
# ANALYSE GRAPHIQUE DES TRAVAUX DE RECHERCHE SUR L'HISTOIRE DU FÉMINISME

## UE 3 - *Réseaux scientométriques*

Master EISO Epistémologie et Ingenierie de la Science Ouverte

M2 IMST EISO -Novembre 2023

Master IMST

# SOMMAIRE

## Introduction

Ce document présente l'analyse d'un réseau de données issues de la base de données HAL, archive ouverte en ligne. Notre sujet porte sur l'histoire du féminisme. Nous avons souhaité travailler sur ce sujet en raison de son actualité et de son importance à nos yeux. C'est un sujet que nous traitons à l'occasion de notre cours "Genre et Academia" et que nous avons voulu analyser et visualiser grâce à ce cours "Réseaux scientométriques". Quand ce mouvement politique est-il apparu ? Est-ce qu'il a été impacté par l'arrivée des nouvelles technologies ? Nous avons souhaité analyser en particulier l'arrivée des médias et des réseaux sociaux dans l'histoire du féminisme. Que ces derniers aient un impact positif ou négatif, une analyse du corpus extrait de HAL à l'aide de Python et Gephi permet d'observer si ces deux mots-clés reviennent régulièrement, ou non, dans les travaux sur l'histoire du féminisme.

Pour la partie requête de bases de données bibliométriques ainsi que pour la production de graphes, nous utilisons python. La manipulation des réseaux se fait sur Gephi qui permet une visualisation claire et une interprétation à posteriori.

Pour développer notre travail sur la question de départ, nous avons souhaité savoir de quelle manière l'histoire du féminisme faisait l'objet de notre corpus sur HAL. Y a-t-il une majorité d'articles travaillant sur cette question ? Ou le sujet fait-il simplement l'objet de blogs ou de quelques communications en conférence ? Nous voulons donc connaître l'importance de ce sujet dans l'espace public à travers un graphique. Nous avons ensuite souhaité produire un second graphique montrant les domaines dans lesquels cette question est la plus travaillée. L'histoire du féminisme est-elle davantage une question de genre, d'histoire, de sociologie ou encore de religion ? Nous allons pouvoir observer quels sont les domaines disciplinaires engagés dans ces objets d'études.

A l'occasion de ce travail, nous allons pouvoir faire une première analyse des réseaux de mots-clés puis ajouter quelques détails sur le périmètre de notre sujet à l'aide de graphiques. Un second réseau bipartite va nous permettre d'observer l'évolution au cours du temps (en années) de l'apparition de mots-clés ciblés autour des réseaux sociaux et des médias. Nous permettant ainsi de répondre à notre question précédemment énoncée : Est-ce que le mouvement féministe a été impacté par l'arrivée des nouvelles technologies ?

## 1. Les requêtes utilisées

Pour commencer notre requête, nous l'avons construite à partir du terme "féminisme". Nous avons ensuite ajouté le mot-clé "histoire" pour cerner notre sujet. En revanche, nous n'avons pas mis de périmètre temporel afin de voir notamment en quelle année le sujet a-t-il commencé à être travaillé (2002).
Les informations qui nous importent pour notre analyse sont le titre, la date de publication, les mots-clés du document, le domaine d'étude et le type de document.
Voici les requêtes principales qui nous ont permis de définir notre sujet avec Python :
Nous avons entré une URL qui contient des paramètres de requête (ligne de code 1 en annexe) :

- *q=Féminisme* : le terme principal de notre recherche est "Féminisme" ;
- *rows=135* : nous avons demandé à l'API de retourner au maximum 135 résultats dans la réponse (ce qui correspond au nombre de résultats à notre requête sur HAL) en sachant que si nous ne mettons pas un nombre précis, le nombre de retours sera 30 par défaut ;
- *keyword_t=Histoire* : nous avons ajouté un mot-clé pour spécifier et préciser notre recherche;
- *response*: c'est la variable dans laquelle la réponse de la requête *http* est stockée. Cette réponse contiendra des informations renvoyées par l'API, telles que des données au format *json* etc.

Dans un second temps, nous avons spécifié les champs de données que nous souhaitions obtenir grâce à la ligne de code 2 (annexes).

Nous avons ensuite construit un tableau avec ces données. Pour que le résultat soit plus lisible, on peut demander les résultats en .csv et les afficher avec pandas avec ces commandes (ligne de code 3 en annexe).

