

LEGAL DATA ANALYTICS

THE MOST CONVICTED STATES BY THE CJEU IN TAX MATTERS

Les Pythons de la Fournaise

GOALS

- Find the Top 5 countries that are the most condemned by the Court of Justice of the European Union in terms of tax law.
- Determine the Top 5 tax areas in which they are the most condemned.

ROADMAP

1

THE CODE

- Database
- Scraping
- Dataframe
- Visualizer

2

ANALYSIS

- Top 5 : The worst performers
- Contentious matters

1 – THE CODE

A – Database


CJEU website : https://european-union.europa.eu/institutions-law-budget/institutions-and-bodies/institutions-and-bodies-profiles/court-justice-european-union-cjeu_en

And : <https://curia.europa.eu/juris/recherche.jsf?language=en#>

Inspect the structure of the website (div and element)

```
In [6]: webpage = requests.get("https://european-union.europa.eu/institutions-la
...: w-budget/institutions-and-bodies/institutions-and-bodies-profiles/court-
...: justice-european-union-cjeu_en")
```

```
[10]: webpage2 = requests.get("https://curia.europa.eu/juris/recherche.jsf?language=en#n")
```



InfoCuria Case-law

English (en) ▾

Home > Search form

last update 25/03/2023

LIST OF RESULTS ?

Display preferences

☒ Automatic

☐ List of cases

☐ List of documents

Listing preferences

☒ Case numbers in descending order

☐ Case numbers in ascending order

☐ Dates in descending order

☐ Dates in ascending order

GENERAL INFORMATION

Help

Disclaimer

LINKS

Curia

Diary

EUR-Lex

Case status	? <input checked="" type="radio"/> All cases <input type="radio"/> Cases closed <input type="radio"/> Cases pending
Court	? <input checked="" type="checkbox"/> All <input checked="" type="checkbox"/> Court of Justice <input checked="" type="checkbox"/> General Court <input checked="" type="checkbox"/> Civil Service Tribunal
Case number	? <input type="text"/> (ex: C-17/05; 17/05)
Name of the parties	? <input type="text"/>
Documents	? All <input type="button" value="..."/>
Provisions of national law referred to	? <input type="text"/>
Provisions of international law referred to	? <input type="text"/>
ECLI	? ECLI:EU:
Text	? <input type="text"/>
Period of	? <input type="text"/>

DevTools is now available in French!

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

Elements Console Sources Network Performance >> 1

<div id="procedureFilterPanel" style="display: none;">...</div>

<div id="jugeFilterPanel" style="display: none;">...</div>

<div id="avocatFilterPanel" style="display: none;">...</div>

<div id="langueFilterPanel" style="display: none;">...</div>

<div id="nationaliteFilterPanel" style="display: none;">...</div>

<div id="origineFilterPanel" style="display: none;">...</div>

<script type="text/javascript" src="https://curia.europa.eu/juris/js/origine.js">

</script>

<div id="planClassementFilterPanel" style="display: none;">...</div>

<div id="cookiesModal" class="cc-window" style="visibility: hidden; opacity: 0;">...

</div>

<script type="text/javascript">...</script>

</div>

<div id="jstree-marker" style="display: none;">...</div>

<div id="jstree-marker-line" style="display: none;">...</div>

<div id="vakata-contextmenu"></div>

</body>

</html>

... ale form#mainForm div#main div#content div.recherche table tbody tr td.rechercheChamps ...

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

}

div.recherche td {

vertical-align: top;

}

style_defau_v=2.3.0:486

Console What's New

top Filter Default levels 1 Issue: 1

B – Scraping

Import the libraries

```
from tqdm import tqdm # Pour afficher la barre de progression
import requests # Pour récupérer les pages web
from bs4 import BeautifulSoup # Pour parser les pages web (i.e. les analyser)
import re # Pour utiliser des expressions régulières (regex)
```

Text files and List

```
# Création d'une liste de noms de fichiers préexistants à vider
noms_fichiers = ['affaires_dataframe.txt', 'cadres_dataframe.txt', 'cadres_raccourcis.txt', 'dates_dataframe.txt',
                 'documents_dataframe.txt', 'matieres_final.txt', 'liens_dataframe.txt', 'matieres.txt', 'matieres_']

# Pour chaque nom de fichier dans la liste des noms de fichiers
for nom_fichier in noms_fichiers:
    with open(nom_fichier, 'r+') as fichier: # Ouverture du fichier en mode lecture et écriture
        fichier.truncate(0) # Troncature du fichier à 0 byte (i.e. suppression de son contenu)
```

