



HEC
PARIS

LEGAL DATA ANALYTICS

Oral presentation - 27/03/2023

Does the unconstitutionality of Laws passed in French Parliament depend on the political majority ?

Joëlle Adedze - Theo Benichou - Yanis Gandrillon - Ezio Martinez : Pycharmeurs



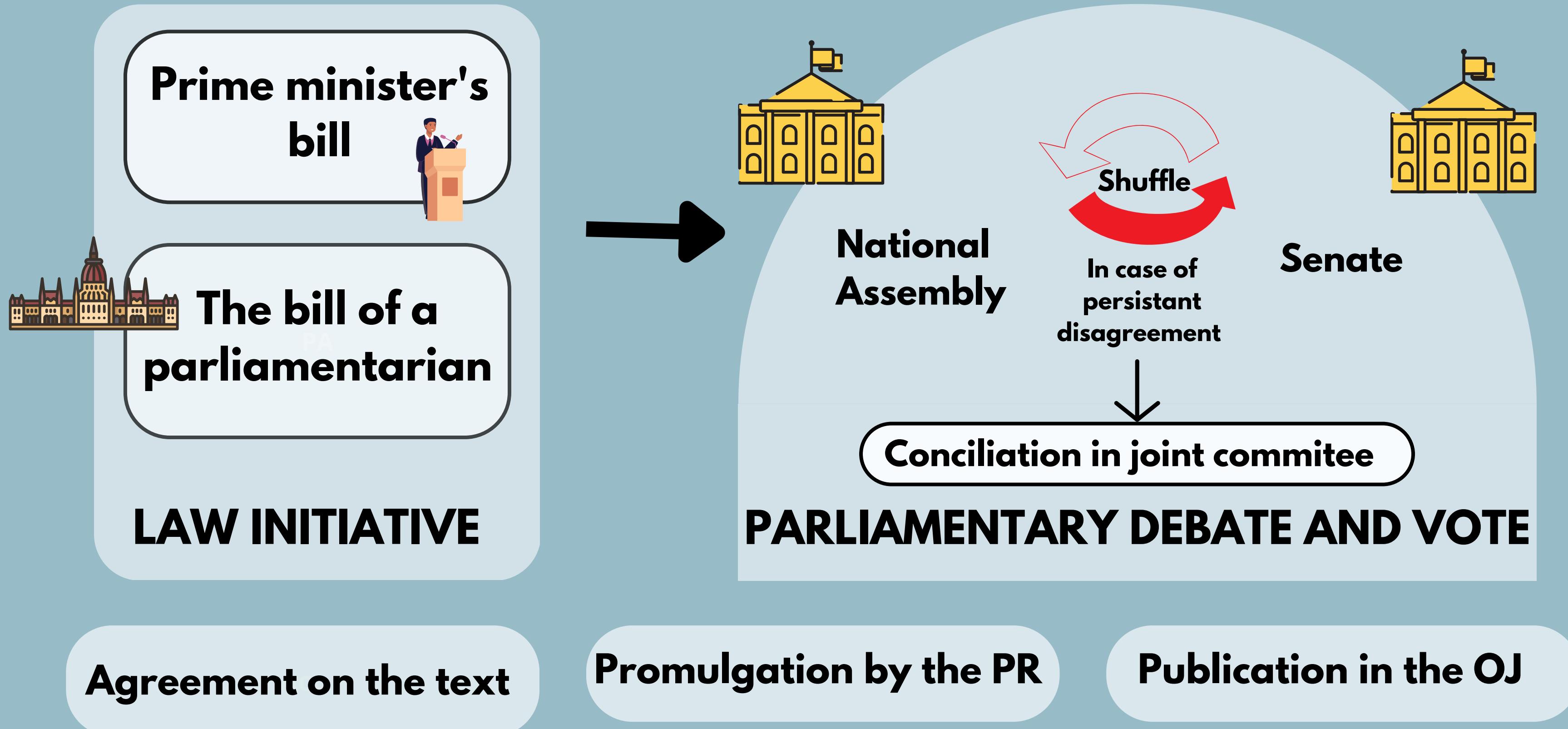
Table of contents

- 1 Introduction to our topic - Legal context, issues and choices**
- 2 Our solution - Key elements of code and technical choices**
- 3 Graphs, Conclusion and Rationales**

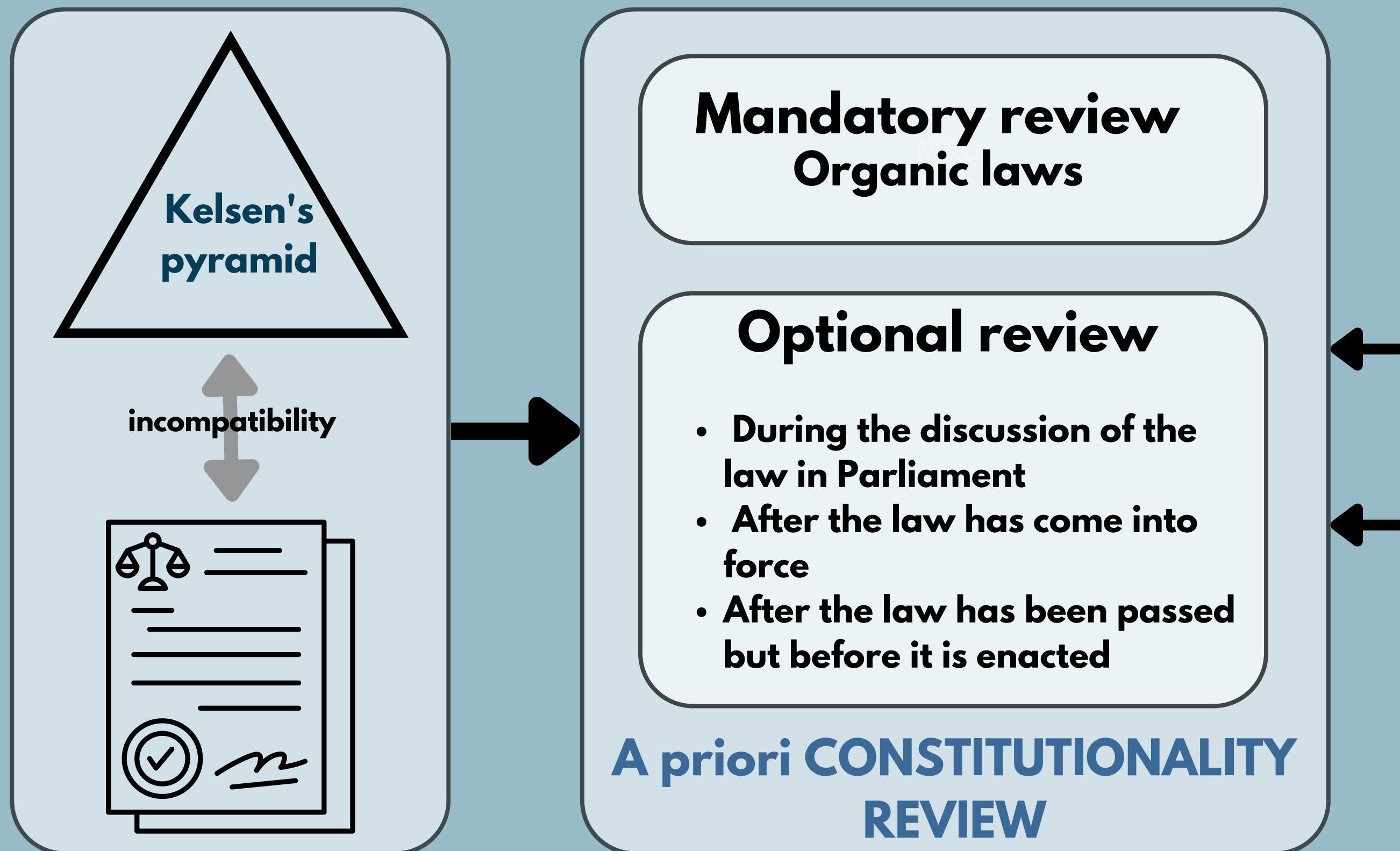
1 - Introduction : legal context, issues and choices