Cette ligne de code nous a bien permis de trier le DataFrame as_csv en fonction des valeurs de la colonne "publicationDate_s" dans un ordre décroissant (ligne de code 4 en annexe).

## 2. Création de graphiques avec Python

**Types de documents traitant de l'histoire du féminisme**

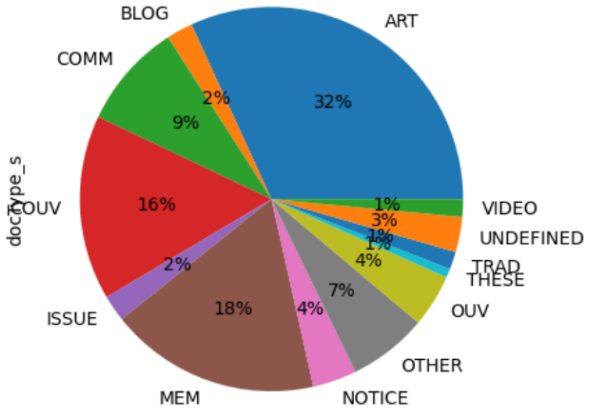
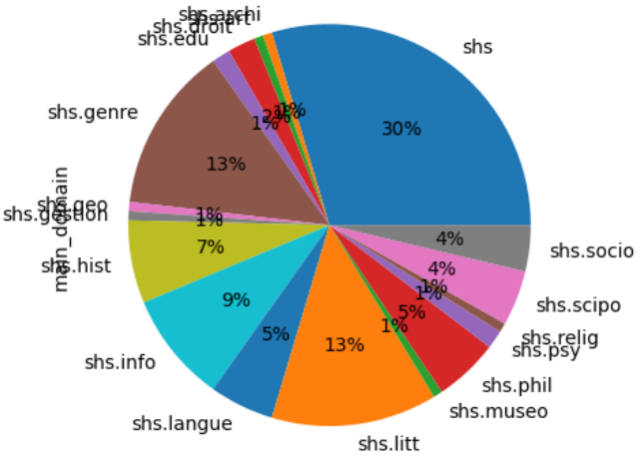



***Figure 1*** *- Représentation graphique de la répartition des types de documents. COMM correspond à des communications dans des conférences, ART à des articles de revues, COUV à des chapitres d'ouvrages, OUV à des ouvrages et MEM à des mémoires.*

Cette figure présente la répartition des types de documents dans le résultat de notre recherche. Les articles de revues sont majoritaires, puisqu'ils composent presque le tiers de notre corpus (32 %). En deuxième position par ordre d'importance, nous retrouvons les mémoires qui composent 18 % de notre corpus, puis viennent les chapitres d'ouvrages qui constituent 16 % du corpus.
Nous pouvons aussi noter que les communications dans les conférences représentent 9% de notre corpus. Les 25 % restants correspondent à différents autres types de documents (blogs, thèses, ouvrages...) qui sont présents dans des proportions plus minimes. La composition de notre corpus est donc relativement hétérogène avec une dominance d'articles montrant l'intérêt scientifique pour cette question sociétale.

**Domaines de recherche concernés**

***Figure 2 –*** *Répartition des principales disciplines pour les travaux de notre requête.*




La figure 2 nous permet d'avoir un aperçu de la répartition des travaux par discipline. Sur HAL, à chaque dépôt peut être associé un ou des domaines et un ou plusieurs sous domaines disciplinaires. Nous avons fait en sorte que le graphique n'affiche que les domaines principaux pour chaque article.
Les disciplines les plus représentées sont liées aux sciences humaines avec notamment la catégorie sciences humaines et sociales (shs) avec une présence de 30% dans notre corpus. Les autres disciplines dominantes du corpus sont les études du genre (shs.genre), la littérature (shs.litt), l'information (shs.info) et l'histoire (shs.hist).
De manière générale, ce graphique montre que notre requête était très vaste et s'inscrit dans plusieurs domaines des sciences humaines.