List of the 9 internet pages

```
# Récupération des liens des pages web contenant les documents de la CJUE et stockage dans une liste
urls = [
```

Python

Empty list and progress bar

```
liens = [] # Liste vide pour stocker les liens des documents
```

```
barre_progression_urls = tqdm(
    total=len(urls), desc='Traitement des urls ', position=0, leave=True) # Barre de progression
```

Create the main loop

```
for url in urls: # Pour chaque URL dans la liste des URLs
    reponse = requests.get(url) # Récupération de la page web

    # Utilisation de BeautifulSoup pour récupérer le contenu de la page web en html
    soup = BeautifulSoup(reponse.content, 'html.parser')

    # Trouver tous les éléments td de la classe 'table_cell_nom_usuel' pour récupérer les noms des pays
    for td in soup.find_all('td', {'class': 'table_cell_nom_usuel'}):
        # Récupération du nom du pays
        pays = td.text.strip().split('/')[1].strip()

        # Suppression des parenthèses et du contenu entre parenthèses dans le nom du pays (ex: Royaume-Uni (Angleterre))
        pays = re.sub(' \([^)]+\)', '', pays)

        # Écriture du nom du pays dans le fichier pays_dataframe.txt
        # Ouverture du fichier pays_dataframe.txt en mode écriture et encodage en utf-8 pour les caractères spéciaux
        with open('pays_dataframe.txt', 'a', encoding='utf-8') as f:
            # Écriture du nom du pays dans le fichier pays_dataframe.txt sans le dernier caractère (espace)
            f.write(pays + '\n')
```


Close the 1st process bar and open a 2nd one

```
barre_progression_urls.close() # Fermer la barre de progression des liens eur-lex  
  
print() # Saut de ligne entre les barres de progression  
  
barre_progression_documents = tqdm(  
|     total=len(liens), desc='Traitement des documents ', position=0, leave=True)
```

Find all the key words

```
# Parcourir tous les liens de la liste des liens eur-lex pour récupérer les mots clés des documents
for lien in liens:
    response = requests.get(lien) # Récupération de la page du document
    soup = BeautifulSoup(response.content, 'html.parser',
                          from_encoding='utf-8') # Récupération du contenu de la page du document

    # Ouverture du fichier mots_cles.txt en mode lecture
    with open('mots_cles.txt', 'r', encoding='utf-8') as f:
        b = True # Variable booléenne pour vérifier si le mot clé est trouvé ou non

        # Parcourir tous les mots clés du fichier mots_cles.txt
        for ligne in f:
            # Vérifier si le mot clé est trouvé dans le contenu du document
            if ligne.strip().lower() in str(soup).lower():
                # Écriture du mot clé dans le fichier cadres_dataframe.txt
                with open('cadres_dataframe.txt', 'a', encoding='utf-8') as f:
                    # Écriture du mot clé dans le fichier cadres_dataframe.txt
                    f.write(ligne.strip().replace('.', '') + ';')
                    b = False # Modifier la valeur de la variable booléenne
            # Écriture de 'None' dans le fichier cadres_dataframe.txt si le mot clé n'est pas trouvé
            if b:
                # Ouverture du fichier cadres_dataframe.txt en mode écriture
                with open('cadres_dataframe.txt', 'a') as f:
                    # Écriture de 'None' dans le fichier cadres_dataframe.txt
                    f.write('None;')
```

Close the process bar

```
# Écriture d'un saut de ligne dans le fichier cadres_dataframe.txt
with open('cadres_dataframe.txt', 'a') as f:
    # Écriture d'un saut de ligne dans le fichier cadres_dataframe.txt
    f.write('\n')

# Incrémenter la barre de progression
barre_progression_documents.update(1)

barre_progression_documents.close() # Fermer la barre de progression
```

Make the research easier by replacing some words or sentences

```
# Remplacer la phrase 'Union douanière' par 'Douane'
contenu = contenu.replace('Union douanière', 'Douane')

# Remplacer la phrase 'Tarif douanier commun' par 'Douane'
contenu = contenu.replace('Tarif douanier commun', 'Douane')

# Remplacer la phrase 'Importation' par 'Douane'
contenu = contenu.replace('Importation', 'Douane')

# Remplacer la phrase 'Exportation' par 'Douane'
contenu = contenu.replace('Exportation', 'Douane')

# Remplacer la phrase 'Impositions intérieures' par 'Douane'
contenu = contenu.replace('Impositions intérieures', 'Douane')
```

