



European Court of Human Rights case law analysis

Garance ANJUBAULT, Maxime DESPRES, Lisa KHAZAR, Jean-Baptiste PORTIER

TABLE OF CONTENTS

01

Introduction

02

Scraping Method

03

Results

04

Analysis



01

Introduction



Analysis of the case law of the
European Court of Human Rights over
time and according to the States
involved and the rights violated

Approx. 25,000 decisions



02

Scraping method

FOUR STEPS IN CODING

- 1) Get the cases filtered by languages
- 2) Get the page to scroll down
- 3) Check if the same case has already been processed
- 4) Extract the relevant information from each case



Step 1: Get the cases filtered by languages

How to get all the decisions without fail?

⇒ Subdivisions by loops

⇒ Potential criteria: dates or languages?

EXTRACT OF THE CODE

```
LANGUAGES_TO_PROCESS = ["english", "french", "not english french"]
```

CLICK ON THE RIGHT LANGUAGES

FILTERS

LANGUAGE ?	IMPORTANCE ?
French (25939)	3 (46119)
English (25938)	2 (18875)
Turkish (4083)	Key cases (9754)
Russian (2987)	1 (5473)
More...	More...

STATE ?	ARTICLE ?
Türkiye (12176)	6 (39620)
Russia (8883)	41 (34956)
Italy (6316)	6-1 (34767)
Ukraine (5353)	35 (15049)
More...	More...

NON-VIOLATION ?	VIOLATION ?
6 (4854)	6 (32455)
6-1 (4129)	6-1 (27909)
3 (3234)	3 (11001)
8 (2975)	5 (9859)
More...	More...

Violation of Article 11 - Freedom of assembly and association (Article 11-1 - Freedom of assembly and association) (Article 11-1 - Freedom of assembly and association)

[Case Details](#) [Press Release](#) [Related](#)

new CASE OF ROGACHEV AND OTHERS v. RUSSIA

64754/14 54040/17 79261/17... [17 more...](#) | Available only in [English](#) | Judgment (Merits and Just Satisfaction) | 23/03/2023

Violation of Article 11 - Freedom of assembly and association (Article 11-1 - Freedom of assembly and association)

[Case Details](#) [Press Release](#) [Related](#)

new CASE OF ABISHOV v. AZERBAIJAN

46419/16 | Available only in [English](#) | Judgment (Merits and Just Satisfaction)

Violation of Article 3 - Prohibition of torture (Article 3 - Degrading treatment Inhuman and Degrading Treatment)

[Case Details](#) [Press Release](#) [Related](#)

new CASE OF BAGAUTDINOV AND OTHERS v. RUSSIA

48041/16 67362/17 69436/17... [15 more...](#) | Available only in [English](#) | Judgment (Merits and Just Satisfaction) | 23/03/2023

Violation of Article 3 - Prohibition of torture (Article 3 - Degrading treatment) (Su

Three situations covered:

⇒ decisions in English;

⇒ decisions in French;

⇒ decisions neither in English nor in French.

Sort by: [Alphabetical Order \(A-Z\)](#)

☒ OF ☐ NOT

- ☐ Albanian (799)
- ☐ Arabic (98)
- ☐ Armenian (846)
- ☐ Azerbaijani (757)
- ☐ Belarusian (1)
- ☐ Bosnian (277)
- ☐ Bulgarian (856)
- ☐ Croatian (805)
- ☐ Czech (1200)
- ☐ Dutch (57)
- ☒ English (25938)
- ☒ French (25939)
- ☐ Georgian (1053)
- ☐ German (1472)
- ☐ Greek (875)
- ☐ Hungarian (491)
- ☐ Icelandic (858)
- ☐ Italian (596)
- ☐ Korean (20)
- ☐ Latvian (17)
- ☐ Lithuanian (187)
- ☐ Macedonian (816)
- ☐ Montenegrin (76)
- ☐ Persian (1)
- ☐ Polish (964)
- ☐ Portuguese (152)
- ☐ Romanian (2488)
- ☐ Russian (2987)
- ☐ Serbian (513)
- ☐ Slovak (85)

[OK](#) [CLOSE](#)

