SOULIER Marion, BORIE Emma, VALLI Pauline

Analysis of bilateral tax treaties

Taxation of interest

01 The project

Context and issues

02 A multi-stage code

Analysis of the code and difficulties encountered

03 The result

Analysis of the result, concrete consequences for companies

01 **The project** – Background and issues

121 bilateral tax treaties signed by France

Tax treaties help **prevent double taxation and tax evasion.** They promote **cooperation** between France and other international tax authorities in applying their respective tax laws.

Distribution of taxing power

Tax treaties **cover cross-border flows** and **allow taxation to be attributed** to one or other of the signatory states, or to both.

Risk of double taxation

Elimination of double taxation

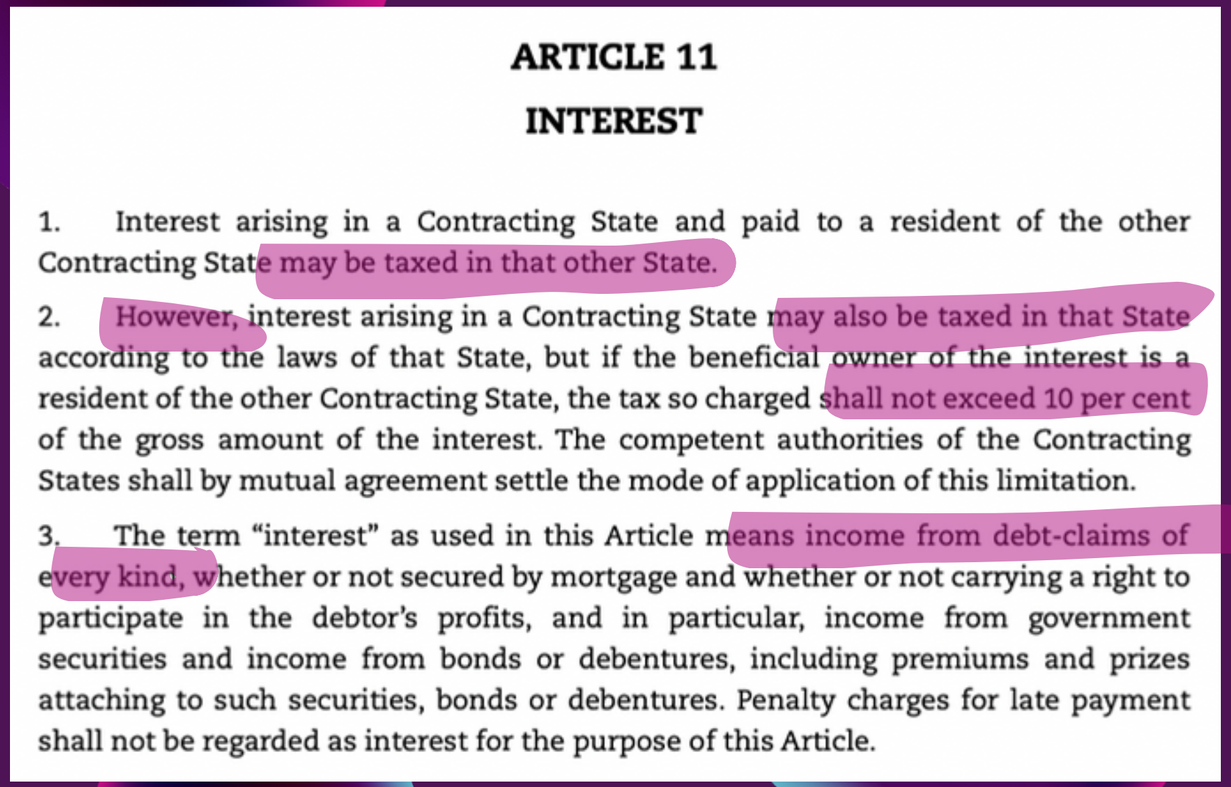
Two main methods of **eliminating double taxation :**

1° **The exemption method**
2° **The imputation method** : in this case both States tax, but one of them agrees downstream to neutralise all or part of the tax that it would be entitled to levy by granting a tax credit.

