Lion in the Room

Loading the necessary packages for the project

```
library(tidyverse)  # Loads ggplot2, dplyr, tidyr, readr, purrr, tibble, and stringr
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readxl)     # For reading Excel files
library(rvest)      # For web scraping
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
library(httr)       # For working with HTTP
library(rcrossref)  # For using CrossRef's API
library(janitor)    # For cleaning data
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(reshape2)   # For reshaping data
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(igraph)     # For graphing
```

```
##
## Attaching package: 'igraph'
```

```
##
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
##
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```
library(ggraph)     # For graphing
library(scales)     # For graphing
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(lubridate)  # For time series work
library(gridExtra)  # For Grid making
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(VennDiagram)# For Venn Diagrams
```

```
## Loading required package: grid
```

```
## Loading required package: futile.logger
```

Loading the data-set sourced from Retraction Watch on 12-Jan-2024

```
data<- read.csv("retraction_watch.csv")
abdc<- read.csv("abdc.csv")
predatory<- read.csv("predatory.csv")
```

Doing some wrangling of the data

```
# Convert dates to POSIX format for the pupose of computation
data$OriginalPaperDate <- as.POSIXct(data$OriginalPaperDate, format = "%m/%d/%Y %H:%M")
data$RetractionDate <- as.POSIXct(data$RetractionDate, format = "%m/%d/%Y %H:%M")

# Compute the difference in months
data$DurationInMonths <- interval(start = data$OriginalPaperDate, end = data$RetractionDate) / months(1)

# Extract year from RetractionDate
data$RetractionYear <- year(data$RetractionDate)

#Creating Clean names
data<- data %>%
  clean_names()

abdc<- abdc%>%
  clean_names()

predatory<- predatory %>%
  clean_names()
```

Creating the lists

```
abdc_list <- unique(tolower(abdc$journal))
retraction_list <- unique(tolower(data$journal))
predatory_list <- unique(tolower(predatory$journal))
```

Trying to create a venn diagram to see how these sets intersect

```
# Define a list of lists for the Venn diagram
list_of_lists <- list(
  ABDC = abdc_list,
  Retraction = retraction_list,
  Predatory = predatory_list
)

# Specify the filename for the output
output_filename <- "venn_diagram.png"

# Create and save the Venn diagram
venn.plot <- venn.diagram(
  x = list_of_lists,
  category.names = c("ABDC", "Retraction", "Predatory"),
```

```
  filename = output_filename
)
```

No Direct relationship between predatory practices and retractions: The fact that there are many retractions not linked to predatory journals might suggest that predatory practices are not the only reason for retractions. Retractions can occur in any journal and may be due to factors like data fabrication, plagiarism, or other ethical issues.

Overlap between predatory and retraction: There is an overlap of just one journal between the predatory list and the ABDC list, suggesting that the ABDC list is largely successful in avoiding predatory journals. However, the presence of that one journal indicates that no vetting process is completely foolproof.

Retractions are not limited to Predatory Journals: The larger number of retractions (7820) not associated with predatory journals implies that retractions are a broader issue in academic publishing, potentially due to factors like honest errors, misconduct, or problems in the peer-review process even in non-predatory journals.

Considering only one predatory journal of the 1147 that are listed, has ever had a retracted article in it, is there something that we are overseeing?

Filter data for rows where article_type is "Research Article" or "Article in Press"

```
# This is to ensure that there are no conference papers, book chapters, and others that are there in th
filtered_data <- data[grepl("Research Article", data$article_type) | grepl("Article in Press", data$arti
```

Looking only at the intersection of ABDC and Retracted.

```
ret_int_abdc <- filtered_data %>%
  filter(
    tolower(filtered_data$journal) %in% tolower(abdc$journal) # Converting to lower case to ensure that
  )

write.csv(ret_int_abdc, "ret_int_abdc.csv") # This is so that Pari can see the data
```

There are 951 data points with the conference articles, and 883 without the conference and other types of articles. It is these 883 articles that we focus our attention on.