CLICK ON THE RIGHT LANGUAGES

```
# Retrieve the more button in FILTERS/LANGUAGE section
button = driver.find_element(By.XPATH, "//*[@id='refinement-filters-row1-left']")
button = button.find_element(By.XPATH, ".//button[@class='morebutton']")
# Scroll until the button is in the view to be able to click on it
button.location_once_scrolled_into_view
button.click()
time.sleep(2)

languages = languageToProcess.split()
for language in languages:
    if language.lower() == "not":
        # We want to ignore the given languages
        button = driver.find_element(By.XPATH, "//*[@id='radio_choices_NOT']")
        button.click()
    elif language.lower() == "english":
        # Click on the given language
        button = driver.find_element(By.XPATH, "//*[@id='ENG']/h4")
        button.location_once_scrolled_into_view
        button.click()
    elif language.lower() == "french":
        button = driver.find_element(By.XPATH, "//*[@id='FRE']/h4")
        button.location_once_scrolled_into_view
        button.click()
```

Step 2: Get the page to scroll down

How many **scrolls** do we need to get all the decisions displayed?

⇒ Define the **number of cases to get**:

```
nb_cases_to_get = int(soup.find("span", class_="resultNumber").contents[0])
```

⇒ Define the **number of cases displayed per scroll**:

```
nb_cases_per_scroll =  
len(soup.find("div", class_="results-list-block").findChildren("div", title="Results"))
```

⇒ Define the **number of scroll needed**:

```
for x in range(int(nb_cases_to_get/nb_cases_per_scroll)+10): # add a small margin  
    driver.find_element(By.CSS_SELECTOR, "body").send_keys(Keys.CONTROL, Keys.END)  
    time.sleep(0.5)
```

Step 3: Check if the same case has already been processed



- 1) Initialize a list: `caseids.list`
- 2) Add each case id in the list if it is not already present and process the case
- 3) Skip the entry if the case has already been processed

EXTRACT OF THE CODE

```
if caseid in caseids_list:  
    continue # Skip this entry, case already processed
```



03

Results

Step 4: Extract the relevant information from each case

How to extract the useful information?

⇒ Find where the information is available

The image shows a screenshot of the HUDOC website (Cour européenne des droits de l'homme) and its DevTools interface. The website displays search results for 'AFFAIRE ROGALSKI c. POLOGNE' and 'AFFAIRE UDOVYCHENKO c. UKRAINE'. The DevTools interface shows the HTML structure of the page, with specific elements highlighted in red boxes. The first red box highlights the date column: `23/03/2023`. The second red box highlights the full text of the case: ` " Violation de l'article 10 - Liberté d'expression-général) (Article 10-1 - Liberté d'expression) Domage matériel et préjudice moral - réparation (Article 41 - Préjudice moral Domage matériel Satisfaction équitable) "`.

⇒ Extract the States:

```
states = lineone.find("a", class_="document-link headline").get_text()
```

```
x = re.search(r" [vc]\. ", states) ➡ Following "v." or "c."
```

```
states = re.findall(r"[\w\s]+", states[x.end():]) ➡ Starting from the end, we select the  
State's name which are words made  
up of letters and spaces
```

⇒ Extract the dates:

```
date = linetwo.find("span", class_="column dateColumn").get_text()
```

⇒ Extract the articles violated and not violated:

Cleaning the conclusions

- 1) Clearly distinguish articles that were NOT violated from those who effectively were

```
articles = re.sub("No violation Of |Non-violation de l'", "ViolationNot",  
conclusions, 0, re.I)  
articles = re.sub("violation Of | violation de l'", "Violation", articles, 0, re.I)
```

- 2) Eliminate abbreviations

```
articles = re.sub("art\\.", "Article", articles, 0, re.I)
```

Extract the articles' numbers

```
articles_not_violated = re.findall("(?<=ViolationNot)Article \d+ du Protocole n°  
\s?\d+|(?<=ViolationNot)Article \d+ of Protocol  
No.\s?\d+|(?<=ViolationNot)Article [P?\d+~?\d*\d+]\"", articles, re.I)  
articles_violated = re.findall("(?<=Violation)Article \d+ du Protocole n°  
\s?\d+|(?<=Violation)Article \d+ of Protocol No.\s?\d+|(?<=Violation)Article  
[P?\d+~?\d*\d+]\"", articles, re.I)
```

NB:

- Articles can come from the European Convention on Human Rights but also from its protocols
- Articles' paragraphs can also be mentioned using an hyphen
- Juxtaposition of articles can be phrased differently (for example: Article P1-1-1+6-1+11)