Taxation of interest

- Interest is income from **all types of debt** (often loans).
- It corresponds to income from "**fixed-income investments**", which distinguishes them from "risky" investments such as shares, whose income is uncertain because it depends on the results of the issuing company.
- These "fixed-income" investments actually include several types: **loans; bonds; government bonds and treasury bills; deposits, etc.**

Taxation of interest



ARTICLE 11

INTEREST

1. Interest arising in a Contracting State and paid to a resident of the other Contracting State may be taxed in that other State.

2. However, interest arising in a Contracting State may also be taxed in that State according to the laws of that State, but if the beneficial owner of the interest is a resident of the other Contracting State, the tax so charged shall not exceed 10 per cent of the gross amount of the interest. The competent authorities of the Contracting States shall by mutual agreement settle the mode of application of this limitation.

3. The term "interest" as used in this Article means income from debt-claims of every kind, whether or not secured by mortgage and whether or not carrying a right to participate in the debtor's profits, and in particular, income from government securities and income from bonds or debentures, including premiums and prizes attaching to such securities, bonds or debentures. Penalty charges for late payment shall not be regarded as interest for the purpose of this Article.

The OECD model convention advocates **shared taxation** of cross-border interest flows. This implies :

-Taxation in the **State of residence**

-Taxation by the **source State** by means of a **withholding tax** at a **rate not exceeding 10%**

However, not all tax treaties follow this model, hence the interest of our project: **to determine for each tax treaty whether taxation is exclusive or shared, and if so what the maximum rate of withholding tax is.**

Are bilateral tax treaties consistent as regards the taxation of interest ?

If there is a multitude of systems, is it nevertheless possible to identify a predominant tax system ?

02 A multi-stage code

1) The Data Collection

- **Web scrapping** to extract the treaties
- Extract **the right article** in each treaty
- Extract **the right information in the relevant article**

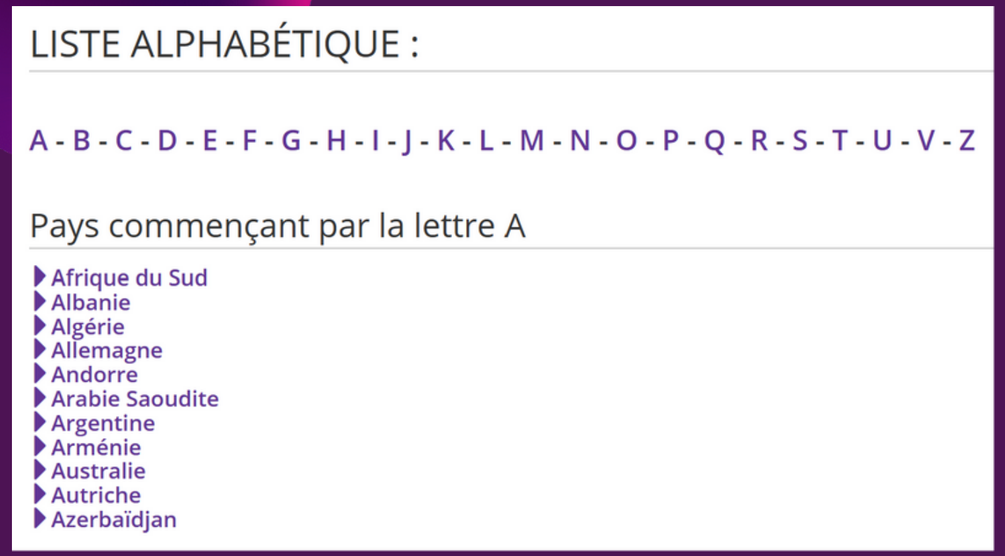
2) The Dataframe formulation

- Creation of an object to **stock all the data**
- **Display collected data**

3) The Data interpretation

- Create **charts to analyze** the Data

1) The Data Collection



All the treaties are in **pdf format** on the website **impot.gouv** so we had to scrap to access to theses treaties.