## 3. Visualisation de notre réseau sur GePhi

### Les mots-clés des travaux étudiés

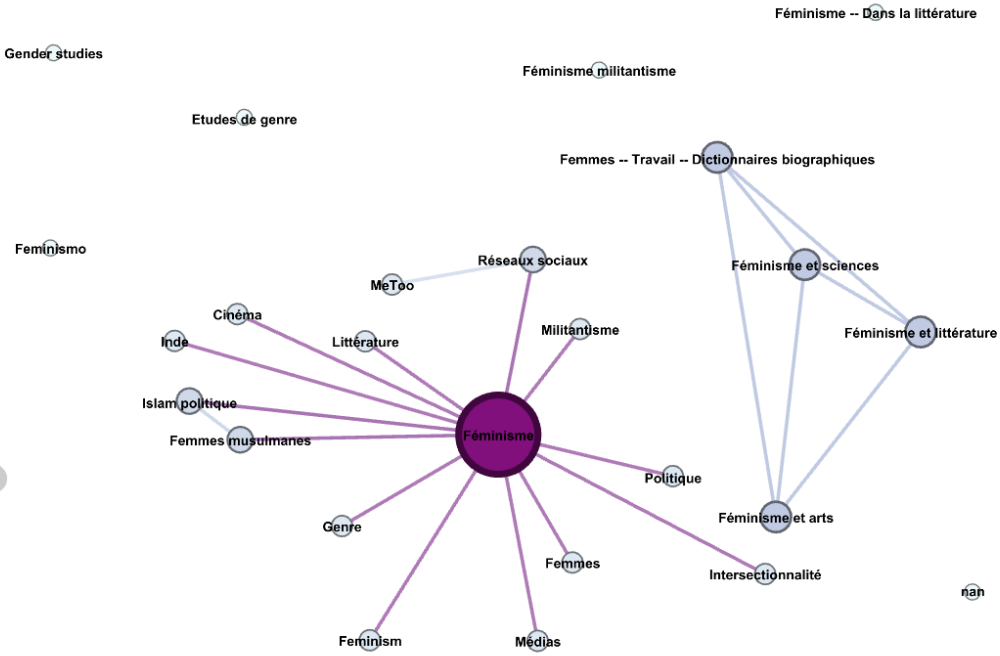



***Figure 3*** *- Réseau des mots-clés* représentant la betweenness (taille) et la closeness (intensité de couleur) des différents nœuds.*
**Ici les mots apparaissent au moins deux fois dans les 135 travaux étudiés.*

Ce réseau (**Figure 3**) nous indique beaucoup d'informations. Naturellement, "Féminisme" possède la betweenness la plus élevée. Nous le remarquons par la taille de ce nœud, qui est largement plus imposante que celle des autres termes présents dans le réseau. Notons par ailleurs, la grande majorité des mots-clés est en lien avec notre noeud central, une connexion unique qui nous explique que l'on ne peut pas parler de "Genre", de "Cinéma" ou encore de "Politique" sans avoir abordé le terme de "Féminisme" (pour notre sujet).

Les mots-clefs gravitant autour de ce nœud central ("Féminisme") n'ont pas réellement de betweenness qui se démarque plus que les autres, leur niveau de centralité est quasiment identique. Cette centralité presque neutre pour le reste des termes est tout de même étonnante. On souligne ici, une betweenness identique entre deux mots-clés "Femmes" et "Inde", c'est-à-dire qu'il n'y a pas de différence au niveau des liens de ces termes avec les autres nœuds présents dans le réseau. Ce résultat est plutôt étonnant puisqu'on va naturellement et plus facilement relier les femmes au féminisme, que l'Inde. Cette incohérence pourrait être expliquée pour plusieurs raisons. Tout d'abord notre sujet est très niche, on retrouve peu de documentation sur le sujet étudié dans la base de donnée HAL (135 résultats obtenus), ce qui pourrait expliquer la faible betweenness de certains mots-clés alors qu'ils nous apparaissent essentiels au sujet. De plus, nombreux sont les travaux n'ayant pas indiqué de mots-clés. Cette erreur dans le référencement correspond à "nan" présent et isolé sur notre réseau avec une occurrence de 15 donc environ 11,11% de nos résultats n'affichent pas de mots-clés.