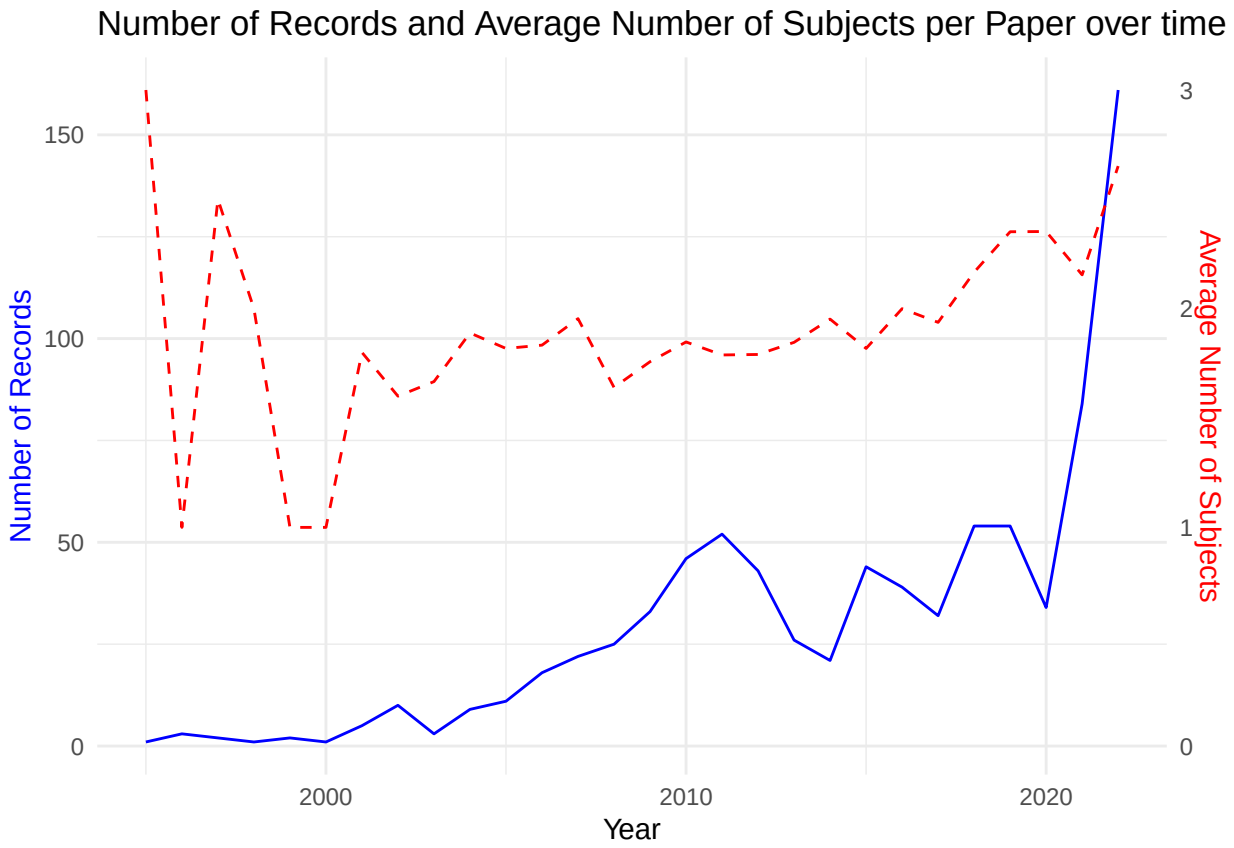
Let's just describe this dataset. This is for us to mention in the introduction section

```
# Select the specified columns
our_interest <- select(ret_int_abdc, record_id, subject, title, original_paper_date, reason, author)
```

Plot Average Number of Subjects per Paper Over Time, and the number of retracted artciles in the ABDC

```
# Calculate the number of subjects per record
our_interest$subject_count <- sapply(strsplit(our_interest$subject, ";"), function(x) sum(nzchar(x)))

# Extract the year from the original paper date
our_interest$year <- format(our_interest$original_paper_date, "%Y")

# Ensure year is treated as a continuous variable
our_interest$year <- as.numeric(our_interest$year)
```

```r
# Create a summary data frame
yearly_summary <- our_interest %>%
  group_by(year) %>%
  summarize(
    record_count = n(),
    avg_subject_count = mean(subject_count, na.rm = TRUE)
  ) %>%
  ungroup()

# Ensure we have a row for every year in the range from 2000 to 2022
yearly_summary <- data.frame(year = 1995:2022) %>%
  left_join(yearly_summary, by = "year") %>%
  replace_na(list(record_count = 0, avg_subject_count = 0))

# Find the maximums for scaling the secondary axis
max_records <- max(yearly_summary$record_count)
max_avg_subjects <- max(yearly_summary$avg_subject_count)

# Create the line plot with two y-axes and colored axis labels
ggplot(yearly_summary, aes(x = year)) +
  geom_line(aes(y = record_count), color = "blue") +
  labs(x = "Year", y = "Number of Records", title = "Number of Records and Average Number of Subjects p
  theme_minimal() +
  theme(axis.title.y = element_text(color = "blue")) +
  scale_y_continuous(
    "Number of Records",
    sec.axis = sec_axis(~ . * max_avg_subjects / max_records, name = "Average Number of Subjects", label
  ) +
  geom_line(aes(y = avg_subject_count * max_records / max_avg_subjects), color = "red", linetype = "dash
  theme(axis.title.y.right = element_text(color = "red"))
```