For this purpose : creation of a **“soup”** that target files in pdf format

```
webpage = requests.get("https://www.impots.gouv.fr/les-conventions-internationales")


if webpage.status_code == 200:
    soup = BeautifulSoup(webpage.text, 'html.parser')
    url_convention = soup.find("article").find_all("a", href=re.compile("\.pdf"))
```

1) The Data Collection

```
# Créer un répertoire pour les fichiers PDF téléchargés
pdf_directory = os.path.join(os.getcwd(), 'conventions_pdf')
if not os.path.exists(pdf_directory):
    os.makedirs(pdf_directory)


# Initialiser une liste pour stocker les données de chaque convention
convention_info_list = []
```

- **Create a folder** on the computer so that all conventions can be **saved** in this folder for easy access and to check that the right conventions are being saved

- Create a **list to store the data** to be extracted from each convention

⚠ **Difficulty alert :** This step was essential, as the site contained a large number of conventions, but we were **only interested in those relating to income tax**. We therefore had to find criteria to select only these, but to check that the criteria were working properly, we had to go and check that the right conventions were being registered.

117 treaties extracted

1) The Data Collection

```python
# Fonction pour extraire le texte d'un fichier PDF
def extract_text_from_pdf(pdf_path):
    text = ""
    with open(pdf_path, 'rb') as file:
        pdf_reader = PdfReader(file)
        for page in pdf_reader.pages:
            text += page.extract_text()
    return text
pdf_text = extract_text_from_pdf(file_name)
```

- **Download** each convention in pdf format in the folder created above
- **Extract the text form the pdf**

⚠ **Difficulty Alert :** found criteria to download only conventions dealing with income and wealth tax, but there was no title template, each convention was listed under a different name.

```python
# Extraire le texte de chaque fichier PDF dans le répertoire spécifié
for x in url_convention:
    if any(word in x.text.lower() for word in ["consolidée", "avenant", "accord", "travailleurs"
, "donations", "successions", "non entrée en vigueur", "administrations", "mutation", "aéroport"
]):
        print("Filtré :", x.text)
    else:
        print(x)
        href = x['href']
        # Ajoutez "https://" avant l'URL relative
        # Utilisation de la fonction pour extraire le texte d'un fichier PDF depuis une URL
        url_complet = "https://www.impots.gouv.fr" + href
        file_name = os.path.join(pdf_directory, os.path.basename(href))
        if not os.path.exists(file_name):
            wget.download(url_complet, file_name)
            print("Fichier téléchargé :", file_name)
        else:
            print("Le fichier existe déjà :", file_name)

        pdf_text = extract_text_from_pdf(file_name)
```

1) The Data Collection

```
# Split the convention in as many articles as there are; for this,harmonise the text, and use a regex
 to split by ARTICLE
        spls = re.split("(\n|\. ?)ARTICLE (?=\d+)", pdf_text.upper(), re.S)
        spls = [x for x in spls if len(x) > 3]  # This line to remove all splits that are too short (
double spaces, for instance)

        # Initialize article variable here
        article = ""
        for spl in spls:  # We look for the article that is the most likely candidate, i.e., the one
article that
            # mentions "interest" in the first 100 characters (could be changed)
            if re.search("Int.r.s?ts", spl[:100], re.I | re.S):
                article = spl
                break
            elif re.search("revenus\s*de\s*cr[ée]ances", spl[:100], re.I | re.S):
                article = spl
                break
        if article == "": # If this did not work, we look for it in the next 500 characters
            for spl in spls:
                if re.search("Int.r.êts?", spl[:500], re.I | re.S):
                    article = spl
                    break
                elif re.search("revenus\s*de\s*cr[ée]ances", spl[:500], re.I | re.S):
                    article = spl
                    break
```

⚠ **Difficulty alert :** in some agreements, the **term "interest" was not used**, but "debt income", so this term had to be included in the search.

- **Split each part of the treaty by article** in order to **search** for the one dealing with **interests** and target it exclusively later.

⚠ **Difficulty alert : sometimes in the conventions there is a title to the article, sometimes** not. So we need to find out if the term "interest" appeared in the first 100 characters of each article, and if not, to make sure we didn't miss the right article, he searched the first 500 characters.