Summarize into a dataframe

6 separate columns: CaseIds, State, Date, Full conclusions, Articles identified as violated and Articles identified as NOT violated

```
df = pd.DataFrame(data_scrapped, columns=["Ids", "State", "Date",  
    "Conclusions", "Articles violated", "Articles not violated"])
```

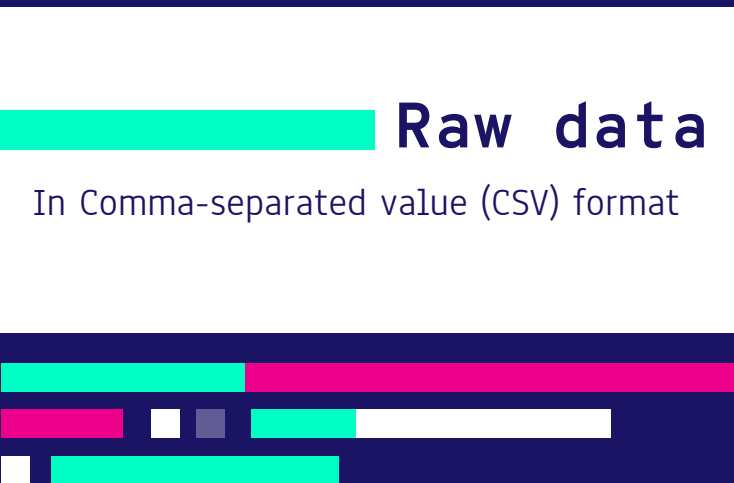
⇒ Use the extracted information summarized in the excel file to start our analysis



04

Analysis

Extracting insights from data



Assessment

Flaws inherent to a scraped dataset

Cleanup

Applying filters and “manual” intervention

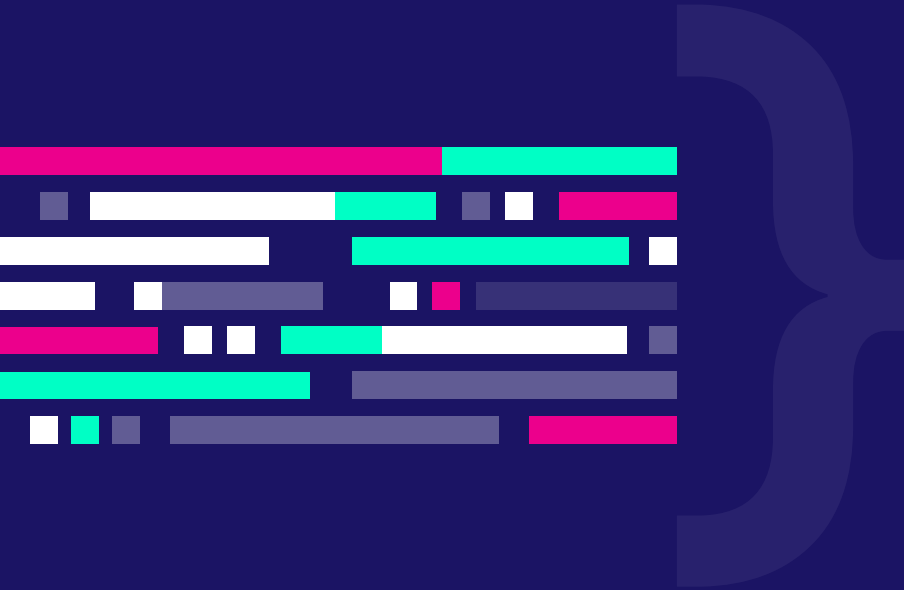
Processing

From numeric values to charts

Interpretation

Insights emerging from visual analysis

Step 1: Assessing the scraped data



- 1) Visualizing the CSV file with appropriate software (Sheets, Excel)
- 2) Comparing the dataset to available statistics and orders of magnitude
- 3) Estimating the necessary resources to process the data and adjusting scope

Step 2: Cleaning up the dataset

States

- Merging duplicates
- Removing outliers

ECHR Violations

- Fixing inconsistencies
- Unifying references

Fixing inconsistencies in States names

```
df['State'] = df['State'].replace('THE REPUBLIC OF MOLDOVA', 'MOLDOVA')  
df['State'] = df['State'].replace('TÜRKİYE', 'TURKEY')  
df['State'] = df['State'].replace('TÜRKİYE', 'TURKEY')  
df['State'] = df['State'].replace('TURQUIE', 'TURKEY')  
df['State'] = df['State'].replace('LUXEMBURG', 'LUXEMBOURG')  
df['State'] = df['State'].replace('ITALIE', 'ITALY')
```