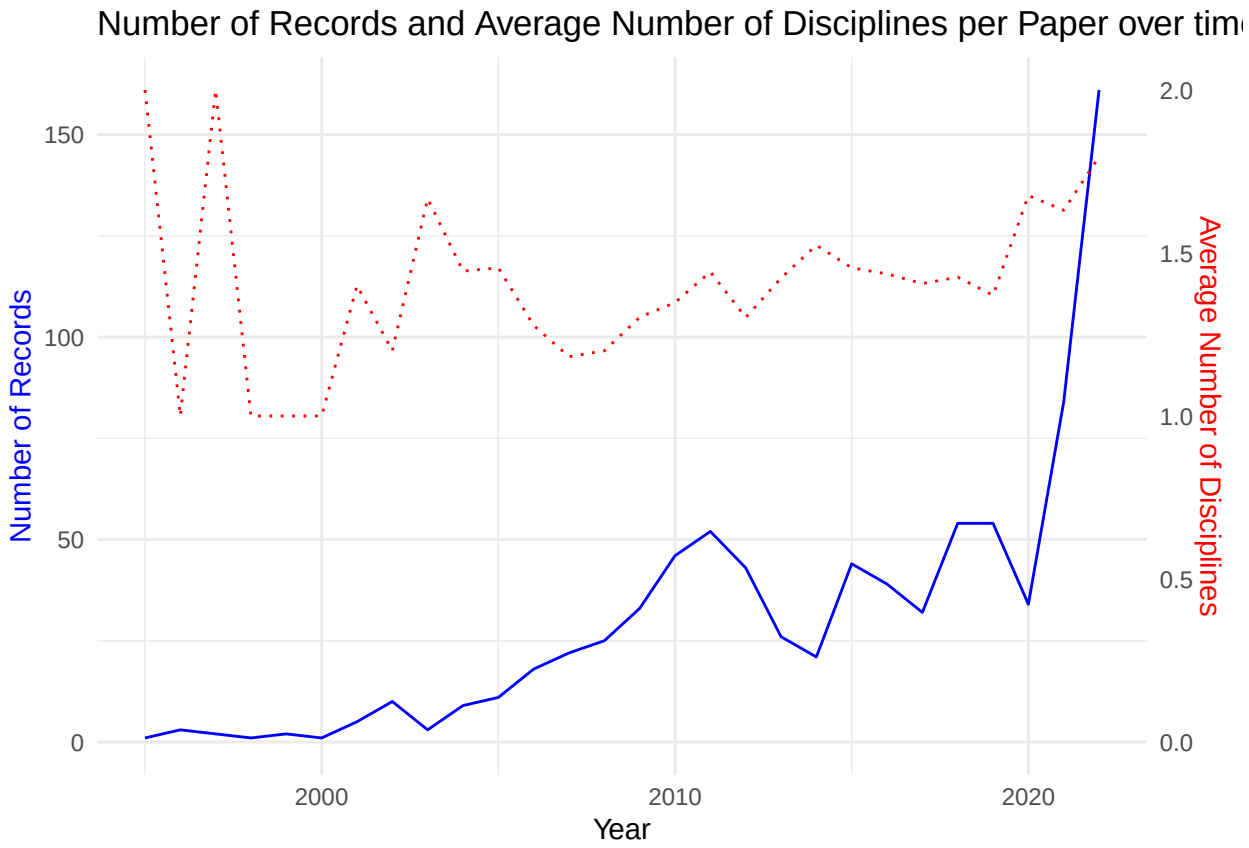


This chart shows us that from the turn of the century, the idea of inter disciplinary research (defined as research being carried out in multiple subjects within the same meta subject) has increased from just over 1 to nearly 3. This is a massive shift - and somewhat concerning.

Checking if the same trend also appears in the metasubject realm

```
# Define the mapping of codes to metasubjects
metasubjects_map <- c(
  "B/T" = "Business and Technology",
  "BLS" = "Basic Life Sciences",
  "ENV" = "Environmental Sciences",
  "HSC" = "Health Sciences",
  "HUM" = "Humanities",
  "PHY" = "Physical Sciences",
  "SOC" = "Social Sciences"
)

# Extract and map metasubjects
our_interest <- our_interest %>%
  mutate(
    year = as.numeric(format(original_paper_date, "%Y")),
    metasubject_codes = str_extract_all(subject, "\\(\\w+/\\w+\\)|\\(\\w+\\)"),
    metasubjects = lapply(metasubject_codes, function(codes) {
      unique_codes <- unique(codes)
      sapply(unique_codes, function(code) metasubjects_map[substr(code, 2, nchar(code) - 1)])
    }),
    metasubject_count = sapply(metasubjects, length)
```

```r
  )

# Create a summary data frame for metasubjects
yearly_metasubject_summary <- our_interest %>%
  group_by(year) %>%
  summarize(
    avg_metasubject_count = mean(metasubject_count, na.rm = TRUE),
    record_count = n()  # Assuming you want to also plot the number of records
  ) %>%
  ungroup()

# Ensure we have a row for every year in the range
yearly_metasubject_summary <- data.frame(year = 1995:2022) %>%
  left_join(yearly_metasubject_summary, by = "year") %>%
  replace_na(list(avg_metasubject_count = 0, record_count = 0))
# Find the maximum value to set as the upper limit for both y-axes
max_count <- max(yearly_metasubject_summary$record_count, na.rm = TRUE)
max_avg_metasubjects <- max(yearly_metasubject_summary$avg_metasubject_count, na.rm = TRUE)
common_limit <- max(max_count, max_avg_metasubjects)

# Find the maximum value to use as the upper limit for the secondary y-axis
max_records <- max(yearly_metasubject_summary$record_count, na.rm = TRUE)
max_avg_metasubjects <- max(yearly_metasubject_summary$avg_metasubject_count, na.rm = TRUE)

# Create the line plot with adjusted axis scales
ggplot(yearly_metasubject_summary, aes(x = year)) +
  geom_line(aes(y = record_count), color = "blue") + # Plot number of records in blue
  labs(x = "Year", y = "Number of Records", title = "Number of Records and Average Number of Disciplines
  theme_minimal() +
  theme(
    axis.title.y = element_text(color = "blue"),
    axis.title.y.right = element_text(color = "red")
  ) +
  scale_y_continuous(
    name = "Number of Records",
    limits = c(0, max_records), # Set limits for the primary y-axis
    sec.axis = sec_axis(~ . * max_avg_metasubjects / max_records, name = "Average Number of Disciplines
  ) +
  geom_line(aes(y = avg_metasubject_count * max_records / max_avg_metasubjects), color = "red", linetyp
  scale_x_continuous(limits = c(1995, 2022)) # Set x-axis limits from 1980 to 2022
```