Lorsque l'on se penche sur la closeness des nœuds, c'est-à-dire la distance moyenne d'un nœud avec les autres (représentée par la couleur du réseau de la Figure 3 : plus la couleur est foncée, plus la distance avec les autres nœuds est faible). Encore une fois, rien d'étonnant car nous retrouvons "Féminisme" en violet très foncé, étant le mot-clé le plus présent, c'est alors le plus proche des autres nœuds. Des thématiques se dégagent également, avec des liaisons logiques entre certains noeuds : "Réseaux sociaux" et "MeToo"; 'Islam politique" et "Femmes musulmanes" ou encore avec ce graphe qui se dégage du réseau "Femmes-Travail-Dictionnaires biographique", "Féminisme et sciences", "Féminisme et littérature" et "Féminisme et arts". Ce graphe est certes isolé en n'étant pas rattaché au nœud central, mais pour autant sa closeness n'est pas faible puisque ces 4 nœuds sont reliés les uns avec les autres et ainsi leur distance est moindre. Contrairement à d'autres nœuds isolés du nœud central : "nan", "Études de genre" ou encore "Féminisme militantisme" qui ont une closeness extrêmement faible. Nous aurions pu aller plus loin et affiner notre travail en mettant en place une ligne de code regroupant certains mots présents en doublon mais afficher dans une autre langue, ou encore en dissociant certains mots-clés en réalité composés de 2 mots-clés :

- "Gender studies" = "Études de genre"
- "Feminismo" = "Féminisme"
- "Féminisme militantisme" = "Féminisme" + "Militantisme"
- "Féminisme - Dans la littérature" = "Féminisme et littérature" = "Féminisme" + "Littérature"

**L'évolution de l'apparition des médias et des réseaux sociaux**

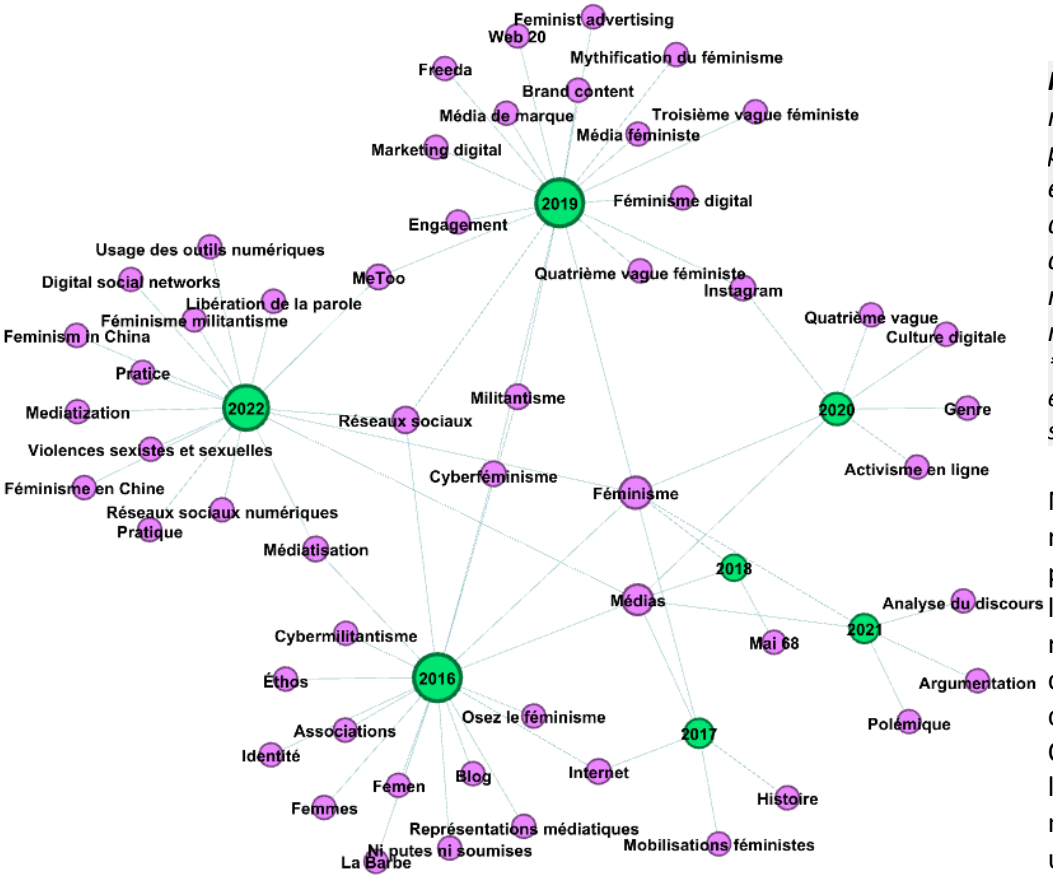



***Figure 4*** *- Réseau bipartite mots-clés*/années de publications. Les nœuds en vert représentent les années de publication, ceux en mauve représentent les mots-clés.*
**Ici les mots-clefs choisis étant : Médias et Réseaux sociaux*

