Quantifying interactions between economics of climate change, the rest of economics and other disciplines

10/06/24

Preliminary data analysis

I. Authorship data: gender

```
# First load data: Corpus_Short
load(here(dir$prep.data,"Corpus_Short.Rdata"))

#10% of the original data : Corupus.Short.Small
Corpus.Short.Small <- Corpus.Short %>%
  slice_sample(n = 60000)
```

Step 1: Creation of new columns for each author and for each article

Separation of the column "AF" into several columns. New columns are entitled "author_" and goes from 1 to 15 (1 if there is only 1 author and 15 if there are 15 authors). We thus discard of the analysis paper with more than 15 authors.

```
# Creation of a new column "nb_authors" to count the number of authors per article

Corpus.Short$nb_authors <- str_count(Corpus.Short$AF, ";") + 1 #for the original data base
Corpus.Short.Small$nb_authors <- str_count(Corpus.Short.Small$AF, ";") + 1 #for the reduced data base

# Display the 1000 largest values in the "nb_authors" column to see the maximum number of authors in th

top_1000_max_authors <- head(sort(Corpus.Short$nb_authors, decreasing = TRUE), 1000)
print(top_1000_max_authors)
```

```
##    [1] 281  93  74  72  66  61  59  52  51  49  49  47  47  47  46  46  45  44
##   [19]  43  42  41  40  40  38  36  36  36  36  35  35  35  34  34  34  33  33
##   [37]  33  33  33  33  33  33  33  32  32  31  31  31  30  30  30  30  29  29
##   [55]  29  28  28  28  28  28  28  28  28  27  27  27  27  27  27  27  27  27  27
##   [73]  27  27  26  26  26  26  26  26  26  26  26  26  25  25  25  25  25  25  25
##   [91]  25  25  25  25  25  24  24  24  24  24  24  24  24  24  24  24  24  24  24
##  [109]  24  24  24  24  24  24  24  24  24  23  23  23  23  23  23  23  23  23
##  [127]  23  23  23  23  23  23  23  22  22  22  22  22  22  22  22  22  22  22  22
##  [145]  22  22  22  22  22  22  21  21  21  21  21  21  21  21  21  21  21  21
##  [163]  21  21  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20
##  [181]  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  19  19  19  19
##  [199]  19  19  19  19  19  19  19  19  19  19  19  19  19  19  19  19  19  19  19
##  [217]  19  19  19  19  19  19  19  19  18  18  18  18  18  18  18  18  18  18
```

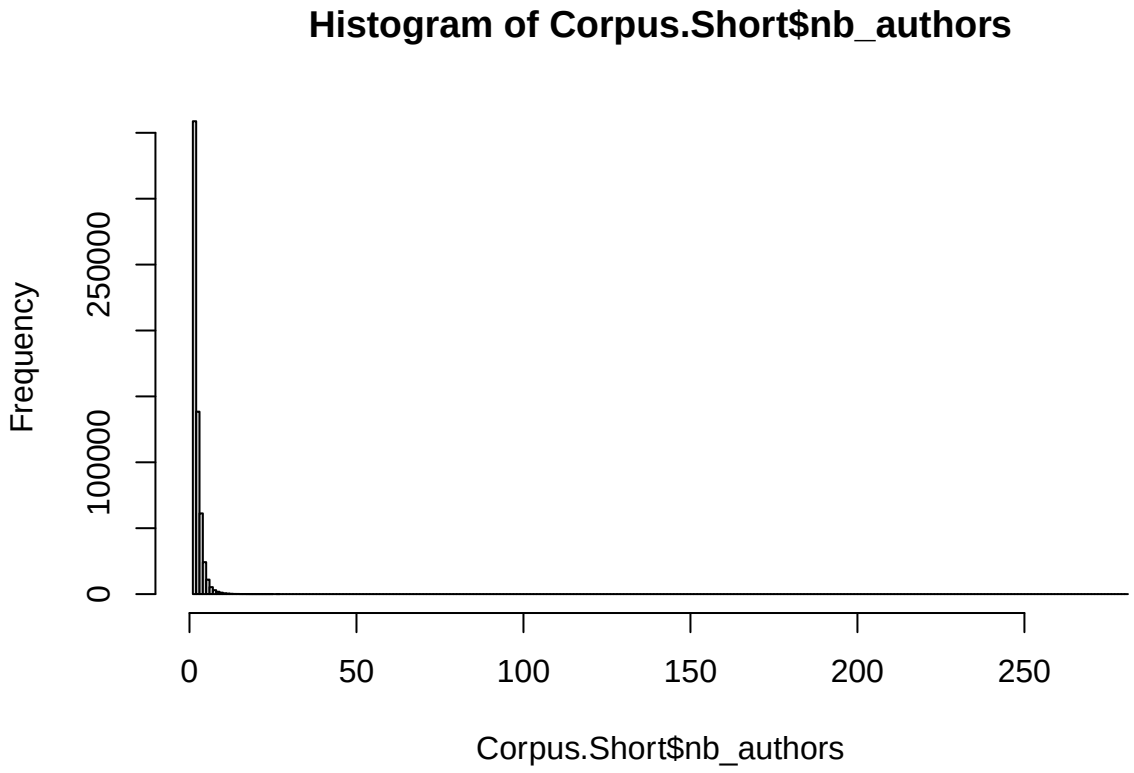
```
##  [235]  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18
##  [253]  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18  18  17  17
##  [271]  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17
##  [289]  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17
##  [307]  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17  17
##  [325]  17  17  17  17  17  16  16  16  16  16  16  16  16  16  16  16  16  16
##  [343]  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16
##  [361]  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16
##  [379]  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  16  15  15
##  [397]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [415]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [433]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [451]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [469]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [487]  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15  15
##  [505]  15  15  15  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [523]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [541]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [559]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [577]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [595]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [613]  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14  14
##  [631]  14  14  14  14  14  14  14  14  14  14  14  13  13  13  13  13  13  13
##  [649]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [667]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [685]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [703]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [721]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [739]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [757]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [775]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [793]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [811]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [829]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [847]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [865]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [883]  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13  13
##  [901]  13  13  13  13  12  12  12  12  12  12  12  12  12  12  12  12  12  12
##  [919]  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
##  [937]  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
##  [955]  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
##  [973]  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12  12
##  [991]  12  12  12  12  12  12  12  12  12  12
```

```
# Or alternatively:
table(Corpus.Short$nb_authors) # Number of articles by number of authors
```

```
##
##      1      2      3      4      5      6      7      8      9     10     11
## 172171 186621 138439  61243  24263  11009   5310   2986   1796   1115    655
##     12     13     14     15     16     17     18     19     20     21     22
##    440    263    134    113     65     61     44     30     30     14     17
##     23     24     25     26     27     28     29     30     31     32     33
##     16     22     12      9     12      7      3      4      3      2      9
```

```
##     34     35     36     38     40     41     42     43     44     45     46
##      3      3      4      1      2      1      1      1      1      1      2
##     47     49     51     52     59     61     66     72     74     93    281
##      3      2      1      1      1      1      1      1      1      1      1
```

```
hist(Corpus.Short$nb_authors, breaks = 281) # Same thing but as a histogram
```




```
sum(table(Corpus.Short$nb_authors)[1:15])/length(Corpus.Short$nb_authors) # Let's focus on articles wit
```

```
## [1] 0.9993509
```

```
# Creation of the columns for the authors : each column correspond to an additional author for each art

Corpus.Short.Small.Filt <- Corpus.Short.Small %>%
  filter(nb_authors < 16) %>% # Only keep articles with 15 authors at max (99.95% of the corpus)
  separate(col = AF, into = paste0("author_", 1:15), sep = ";", remove = FALSE, extra = "warn") # I've
```

```
## Warning: Expected 15 pieces. Missing pieces filled with `NA` in 59956 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
# Just to check the number of articles with exactly 15 authors (code below is commented, but possible t
# Corpus.Short.Small.Filt %>%
#   filter(nb_authors == 15) %>%
```

```
#   dim()

# Last check: number of articles by authors, but using the random sample of 60 000 articles:
table(Corpus.Short.Small.Filt$nb_authors)
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 17085 18395 13750  6022  2416  1021   514   304   180   122    74    48    14
##    14    15
##    11    12
```

Check to see if there are no missing values in one of the new column created : OK

```
results<- vector("logical", length = 15)

for (i in 1:15) {
  results[i] <- all(is.na(Corpus.Short.Small.Filt[[paste0("author_", i)]]))
}

print(results)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE
```

Step 2 : Create a column for each last name (l_name) and first name (f_name).

f_name will be used during all the analysis of the gender.

```
for (i in 1:15) {
  Corpus.Short.Small.Filt <- Corpus.Short.Small.Filt %>%
    separate(col = paste0("author_", i), into = c(paste0("l_name_", i), paste0("f_name_", i)), sep = ",
}
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 341 rows [46, 179, 240,
## 339, 459, 979, 1099, 1650, 1944, 2168, 2304, 2677, 2717, 2826, 3026, 3202,
## 3283, 3384, 3391, 3451, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 199 rows [739, 837, 954,
## 2168, 2246, 2304, 2407, 2608, 2661, 2809, 3511, 4016, 4192, 4219, 4235, 4433,
## 4870, 4871, 4910, 5177, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 142 rows [1513, 2039,
## 2304, 3616, 4771, 4870, 4915, 5185, 6183, 6375, 6530, 6620, 6849, 7585, 8641,
## 8705, 8905, 9353, 9454, 9552, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 77 rows [2304, 4870,
## 5185, 5666, 6169, 6849, 7497, 8003, 9360, 9552, 9737, 11137, 12439, 12747,
## 13475, 15888, 16668, 17778, 19720, 21790, ...].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 31 rows [4235, 4870,
## 6849, 8279, 9552, 13028, 23058, 23268, 24052, 26176, 29420, 31640, 31679,
## 33449, 33786, 35650, 36766, 36977, 39951, 40781, ...].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 19 rows [1035, 4870,
## 6849, 23165, 23268, 26790, 30020, 31640, 33786, 36766, 44276, 45746, 49190,
## 49231, 50665, 51645, 54391, 54563, 55670].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [14496,
## 48768].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 3 rows [20273, 28998,
## 31613].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [20273,
## 47651].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1 rows [46].
```

The warning message signals that some authors have no first name or last name. Ideed some authors don't have first names, that we need to keep in mind in the rest of the analyis. Two reasons for the absence of first name: some authors have no firstname in the data (they only mention one name, classified as last name, and no first name), and some authors have their names not separated with comma ",", thus not making possible to separate last name and first name. This is a mistake from the raw data from the bibliographic database, that seems to occur not that much. We can set this problem aside for the rest of the analysis.

```
# Check the case of an article with multiple missing authors (replace the number 27162 with one of the
Corpus.Short.Small.Filt %>%
  slice(27162) %>%
  select(matches("\\d$")) # To select columns whose name ends with a number (and therefore only view co
```

```
##                                  author_1        l_name_1
## 1 Koojaroenprasit, Sauwaluck (56732978300) Koojaroenprasit
##                   f_name_1 author_2 l_name_2 f_name_2 author_3 l_name_3
## 1  Sauwaluck (56732978300)     <NA>     <NA>     <NA>     <NA>     <NA>
##   f_name_3 author_4 l_name_4 f_name_4 author_5 l_name_5 f_name_5 author_6
## 1     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>
##   l_name_6 f_name_6 author_7 l_name_7 f_name_7 author_8 l_name_8 f_name_8
## 1     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>     <NA>
##   author_9 l_name_9 f_name_9 author_10 l_name_10 f_name_10 author_11 l_name_11
## 1     <NA>     <NA>     <NA>      <NA>      <NA>      <NA>      <NA>      <NA>
##   f_name_11 author_12 l_name_12 f_name_12 author_13 l_name_13 f_name_13
## 1      <NA>      <NA>      <NA>      <NA>      <NA>      <NA>      <NA>
##   author_14 l_name_14 f_name_14 author_15 l_name_15 f_name_15 JEL1 Top25 Top10
## 1      <NA>      <NA>      <NA>      <NA>      <NA>      <NA> <NA>     0     0
##   Top30
## 1     0
##                                                                          C1
## 1 Koojaroenprasit S., Department of Economics, Kasetsart University, Thailand
##     Z9   U1   U2   J9 Q54
## 1 <NA> <NA> <NA> <NA>  NA
```

Step 3 : Cleaning in two steps: - first step -> removing spaces and uniformize character encoding of first names and last names to then generate a column "fullname_i" which is made of first name and last name.