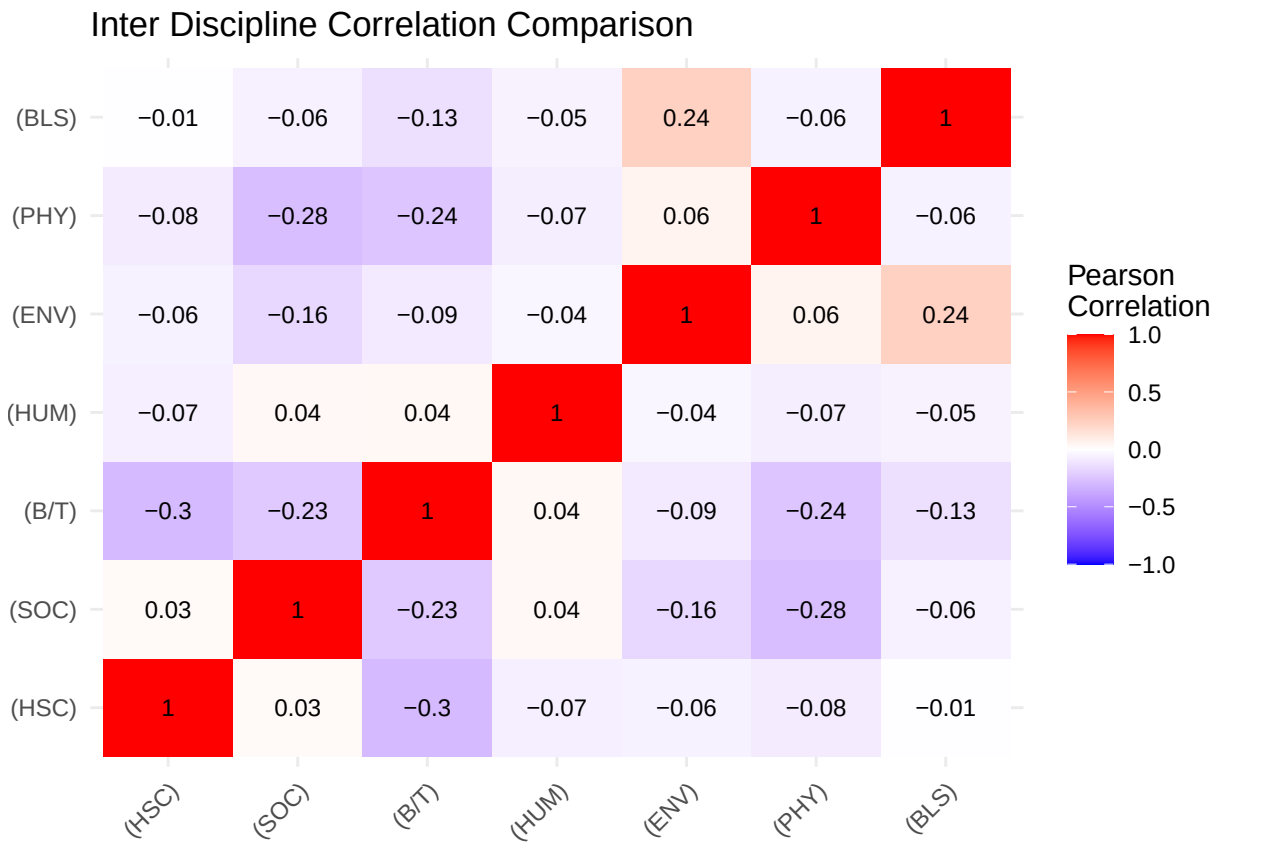


The trend here also appears somewhat consistent with the subjects. This means that retracted papers are slowly becoming inter disciplinary in nature.

Now, ww create a correlation plot to see which Disciplines keep occuring together. I am not doing the regular subjects becasue that would be too cluttered in my opinion.

```
# Extract meta-subjects from the 'subject' column
our_interest$meta_subjects <- str_extract_all(our_interest$subject, "\\(.*?\\)")

# Create a unique list of all meta-subjects
all_meta_subjects <- unique(unlist(our_interest$meta_subjects))

# Function to create dummy variables
create_dummy <- function(meta_subjects, subject) {
  as.integer(subject %in% meta_subjects)
}

# Apply the function to create dummy variables for each meta-subject
for (meta_subject in all_meta_subjects) {
  our_interest[[meta_subject]] <- sapply(our_interest$meta_subjects, create_dummy, subject = meta_subje
}

# Select only the meta-subject dummy variables
meta_subject_cols <- all_meta_subjects

# Calculate the correlation matrix
```

```r
cor_matrix <- cor(our_interest[, meta_subject_cols], use = "complete.obs")


# Melt the correlation matrix for visualization
melted_cor_matrix <- melt(cor_matrix)

# Plot the correlation matrix
ggplot(data = melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(label = round(value, 2)), color = "black", size = 3) +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Pearson\nCorrelation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title = element_blank()) + ggtitle("Inter Discipline Correlation Comparison")
```