Nous avons réalisé un réseau bipartite permettant de voir l'évolution des médias et réseaux sociaux au cours du temps, dans l'histoire du féminisme (Figure 4). On remarque alors l'arrivée des médias dans notre sujet en 2016, avec un vrai mouvement féministe mis en avant avec de nombreuses associations. Puis les publications diminuent jusqu'en 2019, de nouveau nous avons beaucoup de publications avec des mots-clés autour d'un nouveau mouvement féministe : MeToo, qui est majoritairement en action sur les réseaux sociaux et a été très médiatisé. Puis de nouveau une diminution de publications autour du féminisme et des réseaux sociaux, des médias jusqu'en 2022 où le nombre de travaux publiés augmente. Nous avons donc des pics de publication qui sont les nœuds verts les plus gros : en 2016, 2019 et 2022. De plus, ce réseau nous permet d'observer de nouveaux termes liant le féminisme et sa lutte, avec les réseaux sociaux et la médiatisation : "CyberMilitantisme", "Féminisme digital" et "Activisme en ligne".

## 4. Interprétation de l'analyse et conclusion

L'objectif de ce travail était d'aborder la question du féminisme, sujet d'actualité. Étant un sujet assez large, nous avons souhaité l'affiner en y ajoutant un mot-clé lors de notre requête dans HAL. Nous avons donc décidé de travailler sur l'histoire du féminisme, nous permettant d'obtenir un nombre de documents moins important (en passant de 1950 à 135 résultats) sur une problématique plus précise. Dans un premier temps, nous avons voulu nous renseigner sur le type de documents qui composaient le corpus, formant l'objet de notre premier graphique. Une des limites de ce graphique est que nous avons deux catégories ne nous permettant pas d'avoir des informations sur certains types de documents, avec "autre" à 7% et "indéfini" à 3%. Nous avons tout de même pu constater l'importance de ce sujet dans le monde de la recherche. Le second graphique nous a permis d'avoir un bon regard sur les domaines, et surtout les sous-domaines, présents dans le corpus avec une majorité des études du genre et de la littérature par exemple. A noter que pour obtenir cette représentation, nous avons dû ajouter une colonne dans notre tableau faisant ressortir le sous-domaine principal de chaque document. Chaque document avait en effet plusieurs sous-domaines de référence à la base, nous faisant obtenir un graphique lourd et illisible (voir lignes de code 5 et 6 en annexes).

Nous nous sommes ensuite concentrées sur une analyse à base de mots-clés, or un biais important à prendre en compte est que nous avons 15 documents sur 135 dans notre corpus qui n'ont pas de mots-clés.
A partir de ces mots-clés, nous avons pu obtenir 2 réseaux : un premier plutôt général faisant apparaître tous les mots présents au moins 2 fois dans les documents étudiés, et un second réseau étudiant l'arrivée des médias et des réseaux sociaux dans l'histoire du féminisme.

Lors de l'étude de notre premier réseau, nous avons pu constater rapidement que les termes de "Médias" et "Réseaux sociaux" ne se démarquent pas plus que les autres mots-clés présents. Nous avons également observé certaines limites qui rendent sans doute notre étude moins pointue, notamment par le choix de notre sujet qui est très précis et spécifique ; mais également à cause de nombreux travaux qui ne sont pas correctement, voire pas du tout, répertoriés au niveau de leurs mots-clés. Le dernier réseau, permet d'établir une réelle conclusion sur notre problématique de base. Les médias ainsi que les réseaux prennent une place importante dans les travaux scientifiques sur le féminisme, mais cela depuis seulement 2016. Des années ressortent d'autant plus avec des travaux en lien avec l'actualité de l'année :

- 2016 : "Femen", "Ni putes ni soumises", "Osez les femmes", "Association"
  qui correspondent aux actions et mobilisations des associations féministes.
- 2019-2022 : "MeToo", "Libération de la parole", "4ème vague féministe"
  qui correspondent à l'utilisation des réseaux sociaux ayant permis aux femmes de libérer la parole et de dénoncer les actions de harcèlement et de violence qu'elles ont pu subir.

En plus du lien entre les mouvements féministes et les médias, nous pouvons voir que les réseaux sociaux deviennent une thématique à part entière avec la mise en place d'un lexique qui lui est propre.