This column will be the one used to match with the list of female economists (which is made of female economists full names) - second step -> continue cleaning of the firstnames columns (remove special characters, punctuations and isolated letters). This will give us first names columns that are cleaned and ready to be used with the gender command

```
Corpus.CleanedNames <- Corpus.Short.Small.Filt

for (i in 1:15) {
  print(i)
  # Supprimer les textes entre parenthèses :
  # Pour le prénom
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- gsub(pattern = "\\s*\\([^\\)]+\\)",
                                                      replacement = "",
                                                      x = Corpus.CleanedNames[[paste0("f_name_", i)]])
  # Pour le nom
  Corpus.CleanedNames[[paste0("l_name_", i)]] <- gsub(pattern = "\\s*\\([^\\)]+\\)",
                                                      replacement = "",
                                                      x = Corpus.CleanedNames[[paste0("l_name_", i)]])

  # Supprimer les espaces supplémentaires :
  # Pour le prénom
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- gsub(pattern = "\\s+",
                                                      replacement = " ",
                                                      x =  Corpus.CleanedNames[[paste0("f_name_", i)]])
  # Pour le nom
  Corpus.CleanedNames[[paste0("l_name_", i)]] <- gsub(pattern = "\\s+",
                                                      replacement = " ",
                                                      x =  Corpus.CleanedNames[[paste0("l_name_", i)]])

  # Supprimer les espaces de début et de fin :
  # Pour le prénom
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- trimws(Corpus.CleanedNames[[paste0("f_name_", i)]])
  # Pour le nom
  Corpus.CleanedNames[[paste0("l_name_", i)]] <- trimws(Corpus.CleanedNames[[paste0("l_name_", i)]])

  # Remove all accents and uniformize character encoding
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- stri_trans_general(Corpus.CleanedNames[[paste0("f_name
                                                                    "Latin-ASCII")
  Corpus.CleanedNames[[paste0("l_name_", i)]] <- stri_trans_general(Corpus.CleanedNames[[paste0("l_name
                                                                      "Latin-ASCII")

  # Créer une colonne fullname_i qui combine firstname et lastname avec un espace:

  Corpus.CleanedNames <- Corpus.CleanedNames %>%
    mutate(!!paste0("fullname_", i) := ifelse(test = is.na(.[[paste0("f_name_", i)]]) & is.na(.[[paste0
                                          yes = NA,
                                          no = ifelse(
                                            test = is.na(.[[paste0("f_name_", i)]]),
                                            yes = .[[paste0("l_name_", i)]],
                                            no = ifelse(
```

```
                                                test = is.na(.[[paste0("l_name_", i)]]),
                                                yes = .[[paste0("f_name_", i)]],
                                                no = paste(.[[paste0("f_name_", i)]],
                                                           .[[paste0("l_name_", i)]],
                                                           sep = " ")
                                              )
                                            )
                                            )
          )


# Second step cleaning: nettoyer la colonne first name pour améliorer l'identification des prénoms avec
# Supprimer les lettres suivies d'un point :
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- gsub(pattern = "[A-Za-z]\\.",
                                                      replacement = "",
                                                      x =  Corpus.CleanedNames[[paste0("f_name_", i)]])

  # Remplacer les ponctuations par des espaces (exemple des noms composés) :
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- gsub(pattern = "[[:punct:]]",
                                                      replacement = " ",
                                                      x = Corpus.CleanedNames[[paste0("f_name_", i)]])

  # Supprimer les espaces supplémentaires (double espaces potentiellement dûs au remplacement de la pon
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- gsub(pattern = "\\s+",
                                                      replacement = " ",
                                                      x =  Corpus.CleanedNames[[paste0("f_name_", i)]])

  # Supprimer les espaces de début et de fin :
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- trimws(Corpus.CleanedNames[[paste0("f_name_", i)]])

  # If several first names only keep the first one:
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- map_chr(.x = Corpus.CleanedNames[[paste0("f_name_", i)]
                                                         .f =  ~ strsplit(.x, " ")[[1]][1])


  # Supprimer les espaces de début et de fin (à nouveau, au cas où la dernière opération en a introduit
  Corpus.CleanedNames[[paste0("f_name_", i)]] <- trimws(Corpus.CleanedNames[[paste0("f_name_", i)]])


}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
```

```
## [1] 13
## [1] 14
## [1] 15
```

```
# Check how it looks like
check <- Corpus.CleanedNames %>%
  slice(1:20) %>%
  select(matches("\\d$"))
```

Jean-Paul devient Jean Paul, les parethèses et les chiffres sont supprimées ainsi que les lettres suivies d'un point.

Step 4 : Dataframe for probabilities of gender according to first name

La méthode "ssa" utilise les noms de bébés de la US Census List des États-Unis. Elle se restreint donc seulement aux EU. La méthode "ipums" recherche les noms à partir des données du recensement américain Integrated Public Use Microdata Series. La méthode "napp" utilise les microdonnées de recensement du Canada, de la Grande-Bretagne, du Danemark, de l'Islande, de la Norvège et de la Suède de 1801 à 1910 créées par le North Atlantic Population Project. La méthode "kantrowitz" utilise le corpus Kantrowitz de noms masculins et féminins. La méthode "genderize" utilise l'API Genderize.io https://genderize.io/, qui est basée sur les "profils d'utilisateurs des principaux réseaux sociaux".

En revanche, je ne sais pas comment le package traite les prénoms qu'il ne connait pas. -> A priori: Il renvoie une valeur manquante / pas de valeur. Par exemple

```
  test1 = gender("Loic")
  test2 = gender("Jeanu")
```

Les différentes méthodes donnent bien des résultats différents. Chaque méthode ont des listes de noms différents, et entre méthode, un même prénom peut avoir une probabilité de genre différente.

Nous démarrons par assigner le genre au prénom des auteurs avec la méthode ssa qui semble la plus adaptée pour notre corpus (car issue de données américaines relativement récente, et les Etats-Unis sont composés d'un nombre important de cultures/civilisations, augmentant ainsi l'ensemble de prénoms possibles).

Puis nous utiliserons la méthode "napp".

```
#Avec une seule méthode (ssa par défaut)

#Définition de la fonction
get_gender_prob_ssa <- function(names)

{
  gender(names, method = "ssa")
}

#Application de la fonction get_gender_prob aux colonnes de Corpus.Short qui commencent par f_name

gender_proba_ssa <- lapply(Corpus.CleanedNames[, grepl("^f_name_", colnames(Corpus.CleanedNames))], get_

#Combiner les résultats du code précédent en un seul dataframe, en les ajoutant lignes par lignes

gender_proba_ssa_df <- do.call(rbind, gender_proba_ssa)
```

```r
#Garder les valeurs uniques

gender_proba_ssa_df_2 <- gender_proba_ssa_df %>%
  group_by(name) %>%
  slice(1)
```

```r
# Je propose d'améliorer l'approche: récupérer les cas uniques de prénoms par l'approche nap non repéré

# D'abord avec la méthode napp qui a l'avantage d'ajouter un bon nombre de prénoms "nordiques"
get_gender_prob_napp <- function(names)
{
  gender(names,
         method = c("napp"),
         countries = c("Canada", "United Kingdom", "Denmark", "Iceland","Norway", "Sweden")
         )
}

gender_proba_napp <- lapply(Corpus.CleanedNames[, grepl("^f_name_", colnames(Corpus.CleanedNames))], ge

gender_proba_napp_df <- do.call(rbind, gender_proba_napp)

gender_proba_napp_df_2 <- gender_proba_napp_df %>%
  group_by(name) %>%
  slice(1)

# Only keep names not in gender_proba_df_2 and add them:
gender_proba_napp_df_3 <- gender_proba_napp_df_2 %>%
  filter(!(name %in% gender_proba_ssa_df_2$name))

gender_proba_df <- gender_proba_ssa_df_2 %>%
  bind_rows(gender_proba_napp_df_3) %>% # Add them to the first names from ssa and put that in a new ob
  arrange(name)

# Puis tentative avec la méthode kantrowitz
get_gender_prob_kant <- function(names)
{
  gender(names,
         method = "kantrowitz"
         )
}

# Suite méthode kantrowitz, pour voir les prénoms possiblement ajoutés mais abandon car long et peu de
# Attention: la ligne de code suivante prend une 10aine de minutes et ne rajoute qu'une 20 aine de prén
# gender_proba_kant <- lapply(Corpus.CleanedNames[, grepl("^f_name_", colnames(Corpus.CleanedNames))],
#
# gender_proba_df_kant <- do.call(rbind, gender_proba_kant)
#
# gender_proba_df_kant <- gender_proba_df_kant %>%
#   group_by(name) %>%
#   slice(1)
#
# gender_proba_df_6 <- gender_proba_df_kant %>%
#   filter(!(name %in% gender_proba_df$name), !is.na(gender), gender %in% c("male","female")) %>%
```

```
#   mutate(proportion_male = ifelse(gender == "male", 1.0000, 0.0000),
#          proportion_female = ifelse(gender != "male", 1.0000, 0.0000))

# Objet final: tous les prénoms et leur proba de genre:
names_proba_df <- gender_proba_df
# %>%
#   bind_rows(gender_proba_df_6) %>%
#   arrange(name)
```

Problème corrigé: On a maintenant un annuaire de 10400 prénoms environ. Avant de faire l'algorithme d'appariement du genre au prénom, nous pouvons regarder la quantité de prénoms d'auteurs déifférents dans le corpus. Et la comparer à nos 10400 prénoms différents.

```
all_names_incorpus <- unlist(Corpus.CleanedNames[, grepl("^f_name_", colnames(Corpus.CleanedNames))]) %
  unique() # Vecteur qui contient tous les prénoms différents dans le corpus
length(all_names_incorpus)
```

```
## [1] 27273
```

Environ 27300 prénoms. Nous avons donc un peu plus d'1/3 des prénoms dont nous pouvons déterminer le genre à partir des méthodes SSA et NAPP, et 2/3 qui ne seront pas reconnus par cette méthode. Quels sont les noms que nous ne pourrons pas apparier, et combien y-en-a-t-il?

```
unmatched_names <- all_names_incorpus[!(all_names_incorpus %in% names_proba_df$name)]
length(unmatched_names)
```

```
## [1] 16910
```

```
unmatched_names[1:100]
```

```
##   [1] "Sedef"        "Rusni"        "SangHyun"     "Venkata"      "Youngchan"
##   [6] "Prabha"       NA             "Maiju"        "Xuehong"      "Mudasir"
##  [11] "Achmad"       "Asraul"       "Zhengguang"   "Lemeng"       "Guzel"
##  [16] "Hamedi"       "Manthos"      "Shigeki"      "Seo"          "Junjian"
##  [21] "Yoichiro"     "Girma"        "Hangsuck"     "Yunfei"       "Takuji"
##  [26] "Rufei"        "Guiwu"        "Gulshan"      "Ngonn"        "Anqi"
##  [31] "Jianhong"     "Pushpesh"     "Rati"         "Shalendra"    "Hua"
##  [36] "Baiyu"        "Rahmat"       "Yiting"       "Vivekananda"  "Hsiu"
##  [41] "Muaayed"      "Minquan"      "Yosra"        "Junfeng"      "Gongming"
##  [46] "Hurvashee"    "WenShwo"      "Mahour"       "Hakki"        "Gonenc"
##  [51] "Zhongqi"      "Shrabani"     "Cemil"        "Gaurab"       "Xingtang"
##  [56] "Yugu"         "Yangwen"      "Abdusalam"    "Olusesan"     "Anastasiia"
##  [61] "Mhamed"       "Volker"       "Ratnam"       "Yihong"       "Balogun"
##  [66] "Zhuo"         "Y"            "Piruna"       "Koresh"       "Xuanjuan"
##  [71] "Iraklis"      "Macartan"     "Yazhuo"       "Frane"        "Viacheslav"
##  [76] "Vinicios"     "Zayyana"      "Siyu"         "Wahyu"        "Yongqin"
##  [81] "Biola"        "Chukwumerije" "Kudakwashe"   "Sehwan"       "Qingmin"
##  [86] "Jingchao"     "Tetiana"      "Madhur"       "Jitendra"     "Mengyi"
##  [91] "Feilong"      "Nombulelo"    "Haonan"       "Jongsub"      "Akhand"
##  [96] "Yangjun"      "Xikai"        "Santanu"      "Dmitrii"      "Ruizhan"
```

Environ 17 000 prénoms que nous ne pourrons pas apparier. Beaucoup de prénoms qui ne semblent pas être issues de culture occidentales. Nous verrons plus tard qu'il s'agit aussi de prénoms très rare et qui reviennent peu régulièrement dans les articles. Sur les 60000 articles, ces prénoms reviennent finalement peu de fois, ce qui affecte assez peu l'appariement. En revanche, cela introduit un biais de sélection important: nous sommes incapable de genrer les auteurs avec des prénoms non occidentaux.

Dernière tentative d'augmenter la liste de prénoms et le genre associés. La méthode genderize du package gender. Est-il possible d'utiliser la méthode avec genderize? Il s'agit d'un essai ci-dessous, mais cela ne fonctionne pas: le site genderize refuse la nombre trop grand de requêtes qu'il faudrait faire

```
get_gender_prob_genderize <- function(names)
{
  gender(names,
         method = "genderize"
         )
}

# Initialize an empty vector to store the results
results <- tibble()

# Apply the function with a delay of 0.1 seconds between each application
for (i in 1:30) {
  print(i)
  result <- get_gender_prob_genderize(unmatched_names[i])
  results <- bind_rows(results, result)
  Sys.sleep(1)  # Pause for 0.01 seconds
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
```

```
## [1] 27
## [1] 28
## [1] 29
## [1] 30
```

L'API de genderize bloque après trop de requêtes.

Step 5 : Creation of an algorithm to assign first names to gender

We assign gender to a name using the following algorith: 1. Test If first name is missing (NA): If missing: no gender is assigned (NA) If not: 2. test if first name is among the list of common chinese first names (for construction of `common_chinese_namessee` Rscript) If first name present: gender is unknown If not present: 3. test if the full name is in the list of the top10% of fermale economists If full name is present: gender is female If not present: 4. test if first name is missing in the list of first name with gender probability (`names_proba_df$name`): If absent: no gender is assigned (NA) If not: assign gender according to probability threshold. Rule is the following: If probability of a male first name is larger than 0.9, gender is male If probability of a male first name is smaller than 0.1, gender is female If probability of a male first name is between 0.1 and 0.9, then gender is unknown.