Law-making process in France



Primacy of the Constitution in French Law



The right of referral

- President of the republic
- Prime minister
- President of the NA
- President of the Senate
- **60 deputies or 60 senators**

Our assumptions for this study

**Consideration of "a priori"
constitutionality review only**

**Consideration of ordinary and
organic laws**

Mandate	Time period	Parliament majority	Gouvernement
1	1993-1997	Right wing	F. Mitterrand - E. Balladur J. Chirac - A. Juppé
2	1997-2002	Left wing	J. Chirac - L. Jospin
3	2002-2007	Right wing	J. Chirac - JP. Raffarin / D. Villepin
4	2007-2012	Right wing	N. Sarkozy - F. Fillon
5	2012-2017	Left wing	F. Hollande - JM. Ayrault / M. Valls / B. Cazeneuve
6	2017-2022	Center	E. Macron - E. Philippe



Our issue :

**To what extent does the unconstitutionality
of Laws passed by Parliament vary
according to the political majority?**



2 - Our solution : Key elements of code and technical choices

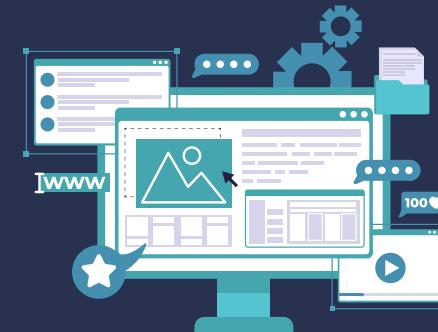


OUR SOLUTION : 4 STEPS

A - Libraries : import all functions needed



B - Scraping : gather the data we need from the Conseil Constitutionnel's website



C - Data processing : create a clean dataframe with the data collected



D - Designing : create the graphs to synthesize the data and draw conclusions



A - Libraries

```
1 # Libraries  
2  
3 import re  
4 from bs4 import BeautifulSoup  
5 import pandas as pd  
6  
7 from selenium import webdriver  
8 from selenium.webdriver.chrome.service import Service  
9 from webdriver_manager.chrome import ChromeDriverManager  
10 from selenium.webdriver.common.keys import Keys as KeysBrowser  
11  
12 import matplotlib.pyplot as plt  
13 import seaborn as sns  
14  
15 import locale  
16 locale.setlocale(locale.LC_ALL, "fr_FR")  
17
```

**USED FOR SCRAPING
AND DATA PROCESSING**

**USED FOR DESIGNING
GRAPHS**

B - Scraping

First, we need to find the correct URL for our study

**FIRST STEP : GO ON
THE WEBSITE'S
DATABASE**

The screenshot shows a web browser displaying the official website of the French Constitutional Council (conseil-constitutionnel.fr). The page is in French (FR) and features a dark blue header with the council's logo. The main navigation menu includes links for 'Actualités et événements', 'Le Conseil', 'Les décisions' (which is underlined in yellow), and 'La Constitution'. A search bar at the top right contains the placeholder 'Ex : décision n° 2011-111 QPC ...'. Below the header, there is a large dark sidebar with several links: 'Toutes les décisions >', 'Comprendre le contentieux électoral >', 'Outils et ressources >', and 'Affaires en instance'. To the right of this sidebar, there are two more sections: 'Décisions par année' and 'Décisions par type'. At the bottom of the page, there are three columns: 'LES AUDIENCES PUBLIQUES QPC' (with a thumbnail of a courtroom and links to 'Affaire n° 2023-1042-QPC du 21 mars 2023' and 'Affaires n° 2023-1040 et 2023-1041 QPC'), 'LES DERNIÈRES DÉCISIONS' (listing 'Décision n° 2023-1039 QPC du 24 mars 2023' and 'Décision n° 2023-1038 QPC du 24 mars 2023' with brief descriptions), and 'Décision n° 2023-6190 AN du 24 mars 2023' (with a link to 'EN SAVOIR PLUS'). The footer of the page includes a 'Titre VII : Accédez au dernier numéro' section and navigation icons.

B - Scraping

SECOND STEP : SET THE RESEARCH WITH THE CRITERIA WE CHOSE

The screenshot shows a web browser window for the Constitutional Council's search interface at recherche.conseil-constitutionnel.fr. The page title is "Les décisions du Conseil Constitutionnel | Conseil constitutionnel". The main navigation menu includes "Actualités et événements", "Le Conseil", "Les décisions", "La Constitution", and a search bar with placeholder text "Ex : décision n° 2011-111 QPC ...". A language selector "FR" is also present.

The main content area is titled "Date de la décision" and features two date selection tools: "CHOISIR UNE DATE" (with a date input field showing "jj/mm/aaaa" and a calendar icon) and "CHOISIR UNE PÉRIODE" (with date inputs for "03/04/1993" and "au" followed by "21/06/2022" and another calendar icon). A large red curved arrow points from the left margin towards this section.

Below the date selection is a section titled "Type de la décision" with a heading "□ CONTRÔLE DE CONSTITUTIONNALITÉ (SAUF LP)". It lists several checkbox options under this heading:

- DC, Loi ordinaire
- DC, Loi organique
- DC, Traité
- DC, Règlement des assemblées
- QPC
- Renvoi Conseil d'État
- Renvoi Cour de cassation
- Référendum d'initiative partagée
- Tout sélectionner

A red curved arrow points from the bottom of the "SECOND STEP" text on the left towards the "Type de la décision" section.

At the bottom of the page is a dark footer bar with a magnifying glass icon and the text "VOTRE RECHERCHE".

B - Scraping

RESULT



TADAAAAAA

474 RESULTS OUT OF
48 PAGES ??!!

The screenshot shows a web browser window with two tabs open: "Les décisions du Conseil Constitutionnel" and "Recherche experte décisions". The main content area displays search results for the query "recherche.conseil-constitutionnel.fr/?q=&filtres3=&type_doc=versionHTML&datepicker=&date-from=1993-04-03&date-to=2022-06-21&filtres1%5B%5D=sous_t...". The results page includes a sidebar for filtering by period (from 03/04/1993 to 21/06/2022) and type (DC, Loi ordinaire, DC, Loi organique, Tous les types). The results are sorted by date (antéchronologique), showing 1 result on page 1 of 48. The first result is a decision from March 17, 2022, regarding the protection of whistleblowers.

Il y a 474 résultats correspondant à votre recherche

FILTRER VOTRE RECHERCHE

PAR PÉRIODE

03/04/1993

21/06/2022

TRIER PAR Date (antéchronologique)

1 sur 48 > »

Décision - 2022-839 DC - Loi visant à améliorer la protection des lanceurs d'alerte - du 17 mars 2022

[Non conformité partielle] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI, dans les conditions prévues au deuxième alinéa de l' article 61 de la Constitution, de la loi visant à améliorer la protection des lanceurs d' alerte, sous le n° 2022- 839 DC, le 18 février 2022, par le Premier ministre. .Au vu des textes suivants:...]

