statutaire d'exclusion d'un associé d'une société par actions simplifiée] - du 09 décembre 2022

[Conformité] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 13 octobre 2022 par la Cour de cassation (chambre commerciale, arrêt n° 699 du 12 octobre 2022), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour...]

Décision - 2022-1027/1028 QPC - Conseil national de l'ordre des médecins [Dispositif de non-concurrence applicable à certains praticiens exerçant dans un établissement public de santé] - du 09 décembre 2022

[Conformité] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 28 septembre 2022 par le Conseil d'Etat (décisions nos 462977 et 462978 du même jour), dans les conditions prévues à l'article 61-1 de la Constitution, de deux questions prioritaires de constitutionnalité. Ces questions ont été posées pour le...]

Décision - 2022-1026 QPC - Association France horizon [Assujettissement de certaines associations à la taxe pour la création de locaux à usage de bureaux, de commerce ou de stockage en île-de-France] - du 25 novembre 2022

[Conformité] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 23 septembre 2022 par le Conseil d'Etat (décision n° 452256 du même jour), dans les conditions prévues à l'article 61-1 de la Constitution, d'une question prioritaire de constitutionnalité. Cette question a été posée pour l'association...]

Décision - 2022-1021 QPC - Mme Marie P. [Requête en nullité d'un acte d'investigation déposée par un journaliste n'ayant ni la qualité de partie à la procédure ni celle de témoin assisté] - du 28 octobre 2022

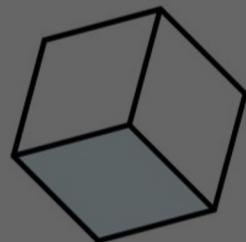
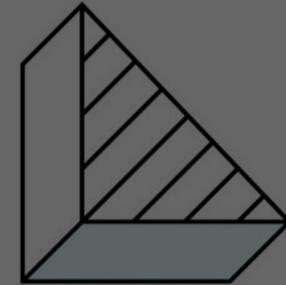
[Conformité] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 28 juillet 2022 par la Cour de cassation (chambre criminelle, arrêt n° 1125 du 27 juillet 2022), dans les conditions prévues à l'article 61-1 de la Constitution, d'une

```
<!DOCTYPE html>
<html lang="fr" dir="ltr" prefix="og: https://ogp.me/ns#> class=" js">
  <head></head>
  <body class="path-gabarit-decisions eu-cookie-compliance-status-0" data-eu-cookie-compliance-once="true">
    <a id="top" href="#hautDePage" class="visually-hidden focusable skip-link"> Aller au contenu principal </a>
    <div class="dialog-off-canva... main-canva... data-off-canva... main-canva...">
      <header role="banner"></header>
      <div class="region region-breadcrumb"></div>
      <div class="region region-highlighted"></div>
      <main role="main">
        <a id="hautDePage" tabindex="-1"></a>
        <div class="layout-content">
          <div class="region region-content">
            <div id="block-cc-content" class="block block-system block-system-main-block">
              <div class="recherche-synomia">
                <script></script>
                <style></style>
              <div class="recherche-synomia">
                <header></header>
                <section> flex
                  <div class="content">
                    <div class="actions" flex
                      <div class="order-by" flex></div>
                    <div class="cc-search-views-custom-form">
                      <nav class="pager" role="navigation" aria-labelledby="pagination-heading" flex>
                        <h4 id="pagination-heading" class="visually-hidden">Pagination</h4>
                        <ul class="pager-items" flex>
                          <li class="pager-item input-number"></li>
                          <li class="pager-item pager-item-next">
                            <a href="custom_search_conseil-constitutionnel.php?id=a35262a4dccb2f69a36693ec74e69_d...&sortBy=date&typeQuery=4&dateBefore=&dateAfter=&xtrmc=&xtnp=p2&rech_ok=1" title="Aller à la page suivante" rel="next"></a>
                          </li>
                          <li class="pager-item pager-item-last"></li>
                        </ul>
                      </nav>
                    </div>
                  </div>
                </section>
              </div>
            </div>
          </div>
        </main>
      </div>
    </body>
```

```
if (x!=91): # si x=91 on ne boucle pas la partie suivante (car pas de page suivante à la dernière page)
    li=soup.find_all("li", class_="pager__item pager3__item--next")#liste simple, avec deux li
    dom="https://recherche.conseil-constitutionnel.fr/"
    truc=li[0].find("a").get("href")#récupération href du tout premier élément du tableau
    rest=re.split('?', truc)#en regex '?' est une caractère spéciale donc mettre backslash avant pour avoir le caractère simple
    #split divise une string donnée selon un caractère en l'occurrence le point d'interrogation
    url=dom+'?' +rest[1]#construction url =>n... de domaine+?+queryparams

print_(cassdf)
```