We code it as follows:

```
# Garder uniquement les colonnes qui nous intéressent dans le dataframe gender_proba_df_2
# Je propose de garder toutes les variables pour le moment
head(names_proba_df)
```

```
## # A tibble: 6 x 6
## # Groups:   name [6]
##   name      proportion_male proportion_female gender year_min year_max
##   <chr>               <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 AL                  0.987            0.0132 male       1932     2012
## 2 Aadil               1                0      male       1932     2012
## 3 Aaditya             1                0      male       1932     2012
## 4 Aadne               1                0      male       1758     1910
## 5 Aagoth              0                1      female     1758     1910
## 6 Aakanksha           0                1      female     1932     2012
```

```
# Récupérer les noms des femmes économistes et des prénoms chinois communs
load(here(dir$prep.data, "FemaleEconomists_Names.Rdata"))
load(here(dir$prep.data, "Common_Chinese_Names.Rdata"))
load(here(dir$prep.data, "MaleEconomists_Names.Rdata"))

common_chinese_names <- common_chinese_names[common_chinese_names!="Juan"]

Corpus.CleanedNames.2 <- Corpus.CleanedNames
# Algorithm that create a gender variable. But it also adds for all authors their probability of male f
for(i in 1:15) {
  print(i)
  Corpus.CleanedNames.2 <- Corpus.CleanedNames.2 %>%
    # First create variable gender (takes value: NA, unknown, male and female)
    mutate(
      !!paste0("gender_", i) := ifelse(
        test = is.na(.[[paste0("f_name_", i)]]), # 1. if first name is missing in Corpus.CleanedNames.2
        yes = NA,
```

```
      no = ifelse(
        test = .[[paste0("f_name_", i)]] %in% common_chinese_names, # 2. if first name is in the list
        yes = "unknown",
        no = ifelse(
          test = .[[paste0("fullname_", i)]] %in% top10_female_economists, # 3. If full name is in th
          yes = "female",
          no = ifelse(
            test = .[[paste0("fullname_", i)]] %in% top10_male_economists, # 4. If full name is in th
            yes = "male",
            no = ifelse(
              test = !(.[[paste0("f_name_", i)]] %in% names_proba_df$name), # 5. If first name is abs
              yes = NA,
              no = case_when(
                names_proba_df$proportion_male[match(.[[paste0("f_name_", i)]], names_proba_df$name)]
                names_proba_df$proportion_male[match(.[[paste0("f_name_", i)]], names_proba_df$name)]
                names_proba_df$proportion_male[match(.[[paste0("f_name_", i)]], names_proba_df$name)]
              )
            )
          )
        )
      )
    )
  ) %>%
  mutate(
    !!paste0("proportion_male_", i) := ifelse(
      test = is.na(.[[paste0("f_name_", i)]]), # Create the variable that gives the probability that
      yes = NA,
      no = ifelse(
        test = .[[paste0("f_name_", i)]] %in% common_chinese_names,
        yes = NA,
        no = ifelse(
          test = .[[paste0("fullname_", i)]] %in% top10_female_economists,
          yes = 0,
          no = ifelse(
            test = .[[paste0("fullname_", i)]] %in% top10_male_economists,
            yes = 1,
            no = ifelse(
              test = !(.[[paste0("f_name_", i)]] %in% names_proba_df$name),
              yes = NA,
              no = ifelse(
                test = .[[paste0("gender_", i)]] %in% c("male", "female"),
                yes = names_proba_df$proportion_male[match(.[[paste0("f_name_", i)]], names_proba_df$
                no = NA
              )
            )
          )
        )
      )
    ),
    !!paste0("proportion_female_", i) := ifelse(
      test = is.na(.[[paste0("f_name_", i)]]), # Create the variable that gives the probability that
      yes = NA,
      no = ifelse(
```

```
          test = .[[paste0("f_name_", i)]] %in% common_chinese_names,
          yes = NA,
          no = ifelse(
            test = .[[paste0("fullname_", i)]] %in% top10_female_economists,
            yes = 1,
            no = ifelse(
              test = .[[paste0("fullname_", i)]] %in% top10_male_economists,
              yes = 0,
              no = ifelse(
                test = !(.[[paste0("f_name_", i)]] %in% names_proba_df$name),
                yes = NA,
                no = ifelse(
                  test = .[[paste0("gender_", i)]] %in% c("male", "female"),
                  yes = names_proba_df$proportion_female[match(.[[paste0("f_name_", i)]], names_proba_d
                  no = NA
                )
              )
            )
          )
        )
      ),
      nb_authors_gendered = rowSums(!is.na(select(., starts_with("proportion_male")))), # Count the num
      ratio_identified_gender = nb_authors_gendered / nb_authors # Among authors, the proportion of aut
    )
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
```

```
# Check results
check <- Corpus.CleanedNames.2 %>%
  slice(1:30) %>%
  select(matches("\\d$"), nb_authors_gendered, nb_authors)

# Old code (can be removed later if needed)
# determine_gender <- function(proportion_male) {
#   if (is.na(proportion_male)) {
#     return(NA)  # Si la proportion est NA, retourner NA
#   } else if (proportion_male > 0.90) {
#     return(1)  # Si la proportion de probabilité est > 90%, on considère que c'est un homme
```

```
#   } else if (proportion_male <= 0.10) {
#     return(0)  # Si la proportion de probabilité est <= 10%, on considère que c'est une femme
#   } else {
#     return("Unknown gender")  # Si la proportion est entre 0.1 et 0.9, le genre est inconnu
#   }
# }

# Détermination du genre (nouvelle colonne gender_i) pour chaque colonne f_name_i
# for (i in 1:10)
#   {
#   f_name_col <- paste0("f_name_", i)
#   gender_col <- paste0("gender_", i)

#Création de deux nouvelles colonnes proportion_male (probabilité que le nom soit celui d'un homme) et

# Corpus.Short <- Corpus.Short %>%
#     left_join(gender_proba_df_2, by = setNames("name", f_name_col)) %>%
#     mutate(!!gender_col := sapply(proportion_male, determine_gender)) %>%
#     select(-proportion_male)
# }
```

Step 6 : Vérification de l'algorithme, pour chaque auteur unique

Un auteur unique, est une valeur unique de la variable "author_i" pour tout i allant de 1 à 15.

Regardons à quel point l'algorithme est capable d'assigner un genre aux auteurs par article:

```
ratio_by_nbauthours <- Corpus.CleanedNames.2 %>%
  group_by(nb_authors) %>%
  summarize(mean_ratio = mean(ratio_identified_gender, na.rm = T),
            med_ratio = median(ratio_identified_gender),
            nb_article = n())
```

Regardons à quel point l'algorithme est capable d'assigner un genre à tous les full names d'auteurs:

```
# Store in vectors all columns of information of  authors for i from 1 to 15 : empiler les colonnes - a
all_firstnames_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^f_name_", colnames(Corpus.CleanedName
all_lastnames_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^l_name_", colnames(Corpus.CleanedNames
all_fullnames_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^fullname_", colnames(Corpus.CleanedNam
all_fullnames_incorpus2 <- unlist(Corpus.CleanedNames.2[, grepl("^author_", colnames(Corpus.CleanedName
all_gender_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^gender_", colnames(Corpus.CleanedNames.2)
all_proportionmale_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^proportion_male_", colnames(Corpus
all_proportionfemale_incorpus <- unlist(Corpus.CleanedNames.2[, grepl("^proportion_female_", colnames(C

authors_df <- tibble(firstname = all_firstnames_incorpus, #base de données générale pour les auteurs
                     lastname = all_lastnames_incorpus,
                     fullname = all_fullnames_incorpus, # Clean full name
                     fullname2 = all_fullnames_incorpus2, # Raw full name
                     gender = all_gender_incorpus,
                     proportion_male = all_proportionmale_incorpus,
                     proportion_female = all_proportionfemale_incorpus) %>%
  # filter(!is.na(firstname)) %>%
```

```
  filter(!is.na(fullname2)) %>% # Only keep existing authors
  #mutate(fullname = trimws(fullname)) %>%
  group_by(firstname, lastname) %>% # Consider a row full name to be a unique author (our algorithm wou
  mutate(n_articles = n()) %>%  # Number of articles for the authors with this full name
  slice(1) %>% #valeurs uniues
  arrange(fullname2) %>%
  ungroup()

length(authors_df$fullname2)
```

```
## [1] 110788
```

111 112 different authors for 60 000 articles.

```
table(is.na(authors_df$gender))
```

```
##
## FALSE  TRUE
## 80578 30210
```

```
# Check authors that have written more than 10 articles:
authors_with10articlesormore <- authors_df %>%
  filter(n_articles>9)
# Il s'agit en majorité de noms issus d'asie de l'est

# By number of articles written for each author, how much missing gender?
table(authors_df$n_articles, is.na(authors_df$gender)) #False : auteurs bien assignés
```

```
##
##       FALSE  TRUE
##    1  63648 25774
##    2  10665  3146
##    3   3419   760
##    4   1323   279
##    5    624   141
##    6    329    43
##    7    168    21
##    8    131    15
##    9     73     8
##    10    57     8
##    11    35     3
##    12    22     7
##    13    15     0
##    14    18     1
##    15    10     0
##    16     9     1
##    17    10     1
##    18     6     0
##    19     4     1
##    20     0     1
##    21     3     0
```

```
##    22      1      0
##    23      1      0
##    27      2      0
##    28      2      0
##    32      1      0
##    35      1      0
##    47      1      0
```

Overall we have about 25% of authors with unidentified gender (30 000/80 000). When we will work at the level of the article, we thus will reduce the amount of articles with unassigned gender proportion of authorship as there are articles with multiple authors (and among them, at least some will have an assigned gender). The more the number of authors is important, the higher the prob of assigning gender to the article is higher

For instance, among authors that have written 1 article, about 76 000 have an assigned gender, and 29 000 have a missing info on gender (about 25% of missing gender). La ratio diminue jusqu'aux auteurs avec 6 articles (15% de genre non assignés), puis ensuite le fait de travailler sur un échantillon aléatoire du corpus semble brouill l'information. Mais il semble qu'il y ait des auteurs importants (avec plus de 10 articles), dont le genre n'est pas assigné (souvent des auteurs d'asie de l'est).

Depuis l'aout de top_10_male_economists, les auteurs principaux (qui écrivent beaucoup d'articles) sont correctement assgnés. Rq : disparition de l'auteur qui avait écrit 54 articles.

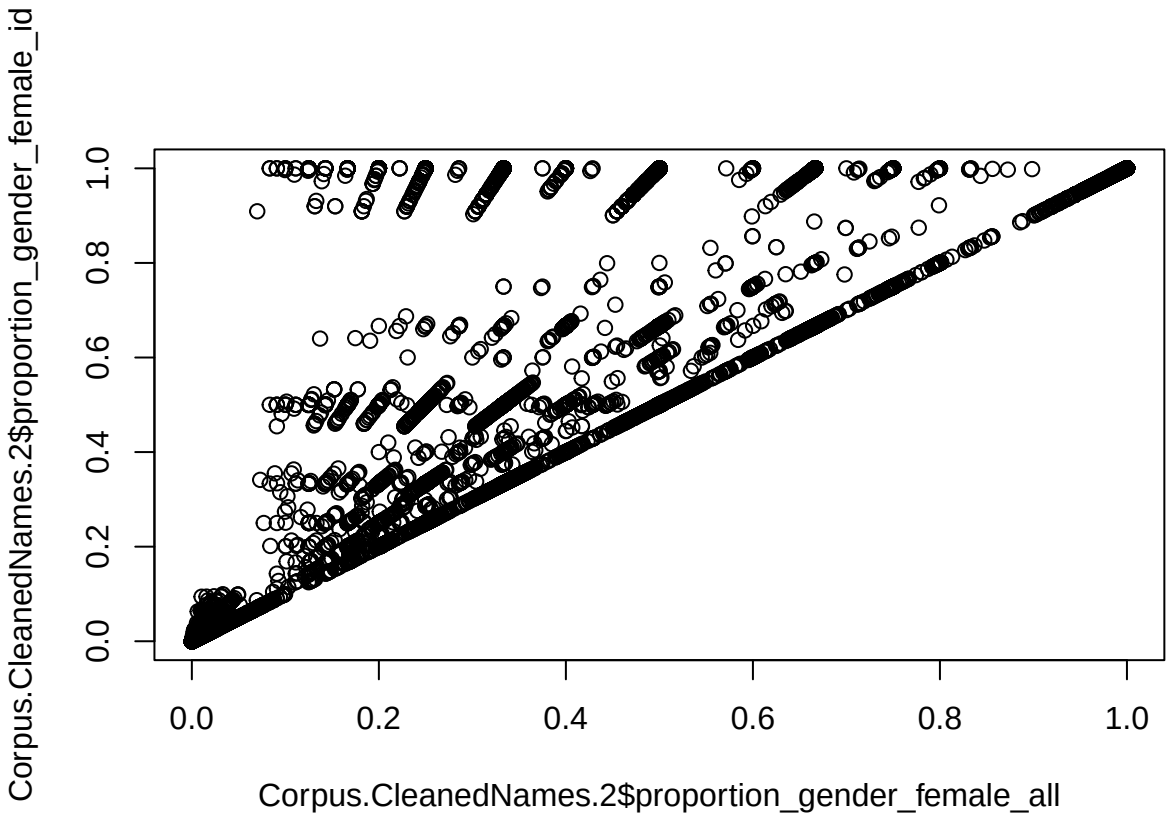
Step 7 : Agregation at the level of the article

```
# Je corrige pour aggréger avec les colonnes proportion_male et proportion_female plutôt que les colonn
# Identifier les colonnes gender_
proportion_male_cols <- grep("^proportion_male_", colnames(Corpus.CleanedNames.2), value = TRUE)
proportion_female_cols <- grep("^proportion_female_", colnames(Corpus.CleanedNames.2), value = TRUE)

# Calculer la somme des genres masculins puis féminin pour chaque ligne
Corpus.CleanedNames.2$sum_gender_male <- rowSums(Corpus.CleanedNames.2[proportion_male_cols], na.rm = TR
Corpus.CleanedNames.2$sum_gender_female <- rowSums(Corpus.CleanedNames.2[proportion_female_cols], na.rm

# Calculer la proportion de genres masculins et féminin sur tous les auteurs, puis sur les auteurs au g
# Homme
Corpus.CleanedNames.2$proportion_gender_male_all <- Corpus.CleanedNames.2$sum_gender_male / Corpus.Clean
Corpus.CleanedNames.2$proportion_gender_male_id <- Corpus.CleanedNames.2$sum_gender_male / Corpus.Cleane
# Femme
Corpus.CleanedNames.2$proportion_gender_female_all <- Corpus.CleanedNames.2$sum_gender_female / Corpus.C
Corpus.CleanedNames.2$proportion_gender_female_id <- Corpus.CleanedNames.2$sum_gender_female / Corpus.CL
```