1) The Data Collection

```
if article:
            print(file_name, article[:100])  # Print the first 1000 characters of the article
            # Better to use regexes to check if some text is the text.
            if re.search('ne sont imposables que dans', article, re.I|re.S):
                tax_kind = 'exclusive'
            elif re.search('sont imposables dans.*sont aussi imposables dans|toutefois', article, re.I|re.S):
                tax_kind = 'partagee'
            else:
                tax_kind = 'indeterminee'
```

Find in the relevant article extracted from the convention **whether taxation is exclusive or shared**.

The same expression is used in each convention :

- when there is "ne sont imposés que dans", taxation is “**exclusive**”
- but when there is "sont imposables dans" or "sont aussi imposables" or "toutefois", taxation is “**partagée**”

And if nothing is found, for the code to continue, the agreement will be classified as "**indetérminée**"

1) The Data Collection

```python
tax_match = tax_regex.search(article)
if tax_match:
    tax_rate = float(tax_match.group(1))
else:
    # Si aucun taux d'imposition n'est trouvé, recherchez un nombre suivi de "%" ou de "p. cent" ou de “pour cent”

    percent_match = re.search(r'(\d+(?:\.\d+)?)\s*%|p\.?\s*cent|\bpour\s*cent\b', article)
    if percent_match:
        tax_rate = float(percent_match.group(1))
    else:
        tax_rate = None
```

- Find the **withholding tax rate** in the article.

⚠ **Difficulty alert :** depending on the convention, the **rate is indicated differently** : sometimes as "%", sometimes as "p. cent" and sometimes as "pour cent".
We had to **include these different possibilities** to ensure that no rate was overlooked.

2) The Data Formulation

```
    # Créer l'objet JSON
            convention_info = {
                'pays': os.path.basename(file_name)[:-4],
                'tax_kind': tax_kind,
                'tax_rate': tax_rate
            }
            print(json.dumps(convention_info, indent=4))
            convention_info_list.append(convention_info)



# Créer un DataFrame à partir de la liste de dictionnaires
df = pd.DataFrame(convention_info_list)



# Afficher le DataFrame
pd.set_option('display.max_rows', 150)
print(df)
```

- **Store each item of information** extracted : tax type, rate and treaty country in a **json object**
- For each convention analyzed, this json object is **added to a list**.
- Once all the treaties have been analyzed, the list is complete. To display the results in a structured way, we **created a dataframe** (with the pandas function) in column form.

2) The Data Formulation

	pays	tax_kind	tax_rate
0	afrique-du-sud_convention-avec-l-afrique-du-su...	exclusive	NaN
1	algerie_convention-avec-l-algerie_fd_1720	partagee	10.0
2	allemagne_convention-avec-l-allemagne-impots-s...	exclusive	NaN
3	andorre_convention-avec-la-principaute-d-andor...	partagee	5.0
4	arabie-saoudite_convention-avec-l-arabie-saoud...	exclusive	NaN
5	argentine_convention-avec-l-argentine_fd_1724	exclusive	20.0
6	armenie_convention-avec-l-armenie_fd_1424	partagee	10.0
7	australie_convention-avec-l-australie-signee-l...	partagee	10.0
8	autriche_convention-avec-l-autriche-impot-sur-...	partagee	NaN
9	azerbaidjan_convention-avec-l-azerbaidjan_fd_3388	exclusive	10.0
10	bangladesh_convention-avec-le-bangladesh_fd_1726	partagee	10.0
11	belgique_convention-avec-la-belgique-impot-sur...	exclusive	15.0
12	fichedescriptive_62371	indeterminee	10.0
13	bolivie_convention-avec-la-bolivie_fd_1731	partagee	15.0
14	convention-avec-l-ex-yougoslavie_fd_1730	exclusive	NaN
15	bresil_convention-avec-le-bresil_fd_1799	partagee	15.0
16	bulgarie_convention-avec-la-bulgarie_fd_1800	partagee	NaN
17	cameroun_convention-avec-le-cameroun_fd_1722	partagee	15.0
18	chili_convention-avec-le-chili_fd_3899	partagee	5.0
19	chypre_convention-avec-chypre_fd_1818	partagee	10.0
20	colombie_25_06_2015	partagee	10.0
21	congo_convention-avec-le-congo_fd_1820	exclusive	NaN
22	coree-du-sud_convention-avec-la-republique-de-...	exclusive	10.0
23	cote-d-ivoire_convention-avec-la-cote-d-ivoire...	partagee	15.0
24	egypte_convention-avec-l-egypte_fd_1822	partagee	15.0
25	emirats-arabes-unis_convention-avec-les-emirat...	partagee	NaN
26	equateur_convention-avec-l-equateur_fd_1823	partagee	10.0
27	espagne_convention-avec-l-espagne-impot-sur-le...	partagee	10.0
28	estonie_convention-avec-l-estonie_fd_1428	partagee	10.0
29	etats-unis_convention-avec-les-etats-unis-impo...	exclusive	NaN