DATAFRAME...



Users > doabiddine > Desktop > version finale.py

Project main.py try.py version finale.py VERSION FINALE QPC.py

127 154

else:
 aas2.append(elt) #si aucun des 2 précédent, il append
for obj in aas2: #on parcours les textes dans aas2
 sublist[4]=re.split('\xa0;'+obj.get_text())[0] # pour chaque texte on le nettoie, et on l'ajoute à la position 4 de la sublist, 5eme élément
 cassdf.loc[i]=sublist #.loc est une fonction qui ajoute la donnée à la location index "i" dans le data frame
 i+=1 # pour mettre une ligne un visa, on augmente i à chaque fois qu'on a ajouté un texte.
if (x!=91): # si x=91 on ne boucle pas la partie suivante (car pas de page suivante à la dernière page)
 lis=soup.find_all("li", class_="pager__item pager__item--next")#liste simple, avec deux li
 dom="https://recherche.conseil-constitutionnel.fr/"
 truc=li[0].find("a").get("href")#é recuperation href du tout premier élément du tableau
 rest=re.split('?',truc)#en regex '?' est une caractère spéciale donc mettre backslash avant pour avoir le caractère simple
 #split divise une string donnée selon un caractère en l'occurrence le point d'interrogation
 url=dom+'?' +rest[1]#construction url => nom de domaine+?+queryparams
print(cassdf)

for x in range (92) > if (x!=91)