Décision - 2022-838 DC - Loi organique visant à renforcer le rôle du Défenseur des droits en matière de signalement d'alerte - du 17 mars 2022

[Conformité - réserve] [LE CONSEIL CONSTITUTIONNEL A ÉTÉ SAISI le 18. février. 2022. par le Premier.

VOTRE RECHERCHE

RÉINITIALISER MA RECHERCHE

B - Scraping

The url we got :

```
'https://recherche.conseil-constitutionnel.fr/  
mid=a35262a4dccb2f69a36693ec74e69d26&filtres[]=  
type_doc%3AversionHTML&filtres[]=  
sous_type_decision%3A%22DC-  
loi%22&filtres[]=  
sous_type_decision%3A%22DC-  
LO%22&offsetCooc=&  
offsetDisplay=10&nbResultDisplay=10&nbCoocDisplay=&  
UseCluster=&cluster=&showExtr=&sortBy=date&type  
Query=4&dateBefore=&dateAfter=&xtmc=&xtnp=p2&rech_ok=1&datepicker=&  
date-from=1993-04-01&date-to=2022-06-30'
```



B - Scraping

```
18  
19 # We open a driver on the correct url  
20 url = 'https://recherche.conseil-constitutionnel.fr/?mid=a35262a4dcc2f69a36693ec74e69d26&filtres[]&filters[]&type_doc%3AversionHTML&filtres[]&filters[]&sous_type_decision%3A%22DC-loc  
21 driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  
22 driver.get(url)  
23 soup = BeautifulSoup(driver.page_source)  
24 # To browse all pages, we first scrape the total number of decisions. The website shows 10 decisions/webpage, so we'll have to iterate  
25  
26 results = soup.find_all(string=re.compile('résultats$', re.IGNORECASE))  
27 number = int(results[0].split()[0])  
28 number_of_pages = number//10  
29
```

- **We use webdriver.Chrome to open a Chrome page that goes automatically to the url**
- **We access to the HTML code with BeautifulSoup**
- **We need the number of pages to scrap all pages = so we need the number of results and divide it by 10 results by page. As there are 2 iteration of results, we get these 2 iterations with findall and split to get the number of results**

B - Scraping

2 ITERATIONS OF RESULTS

The screenshot shows a web browser with two tabs open: "Les décisions du Conseil Cons..." and "Recherche experte décisions |". The main content is a search results page for the query "sous_titre".

The page includes the following elements:

- Header:** CONSEIL CONSTITUTIONNEL logo, navigation menu with links to Actualités et événements, Le Conseil, Les décisions, La Constitution, and a search bar.
- Section Headers:** LES DÉCISIONS, RECHERCHE EXPERTE DÉCISIONS, and RÉSULTATS.
- Text:** Il y a 474 résultats correspondant à votre recherche.
- Filtering:** FILTRER VOTRE RECHERCHE section with PAR PÉRIODE dropdowns set to 03/04/1993 and 21/06/2022, and a VALIDER button.
- Sorting:** TRIER PAR Date (antéchronologique) dropdown.
- Page Navigation:** 1 sur 48 with next/last buttons.
- Result Preview:** A card for "Décision - 2022-839 DC - Loi visant à améliorer la protection des lanceurs d'alerte - du 17 mars 2022" with a detailed description below it.
- Footer:** VOTRE RECHERCHE and RÉINITIALISER MA RECHERCHE buttons.

Two red arrows highlight specific parts of the interface: one pointing to the "Filtrer votre recherche" section and another pointing to the "Résultats" header.

B - Scraping

```
29
30 decisions = []
31 for i in range(number_of_pages+1): # We iterate through all the pages
32     # We open the correct url for the current page
33     url = 'https://recherche.conseil-constitutionnel.fr/?mid=a35262a4dccb2f69a36693ec74e69d26&filtres[]&filters[]&sous_type_decision%3A%22D'
34     driver.get(url)
35     soup = BeautifulSoup(driver.page_source)
36     set_articles = soup.find_all("article", {"class": "type-decision"}) # We get a set of html div of class 'type_decision'
37     for article in set_articles: # Iterate through these articles
38         date = article.find_next('span',class_="date").text
39         article_number = article.find_next('div',class_="title").text.split(' - ')[1] # The article number is in a div with class 'title'
40         outcome = article.find_next('span',class_="type").text.replace('[','').replace(']', '').split('-')[0].rstrip() # the outcome is in a span with class 'type'
41         if(outcome in ['Conformité', 'Non conformité partielle', 'Non conformité totale']):
42             decisions.append([date, article_number, outcome]) # We add our info into a list
43
```

- **We create a list 'decisions' and a loop that goes through each webpage**
- **We access to the HTML code with BeautifulSoup**
- **We create a list 'set_articles' with every decision on each page**
- **We create another loop in that new list to get the date, the decision reference and the outcome of each decision and add it into the list 'decisions'**

B - Scraping

HTML STRUCTURE OF EACH DECISION

B - Scraping

The list created

```
2022-839 DC', 'Non conformité partielle'], ['17 mars 2022', '2022-838 DC', 'Conformité'], ['10 mars 2022', '2022-837 DC', 'Cor
```

```
mars 2022', '2022-836 DC', 'Conformité'], ['21 janvier 2022', '2022-835 DC', 'Non conformité partielle'], ['20 janvier 2022', '2021-834 DC', 'Non conformi
```

```
'Non conformité partielle'], ['23 décembre 2021', '2021-831 DC', 'Non conformité partielle']
```

•••

When we complained about the size
of the url but we saw the size of the list



C - Data treatment

```
45 decisions_df = pd.DataFrame(decisions, columns = ['Date', 'Numero', 'Outcome']) # Transform our list of lists into a dataframe  
46 decisions_df["Date"] = pd.to_datetime(decisions_df["Date"], format="%d %B %Y", errors="coerce") # Convert the date from str to datetime  
47 decisions_df.set_index('Date', inplace = True) # Set the date as index of the dataframe  
48
```

- **We created the Dataframe from our previous list using the 'pd. ' (from Pandas)**
- **We converted the date from a string into a datetime object**
We added errors="coerce" so that any possible error would not block the code
- **We set the date as the index of the dataframe**

C - Data treatment

```
50     mandat = []
51     for i in range(len(decisions_df)):
52         date = decisions_df.index[i]
53         if(date <= pd.to_datetime('1997-04-21', format = "%Y-%m-%d")):
54             mandat.append(str(1))
55         elif(date >= pd.to_datetime('1997-06-12', format = "%Y-%m-%d") and date <= pd.to_datetime('2002-06-18', format = "%Y-%m-%d")):
56             mandat.append(str(2))
57         elif(pd.to_datetime('2002-06-19', format = "%Y-%m-%d") and date <= pd.to_datetime('2007-06-19', format = "%Y-%m-%d")):
58             mandat.append(str(3))
59         elif(date >= pd.to_datetime('2007-06-20', format = "%Y-%m-%d") and date <= pd.to_datetime('2012-06-19', format = "%Y-%m-%d")):
60             mandat.append(str(4))
61         elif(date >= pd.to_datetime('2012-06-20', format = "%Y-%m-%d") and date <= pd.to_datetime('2017-06-20', format = "%Y-%m-%d")):
62             mandat.append(str(5))
63         else:
64             mandat.append(str(6))
65
66     decisions_df['mandat'] = mandat
```