Manual intervention where losing data would substantially impact our findings

Filtering out low-impact outliers for an easier analysis

```
result = result.sort_values(by=['Violations'], ascending=False)  
result = result[result['Total_mentions'] > 9]
```

Fixing inconsistencies in ECHR Articles notation

Any ECHR case with multiple articles breached required splitting for proper analysis

```
# determine the maximum number of elements in the 'Articles violated' column
max_articles_violated = max(df['Articles violated'].str.split(',').apply(len))
new_cols = ['Article ' + str(i+1) for i in range(max_articles_violated)]
new_df = pd.DataFrame(columns=['Nb', 'Ids', 'State', 'Date', 'Conclusions'] + new_cols)

# iterate over each row in the original dataframe
for i, row in df.iterrows():
    articles_violated = row['Articles violated'].strip('')[1:].split(', ')
    new_row = {
        'Nb': row['Nb'],
        'Ids': row['Ids'],
        'State': row['State'],
        'Date': row['Date'],
        'Conclusions': row['Conclusions']
    }
    for j, article in enumerate(articles_violated):
        new_row[new_cols[j]] = article

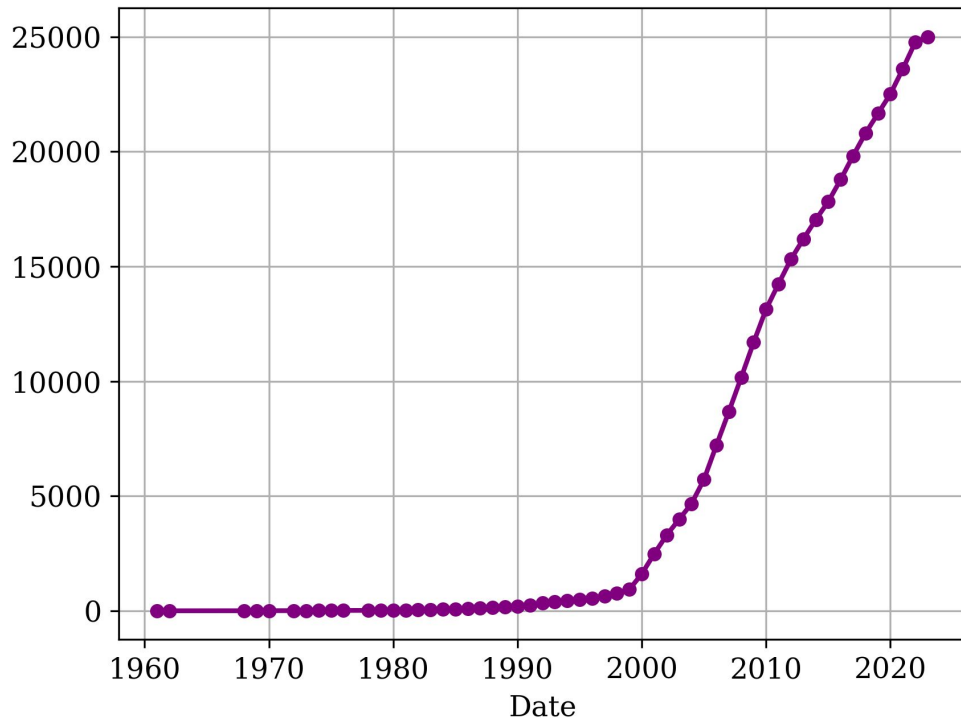
    # append the new row to the new dataframe
    new_df = new_df.append(new_row, ignore_index=True)

# output the new dataframe to a CSV file
new_df.to_csv('echr_articles_violated.csv', index=False)
```

[illegible]