# ANNEXES

## Lignes de codes Python

### Ligne de code n°1 :

```
response = requests.get("http://api.archives-ouvertes.fr/search/index/?q=Féminisme&rows=135&keyword_t=Histoire")
response.json()
```

Cette ligne de code utilise la bibliothèque Python *requests* pour envoyer une requête *http get* à l'URL spécifiée. Voici une description détaillée de la ligne de code :

- *requests.get(...)* : fonction de la bibliothèque *requests* qui envoie une requête *http get* à l'URL spécifiée entre parenthèses ;
- La requête *get* est envoyée à l'URL suivante : "*http://api.archives-ouvertes.fr/search/index/?q=Féminisme&rows=30&keyword_t=Histoire*" : Cette URL pointe vers une API (interface de programmation d'application) web. L'URL contient alors des paramètres de requête :
  - *q=Féminisme* : le terme principal de notre recherche est "Féminisme" ;
  - *rows=135* : nous avons demandé à l'API de retourner au maximum 135 résultats dans la réponse (ce qui correspond au nombre de résultats à notre requête sur HAL) en sachant que si nous ne mettons pas un nombre précis, le nombre de retours sera 30 par défaut ;
  - *keyword_t=Histoire* : nous avons ajouté un mot-clé pour spécifier et préciser notre recherche ;
  - *response*: c'est la variable dans laquelle la réponse de la requête *http* est stockée. Cette réponse contiendra des informations renvoyées par l'API, telles que des données au format *json* etc.
- concernant la ligne response.json () :
  - *.json()* : ce terme correspond à une méthode d'analyse du contenu de la réponse *http* pour la traiter en tant que JSON (JavaScript Object Notation). La méthode *json()* transforme les données JSON de la réponse *http* en un objet Python manipulable dans notre programme. Pour chaque article obtenu, une citation standard sera fournie.

En résumé, cette ligne de code envoie une requête *get* à une API en utilisant l'URL spécifiée avec les paramètres de recherche et stocke la réponse de cette requête dans la variable *response* à laquelle seront extraites les données *json*.

### Ligne de code n°2 :

```
response = requests.get("http://api.archives-ouvertes.fr/search/index/?q=Féminisme&rows=135&keyword_t=Histoire&fl=title_s,keyword_s,publicationDate_s,docType_s")
response.json()
```

Description de cette ligne de code :

- *requests.get(...)* : fonction de la bibliothèque *requests* qui envoie une requête *http get* à l'URL spécifiée entre parenthèses ;
- "*http://api.archives-ouvertes.fr/search/index/?q=Féminisme&rows=135&keyword_t=Histoire&fl=title_s,keyword_s,publicationDate_s,docType_s*" : cette URL pointe vers une API web. L'URL contient alors des paramètres de requête :
  - *q=Féminisme* : le terme principal de notre recherche est "Féminisme" ;
  - *rows=135* : 135 résultats au maximum dans la réponse ;
  - *keyword_t=Histoire* : mot-clé pour spécifier et préciser notre recherche ;
  - *fl=title_s, keyword_s, publicationDate_s, docType_s* : l'option *fl=champ1,champ2,...,champX* nous permet de spécifier les champs de données que nous souhaitons obtenir.

Dans cet exemple, les champs incluent le titre (title_s), les mots-clés (keyword_s), la date de publication (publicationDate_s) et le type de document (docType_s).

- *response*: c'est la variable dans laquelle la réponse de la requête *http* est stockée. Cette réponse contiendra des informations renvoyées par l'API, telles que des données au format *json* etc.

Cette ligne de code nous a envoyé une requête *get* à l'API en utilisant l'URL spécifiée avec les paramètres de recherche et les champs de données à inclure dans la réponse.

Ligne de code n°3 :

```
response =
requests.get("http://api.archives-ouvertes.fr/search/index/?q=Féminisme&rows=1
35&keyword_t=Histoire&wt=csv&rows=10&fl=title_s,publicationDate_s,keyword_s,do
cType_s")
as_csv =pd.read_csv(io.StringIO(response.text),sep=",")
tableau = as_csv
```

Pour que le résultat soit plus lisible, on peut demander les résultats en .csv et les afficher avec pandas avec ces commandes.
La ligne suivant la commande *requests.get* a déjà été explicitée précédemment.
Les deux autres lignes permettent de convertir