Run: version finale

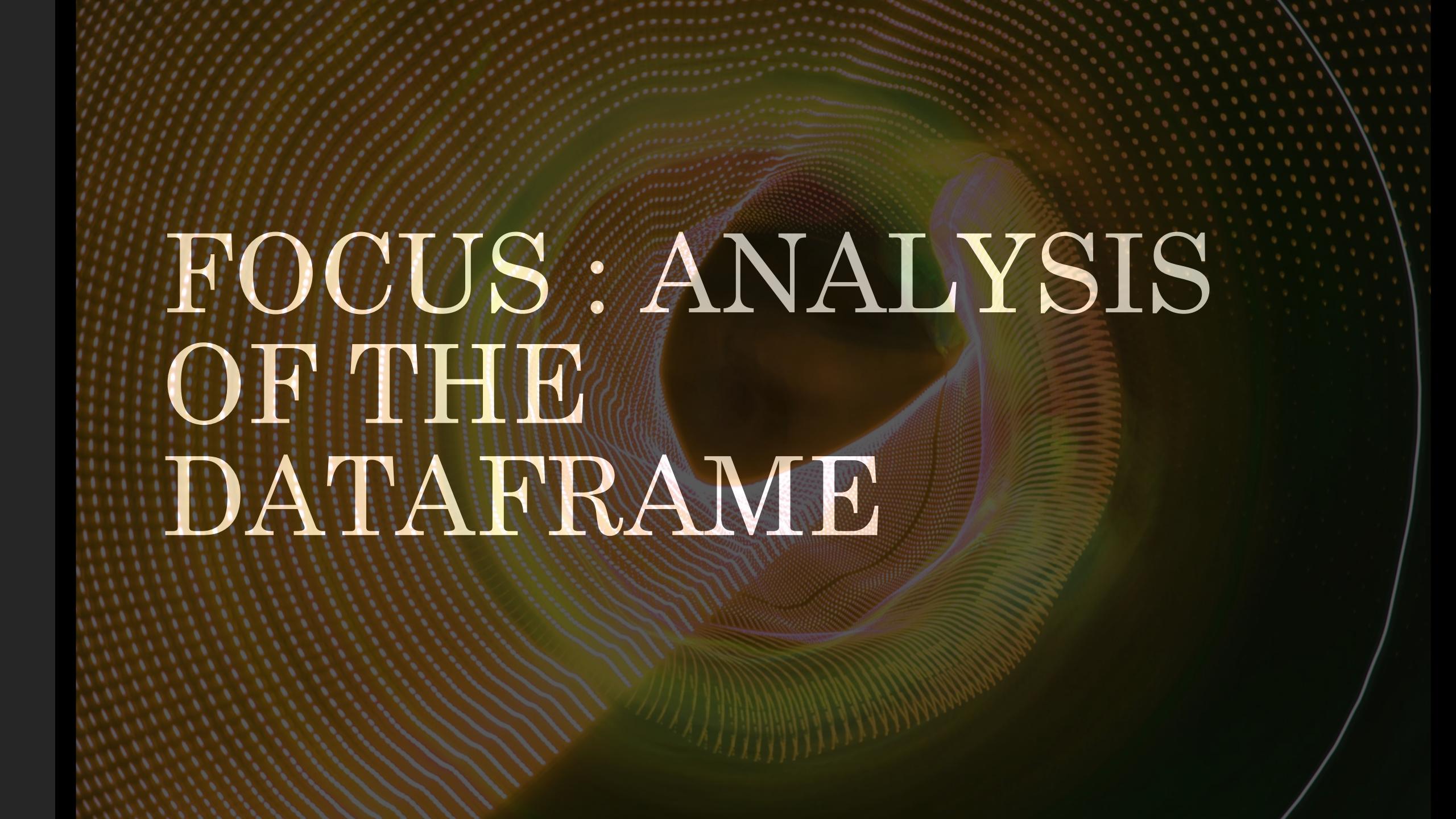
	Numéro-décision	visa
0	2022-1029	la Constitution
1	2022-1029	l'ordonnance n° 58-1067 du 7 novembre 1958 por...
2	2022-1029	le code civil
3	2022-1029	le code de commerce
4	2022-1029	l'ordonnance n° 2000-912 du 18 septembre 2000 ...
...
5046	2010-1	Vu la Constitution
5047	2010-1	Vu l'ordonnance n° 58-1067 du 7 novembre 1958 ...
5048	2010-1	Vu le code des pensions civiles et militaires ...
5049	2010-1	Vu le code des pensions militaires d'invalidit...
5050	2010-1	Vu la loi n° 59-1454 du 26 décembre 1959 de fi...

16851 rows x 5 columns

Version Control Run Python Packages TODO Python Console Problems Terminal Services

PEP 8: E261 at least two spaces before inline comment. PEP 8: E262 inline comment should start with '#'. PEP 8: E501 line too long (131 > 120 characters).