Test : compter le nombre de données non manquantes dans toutes les colonnes gender (nombre d'auteurs qui ont été assignés donc qui prennent la valeur 0 ou 1) et comparaison avec le nombre total d'auteur

Nous pouvons faire une analyse similaire en comparant le pourcentage de genre d'un article sur tous les auteurs, avec le pourcentage de genre d'un article sur tous les auteurs identifiés: comparaison des ratios selon si l'on prend tous les auteurs assignés par article ou tous les auteurs

```
plot(Corpus.CleanedNames.2$proportion_gender_female_all, Corpus.CleanedNames.2$proportion_gender_female_
```



Sur la ligne à 45 degrés, nous avons les articles pour lesquels tous les auteurs ont un genre qui est identifié. Et au dessus de la ligne à 45 degrés, les articles pour lesquels certains auteurs n'ont pas de genre identifiés.

Analyse de l'assignation du genre à l'échelle de l'article :

Il est aussi possible de faire l'analyse à partir de la variable "ratio_identified_gender", qui donne par article, le pourcentage d'auteurs dans le nb d'auteurs totale dont le genre est identifié

```
Gender_of_Articles <- Corpus.CleanedNames.2 %>%
  summarize(Nb_articles = length(ratio_identified_gender), # Nb total d'article
            NoGender = length(ratio_identified_gender[is.na(ratio_identified_gender) | ratio_identified
            AllGender = length(ratio_identified_gender[ratio_identified_gender == 1])/Nb_articles, # Ra
            Gender_Larger0.5 = length(ratio_identified_gender[ratio_identified_gender >= 0.5])/Nb_artic
            Gender_Larger0.9 = length(ratio_identified_gender[ratio_identified_gender >= 0.9])/Nb_artic
            Gender_Larger0.25 = length(ratio_identified_gender[ratio_identified_gender >= 0.25])/Nb_arti
            Mean_IdentifiedGender = mean(ratio_identified_gender), # Moyenne du ratio d'auteurs identif
            Median_identifiedGender = median(ratio_identified_gender))# Médiane

Gender_of_Articles
```

```
##   Nb_articles  NoGender AllGender Gender_Larger0.5 Gender_Larger0.9
## 1       59968 0.1566169 0.5495097         0.774313        0.5503435
##   Gender_Larger0.25 Mean_IdentifiedGender Median_identifiedGender
## 1         0.8345784             0.7067713                       1
```

Nous voyons que pour environ 15% des articles, il nous est impossible d'assigner le genre à l'un des auteurs. Pour plus de 50% des articles nous sommes capable d'identifer le genre de tous les auteurs. Pour 75% des

articles nous sommes capable d'assigner la moitié des auteurs. Pour la moitié des articles nous sommes capable d'assigner le genre de 90% des auteurs. Pour environ 80% des articles, nous pouvons assigner le genre de 25% des auteurs.

Step 8 : Analyse de l'assignation du genre à l'échelle des auteurs :

```
# Check 100 first authors without gender and most articles
NotGendered_authors <- authors_df %>%
  filter(is.na(gender)) %>%
  arrange(firstname)

NotGendered_authors #30 454 auteurs qui n'ont pas été assignés sur 111 116 auteurs différents (27% d'au
```

```
## # A tibble: 30,210 x 8
##    firstname lastname   fullname          fulln~1 gender propo~2 propo~3 n_art~4
##    <chr>     <chr>      <chr>             <chr>   <chr>    <dbl>   <dbl>   <int>
##  1 A         Choi       A-Young Choi      " Choi~ <NA>        NA      NA       1
##  2 A         Godbole    C.A Swati Godbole " Godb~ <NA>        NA      NA       1
##  3 A         Oumlil     A Ben Oumlil      "Oumli~ <NA>        NA      NA       1
##  4 Aa        Nazari     Aa Nazari         "Nazar~ <NA>        NA      NA       1
##  5 Aabgeena  Naeem      Aabgeena Naeem    " Naee~ <NA>        NA      NA       1
##  6 Aadhaar   Chaturvedi Aadhaar Chaturve~ " Chat~ <NA>        NA      NA       1
##  7 Aaheli    Ahmed      Aaheli Ahmed      "Ahmed~ <NA>        NA      NA       1
##  8 Aakil     Caunhye    Aakil M. Caunhye  " Caun~ <NA>        NA      NA       1
##  9 Aam       Bastaman   Aam Bastaman      "Basta~ <NA>        NA      NA       1
## 10 Aam       Rusydiana  Aam S. Rusydiana  "Rusyd~ <NA>        NA      NA       1
## # ... with 30,200 more rows, and abbreviated variable names 1: fullname2,
## #   2: proportion_male, 3: proportion_female, 4: n_articles
```

Step 9 : Creation of categorical variables :

```
#Création de variables binaires : sum_gender_male_bin est défini comme 1 si sum_gender_male est supérie
Corpus.CleanedNames.2 <- Corpus.CleanedNames.2 %>%
  mutate(
    sum_gender_male_bin = ifelse(sum_gender_male > 0, 1, 0),
    sum_gender_female_bin = ifelse(sum_gender_female > 0, 1, 0)
  )

#Création de la variable catégorielle category_gender
Corpus.CleanedNames.2 <- Corpus.CleanedNames.2 %>%
  mutate(
    category_gender = case_when(
      sum_gender_female_bin == 0 ~ "MM",
      sum_gender_male_bin == 0 ~ "FF",
      sum_gender_male_bin == 1 & sum_gender_female_bin == 1 ~ "MF"
    )
  )

# Vérification des résultats
table(Corpus.CleanedNames.2$category_gender)
```

```
##
##    FF    MF    MM
##  3520 35902 20546
```

Step 10 : Descriptive Statistics

First step : determine the absolute number of articles with at least 1 author identified in terms of gender with NoGender

```
OneGender_by_Articles <- Corpus.CleanedNames.2 %>%
  summarize(Nb_articles = length(ratio_identified_gender), # Nb total d'article
            NoGender = length(ratio_identified_gender[is.na(ratio_identified_gender) | ratio_identified_

OneGender_by_Articles
```

```
##   Nb_articles  NoGender
## 1       59968 0.1566169
```

Il y a au moins 85% des articles dont le genre a été assigné à au moins un auteur.

1. Study of the sum_gender_female variable at an aggregate level

```
library(ggplot2)
```

1.1. Proportion of male and female authors

1.1.1. Descriptive statitistics : Proportion of male and female authors at an aggregate level

```
#amélioration de ton code pour faire un tableau unique
# Charger les packages nécessaires
library(dplyr)
library(knitr)

# Calcul des statistiques descriptives
stats_proportion_female_all <- summary(Corpus.CleanedNames.2$proportion_gender_female_all)
stats_proportion_female_id <- summary(Corpus.CleanedNames.2$proportion_gender_female_id)
stats_sum_gender_female <- summary(Corpus.CleanedNames.2$sum_gender_female)
stats_sum_gender_male <- summary(Corpus.CleanedNames.2$sum_gender_male)
stats_proportion_male_all <- summary(Corpus.CleanedNames.2$proportion_gender_male_all)
stats_proportion_male_id <- summary(Corpus.CleanedNames.2$proportion_gender_male_id)

# Extraire les valeurs des statistiques descriptives
extract_summary <- function(summary_obj) {
  c(summary_obj["Min."], summary_obj["1st Qu."], summary_obj["Median"], summary_obj["Mean"], summary_ob
}

# Créer un tableau de données avec les valeurs extraites
```

```r
results_table <- data.frame(
  Statistic = c("Min", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max"),
  Proportion_Female_All = extract_summary(stats_proportion_female_all),
  Proportion_Female_Id = extract_summary(stats_proportion_female_id),
  Sum_Gender_Female = extract_summary(stats_sum_gender_female),
  Sum_Gender_Male = extract_summary(stats_sum_gender_male),
  Proportion_Male_All = extract_summary(stats_proportion_male_all),
  Proportion_Male_Id = extract_summary(stats_proportion_male_id)
)

results_table
```

```
##         Statistic Proportion_Female_All Proportion_Female_Id Sum_Gender_Female
## Min.          Min             0.0000000          0.000000000         0.0000000
## 1st Qu.   1st Qu.             0.0000000          0.001466667         0.0000000
## Median     Median             0.0039000          0.007700000         0.0065000
## Mean         Mean             0.2040620          0.290629812         0.5100596
## 3rd Qu.   3rd Qu.             0.3353667          0.501000000         1.0000000
## Max.          Max             1.0000000          1.000000000        11.9845000
##         Sum_Gender_Male Proportion_Male_All Proportion_Male_Id
## Min.            0.00000           0.0000000          0.0000000
## 1st Qu.         0.00310           0.0022000          0.4990000
## Median          1.00000           0.5000000          0.9923000
## Mean            1.16485           0.5027182          0.7093846
## 3rd Qu.         1.99230           0.9947000          0.9985333
## Max.           11.95790           1.0000000          1.0000000
```

En moyenne, parmi tous les auteurs, il y a 20% de femmes par article. La part d'auteur femme en moyenne est de 30% (en mesurant la part parmi les auteurs dont le genre est identifié, par article).

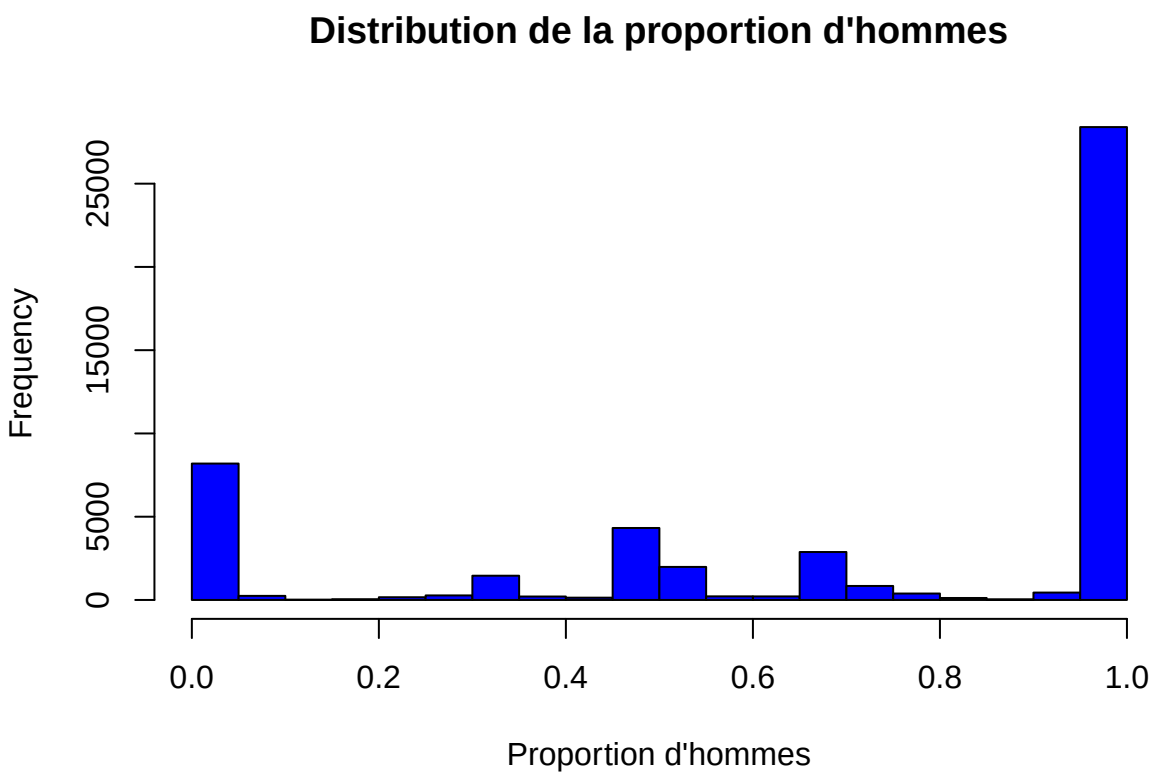
Le summary avec female nous dit qu'il y a 0.5 femmes auteurs par article. La médiane est intéressante: il y a 50% des articles qui sont rédigés sans auteur femme. Et le 3e quartile nous dit que 25% des articles ont une auteur qui est une femme.