- *as_csv = pd.read_csv(io.StringIO(response.text), sep=",")*
  - *response.text* : cette commande nous permet de récupérer le contenu de la réponse *http* sous forme de texte ;
  - *io.StringIO(response.text)* : cette commande permet de créer un objet de type *StringIO* à partir du texte de la réponse et de traiter des chaînes de caractères comme des fichiers texte ;
  - *pd.read_csv(...)* : correspond à une fonction de la bibliothèque pandas qui lit le contenu d'un fichier CSV et le charge dans un DataFrame (une structure de données tabulaire de pandas) ;
  - *sep=", "* : cela permet de spécifier que les données CSV sont séparées par des virgules dans notre cas.
- *as_csv* : c'est le nom de la variable dans laquelle le contenu de la réponse au format CSV est stocké sous forme de DataFrame.

Cette ligne de code nous permet en somme de manipuler facilement les données de la réponse dans un tableau pour l'analyse ultérieure.

Ligne de code n°4 :

```
tableau.sort_values("publicationDate_s",ascending=False)
```

- Pour rendre la lecture encore plus facile, nous avons choisi d'organiser le tableau en fonction des dates (documents les plus récents aux plus anciens). La commande utilisée permet donc de trier les données d'un DataFrame en fonction des valeurs d'une colonne spécifiée. Description détaillée :
- *tableau* : cela correspond au nom du DataFrame sur lequel l'opération de tri est effectuée. Le DataFrame contient les données au format CSV extraites de la réponse de l'API.
- *.sort_values("publicationDate_s", ascending=False)* : cette méthode de pandas trie les données en fonction des valeurs de la colonne spécifiée que nous avons choisie, ici : *publicationDate_s*.
  - *publicationDate_s* : c'est donc le nom de la colonne selon laquelle les données sont triées (contenant les dates de publication des documents) ;
  - *ascending=False* : cela signifie que les données seront triées par ordre décroissant, ici du plus récent (valeur la plus élevée) au plus ancien (valeur la plus basse).

Cette ligne de code nous a bien permis de trier le DataFrame as_csv en fonction des valeurs de la colonne "publicationDate_s" dans un ordre décroissant.

Ligne de code n°5 :

```
def column_most_common(df,previous_name,new_name):
    def most_common(l):
        count = collections.Counter(l)
        most = count.most_common(1)[0][0]
        return most
 df[new_name]=df.apply(lambda row:
most_common(str(row[previous_name]).split(",")), axis=1)
```

- La fonction *column_most_common* nous a permis de créer une nouvelle colonne dans notre tableau, où chaque ligne contient la valeur la plus fréquente parmi les éléments de la colonne spécifiée. Cette valeur est calculée en utilisant la fonction interne *most_common*.

Ligne de code n°6 :

```
column_most_common(tableau,"domainAllCode_s","main_domain")
```

- Nous avons donc utilisé cette ligne de code pour créer la nouvelle colonne appelée "main_domain" dans notre tableau. Cette nouvelle colonne contiendra la valeur la plus fréquente parmi les éléments de la colonne "domainAllCode_s".

Ligne de code pour obtenir le graphique 1 :

```
grouped=as_csv.groupby("docType_s")["docType_s"].count()
print(grouped)
grouped.plot.pie(autopct="%.0f%%")
```

- Cette ligne de code nous a permis de regrouper les données de notre tableau as_csv par type de document, en comptant le nombre d'occurrences de chaque type, pour obtenir un diagramme circulaire illustrant la distribution des types de documents.

Ligne de code pour obtenir la figure 3 :

```
g1 = co_occurence_network (tableau,"keyword_s",threshold=2)
g1
```

Lignes de code pour obtenir la figure 4 :

- Tableau regroupant les mots-clefs et les dates de publication :

```
tableau2 = tableau.loc[:, ['publicationDate_s', 'keyword_s']]
```

- Tableau regroupant les documents contenant les mots-clefs “Médias” ou “Réseaux sociaux” ainsi que leur date de publication :

```
medias = tableau2[tableau2['keyword_s'].str.contains('Média', case=False,
na=False)]
reseau = tableau2[tableau2['keyword_s'].str.contains('Réseaux sociaux',
case=False, na=False)]
tableau3 = pd.merge(medias, reseau, how='outer')
```