We added a new column "Mandat" column to the Dataframe

→ We used a 'for' loop :

- to first iterate though each row and extract the date value
- then check the value using 'if' and 'else if' to determine which mandate it belongs to
- and eventually add the corresponding mandate number to a list

→ We then added the new list as a column to the dataframe

C - Data treatment



Date	Numero	Outcome	mandat
2022-03-17	2022-839 DC	Non conformité partielle	6
2022-03-17	2022-838 DC	Conformité	6
2022-03-10	2022-837 DC	Conformité	6
2022-03-10	2022-836 DC	Conformité	6
2022-01-21	2022-835 DC	Non conformité partielle	6
2022-01-20	2021-834 DC	Non conformité partielle	6
2021-12-28	2021-833 DC	Non conformité partielle	6
2021-12-23	2021-831 DC	Non conformité partielle	6
2021-12-17	2021-830 DC	Conformité	6
2021-12-17	2021-829 DC	Non conformité partielle	6
2021-12-16	2021-832 DC	Non conformité partielle	6
2021-11-09	2021-828 DC	Non conformité partielle	6
2021-10-21	2021-827 DC	Conformité	6
2021-10-21	2021-826 DC	Non conformité partielle	6
2021-08-13	2021-825 DC	Non conformité partielle	6
2021-08-13	2021-823 DC	Non conformité partielle	6
2021-08-05	2021-824 DC	Non conformité partielle	6
2021-06-30	2021-822 DC	Non conformité partielle	6
2021-06-29	2021-821 DC	Conformité	6
2021-05-31	2021-819 DC	Conformité	6
2021-05-21	2021-818 DC	Non conformité partielle	6
2021-05-20	2021-817 DC	Non conformité partielle	6
2021-04-15	2021-816 DC	Conformité	6

D - Designing

Line graph Evolution through the years

```
71 decisions_df['year'] = decisions_df.index.year # add a column 'year' to make our graphs
72
73 # First graph: a chart about the evolution of decisions through the years
74 plt.figure()
75 evolution_years = decisions_df.groupby('year').Outcome.value_counts().unstack().fillna(0)
76 evolution_years.plot().figure.savefig('Courbes_evolution.png')
77 plt.close()
```

- We added a "year" column to the dataframe
- We started the plot using 'plt.figure' (from Matplotlib)
- We grouped the decisions in the DF by Year and Outcome
We then counted the number of each outcome for each year
- We plotted the line graph and saved it as a PNG

D - Designing

Histogram

Repartition of decisions depending on mandates

```
79 # Second graph: an histogram about the repartition of decisions depending on the mandate
80 plt.figure()
81 decisions_by_mandat_hist = sns.histplot(data=decisions_df.reset_index(), x = 'mandat', hue = 'Outcome', multiple="dodge", shrink=.8)
82 fig = decisions_by_mandat_hist.get_figure()
83 fig.savefig('Histogramme.png')
84 plt.close()
```

- We create the histogram using 'sns.histplot' (from Seaborn)
 - We reset the index of the DF
 - We made the Mandat column the x-axis of the histogram
 - We based the color encoding of the bars on the Outcome column
 - We made the bars appear side by side, with a 0.8 width
- We got the underlying histogram and saved it as a PNG

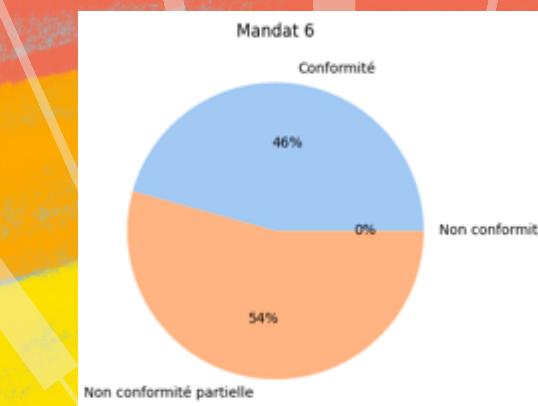
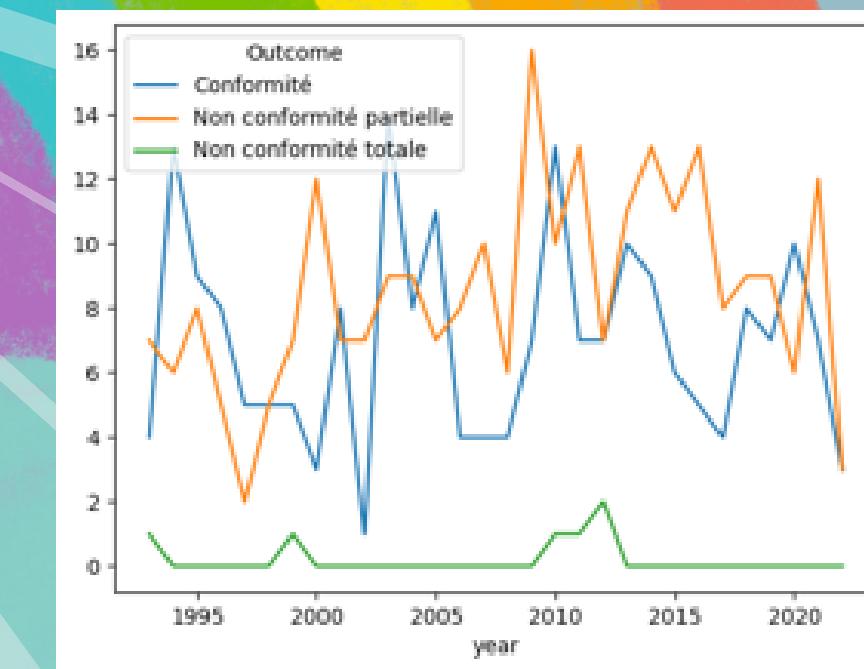
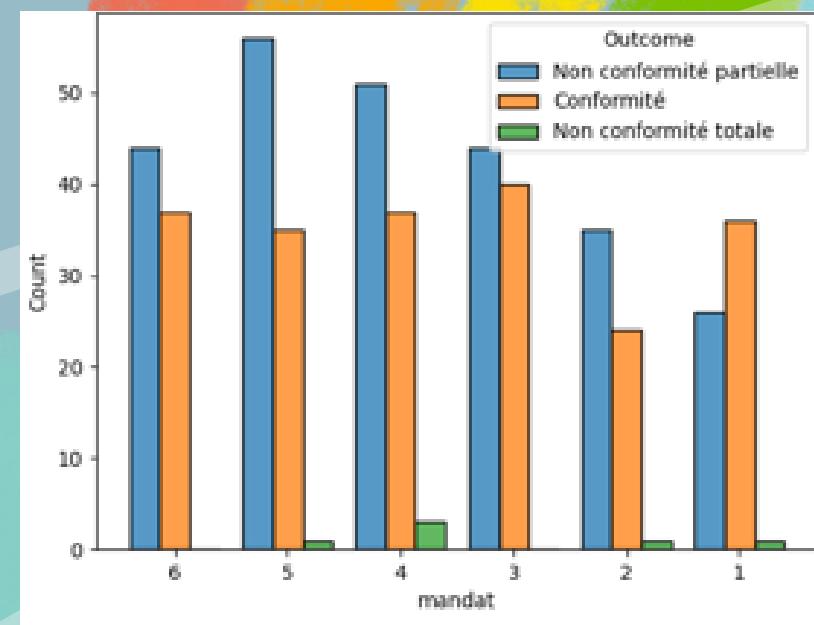
D - Designing