Apply the string search algorithm to obtain more details by categories

Prepare the data to create the dataframe afterwards

```
# Ouverture des fichiers matieres_raccourcis.txt et cadres_raccourcis.txt en mode lecture
with open('matieres_raccourcis.txt', 'r', encoding='utf-8') as matieres, open('cadres_raccourcis.txt', 'r', encoding='utf-8') as cadres:
    # Ouverture du fichier matieres_final.txt en mode écriture
    with open('matieres_final.txt', 'w', encoding='utf-8') as final:
        # Parcours des lignes du fichier matieres.txt et cadres_raccourcis.txt
        for ligne_matieres, ligne_cadres in zip(matieres, cadres):
            # Si la ligne contient '-Fiscalité', on prend la ligne correspondante dans cadres_raccourcis.txt
            if '-Fiscalité' in ligne_matieres:
                final.write('-' + ligne_cadres)
            else:
                # Sinon, on copie la ligne du fichier matieres.txt dans matieres_final.txt
                final.write(ligne_matieres)
```

C – Dataframe

- Data frame file
- Retrieve the information
- 2 lists

```
# Création du dataframe
# Création d'une liste contenant la liste des fichiers à lire
fichiers = ['affaires_dataframe.txt', 'dates_dataframe.txt',
            'documents_dataframe.txt', 'liens_dataframe.txt', 'pays_dataframe.txt']

dfs = [] # Liste des dataframes

# Parcours des fichiers
for fichier in fichiers:
    # Création du dataframe à partir du fichier et ajout dans la liste des dataframes
    df = pd.read_csv(fichier, sep=' ', names=[
                    fichier[:-14].capitalize()], encoding='utf-8')
    # Réinitialiser l'index du dataframe
    df.reset_index(drop=True, inplace=True)
    dfs.append(df) # Ajout du dataframe dans la liste des dataframes
```

Columns to create a table

```
colonnes = ['TVA', 'Droits d\'accises', 'Taxation des produits énergétiques et de l\'électricité', 'Douane',  
            'Droits de succession', 'Fusion', 'Taxe sur les véhicules', 'Libre circulation des capitaux',  
            'Principes, objectifs et missions des Traités', 'Boissons spiritueuses', 'Libre circulation des marchan  
            'Restrictions quantitatives', 'Mesures d\'effet équivalent', 'Liberté d\'établissement', 'Plus-values',  
            'Coopération administrative', 'Tabacs manufacturés', 'Impôts indirects',  
            'Citoyenneté de l\'Union', 'Libre circulation des travailleurs', 'Relations extérieures',  
            'Association européenne de libre-échange', 'Taxe d\'immatriculation', 'Rapprochement des législations',  
            'Adhésion', 'Dispositions financières', 'Ressources propres', 'Transports', 'Agriculture et Pêche',  
            'Environnement', 'Départements français d\'outre-mer', 'Denrées alimentaires',  
            'Monopoles d\'État à caractère commercial', 'Concurrence', 'Aides accordées par les États']
```

```
for colonne in colonnes: # Parcours des colonnes  
    # Création d'une nouvelle colonne dans le dataframe  
    # Initialisation de la colonne à 0  
    df[colonne] = [0 for i in range(len(df))]
```

```
df = pd.concat(dfs, axis=1) # Concaténation des dataframes
```