54:88 LF UTF-8 4 spaces



FOCUS: ANALYSIS
OF THE
DATAFRAME

```
import matplotlib.pyplot as plt
import pandas as pd
import locale

locale.setlocale(locale.LC_ALL, "fr_FR") # On change la locale pour qu'il puisse comprendre que les dates sont au format fr

cassdf.index = pd.to_datetime(cassdf["Date"], format="%d %B %Y") # On réindexe à l'aide de la date transformée au format datetime

# pour compter et plotter tous les visas par période d'un an
cassdf.resample("1Y").visa.value_counts().unstack().plot() # Ce réindexing permet du "resampling", qui prend comme argument une période
# On utilise le value_count() pour obtenir le nombre de visa par période donnée; une fois "unstack" (pour avoir index+colonnes), on peut le plot.
plt.title('Nombre de VISA / an')
plt.grid(True)
plt.show()
```

```
/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/bin/python /Users/maddalenatournier/PycharmProjects/pythonProject1/venv/final?.py
Traceback (most recent call last):
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/final?.py", line 62, in <module>
    cassdf.index = pd.to_datetime(cassdf["date"], format="%d %B %Y") # On réindexe à l'aide de la date transformée au format datetime
                                         ^
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/lib/python3.11/site-packages/pandas/core/tools/datetimes.py", line 1064, in to_datetime
    cache_array = _maybe_cache(arg, format, cache, convert_listlike)
                  ^
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/lib/python3.11/site-packages/pandas/core/tools/datetimes.py", line 229, in _maybe_cache
    cache_dates = convert_listlike(unique_dates, format)
                  ^
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/lib/python3.11/site-packages/pandas/core/tools/datetimes.py", line 430, in _convert_listlike_date
    res = _to_datetime_with_format(
          ^
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/lib/python3.11/site-packages/pandas/core/tools/datetimes.py", line 538, in _to_datetime_with_form
    res = _array.strptime_with_fallback(
          ^
  File "/Users/maddalenatournier/PycharmProjects/pythonProject1/venv/lib/python3.11/site-packages/pandas/core/tools/datetimes.py", line 473, in _array.strptime_with_f
    result, timezones = array.strptime(arg, fmt, exact=exact, errors=errors)
                      ^
  File "pandas/_libs/tslibs/strptime.pyx", line 150, in pandas._libs.tslibs.strptime.array.strptime
ValueError: time data 'Conformité' does not match format '%d %B %Y' (match)
```