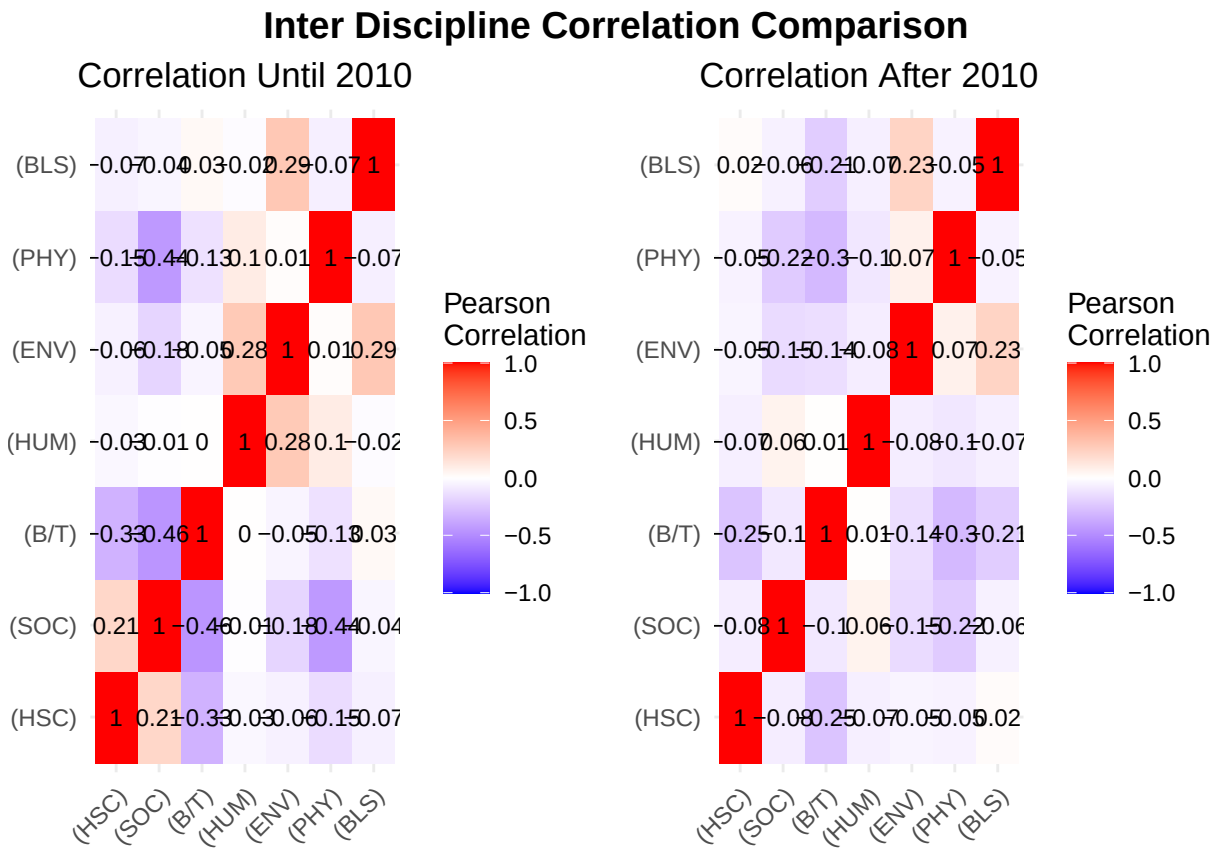
This is the key for the reference of others: (B/T): Business and Technology (BLS): Basic Life Sciences (ENV): Environmental Sciences (HSC): Health Sciences (HUM):Humanities (PHY): Physical Sciences (SOC): Social Sciences