Le summary avec male nous dit qu'il y a 1.1 hommes auteurs par article. La médiane nous dit qu'au moins 50% des articles ont un auteur homme. Et Le 3e quartile nous dit que 25% des articles ont deux auteurs hommes.

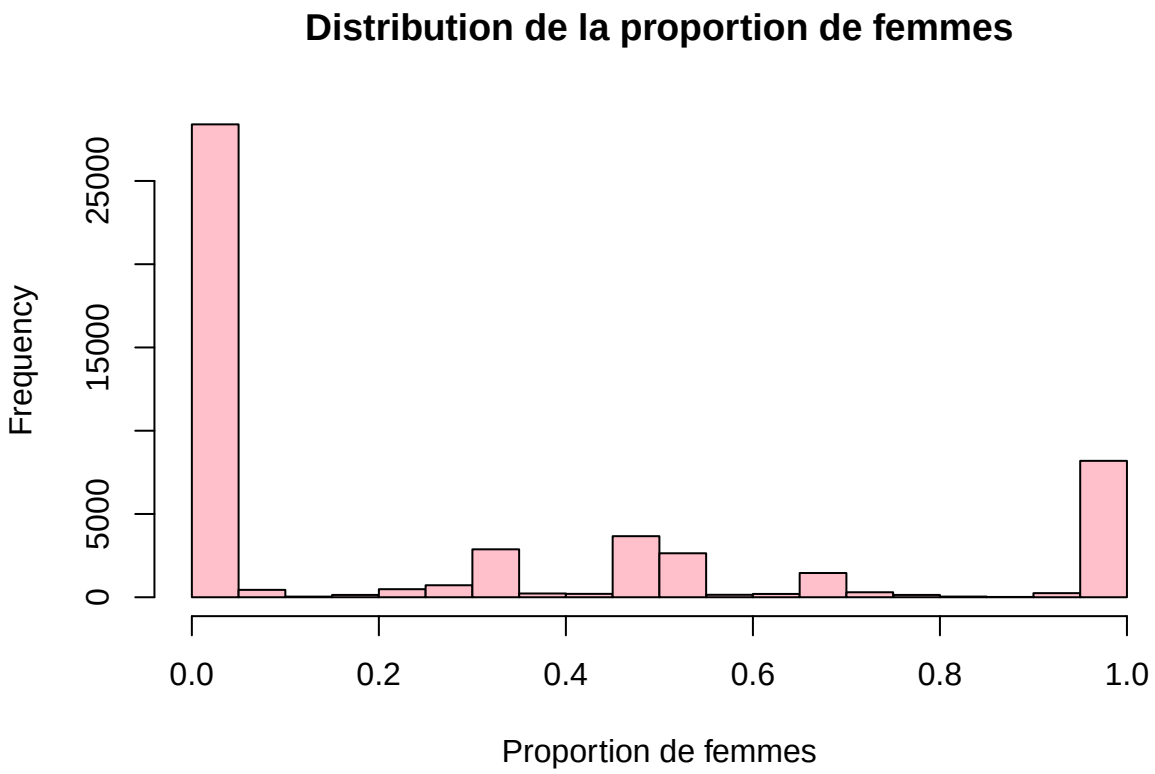
En moyenne 30% et 70% de femmes parmi les auteurs dont le genre a été identifié.

1.1.2. Graphics : Proportion of male and female authors at an aggregate level

```
hist(Corpus.CleanedNames.2$proportion_gender_male_id, main="Distribution de la proportion d'hommes", xla
```




```
hist(Corpus.CleanedNames.2$proportion_gender_female_id, main="Distribution de la proportion de femmes",
```



- Distribution de femmes : la distribution est centrée vers la gauche donc beaucoup d'articles avec aucune femme (+ de 2500) avec très peu de valeurs au milieu de la distirbution et moins d'articles avec seulement des femmes (environ 10 000) - Distribution d'hommes : la distribution est centrée vers la droite donc beaucoup d'articles que des hommes (+ de 2500) avec très peu de valeurs au milieu de la distirbution et moins d'articles avec aucun homme (environ 8 000) - Peu de collaboration car peu de valeurs centrées

```
#Distribution of sum_gender_male : bar chart

#ggplot(data = Corpus.Short, mapping = aes(x = sum_gender_male)) + geom_bar() + labs(title = "Distribut

##Distribution of sum_gender_female : bar chart

#ggplot(data = Corpus.Short, mapping = aes(x = sum_gender_female)) + geom_bar() + labs(title = "Distrib

#Distribution of sum_gender_male : density graph
#ggplot(Corpus.Short, aes(x = sum_gender_male)) + geom_density(linewidth = 0.75) + labs( title = "Distr

#ggplot(Corpus.Short, aes(x = sum_gender_female)) + geom_density(linewidth = 0.75) + labs(title = "Dist
```

1.2. Proportion of female and male authors according to categorical variables

1.2.1 Descriptive statistics: Proportion of female and male authors according to categorical variables

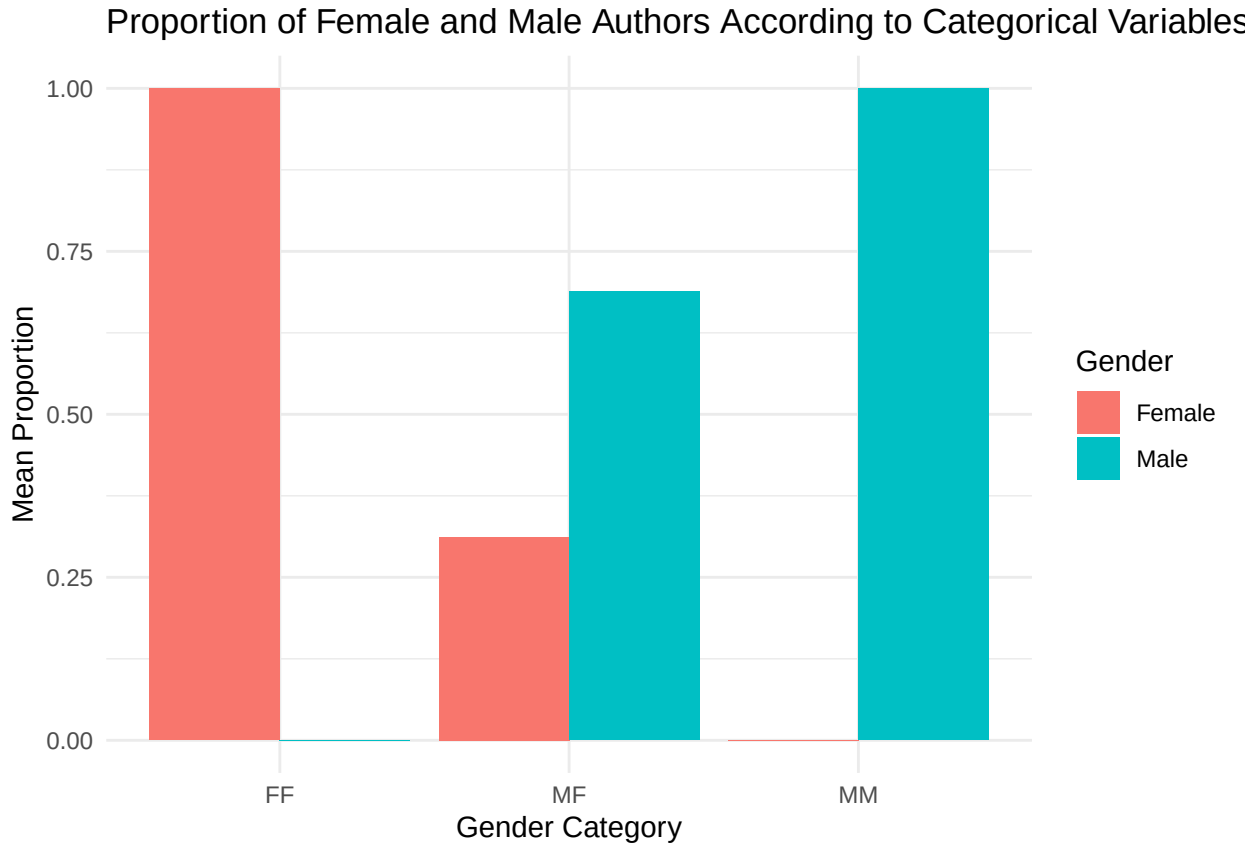
```r
stats_by_category <- Corpus.CleanedNames.2 %>%
  group_by(category_gender) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    Median_Proportion_Female_Id = median(proportion_gender_female_id, na.rm = TRUE),
    Mean_Proportion_Male_Id = mean(proportion_gender_male_id, na.rm = TRUE),
    Median_Proportion_Male_Id = median(proportion_gender_male_id, na.rm = TRUE),
  )
stats_by_category
```

```
## # A tibble: 3 x 5
##   category_gender Mean_Proportion_Female_Id Median_Proportion_~1 Mean_~2 Media~3
##   <chr>                              <dbl>                <dbl>   <dbl>   <dbl>
## 1 FF                                 1                    1       0       0
## 2 MF                                 0.311                0.202   0.689   0.798
## 3 MM                                 0                    0       1       1
## # ... with abbreviated variable names 1: Median_Proportion_Female_Id,
## #   2: Mean_Proportion_Male_Id, 3: Median_Proportion_Male_Id
```

- Catégorie FF : moyenne et médiane de 100% de femmes (logique)
- Catégorie MM : moyenne et médiane de 100% d'hommes (logique)
- Catégorie MF : 30% de femmes en moyenne pour les articles collaboratifs et 70% d'hommes

1.2.2 Graphics : Proportion of female and male authors according to categorical variables

```r
stats_long <- stats_by_category %>%
  pivot_longer(cols = c(Mean_Proportion_Female_Id, Mean_Proportion_Male_Id), #colonne que l'on veut met
               names_to = "Gender", #nouvelle colonne intitulée "Gender"
               values_to = "Proportion") %>% #valeurs dans une nouvelle colonne intitulée "Proportion"
  mutate(Gender = ifelse(Gender == "Mean_Proportion_Female_Id", "Female", "Male")) #Si Gender est égal

# Créer le graphique
ggplot(stats_long, aes(x = category_gender, y = Proportion, fill = Gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportion of Female and Male Authors According to Categorical Variables",
       x = "Gender Category",
       y = "Mean Proportion",
       fill = "Gender") +
  theme_minimal()
```



On retrouve les résultats des statistiques descriptives avec environ 30% de femmes et 70% d'homems pour les articles collaboratifs.

```
#library(ggridges)
#Numerical and categorical variable
#ggplot(Corpus.Short, aes(x=sum_gender_male, y=category, fill=category, color=category)) + geom_density_

#ggplot(Corpus.Short, aes(x=category, y=proportion_gender_male)) + geom_boxplot()

#ggplot(Corpus.Short, aes(x=category, y=proportion_female)) + geom_boxplot()
```

1.3.Overall Temporal evolution

1.3.1. Descriptive statistics : overall temporal evolution

```
annual_stats <- Corpus.CleanedNames.2 %>%
  group_by(PY) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    Mean_Proportion_Male_Id = mean(proportion_gender_male_id, na.rm = TRUE)
  )
annual_stats
```

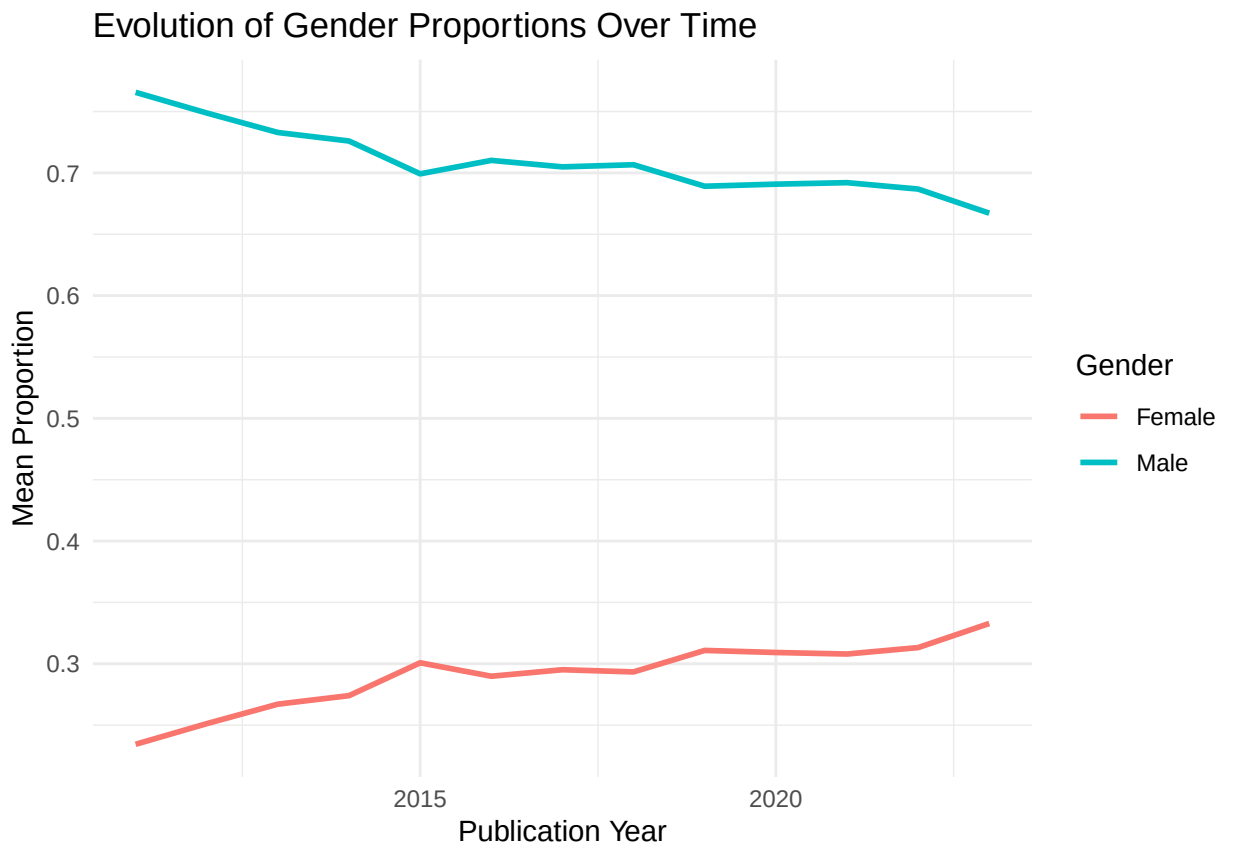
```
## # A tibble: 13 x 3
```

```
##       PY Mean_Proportion_Female_Id Mean_Proportion_Male_Id
##    <dbl>                     <dbl>                   <dbl>
##  1  2011                     0.234                   0.766
##  2  2012                     0.251                   0.749
##  3  2013                     0.267                   0.733
##  4  2014                     0.274                   0.726
##  5  2015                     0.301                   0.699
##  6  2016                     0.290                   0.710
##  7  2017                     0.295                   0.705
##  8  2018                     0.293                   0.707
##  9  2019                     0.311                   0.689
## 10  2020                     0.309                   0.691
## 11  2021                     0.308                   0.692
## 12  2022                     0.313                   0.687
## 13  2023                     0.333                   0.667
```