Pie charts

Repartition of decisions for each mandate

```
86     # Third graph: a pie chart about the repartition of decisions for each mandate
87     decisions_by_mandat = decisions_df.groupby('mandat').Outcome.value_counts().unstack().fillna(0)
88     labels = decisions_by_mandat.columns
89     colors = sns.color_palette('pastel')[0:3]
90
91     # We iterate through all mandates
92     for i in range(1,7):
93         plt.figure()
94         print('Mandat '+str(i))
95         plt.pie(decisions_by_mandat.loc[str(i)], labels = labels, colors = colors, autopct='%.0f%%')
96         plt.title('Mandat '+str(i))
97         plt.savefig('Camembert_mandat_'+str(i)+'.png')
98         plt.close()
```

- We grouped the decisions in the DF by Mandate and Outcome
We then counted the number of each outcome for each mandate
- We created a loop that iterates through the 6 mandates and creates a pie chart for each
- Each pie chart is created using 'plt.pie' (from Matplotlib) :
 - Values are taken from the row of the DF corresponding to the current mandate
 - Labels and colors are set usging the defined variables in lines 88 and 89
 - 'plt.title' sets the title of the current chart

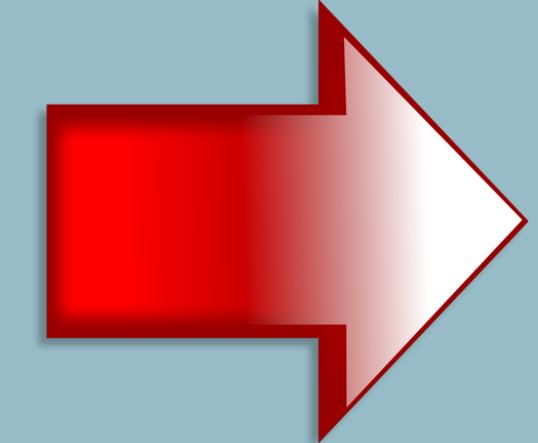


3 - Graphs, Conclusion and Rationales



From Python data frame...

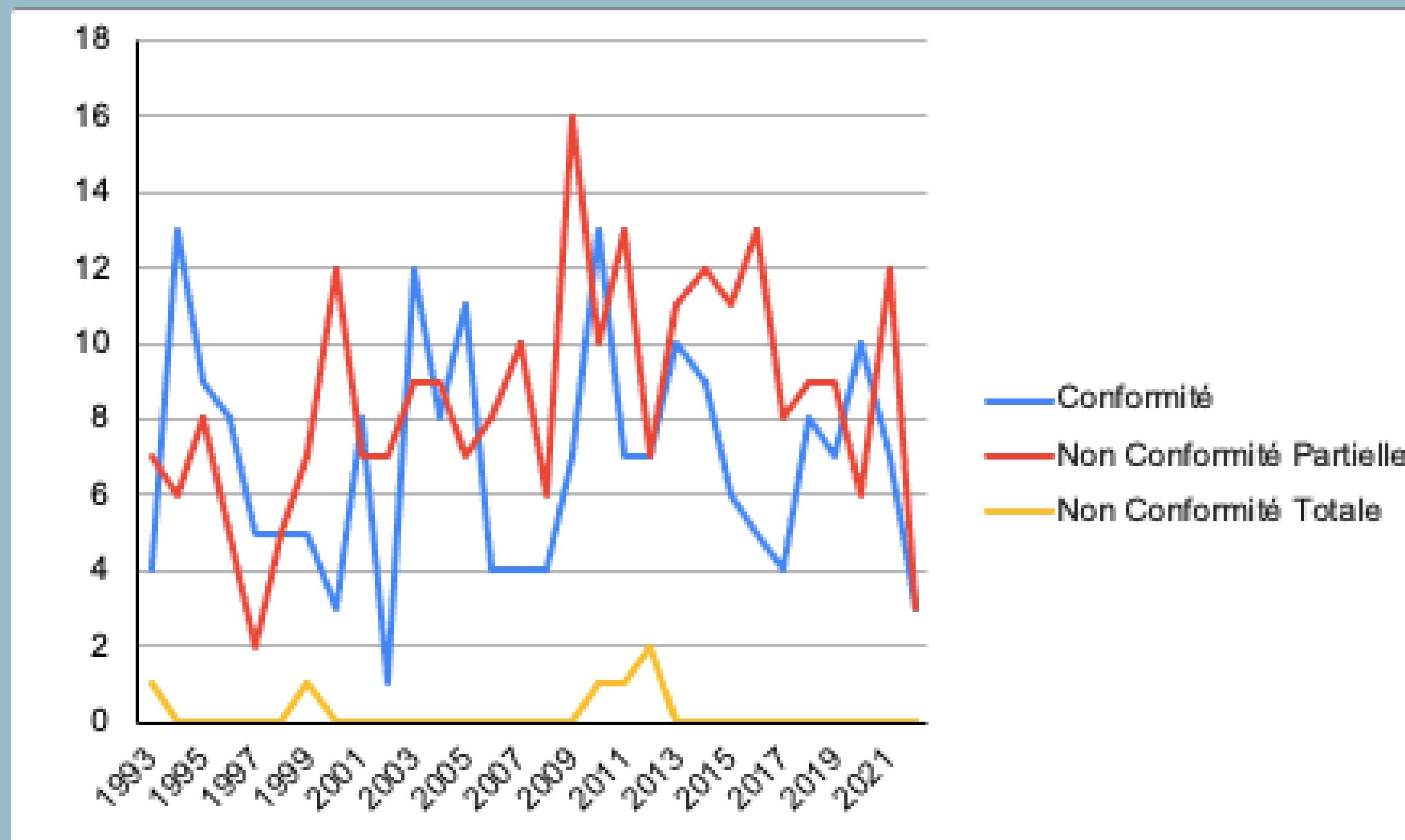
	A
1	Date,Numero,Outcome,mandat
2	2022-03-17,2022-839 DC,Non conformitv© partielle,6
3	2022-03-17,2022-838 DC,Conformitv© ,6
4	2022-03-10,2022-837 DC,Conformitv© ,6
5	2022-03-10,2022-836 DC,Conformitv© ,6
6	2022-01-21,2022-835 DC,Non conformitv© partielle ,6
7	2022-01-20,2021-834 DC,Non conformitv© partielle ,6
8	2021-12-28,2021-833 DC,Non conformitv© partielle,6
9	2021-12-23,2021-831 DC,Non conformitv© partielle ,6
10	2021-12-17,2021-830 DC,Conformitv©,6
11	2021-12-17,2021-829 DC,Non conformitv© partielle ,6
12	2021-12-16,2021-832 DC,Non conformitv© partielle,6
13	2021-11-09,2021-828 DC,Non conformitv© partielle,6
14	2021-10-21,2021-827 DC,Conformitv©,6
15	2021-10-21,2021-826 DC,Non conformitv© partielle,6
16	2021-08-13,2021-825 DC,Non conformitv© partielle,6
17	2021-08-13,2021-823 DC,Non conformitv© partielle ,6
18	2021-08-05,2021-824 DC,Non conformitv© partielle ,6