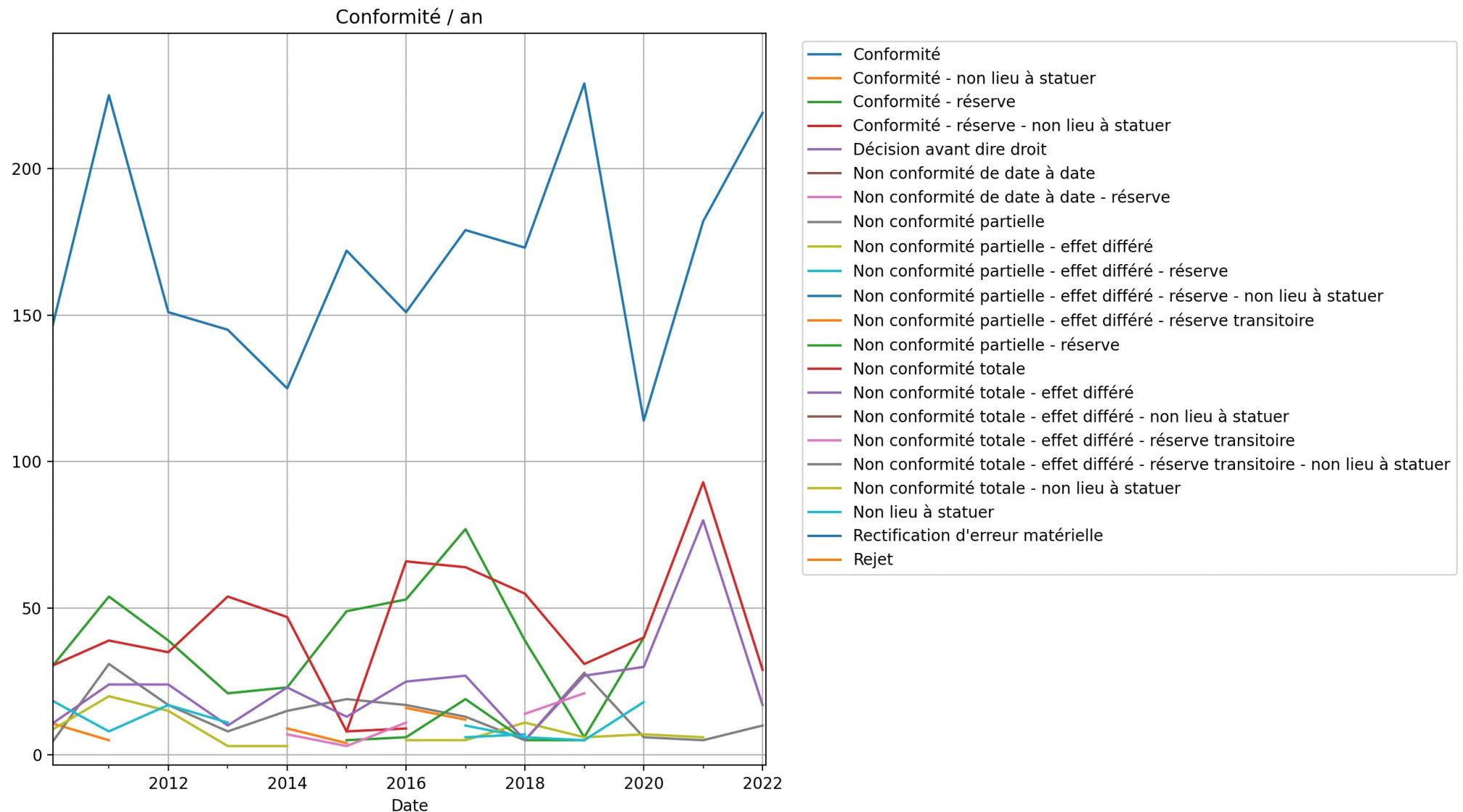
Process finished with exit code 1

```
cassdf.to_csv('cassdf.csv')

import matplotlib.pyplot as plt
import pandas as pd
import locale
locale.setlocale(locale.LC_ALL, "fr_FR") # On change la locale pour qu'il puisse comprendre que les dates sont au format fr

df = pd.read_csv("cassdf4.csv", header="infer")
df.index = pd.to_datetime(df["Date"], format="%d %B %Y") # On réindexe à l'aide de la date transformée au format datetime
df.Date = pd.to_datetime(df["Date"], format="%d %B %Y") # On réindexe à l'aide de la date transformée au format datetime

# pour compter et plotter tous les outcomes par période d'un an
df.resample("1Y").Outcome.value_counts().unstack().plot() # Ce réindexing permet du "resampling", qui prend comme argument une période
# On utilise le value_count() pour obtenir le nombre de visa par période donnée; une fois "unstack" (pour avoir index+colonnes), on peut le plot.
plt.title('Conformité / an')
plt.grid(True)
plt.legend(bbox_to_anchor=(1.04, 1), loc="upper left")
plt.show()
```