- Filling in the values
- A final processing on the line

```
with open('matieres_dataframe.txt', 'r', encoding='utf-8') as fichier:
    lignes = fichier.readlines() # Lecture des lignes du fichier

for i in range(len(df)):
    # Remplacement des doublons par des mots uniques dans la ligne i du fichier
    lignes[i].replace('Alcool', 'Boissons spiritueuses') \
        .replace('Tabac', 'Tabacs manufacturés') \
        .replace('Taxes sur l\'électricité', 'Taxation des produits énergétiques et de l\'électricité') \
        .replace('Taxes d\'effet équivalent', 'Mesures d\'effet équivalent') \
        .replace('Tarif douanier commun', 'Union douanière') \
        .replace('Sucre', 'Denrées alimentaires') \
        .replace('Vin', 'Boissons spiritueuses') \
        .replace('Fruits et légumes', 'Denrées alimentaires').replace('Douane-Douane', 'Douane').replace('Douane;Do

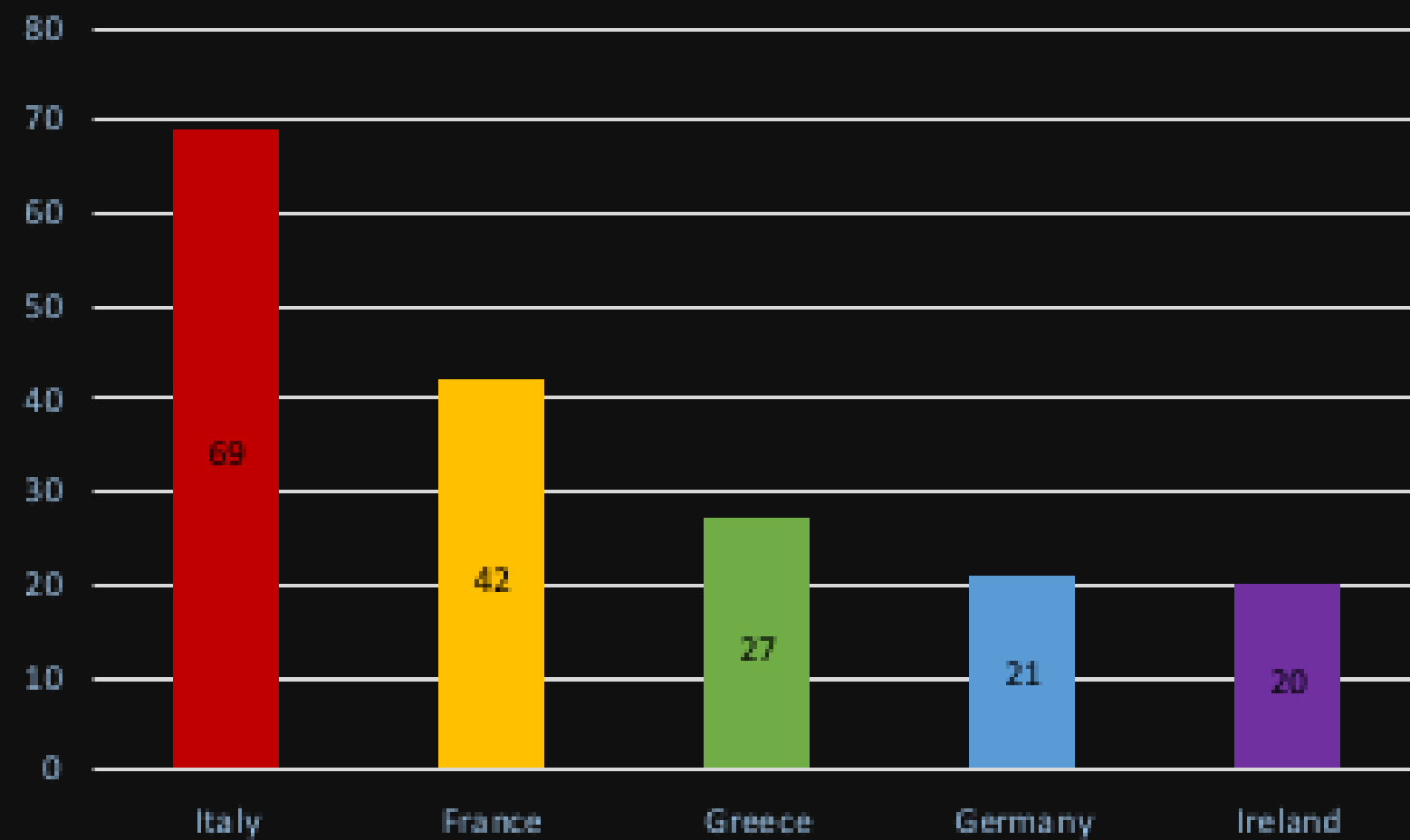
# Boucle sur chaque colonne du dataframe
for colonne in colonnes:
    # Si la colonne est présente dans la ligne i du fichier, on passe la valeur à 1
    if colonne in lignes[i]:
        df.loc[i, colonne] = 1
```