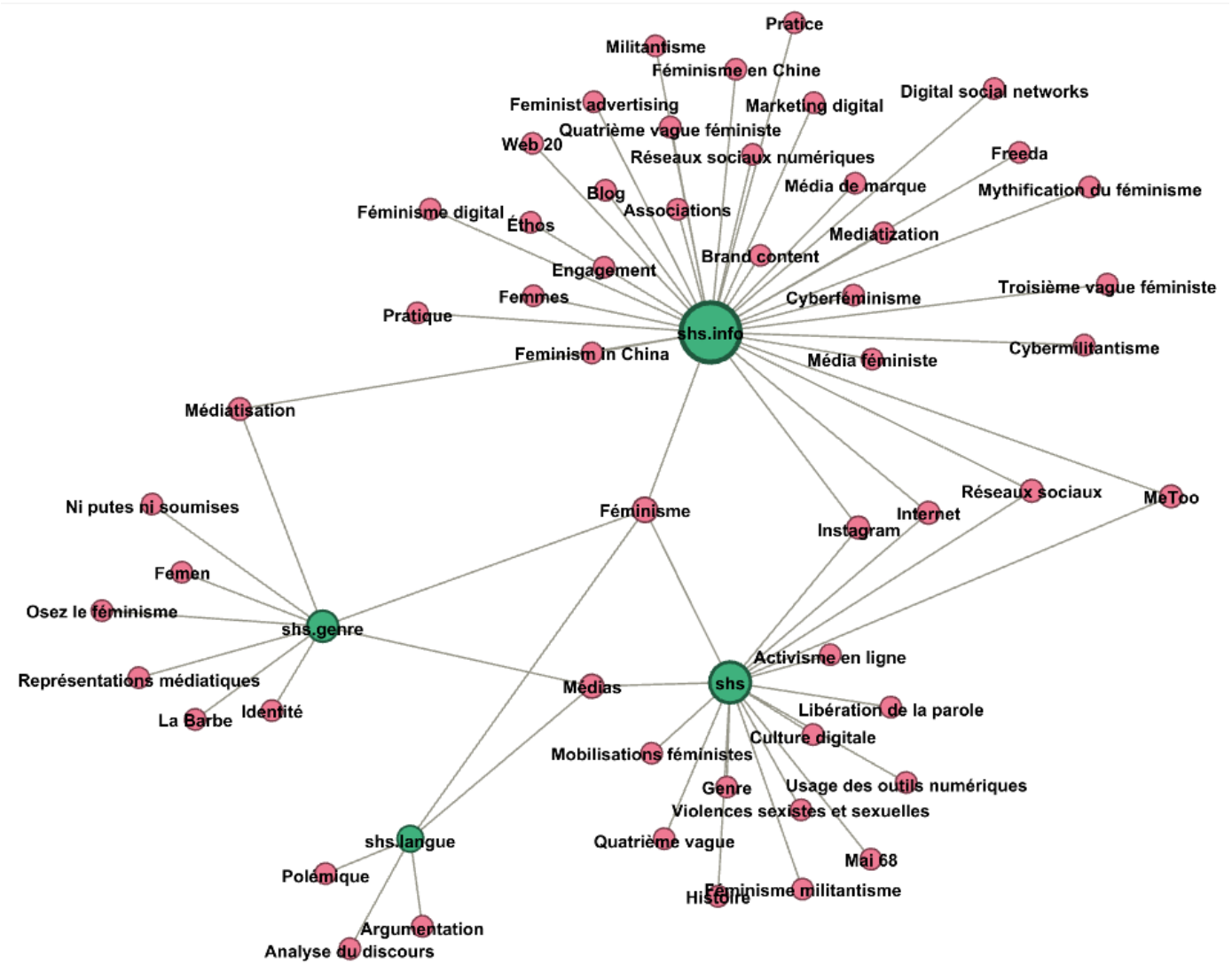
- Tableau regroupant les dates par année ainsi que leur mots-clefs:

```
tableau3['Valeur'] = list(range(len(tableau3)))
tableau3['publicationDate_s'] = pd.to_datetime(tableau3['publicationDate_s'])
tableau3['date'] = tableau3['publicationDate_s'].dt.year
```

```
tableau4 = tableau3.groupby('date')['keyword_s'].sum().reset_index()
```

- Visualisation du réseau mots-clefs en fonction des dates de publications :

```
g2 = bi_partite_network_generic(tableau3,"keyword_s","date")
```




*Autre Réseau bipartite mots-clés*/domaines de publications. Les nœuds en vert représentent aux domaines de publication, ceux en rose représentent les mots-clefs.*
**Ici les mots-clefs choisi étant : Médias et Réseaux sociaux*