This investigation reveals that there are no two subjects that are strongly correlated. However, we can try something interesting here.

```r
library(dplyr)
library(ggplot2)
library(stringr)
library(reshape2)
```

```r
library(gridExtra) # This library is used to arrange ggplots side by side

# Extract meta-subjects from the 'subject' column
our_interest$meta_subjects <- str_extract_all(our_interest$subject, "\\(.*?\\)")

# Create a unique list of all meta-subjects
all_meta_subjects <- unique(unlist(our_interest$meta_subjects))

# Function to create dummy variables
create_dummy <- function(meta_subjects, subject) {
  as.integer(subject %in% meta_subjects)
}

# Apply the function to create dummy variables for each meta-subject
for (meta_subject in all_meta_subjects) {
  our_interest[[meta_subject]] <- sapply(our_interest$meta_subjects, create_dummy, subject = meta_subje
}

# Filter data for the two periods
data_until_2010 <- our_interest %>% filter(year <= 2010)
data_after_2010 <- our_interest %>% filter(year > 2010)

# Select only the meta-subject dummy variables for correlation calculation
meta_subject_cols <- all_meta_subjects

# Calculate the correlation matrix for both periods
cor_matrix_until_2010 <- cor(data_until_2010[, meta_subject_cols], use = "complete.obs")
cor_matrix_after_2010 <- cor(data_after_2010[, meta_subject_cols], use = "complete.obs")

# Melt the correlation matrices for visualization
melted_cor_matrix_until_2010 <- melt(cor_matrix_until_2010)
melted_cor_matrix_after_2010 <- melt(cor_matrix_after_2010)

# Function to create a ggplot of the correlation matrix
create_cor_plot <- function(melted_cor_matrix) {
  ggplot(data = melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    geom_text(aes(label = round(value, 2)), color = "black", size = 3) +
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                         midpoint = 0, limits = c(-1, 1), space = "Lab",
                         name = "Pearson\nCorrelation") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1),
          axis.title = element_blank())
}

# Create the plots
plot_until_2010 <- create_cor_plot(melted_cor_matrix_until_2010)
plot_after_2010 <- create_cor_plot(melted_cor_matrix_after_2010)

library(gridExtra)

# Add individual titles to each plot
```

```
plot_until_2010 <- plot_until_2010 + ggtitle("Correlation Until 2010") + theme(plot.title = element_tex
plot_after_2010 <- plot_after_2010 + ggtitle("Correlation After 2010") + theme(plot.title = element_tex

# Combine the plots side by side with a common title
grid.arrange(plot_until_2010, plot_after_2010, ncol = 2,
             top = textGrob("Inter Discipline Correlation Comparison", gp = gpar(fontface = "bold", fon
```



This shows that there has not really been any significant trend. If anything, the correlation has dropped over time.

This points to the fact that there this issue is not particularly limited to some areas - but much more rampant.

In fact, if we start looking at the titles of the papers, we begin to get a much better picture.

Create a Word Map from the 'Title' Variable

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:httr':
##
##     content
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
# Create a text corpus
corpus = Corpus(VectorSource(our_interest$title))

# Clean up the text
corpus = tm_map(corpus, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents
```

```
corpus = tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents
```

```
corpus = tm_map(corpus, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents
```

```
corpus = tm_map(corpus, removeWords, stopwords("english"))
```

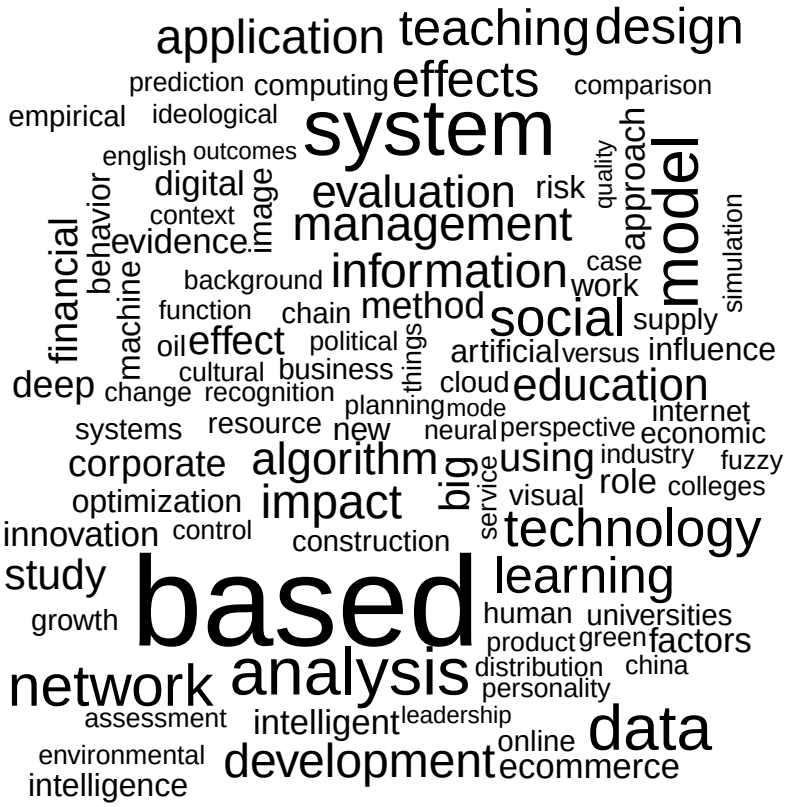
```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")):
## transformation drops documents
```

```
# Create a word cloud
wordcloud(corpus, max.words = 100)
```

```
## Warning in wordcloud(corpus, max.words = 100): performance could not be fit on
## page. It will not be plotted.
```

```
## Warning in wordcloud(corpus, max.words = 100): research could not be fit on
## page. It will not be plotted.
```