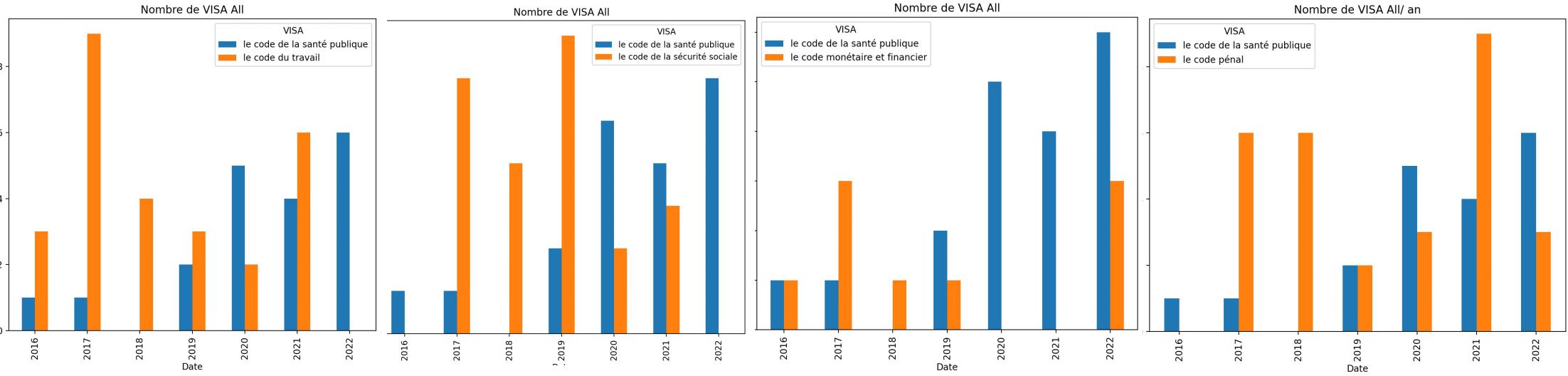
```
# pour compter et plotter les différents codes par rapport au code de la santé publique par période d'un an
X = ["le code de la santé publique","le code civil"]
grouper = df.loc[df.VISA.isin(X)].groupby([pd.Grouper(key='Date', freq='Y'), "VISA"])
ax2 = grouper["VISA"].size().unstack().plot(kind="bar")
ax2.set_title('Nombre de VISA/ an')
ax2.xaxis.set_ticklabels([pd.to_datetime(x).year for x in df.loc[df.VISA == "le code de la santé publique"].resample("1Y").size().index.values.tolist()])
plt.show()

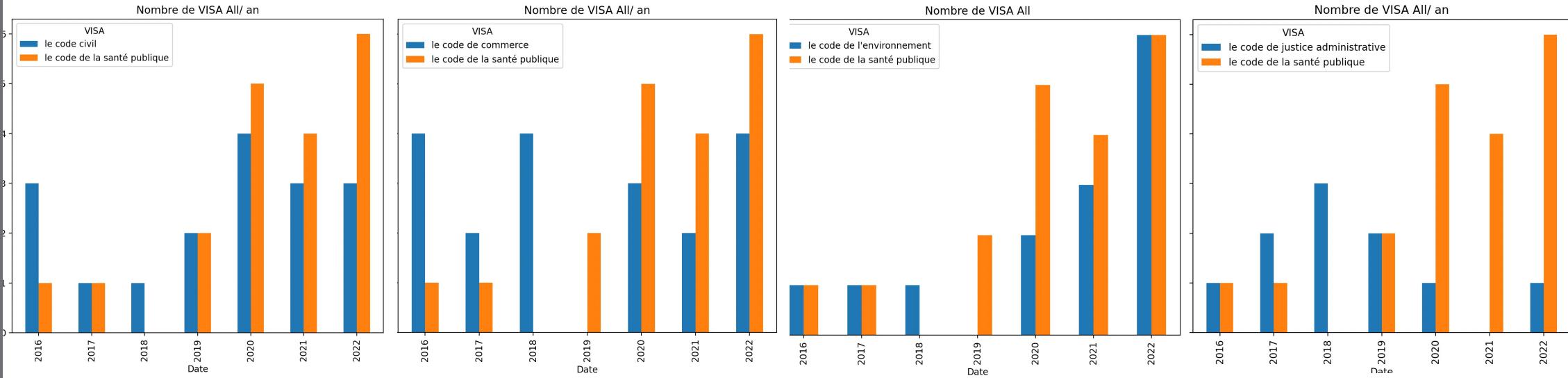
X = ["le code de la santé publique","le code pénal"]
grouper = df.loc[df.VISA.isin(X)].groupby([pd.Grouper(key='Date', freq='Y'), "VISA"])
ax2 = grouper["VISA"].size().unstack().plot(kind="bar")
ax2.set_title('Nombre de VISA/ an')
ax2.xaxis.set_ticklabels([pd.to_datetime(x).year for x in df.loc[df.VISA == "le code de la santé publique"].resample("1Y").size().index.values.tolist()])
plt.show()

X = ["le code de la santé publique","le code de justice administrative"]
grouper = df.loc[df.VISA.isin(X)].groupby([pd.Grouper(key='Date', freq='Y'), "VISA"])