ECHR activity from 1959 to 2023

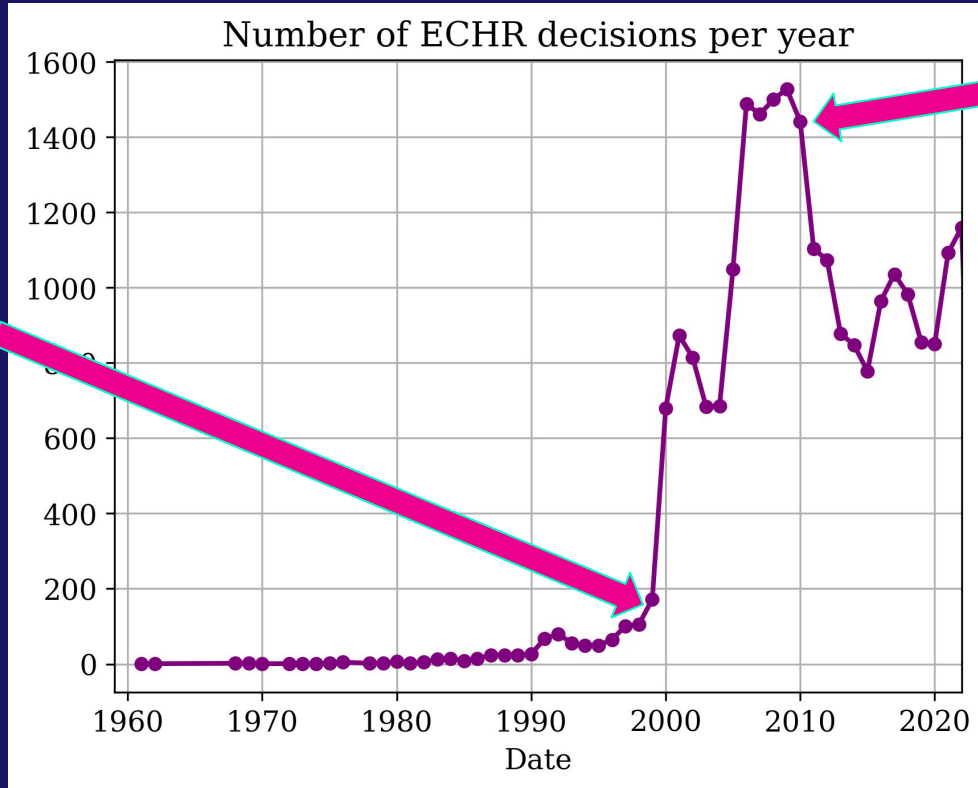
Cumulative number of ECHR decisions



```
df.groupby(df['Date'].dt.year).count()['State'].cumsum().plot(line_color='purple', marker='o', markersize=5, linewidth=2)
plt.title('Cumulative number of ECHR decisions')
plt.grid(True)
plt.savefig('echr_totalcases.png', dpi=300, bbox_inches='tight')
```

Trends presented in a straightforward manner may lack granularity

ECHR activity from 1959 to 2023



2010 : Protocol no. 14 establishes a new admissibility criterion before the Court. It requires the existence of a "significant disadvantage" for the applicant.

1998 : Protocol no. 11 replaced the old system with a full-time court. It enabled applicants to bring their cases directly before the Court, with no prior filtering.

Ranking countries by number of violations

POLAND	1234	1028	189
GREECE	1038	901	129
FRANCE	1056	733	272
BULGARIA	781	675	136
HUNGARY	640	605	31
MOLDOVA	503	458	25
CROATIA	498	398	96
SLOVAKIA	415	368	32
SLOVENIA	364	328	33
THE UNITED	528	296	210
AUSTRIA	396	267	102
AZERBAIJAN	282	266	24
PORTUGAL	360	253	35
SERBIA	235	210	30
BELGIUM	270	189	78
GERMANY	338	183	148
THE CZECH R	222	173	40
ARMENIA	175	161	21
LITHUANIA	240	154	86
SPAIN	198	138	63
FINLAND	183	133	57
THE FORMER	149	128	20
SWITZERLAN	211	123	93
LATVIA	154	119	53
GEORGIA	143	116	50
BOSNIA AND	113	108	11
MALTA	126	93	30
THE NETHER	162	87	59