```
## Warning in wordcloud(corpus, max.words = 100): relationship could not be fit on
## page. It will not be plotted.
```

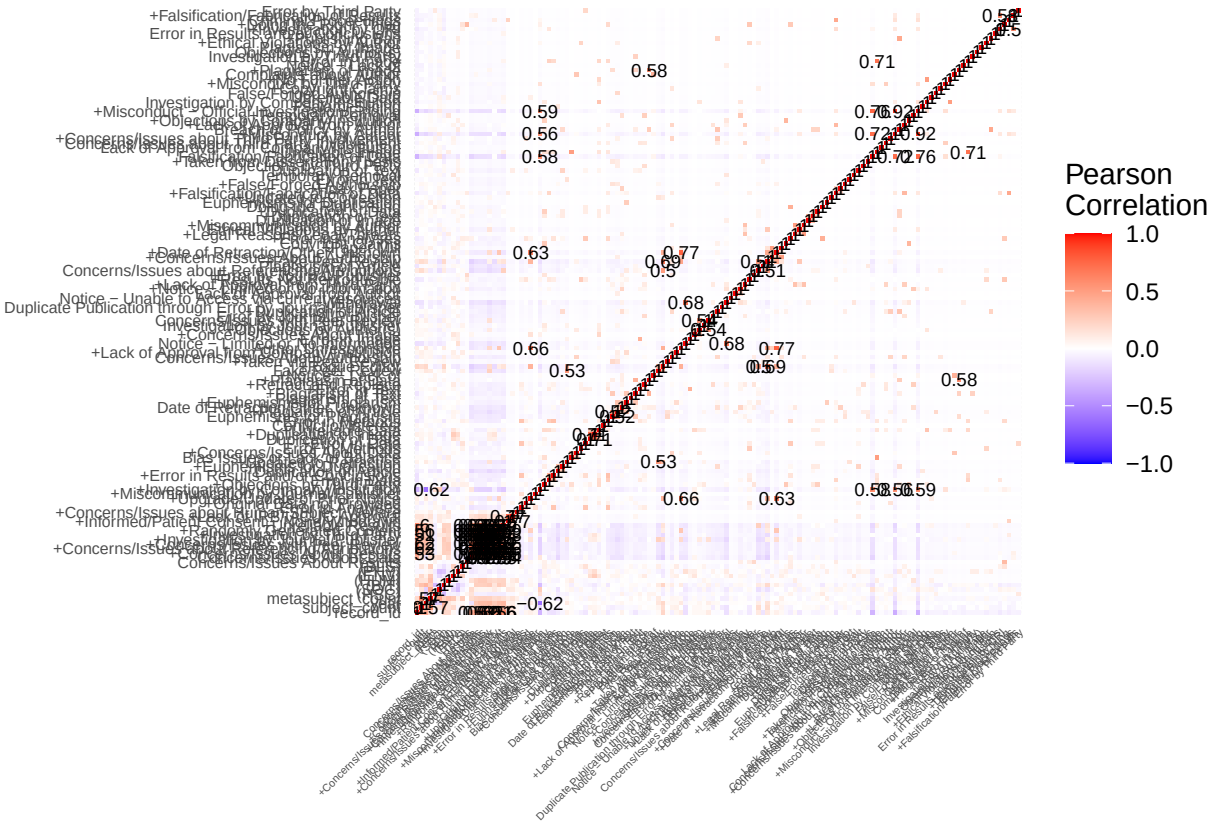


Well, we have to speak about this team!

Let me just find out if there is some correlation between the reasons that are cited for retraction

```
# Step 1: Split the 'reason' field into individual reasons and remove the '+' prefix
reasons_list <- strsplit(gsub("^\\+", "", our_interest$reason), ";")

# Step 2: Identify unique reasons
unique_reasons <- unique(unlist(reasons_list))

# Step 3: Create dummy variables for each unique reason
for(reason in unique_reasons) {
  our_interest[[reason]] <- sapply(reasons_list, function(x) as.integer(reason %in% x))
}

# Select only dummy variables (and any other numeric variables you want to include)
numeric_data <- our_interest[, sapply(our_interest, is.numeric)]