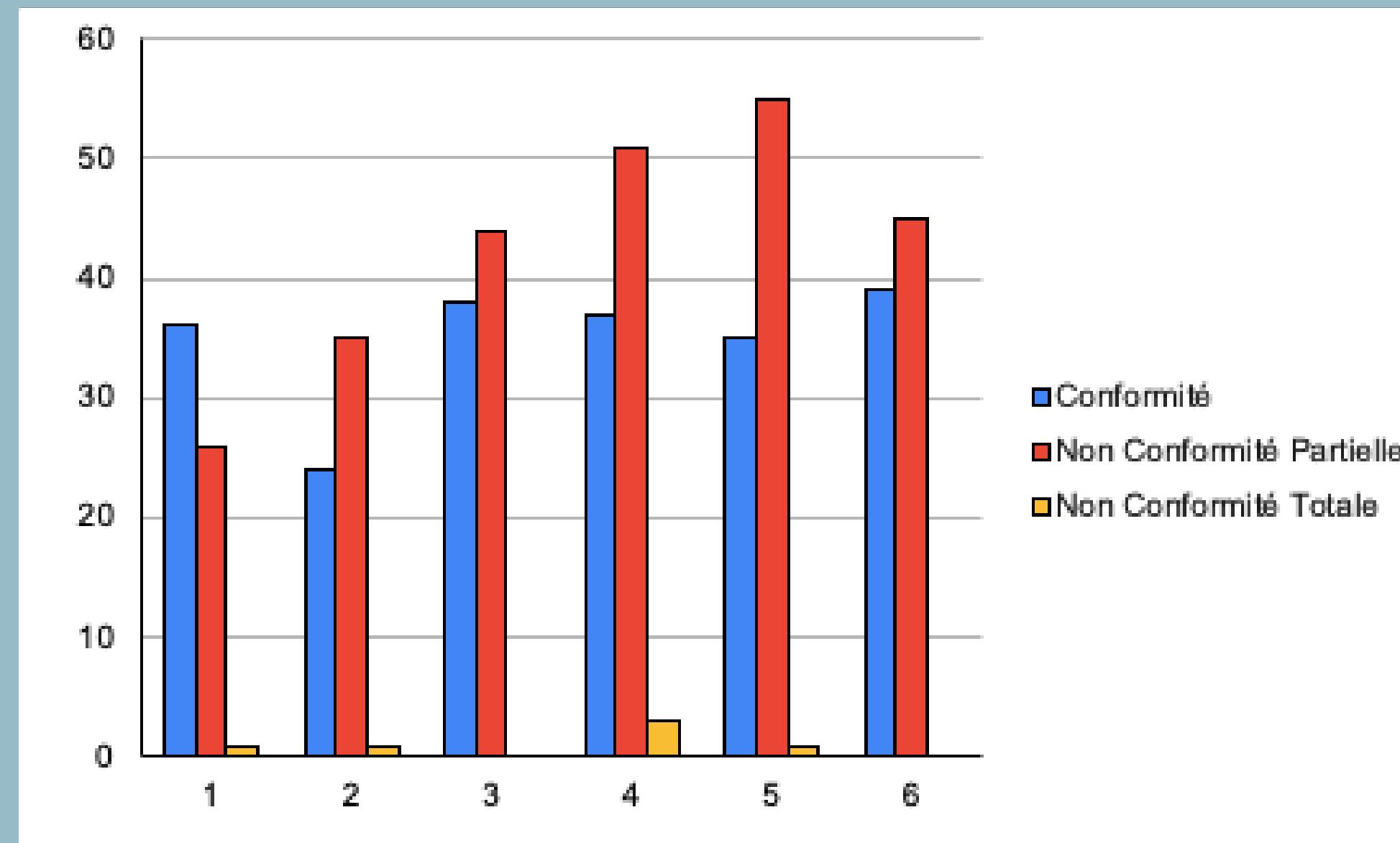
	A	B	C	D	E
1	Date	Number	Outcome	mandat	Year
2	2022-03-17	2022-839 DC	Non conformité	6	2022
3	2022-03-17	2022-838 DC	Conformité	6	2022
4	2022-03-10	2022-837 DC	Conformité	6	2022
5	2022-03-10	2022-836 DC	Conformité	6	2022
6	2022-01-21	2022-835 DC	Non conformité	6	2022
7	2022-01-20	2021-834 DC	Non conformité	6	2022
8	2021-12-28	2021-833 DC	Non conformité	6	2021
9	2021-12-23	2021-831 DC	Non conformité	6	2021
10	2021-12-17	2021-830 DC	Conformité	6	2021
11	2021-12-17	2021-829 DC	Non conformité	6	2021
12	2021-12-16	2021-832 DC	Non conformité	6	2021
13	2021-11-09	2021-828 DC	Non conformité	6	2021
14	2021-10-21	2021-827 DC	Conformité	6	2021
15	2021-10-21	2021-826 DC	Non conformité	6	2021
16	2021-08-13	2021-825 DC	Non conformité	6	2021
17	2021-08-13	2021-823 DC	Non conformité	6	2021
18	2021-08-05	2021-824 DC	Non conformité	6	2021
19	2021-06-30	2021-822 DC	Non conformité	6	2021
20	2021-06-29	2021-821 DC	Conformité	6	2021

Cleaning the Python dataframe

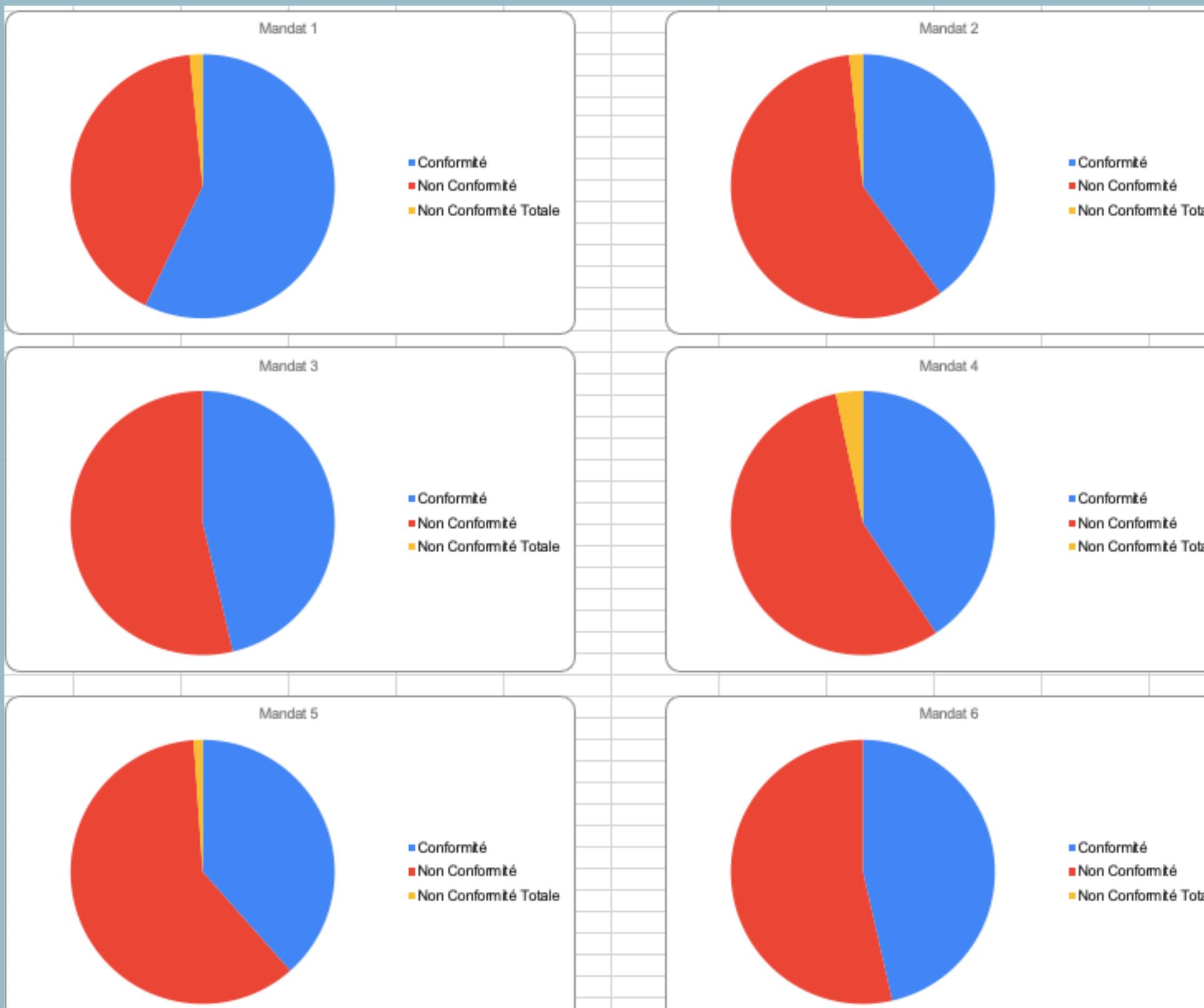
...To Excel graphs (1/3)