La proportion moyenne de femmes dans les articles publiés semble augmenter progressivement, d'environ 24,39% en 2011 à environ 33,17% en 2020. En parallèle, la proportion moyenne d'hommes diminue légèrement sur la même période, passant d'environ 75,61% en 2011 à environ 66,83% en 2020. Evolution vers une plus grande diversité de genres dans les publications au fil du temps, avec une augmentation de la représentation des femmes dans les articles

1.3.2. Graphics : overall temporal evolution

```r
annual_stats_long <- annual_stats %>%
  pivot_longer(
    cols = c(Mean_Proportion_Female_Id, Mean_Proportion_Male_Id),
    names_to = "Gender",
    values_to = "Proportion"
  ) %>%
  mutate(Gender = ifelse(Gender == "Mean_Proportion_Female_Id", "Female", "Male"))

# Créer le graphique
ggplot(annual_stats_long, aes(x = PY, y = Proportion, color = Gender)) +
  geom_line(linewidth = 1) +
  labs(title = "Evolution of Gender Proportions Over Time",
       x = "Publication Year",
       y = "Mean Proportion",
       color = "Gender") +
  theme_minimal()
```




```
# Distribution of female authors accross time
#ggplot(Corpus.Short, aes(x = PY, y = sum_gender_female)) + geom_point() + labs(title = "Proportion of

#ggplot(Corpus.Short, aes(x = PY, y = proportion_gender_male)) + geom_point() + labs(title = "Proportio
```

1.4. Temporal evolution according to categorical variables

1.4.1. Descriptive statistics : temporal evolution according to categorical variables

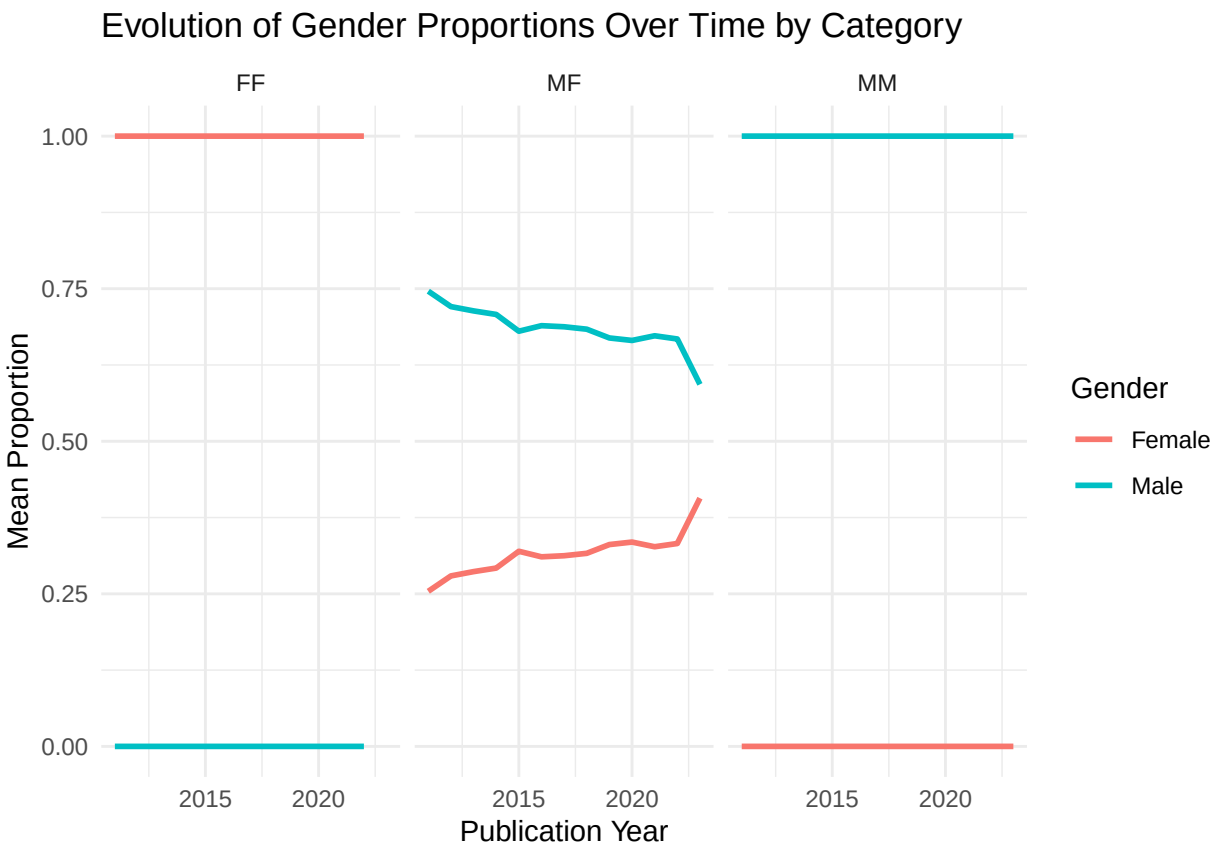
```
# Calculer les moyennes annuelles des proportions de genres par catégorie de genre
# Calculer les moyennes annuelles des proportions de genres par catégorie de genre
annual_stats_by_category <- Corpus.CleanedNames.2 %>%
  group_by(PY, category_gender) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    Mean_Proportion_Male_Id = mean(proportion_gender_male_id, na.rm = TRUE),
    .groups = 'drop' #`summarise()` has grouped output by 'PY'. You can override using the `.groups` ar
  )

annual_stats_by_category
```

```
## # A tibble: 38 x 4
##       PY category_gender Mean_Proportion_Female_Id Mean_Proportion_Male_Id
```

```
##    <dbl> <chr>                                <dbl>                  <dbl>
##  1  2011 FF                                   1                      0
##  2  2011 MF                                   0.254                  0.746
##  3  2011 MM                                   0                      1
##  4  2012 FF                                   1                      0
##  5  2012 MF                                   0.279                  0.721
##  6  2012 MM                                   0                      1
##  7  2013 FF                                   1                      0
##  8  2013 MF                                   0.286                  0.714
##  9  2013 MM                                   0                      1
## 10  2014 FF                                   1                      0
## # ... with 28 more rows
```

```r
# Restructurer les données pour ggplot2
annual_stats_by_category_long <- annual_stats_by_category %>%
  pivot_longer(
    cols = c(Mean_Proportion_Female_Id, Mean_Proportion_Male_Id),
    names_to = "Gender",
    values_to = "Proportion"
  ) %>%
  mutate(Gender = ifelse(Gender == "Mean_Proportion_Female_Id", "Female", "Male"))

# Créer le graphique
ggplot(annual_stats_by_category_long, aes(x = PY, y = Proportion, color = Gender)) +
  geom_line(size = 1) +
  facet_wrap(~ category_gender) +
  labs(title = "Evolution of Gender Proportions Over Time by Category",
       x = "Publication Year",
       y = "Mean Proportion",
       color = "Gender") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```



L'analyse de l'évolution temporelle de la proportion de femmes dans les collaborations mixtes (MF) montre une tendance à la hausse, bien que légèrement variable d'une année à l'autre. En 2011, la proportion moyenne de femmes dans les collaborations MF était d'environ 24,77%, et cette proportion a augmenté progressivement au fil des années pour atteindre environ 31,17% en 2020.

1.4.2. Descriptive statistics : temporal evolution according to categorical variables, by number of articles

```
annual_stats_by_category_nbarticles <- Corpus.CleanedNames.2 %>%
  group_by(PY, category_gender) %>%
  summarize(
    nb_articles=n()
    )
```

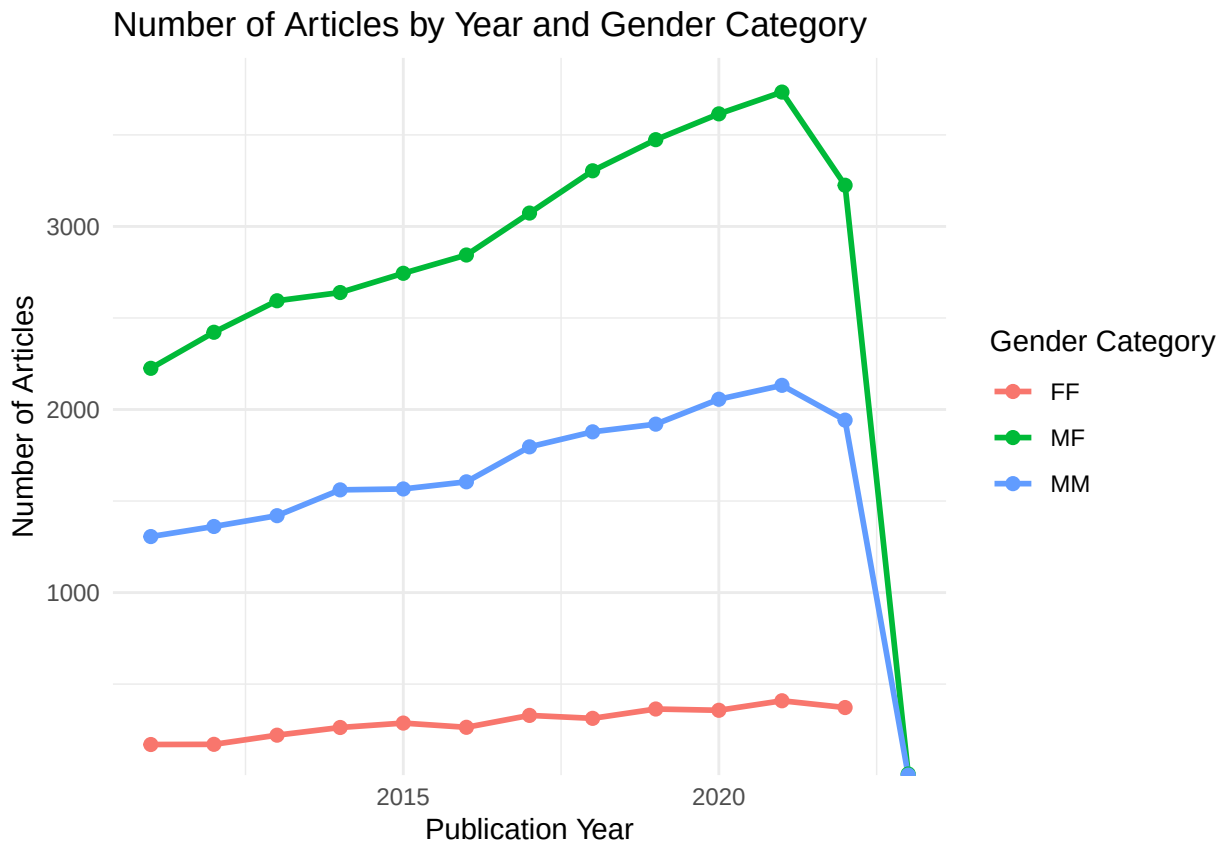
```
## `summarise()` has grouped output by 'PY'. You can override using the `.groups`
## argument.
```

```
annual_stats_by_category_nbarticles
```

```
## # A tibble: 38 x 3
## # Groups:   PY [13]
##       PY category_gender nb_articles
##    <dbl> <chr>                 <int>
##  1  2011 FF                      170
```

```
##  2  2011 MF                    2225
##  3  2011 MM                    1306
##  4  2012 FF                     171
##  5  2012 MF                    2422
##  6  2012 MM                    1361
##  7  2013 FF                     221
##  8  2013 MF                    2594
##  9  2013 MM                    1420
## 10  2014 FF                     263
## # ... with 28 more rows
```