- The datafram displays **data from 100 conventions** in columnar form

03 Results and interpretation

1) Distribution of the power to impose

```
# Analyser le nombre de conventions "exclusive" "partagee" et “interdeterminee”
tax_kind_counts = df['tax_kind'].value_counts()
# Obtenir le décompte des types d'imposition 'exclusive' 'partagee' et ‘indeterminee’
exclusive_count = tax_kind_counts.get('exclusive', 0)
partage_count = tax_kind_counts.get('partagee', 0)
indeterminee_count = tax_kind_counts.get('indeterminee', 0)


# Afficher les décomptes
print("Nombre de conventions 'exclusive':", exclusive_count)
print("Nombre de conventions 'partagee':", partage_count)
print("Nombre de conventions ‘indetermine’:", indeterminee_count)
```

- To get a more precise idea of our results, the code **counts the number of conventions** with **shared** taxation, those with **exclusive** taxation and those which are **indeterminate**.

1° value_counts() method
2° get() method on the tax_kind_counts object

The result :

```
Nombre de conventions 'exclusive': 25
Nombre de conventions 'partagee': 63
Nombre de conventions 'indeterminee': 13
```

1) Distribution of the power to impose

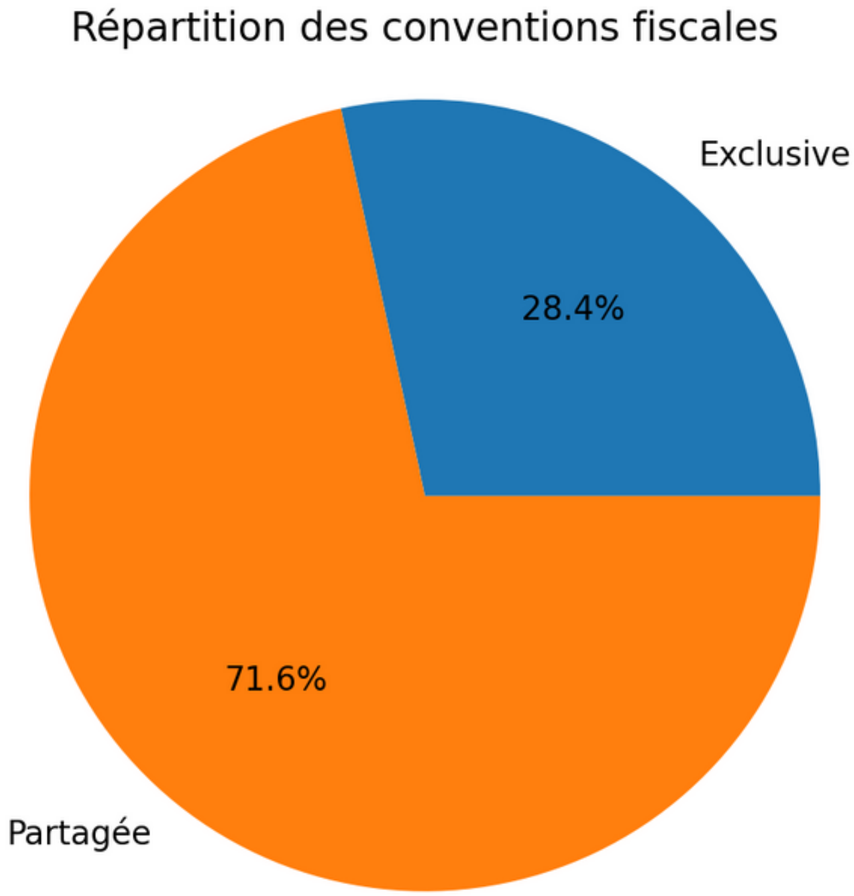
Using the data collected, we ask python to **create a graph** :