Creating simplified
CSV files for easier
chart generation

Ranking countries by number of violations

```
# Read the original CSV file into a pandas dataframe
df = pd.read_csv('echr_clean_data.csv')

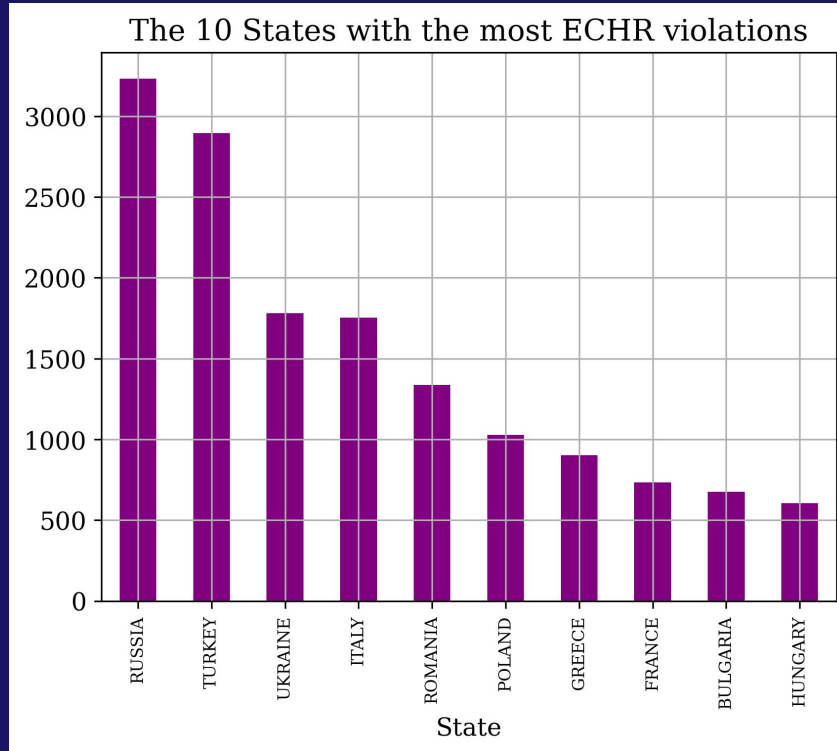
# We set the color palette, font size and style
plt.rcParams['axes.prop_cycle'] = plt.cycler(color=plt.cm.viridis(np.linspace(0, 1, 10)))
plt.rcParams.update({'font.size': 12})
plt.rcParams['font.family'] = 'serif'
```

```
# We create a bar chart that shows the 10 states with the most violations
```

```
df = df.sort_values(by=['Violations'], ascending=False)
df = df.head(10)
df.plot.bar(x='State', y='Violations', color='purple')

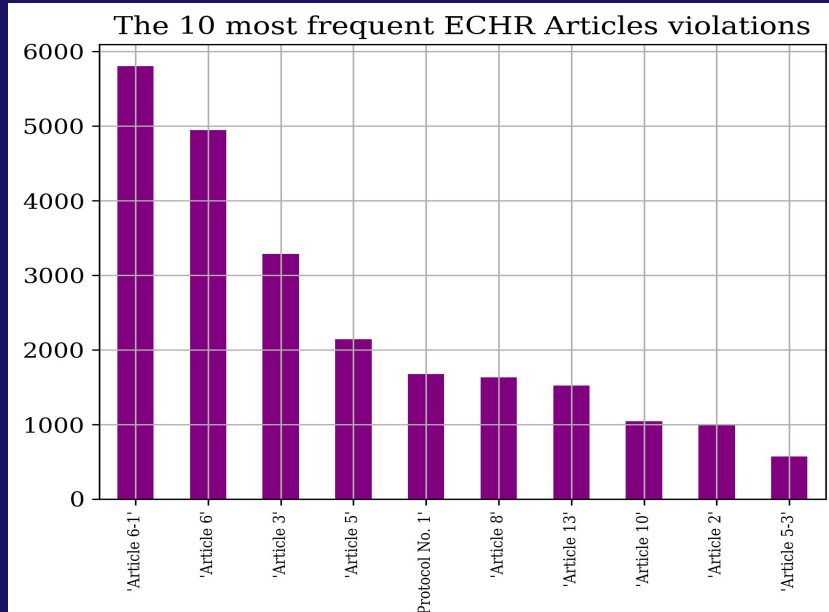
plt.legend().set_visible(False)
plt.xticks(fontsize=8)
plt.title('The 10 States with the most ECHR violations')
plt.grid(True)
plt.savefig('echr_violations_static.png', dpi=300, bbox_inches='tight')
```

Ranking countries by number of violations



Note : The Russian Federation ceased to be a party to the ECHR on 16 September 2022.

A Ranking of ECHR Articles Based on Breach Frequency




6-1 | Right to a fair trial (6-1 | Within a reasonable time)

3 | Prohibition of torture

5 | Right to liberty and security

8 | Respect for private and family life

13 | Right to an effective remedy



Thank you for your attention
Any question?