```
ggplot(annual_stats_by_category_nbarticles, aes(x = PY, y = nb_articles, color = category_gender, group
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Number of Articles by Year and Gender Category",
       x = "Publication Year",
       y = "Number of Articles",
       color = "Gender Category") +
  theme_minimal() +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))
```



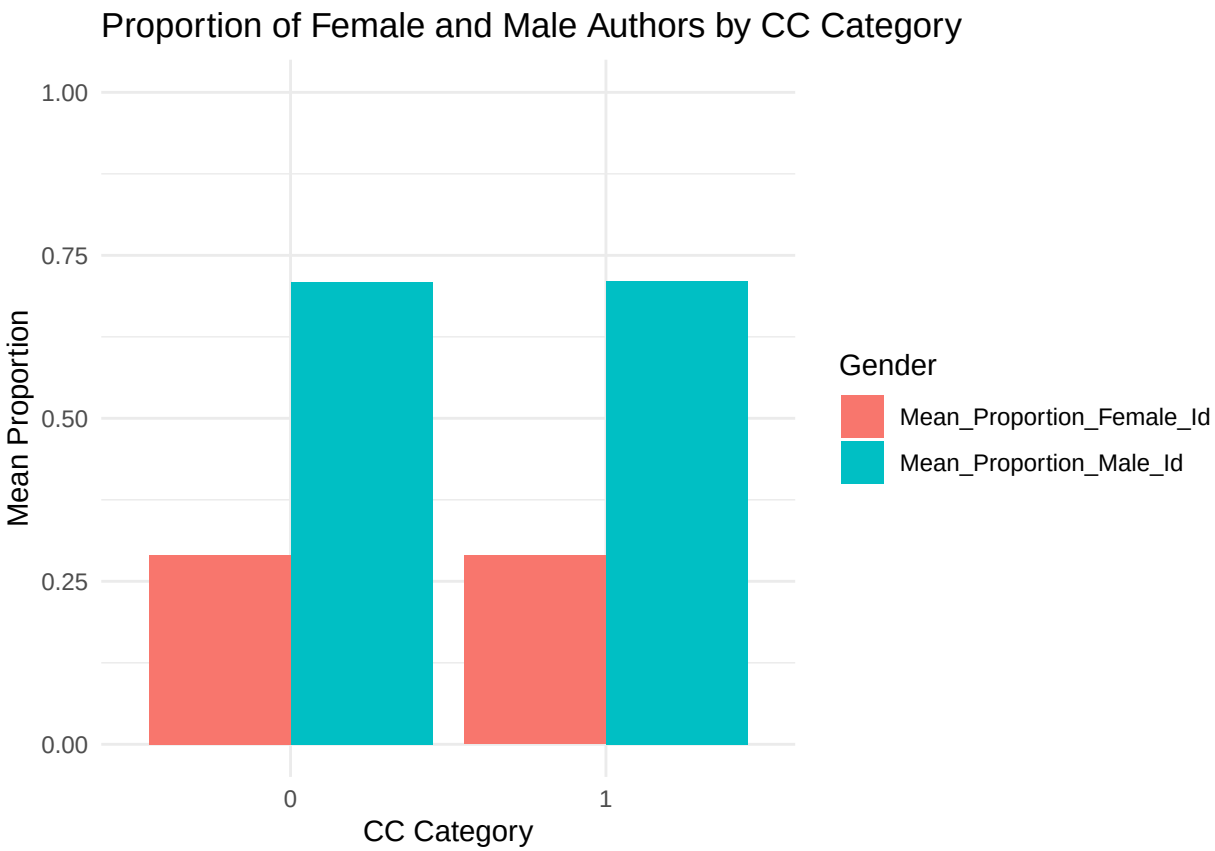
##2.Focus on climate change

2.1 Climate change overall : Descriptive statistics

```
# Calculer les statistiques descriptives pour chaque catégorie de la variable binaire "CC"
stats_by_CC <- Corpus.CleanedNames.2 %>%
  group_by(CC) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    Median_Proportion_Female_Id = median(proportion_gender_female_id, na.rm = TRUE),
    Mean_Proportion_Male_Id = mean(proportion_gender_male_id, na.rm = TRUE),
    Median_Proportion_Male_Id = median(proportion_gender_male_id, na.rm = TRUE)
  )

stats_by_CC
```

```
## # A tibble: 2 x 5
##      CC Mean_Proportion_Female_Id Median_Proportion_Female_Id Mean_Pro~1 Media~2
##   <dbl>                     <dbl>                        <dbl>      <dbl>   <dbl>
## 1     0                     0.291                      0.00755      0.709   0.992
## 2     1                     0.289                      0.0272       0.711   0.973
## # ... with abbreviated variable names 1: Mean_Proportion_Male_Id,
## #   2: Median_Proportion_Male_Id
```

```
# Conversion en format long
stats_long <- stats_by_CC %>%
  pivot_longer(cols = c(Mean_Proportion_Female_Id, Mean_Proportion_Male_Id),
               names_to = "Gender",
               values_to = "Proportion")

ggplot(stats_long, aes(x = factor(CC), y = Proportion, fill = Gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportion of Female and Male Authors by CC Category",
       x = "CC Category",
       y = "Mean Proportion",
       fill = "Gender") +
  scale_y_continuous(limits = c(0, 1)) +
  theme_minimal()
```



Interprétation : en moyenne, la proportion d'auteurs féminins dans les articles liés au changement climatique est légèrement plus faible que dans les articles non liés au changement climatique, avec des moyennes de 29,99% et 29,75% respectivement. En revanche, la proportion d'auteurs masculins est plus élevée dans les articles sur le changement climatique, avec une moyenne de 70,26%, comparativement à 70,06% pour les autres articles.

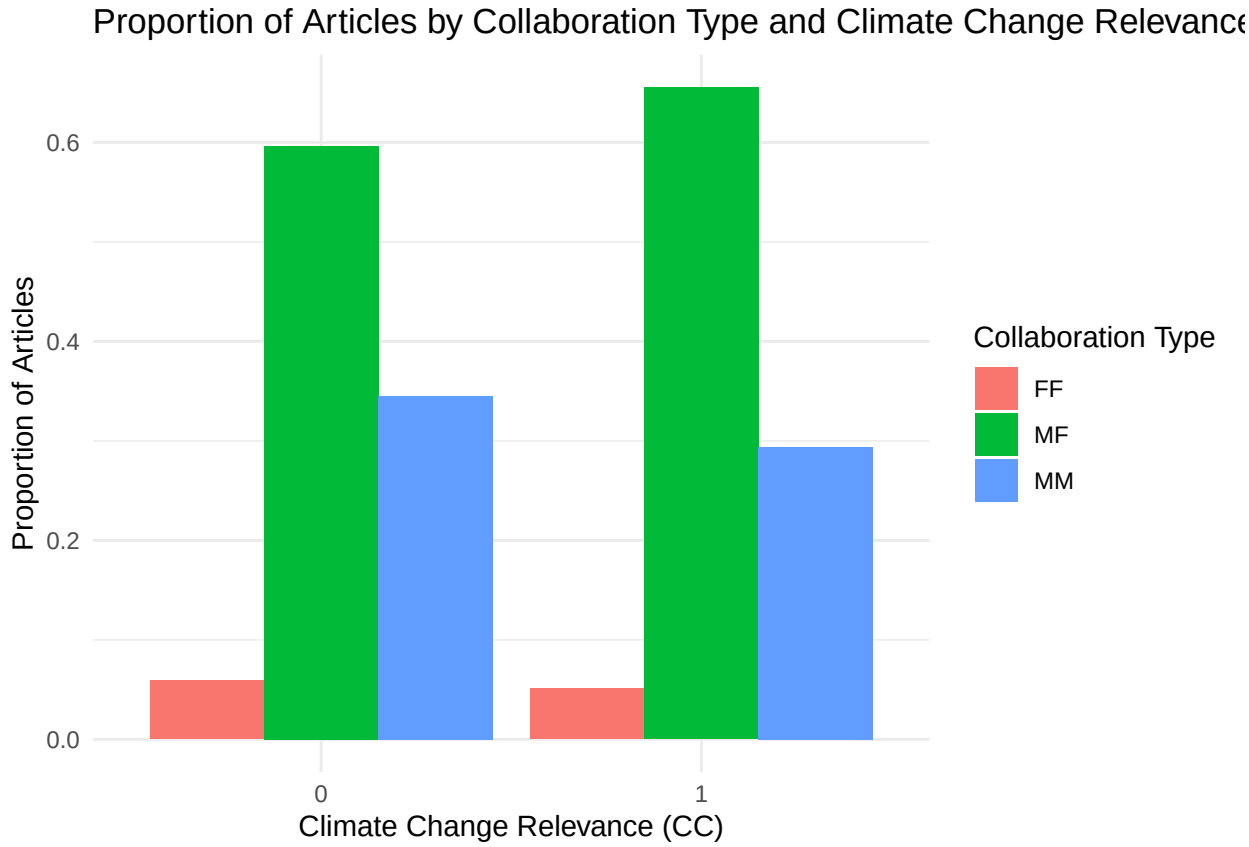
##2.2. Climate change, category gender and number of articles

```
stats_by_CC_category_gender <- Corpus.CleanedNames.2 %>%
  group_by(CC, category_gender) %>%
  summarize(nb_articles = n(), .groups = 'drop')

stats_by_CC_category_gender
```

```
## # A tibble: 6 x 3
##      CC category_gender nb_articles
##   <dbl> <chr>                 <int>
## 1     0 FF                     3391
## 2     0 MF                    34250
## 3     0 MM                    19806
## 4     1 FF                      129
## 5     1 MF                     1652
## 6     1 MM                      740
```

```
# Calculer la proportion des articles pour chaque combinaison de CC et category_gender
stats_by_CC_category_gender <- stats_by_CC_category_gender %>%
  group_by(CC) %>%
```

```
  mutate(proportion_articles = nb_articles / sum(nb_articles))

# Graphique
ggplot(stats_by_CC_category_gender, aes(x = factor(CC), y = proportion_articles, fill = category_gender)
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Proportion of Articles by Collaboration Type and Climate Change Relevance",
       x = "Climate Change Relevance (CC)",
       y = "Proportion of Articles",
       fill = "Collaboration Type") +
  theme_minimal()
```



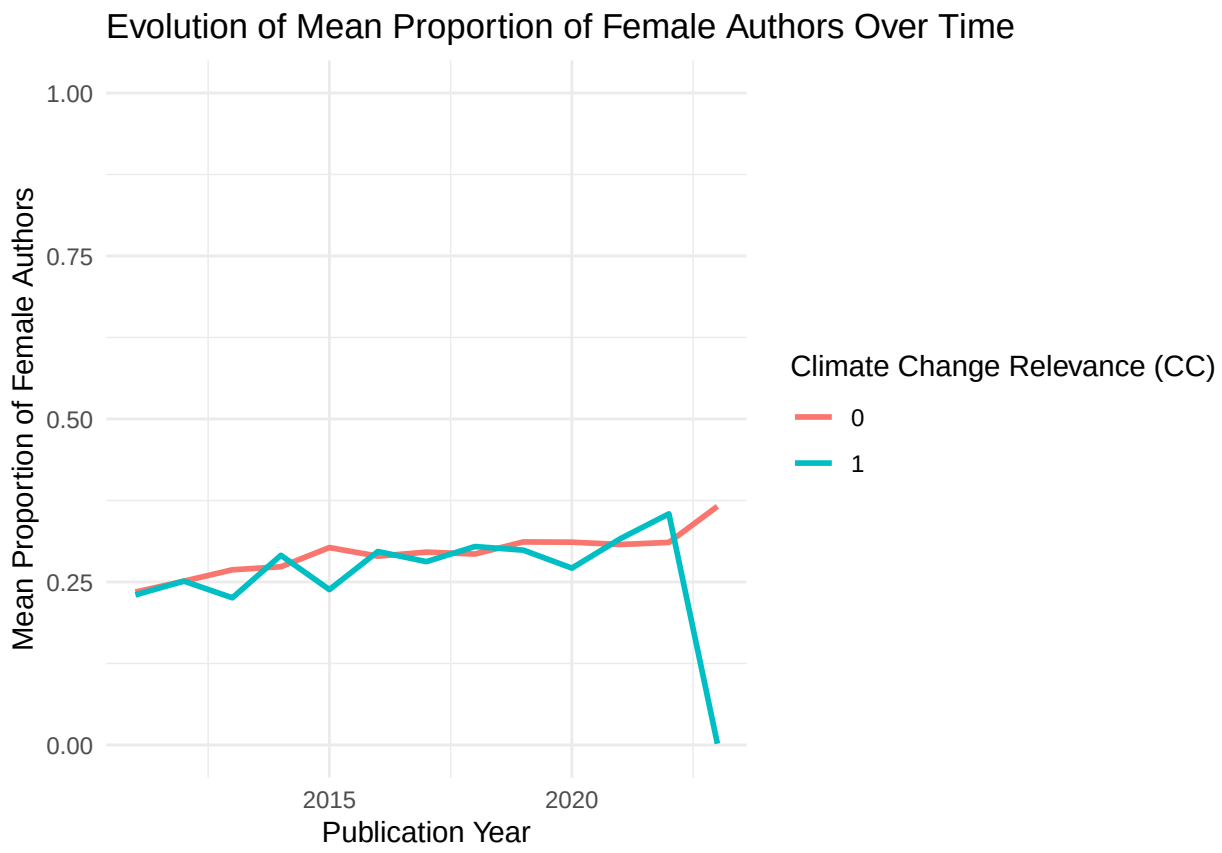
2.3 : Temporal evolution of the proportion of female authors by CC