D – Visualizer

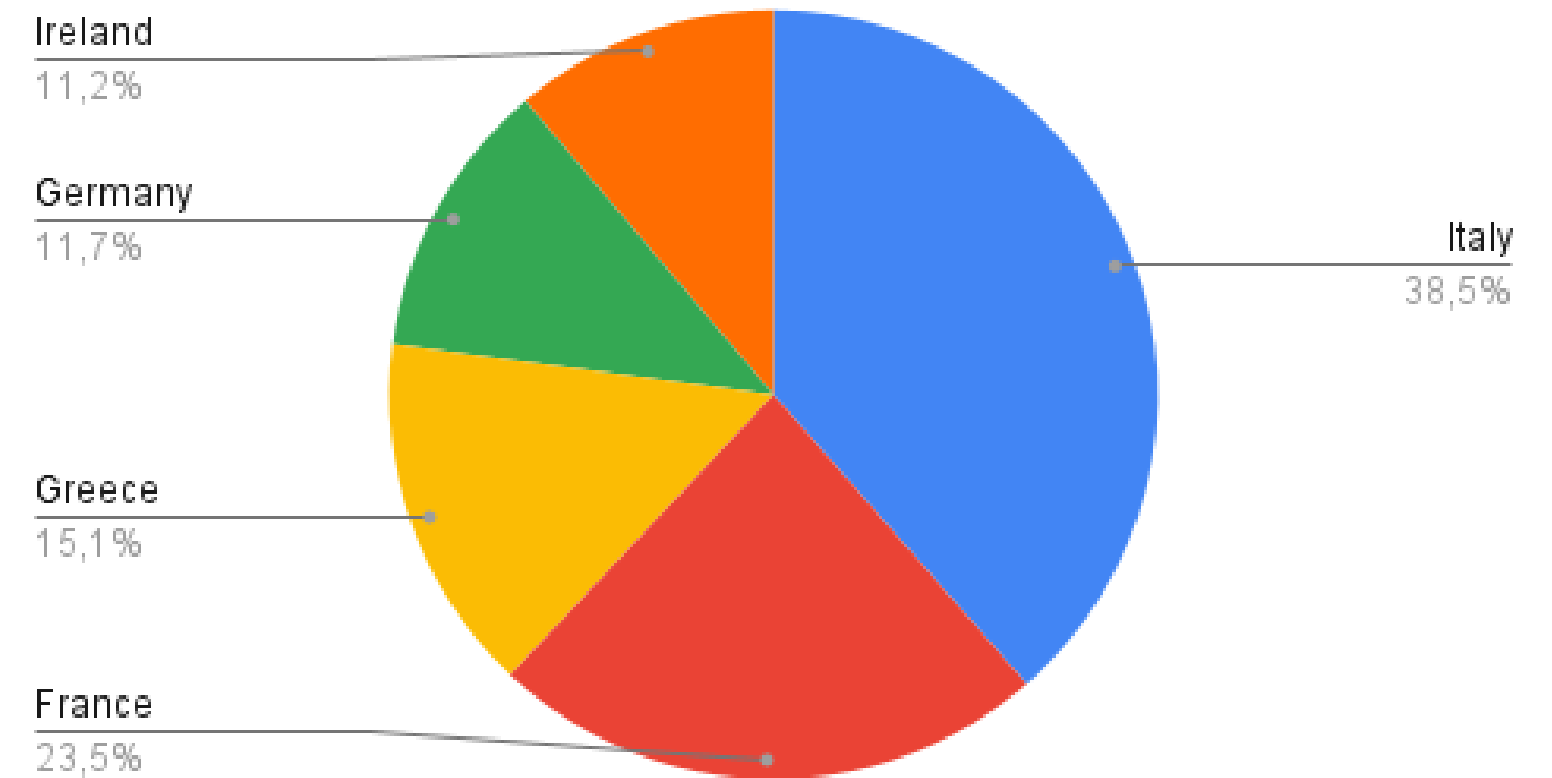
Pays ▾	TVA ▾	Droits d'accises ▾	Taxation des produits énergétiques	Douane ▾	Droits de succe	Fusion ▾
Portugal	0	0	0	1	0	0
Finlande	0	1	0	1	0	0
Finlande	0	1	0	1	0	0
Italie	0	1	1	1	0	0
Autriche	1	0	0	0	0	0
Allemagne	1	0	0	0	0	0
Royaume-Uni	1	0	0	0	0	0
Grèce	0	1	0	0	0	0
Autriche	1	0	0	0	0	0
Grèce	0	1	1	1	0	1
Allemagne	1	0	0	0	0	0
Grèce	0	0	0	0	1	0
Allemagne	1	0	0	0	0	0
Luxembourg	1	0	0	0	0	0
Portugal	1	1	0	1	0	0
Pays-Bas	1	0	0	0	0	0
Suède	1	0	0	0	0	0
Royaume-Uni	0	0	0	0	0	0
Pologne	1	0	0	0	0	0
Royaume-Uni	0	0	0	0	0	0