- we import the **matplotlib.pyplot** library, which is used to create graphical visualizations in Python
- we **filter** the data in the DataFrame **df** to exclude rows where the column '**tax_kind**' has the value "indeterminate", and only count the occurrences of each taxation type ('exclusive' or 'partagée')

```
In [5]: import matplotlib.pyplot as plt
   ...:
   ...: # Filtrer les données pour exclure la catégorie "indéterminée"
   ...: filtered_tax_counts = df[df['tax_kind'] != 'indeterminee']['tax_kind'].value_counts()
   ...:
   ...: # Création du graphique circulaire
   ...: plt.figure(figsize=(8, 8))
   ...: plt.pie(filtered_tax_counts, labels=filtered_tax_counts.index, autopct='%1.1f%%', startangle=140, colors=['skyb
   ...: lue', 'lightgreen'])
   ...:
   ...: # Ajout de titres
   ...: plt.title('Répartition du pouvoir d\'imposer')
   ...:
   ...: # Affichage du graphique
   ...: plt.axis('equal')  # Assure que le graphique soit circulaire
   ...: plt.show()
```

1) Distribution of the power to impose

- there are **more shared taxes** than exclusive taxes
- promoting international cooperation
- **fairness and transparency**



2) Analysis of tax rate

- we filtered the data to exclude missing values in the 'tax_rate' column of the DataFrame **df** and keeps only the valid values.

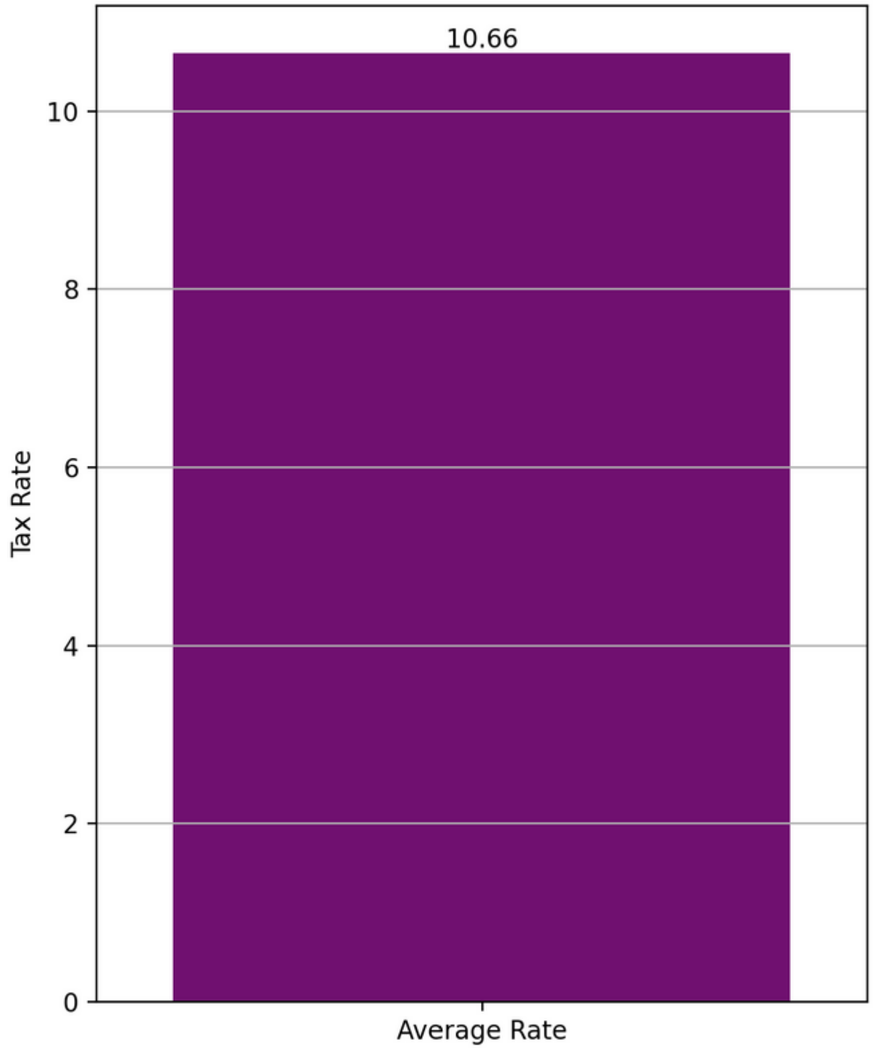
- We **calculated mean of the valid tax rates** (average tax rate)