ax2 = grouper["VISA"].size().unstack().plot(kind="bar")
ax2.set_title('Nombre de VISA/ an')
ax2.xaxis.set_ticklabels([pd.to_datetime(x).year for x in df.loc[df.VISA == "le code de la santé publique"].resample("1Y").size().index.values.tolist()])
plt.show()

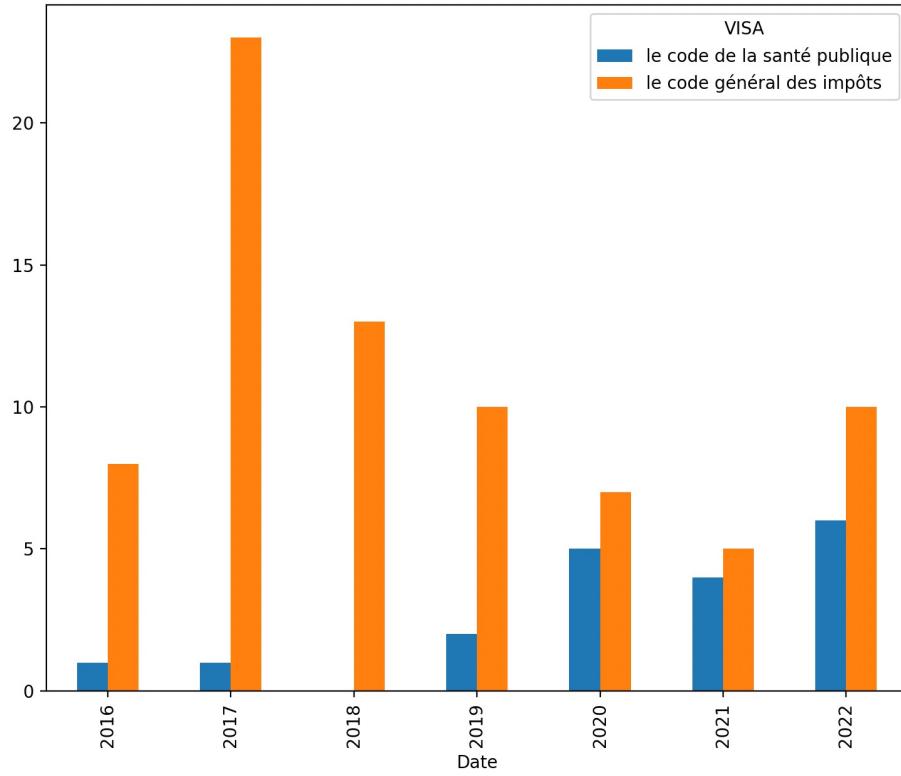
X = ["le code de la santé publique","le code monétaire et financier"]
grouper = df.loc[df.VISA.isin(X)].groupby([pd.Grouper(key='Date', freq='Y'), "VISA"])
ax2 = grouper["VISA"].size().unstack().plot(kind="bar")
ax2.set_title('Nombre de VISA All')
ax2.xaxis.set_ticklabels([pd.to_datetime(x).year for x in df.loc[df.VISA == "le code de la santé publique"].resample("1Y").size().index.values.tolist()])
plt.show()

X = ["le code de la santé publique","le code général des impôts"]
grouper = df.loc[df.VISA.isin(X)].groupby([pd.Grouper(key='Date', freq='Y'), "VISA"])
ax2 = grouper["VISA"].size().unstack().plot(kind="bar")
ax2.set_title('Nombre de VISA All')
ax2.xaxis.set_ticklabels([pd.to_datetime(x).year for x in df.loc[df.VISA == "le code de la santé publique"].resample("1Y").size().index.values.tolist()])
plt.show()
```





Nombre de VISA All



Nombre de VISA All