2 – THE ANALYSIS

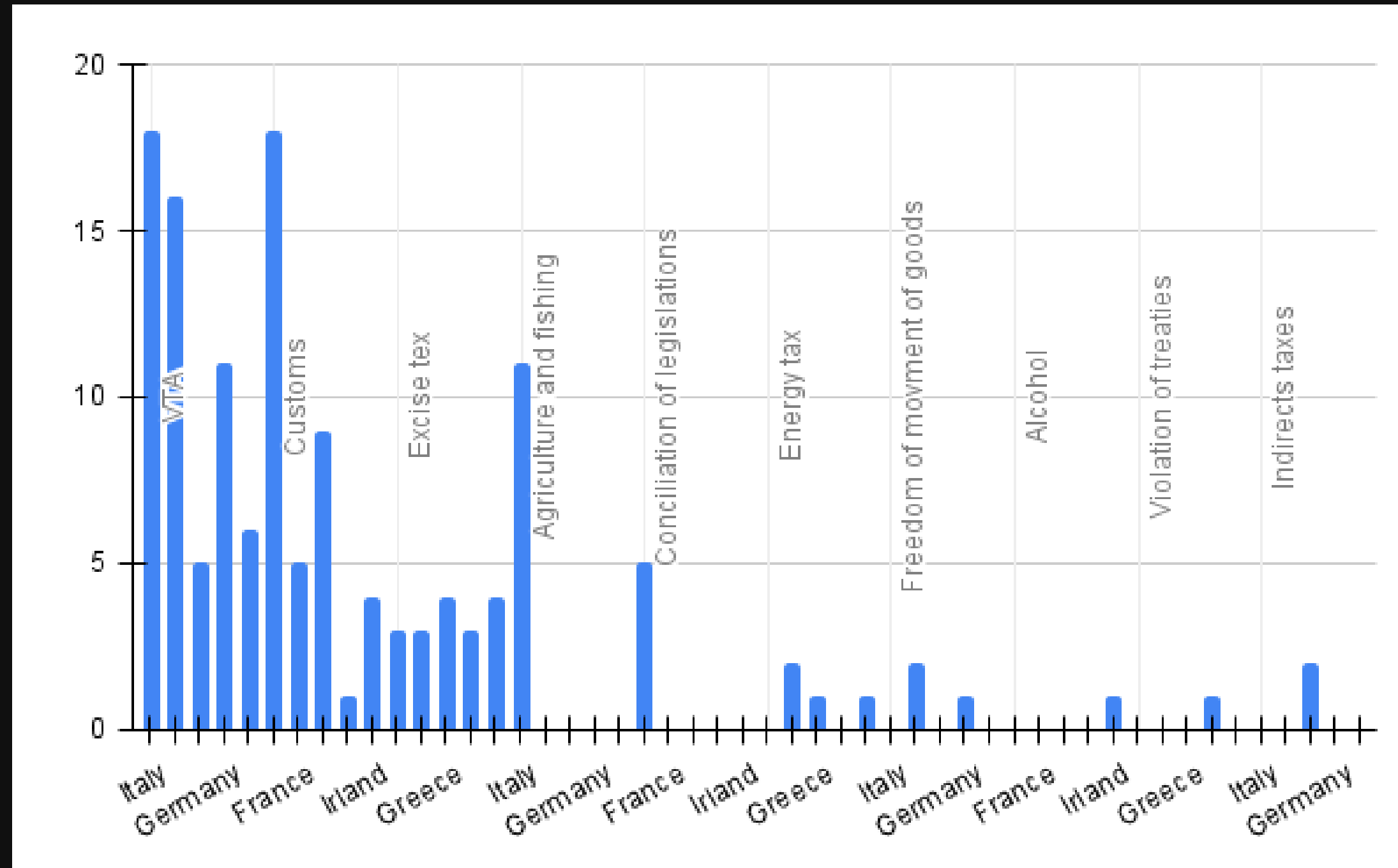
A – Top 5 of the worst performers



Number of convictions



B – The contentious matters



**THANK YOU FOR YOUR
ATTENTION!**

Les Pythons de la Fournaise