- and put it in **graph form**

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Filtrer les données pour exclure les valeurs manquantes et ne conserver que les taxes valides
valid_tax_rates = df['tax_rate'].dropna()

# Calculer la moyenne
average_tax_rate = valid_tax_rates.mean()

# Vérifier si des résultats ont été renvoyés
if not states_average_rate.empty:
    state_average_rate = states_average_rate.iloc[0]
else:
    state_average_rate = None

# Création du graphique à barres
plt.figure(figsize=(5, 6))
sns.barplot(x=['Average Rate'], y=[average_tax_rate], color ='purple')

# Ajout du chiffre de la moyenne sur le graphique
plt.text(x=0, y=average_tax_rate, s=f'{average_tax_rate:.2f}', ha='center', va='bottom')

# Ajout de titres et de labels
plt.ylabel('Tax Rate')

# Affichage du graphique
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

2) Analysis of tax rate

- average tax rate : 10,6%
- this is an indication for companies, but it is not enough, as the average is not representative of practice
- **so we decided to carry out a more in depth analysis**



2) Analysis of tax rate

- Using the same command **matplotlib.pyplot,** we calculate the min, max, mode (most commonly used) tax rate from the filtered data.

```
...:
...: # Calculer le taux minimum, maximum et le mode (taux le plus utilisé)
...: min_tax_rate = valid_tax_rates.min()
...: max_tax_rate = valid_tax_rates.max()
...: mode_tax_rate = valid_tax_rates.mode()[0]  # mode() retourne une série, donc [0] pour obtenir la première vale
...: ur
...:
...: # Trouver les États correspondant au taux minimum, maximum et mode
...: state_min_rate = df[df['tax_rate'] == min_tax_rate]['pays'].iloc[0]
...: state_max_rate = df[df['tax_rate'] == max_tax_rate]['pays'].iloc[0]
...: state_mode_rate = df[df['tax_rate'] == mode_tax_rate]['pays'].iloc[0]
...:
...: # Création du graphique à barres
...: plt.figure(figsize=(10, 6))
...: sns.barplot(x=['Minimum', 'Maximum', 'Mode'], y=[min_tax_rate, max_tax_rate, mode_tax_rate], palette='husl')
...:
```

2) Analysis of tax rate

Then, we ask python to :