```
annual_stats_by_CC <- Corpus.CleanedNames.2 %>%
  group_by(PY, CC) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    .groups = 'drop'
  )

annual_stats_by_CC
```

```
## # A tibble: 26 x 3
```

```
##       PY    CC Mean_Proportion_Female_Id
##    <dbl> <dbl>                      <dbl>
##  1  2011     0                      0.235
##  2  2011     1                      0.230
##  3  2012     0                      0.251
##  4  2012     1                      0.251
##  5  2013     0                      0.269
##  6  2013     1                      0.226
##  7  2014     0                      0.273
##  8  2014     1                      0.291
##  9  2015     0                      0.303
## 10  2015     1                      0.238
## # ... with 16 more rows
```

```
ggplot(annual_stats_by_CC, aes(x = PY, y = Mean_Proportion_Female_Id, color = factor(CC))) +
  geom_line(size = 1) +
  labs(title = "Evolution of Mean Proportion of Female Authors Over Time",
       x = "Publication Year",
       y = "Mean Proportion of Female Authors",
       color = "Climate Change Relevance (CC)") +
  theme_minimal() +
  scale_y_continuous(limits = c(0, 1)) # Ajuster les limites de l'ordonnée pour aller de 0 à 1
```



Increase of female authors regarding climate change articles and decrase for non climate change related articles

##3.Focus on Top5 and Top30 and on climte change

```r
# Calculer les statistiques descriptives pour chaque combinaison de catégorie de journaux et de la vari
stats_by_journal_CC <- Corpus.CleanedNames.2 %>%
  group_by(TopFive, Top30, CC) %>%
  summarize(
    Mean_Proportion_Female_Id = mean(proportion_gender_female_id, na.rm = TRUE),
    Median_Proportion_Female_Id = median(proportion_gender_female_id, na.rm = TRUE),
    Median_Proportion_Male_Id = median(proportion_gender_male_id, na.rm = TRUE),
    Mean_Proportion_Male_Id = mean(proportion_gender_male_id, na.rm = TRUE),
  )
```

```
## `summarise()` has grouped output by 'TopFive', 'Top30'. You can override using
## the `.groups` argument.
```

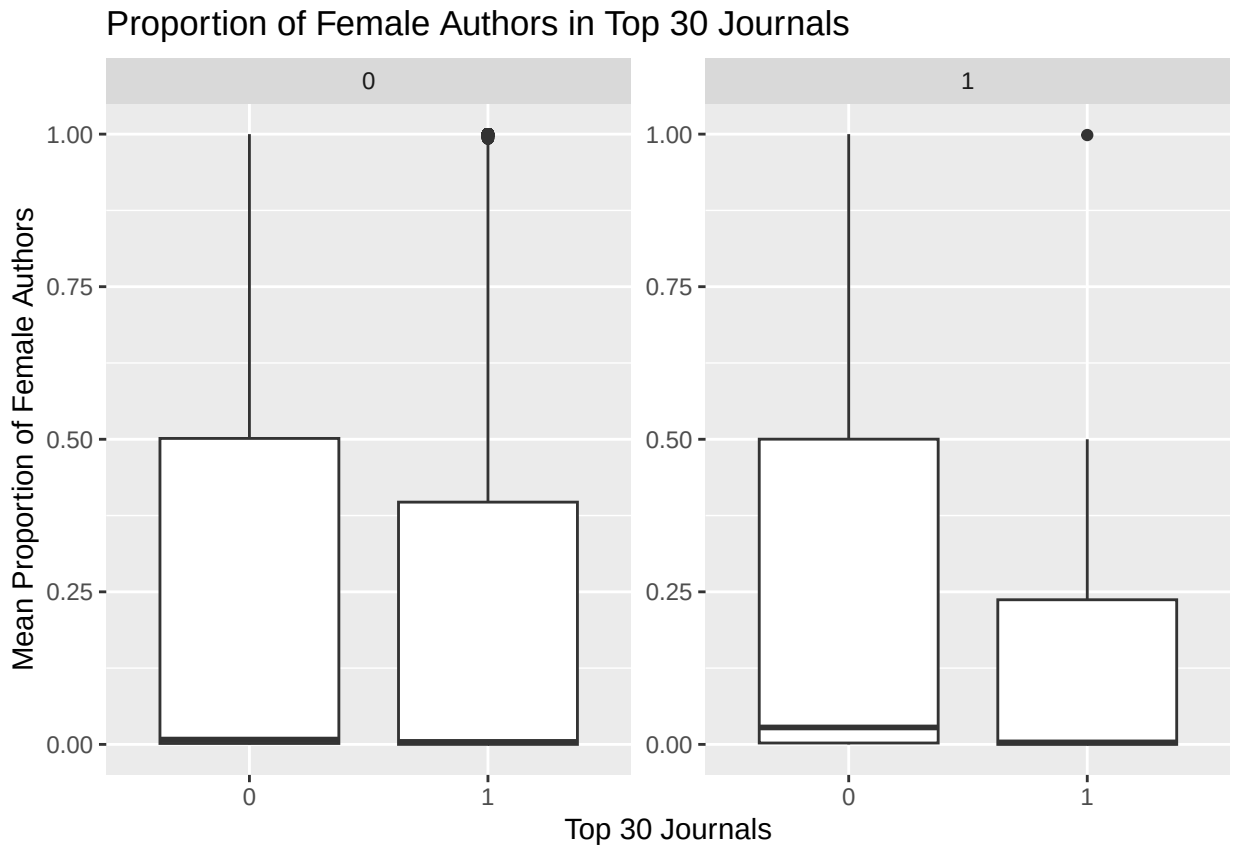
```r
stats_by_journal_CC
```

```
## # A tibble: 6 x 7
## # Groups:   TopFive, Top30 [3]
##   TopFive Top30    CC Mean_Proportion_Female_Id Median_Proport~1 Media~2 Mean_~3
##     <dbl> <dbl> <dbl>                     <dbl>            <dbl>   <dbl>   <dbl>
## 1       0     0     0                    0.295           0.0079    0.992   0.705
## 2       0     0     1                    0.290           0.0278    0.972   0.710
## 3       0     1     0                    0.229           0.0046    0.995   0.771
## 4       0     1     1                    0.197           0.0024    0.998   0.803
## 5       1     1     0                    0.144           0.00258   0.997   0.856
## 6       1     1     1                    0.00284         0.0041    0.996   0.997
## # ... with abbreviated variable names 1: Median_Proportion_Female_Id,
## #   2: Median_Proportion_Male_Id, 3: Mean_Proportion_Male_Id
```

Moyenne et medianens différentes en raison des valeurs extrêmes (beaucoup d'articles où il n'y a pas de femmes / où il n'y a que des hommes)

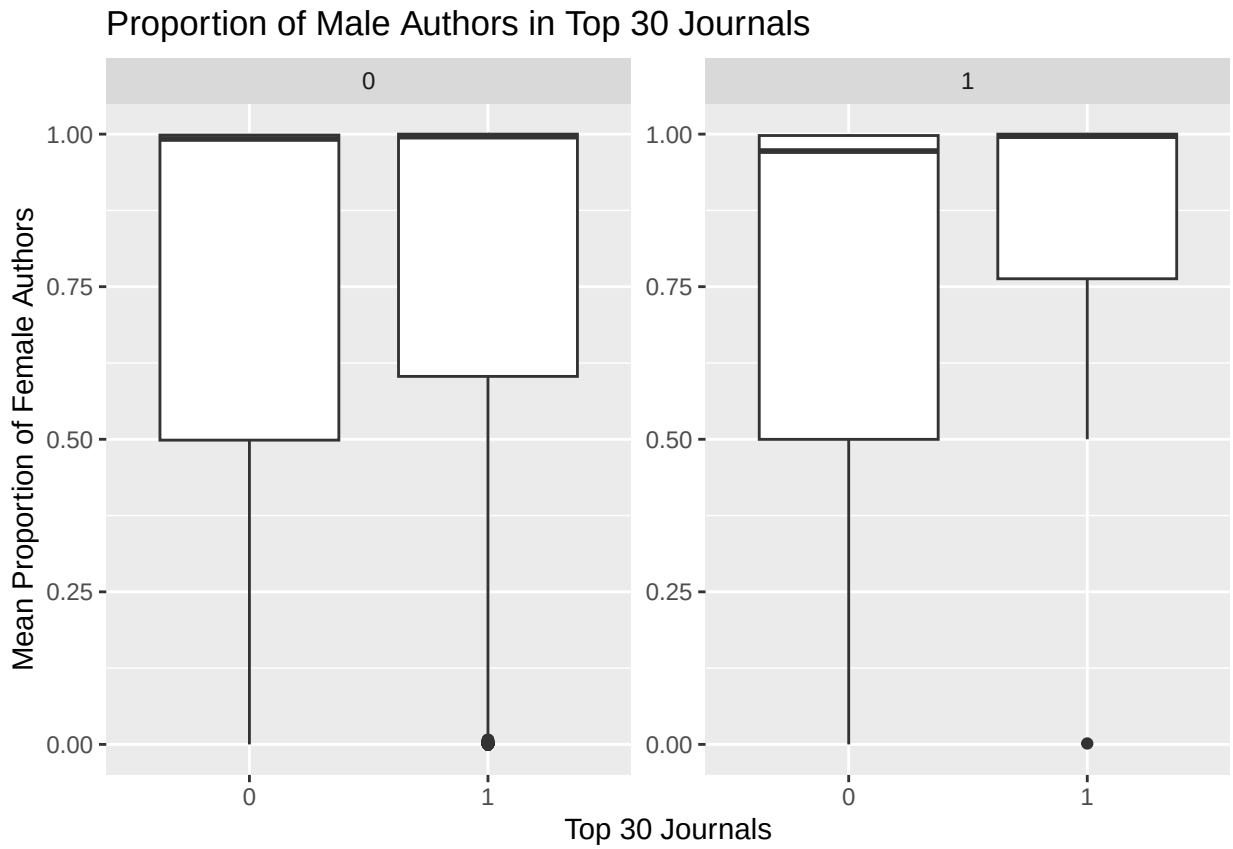
```r
ggplot(Corpus.CleanedNames.2, aes(x = factor(Top30), y = proportion_gender_female_id)) +
  geom_boxplot() +
  labs(title = "Proportion of Female Authors in Top 30 Journals",
       x = "Top 30 Journals",
       y = "Mean Proportion of Female Authors") +
  facet_wrap(~ CC, scales = "free")
```

```
## Warning: Removed 9392 rows containing non-finite values (`stat_boxplot()`).
```




```
ggplot(Corpus.CleanedNames.2, aes(x = factor(Top30), y =proportion_gender_male_id)) +
  geom_boxplot() +
  labs(title = "Proportion of Male Authors in Top 30 Journals",
       x = "Top 30 Journals",
       y = "Mean Proportion of Female Authors") +
  facet_wrap(~ CC, scales = "free")
```

```
## Warning: Removed 9392 rows containing non-finite values (‘stat_boxplot()‘).
```



Articles qui portent sur le CC : la proportion de femme dans le top 30 est beaucoup plus grande Articles qui portent sur le CC et qui ne sont pas dans le top 30 : medianne de la proportion de femmes un peu plus élevée

2. Geography

Affiliation data `C1`and `RP` variables. We'd like to identify each countries in which authors of the articles are affiliated. Thus, we'd like to create a variable "country" which list each country referenced in the `C1`and `RP`variables.

```
# table(Corpus$CC)
table(Corpus.Short$CC)
```

```
##
##      0      1
## 581500  25452
```

Citation data

Clean citation data

Prepare citation data. We want to create a database, where for each article (one raw in `Corpus`), we have a data frame with its cited articles. (one raw by references). We want to separate information on the authors of the cited articles, the title of the article, the year of publication, and the journal.

First work with one of the datasource: let's say Wos.

```
gc()
```

```
##             used   (Mb) gc trigger (Mb) limit (Mb)  max used (Mb)
## Ncells  17973634  959.9   38029058 2031         NA  38029058 2031
## Vcells 349990517 2670.3  599258148 4572      16384 416022172 3174
```

```
load(here(dir$prep.data,"Wos_Short.Rdata"))
head(Wos.Econ.Short$CR)
```

```
## [1] "Apostolato I.-A., 2013, INT REV SOCIAL RES, V3, DOI [10.1515/irsr-2013-0023, DOI 10.1515/IRSR-2
## [2] "Baldwin J. R., 2000, FAILURE RATES NEW CA; Barber BM, 2001, Q J ECON, V116, P261, DOI 10.1162/0
## [3] "Altera Invest, 2020, SAL READ MAD BUS FRA; [Anonymous], 2020, FRANCHISE CAPITAL; BURTON F, 2000
## [4] "Ayanso A, 2014, INFORM TECHNOL DEV, V20, P60, DOI 10.1080/02681102.2013.797378; Polozhentseva Y
## [5] "Ain, 2020, MUCH FREEL EARN THEY; Amigud A, 2020, ASSESS EVAL HIGH EDU, V45, P541, DOI 10.1080/0
## [6] "Ghodsi M, 2016, ESTIMATING IMPORTER; Ghodsi M., 2016, BILATERAL IMPORT ELA; Kee HL, 2009, ECON
```

Test

```
table(is.na(Wos.Econ.Short$CR))
```

```
##
##  FALSE   TRUE
## 149184   4976
```

Test 2

```
table(is.na(Wos.Econ.Short$CR), Wos.Econ.Short$PY)
```

```
##
##          2018  2019  2020  2021  2022  2023
##   FALSE 30163 31178 30738 31229 25774   102
##   TRUE   1463  1742   953   672   146     0
```