...To Excel graphs (2/3)



...To Excel graphs (3/3)

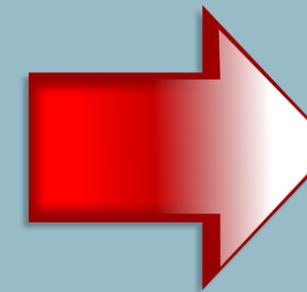
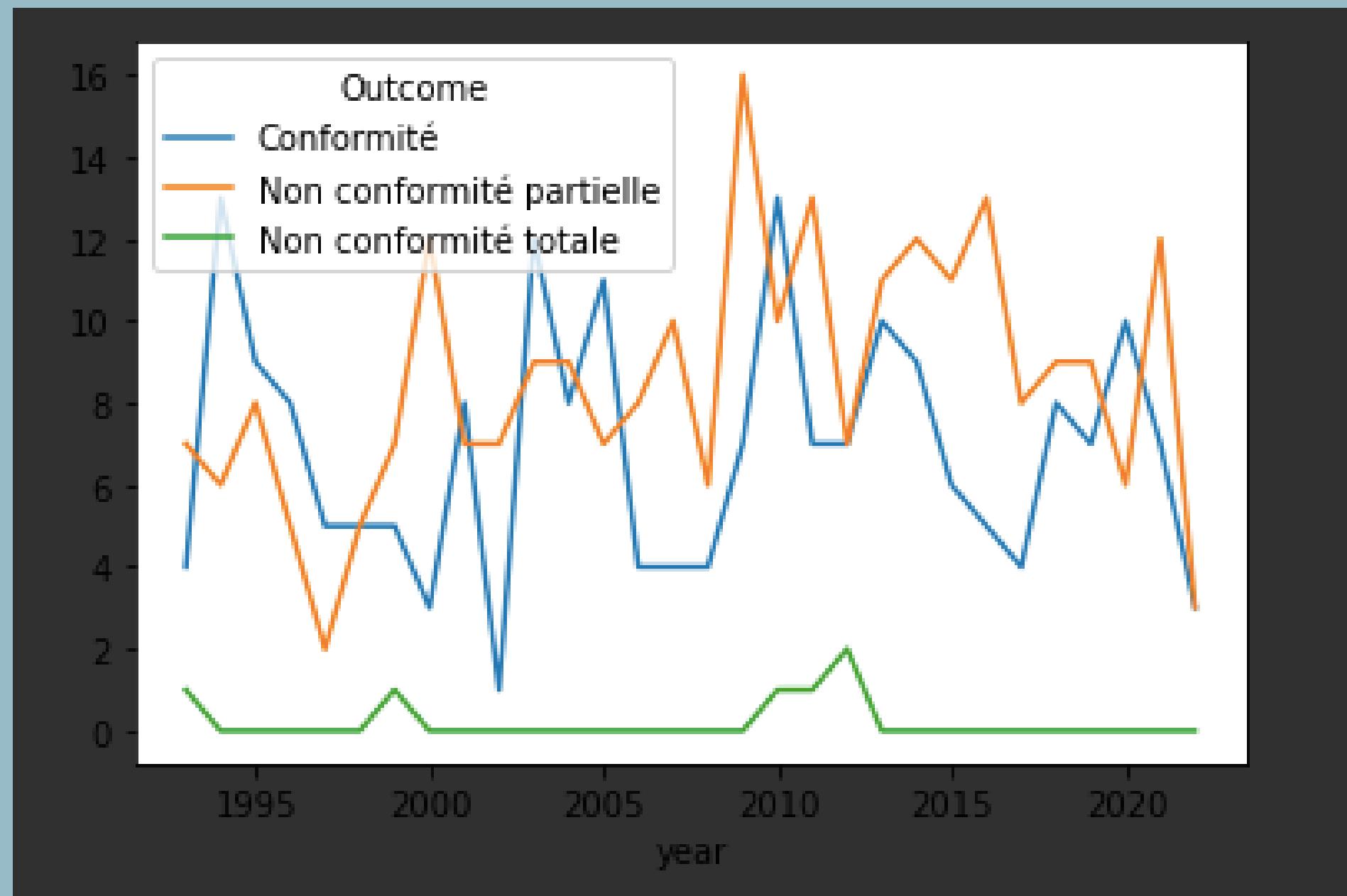


Into Python graphs (1/3)

?

Per year

?