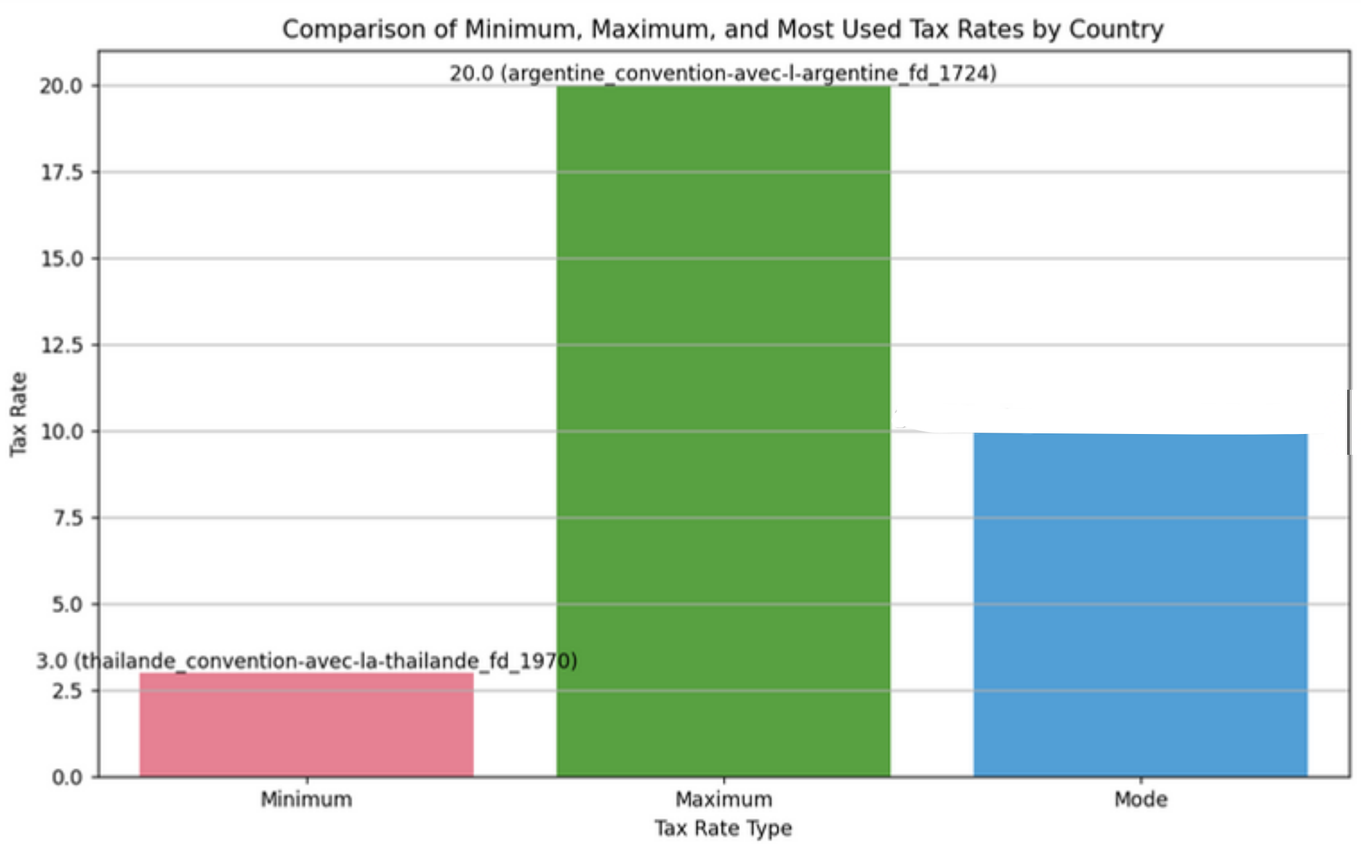
- add annotations to indicate the states corresponding to each tax rate on the bar plot
- add titles to the X-axis and the Y-axis

```
...:
...: # Ajout des annotations pour indiquer les États correspondant à chaque taux
...: plt.text(0, min_tax_rate, f'{min_tax_rate} ({state_min_rate})', ha='center', va='bottom')
...: plt.text(1, max_tax_rate, f'{max_tax_rate} ({state_max_rate})', ha='center', va='bottom')
...: plt.text(2, mode_tax_rate, f'{mode_tax_rate} ({state_mode_rate})', ha='center', va='bottom')
...:
...: # Ajout de titres et de labels
...: plt.title('Comparison of Minimum, Maximum, and Most Used Tax Rates by Country')
...: plt.xlabel('Tax Rate Type')
...: plt.ylabel('Tax Rate')
...:
...: # Affichage du graphique
...: plt.grid(axis='y')
...: plt.tight_layout()
...: plt.show()
```

2) Analysis of tax rate

- **Minimum tax rate :**
  - promotion of investment
  - intenational competitiveness
  - political and social objectives
  - reduce financial cost
  - encouragement of debt
  - effect on capital structure

- **Minimum tax rate:**
  - need for tax revenue
  - deterrence of excessive savings
  - increased financial burden
  - competitive disadvantage
  - impact on investors perception
  - capital allocation considerations



Thank you :)