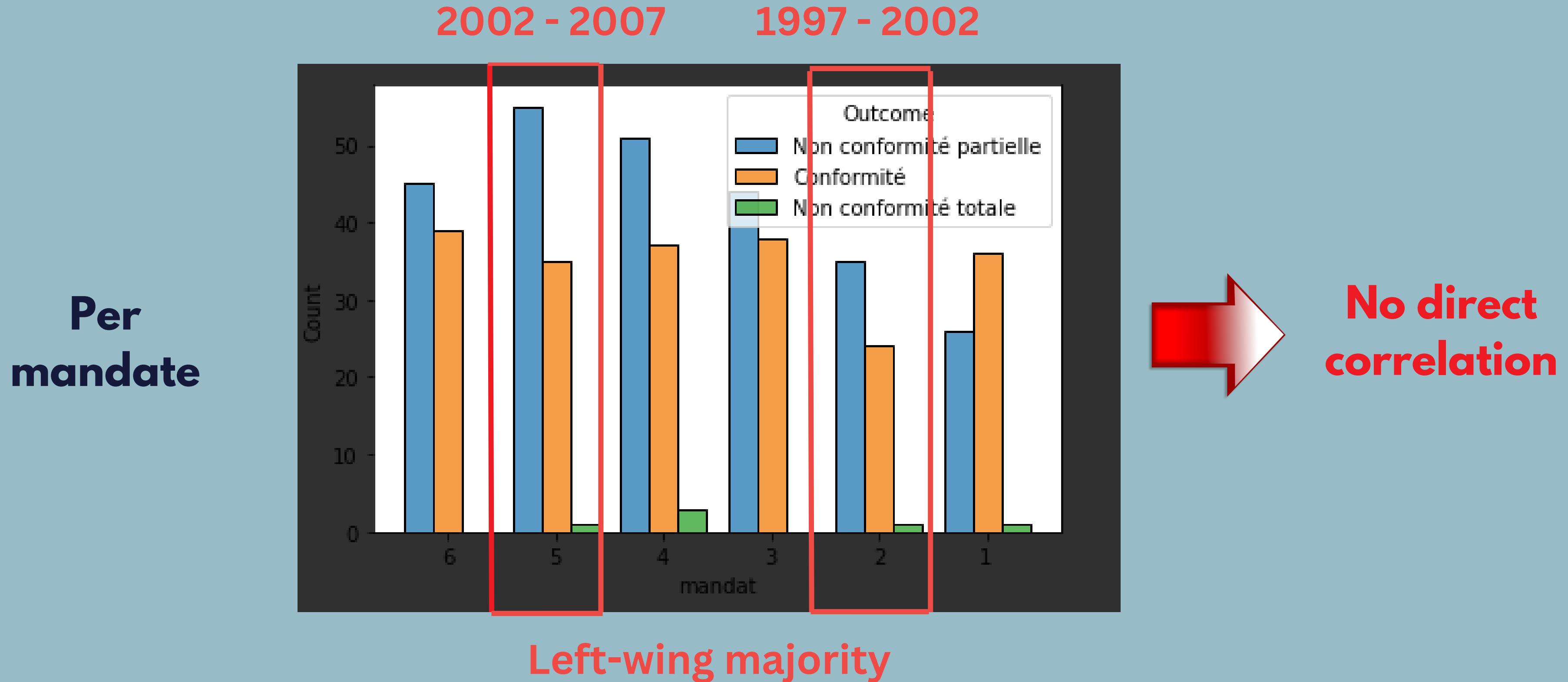


Not
indicative

?

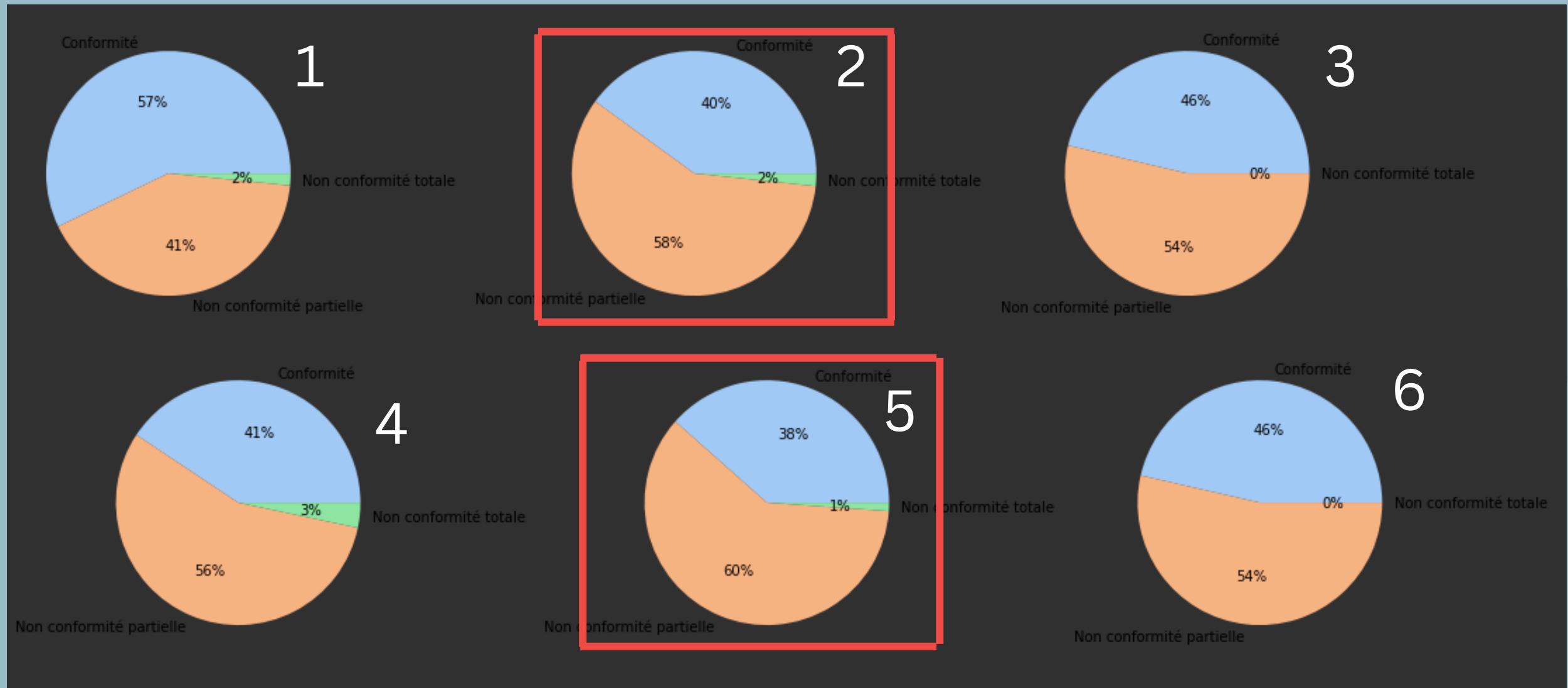
?

Into Python graphs (2/3)



Into Python graphs (3/3)

Per
mandate

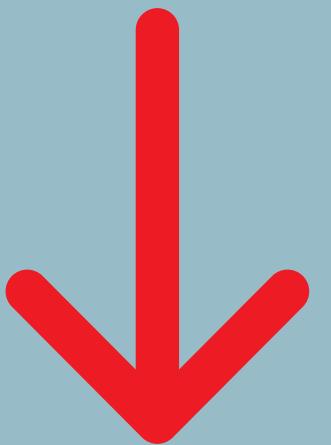


Left-wing majority

No direct correlation

Conclusion

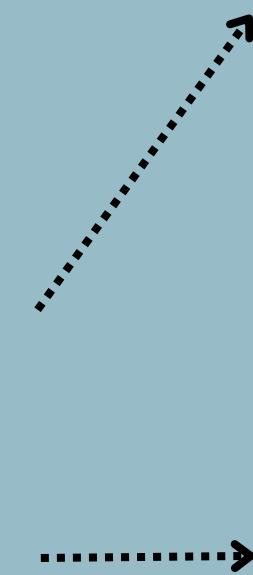
The **unconstitutionality of Laws passed in French Parliament does NOT depend on the political majority thereof**



Why is that ?

Rationales

Lack of correlation



Ability to propose a Law is not reserved for the majority

Ability to defer a Law to the FCC is not reserved for the majority

Inability of FCC members to being reappointed

= A Law declared unconstitutional by the FCC may have been proposed by the minority

= A Law declared unconstitutional by the FCC may have been deferred by the minority

= FCC members have no interest in trying to please the political majority

Thank you all !