# Compute the correlation matrix for numeric data
cor_matrix <- cor(numeric_data)

# Melt the correlation matrix for visualization
melted_cor_matrix <- reshape2::melt(cor_matrix)

# Define a threshold for displaying text
threshold <- 0.5

# Plot the correlation matrix
ggplot(data = melted_cor_matrix, aes(x=Var1, y=Var2, fill=value)) +
```

```
    geom_tile() +
    geom_text(aes(label = ifelse(abs(value) > threshold, round(value, 2), '')), color = "black", size =
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                         midpoint = 0, limit = c(-1,1), space = "Lab",
                         name="Pearson\nCorrelation") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 4),
          axis.text.y = element_text(size = 6),
          axis.title = element_blank()) +
    coord_fixed() # Ensure square cells
```

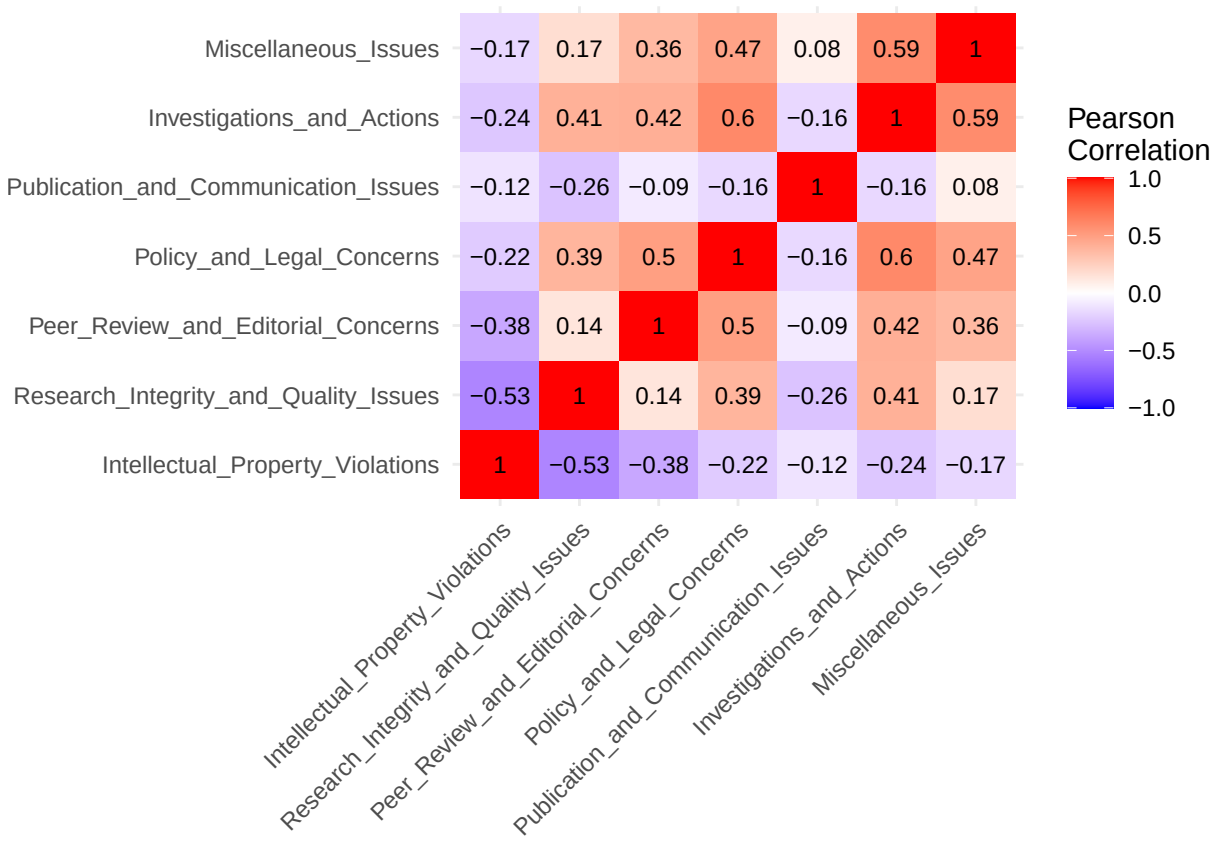


This looks a little difficult to comprehend.I am going to map them to something more easy to understand and check again.

Creating broader themes.

```
# Define the patterns for each category
patterns <- list(
  Intellectual_Property_Violations = c("Plagiarism", "Duplication", "Euphemisms for Plagiarism", "False/
  Research_Integrity_and_Quality_Issues = c("Not Reproducible", "Unreliable Results", "Error in Text", "
  Peer_Review_and_Editorial_Concerns = c("Fake Peer Review", "Rogue Editor", "Investigation by Journal/
  Policy_and_Legal_Concerns = c("Breach of Policy", "Issues about Referencing", "Legal Reasons", "Lack o
  Publication_and_Communication_Issues = c("Withdrawal", "Limited or No Information", "Notice - Lack of
  Investigations_and_Actions = c("Investigation by Third Party", "Doing the Right Thing"),
  Miscellaneous_Issues = c("Date of Retraction", "Randomly Generated Content", "Original Data not Provi
)

# Function to create dummy variables for categories based on the presence of certain patterns
```

```r
create_dummy_vars <- function(df, patterns) {
  for (category in names(patterns)) {
    pattern <- patterns[[category]]
    df[[category]] <- as.integer(sapply(df$reason, function(x) {
      any(sapply(pattern, function(y) str_detect(x, regex(y, ignore_case = TRUE))))
    }))
  }
  return(df)
}

# Apply the function to the data frame
our_interest <- create_dummy_vars(our_interest, patterns)

# Ensure these columns exist in your our_interest dataframe
selected_columns <- c("Intellectual_Property_Violations", "Research_Integrity_and_Quality_Issues",
                      "Peer_Review_and_Editorial_Concerns", "Policy_and_Legal_Concerns",
                      "Publication_and_Communication_Issues", "Investigations_and_Actions",
                      "Miscellaneous_Issues")

# Check if all selected columns are present in our_interest
if(all(selected_columns %in% names(our_interest))) {
    # Select only the specified dummy variable columns
    dummy_data <- our_interest[, selected_columns]

    # Calculate the correlation matrix
    cor_matrix <- cor(dummy_data, use = "complete.obs") # use complete.obs to handle NA values

    # Melt the correlation matrix for visualization
    melted_cor_matrix <- reshape2::melt(cor_matrix)

    # Plot the correlation matrix with numbers
    ggplot(data = melted_cor_matrix, aes(x=Var1, y=Var2, fill=value)) +
        geom_tile() +
        geom_text(aes(label = round(value, 2)), color = "black", size = 3) +
        scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                             midpoint = 0, limit = c(-1,1), space = "Lab",
                             name="Pearson\nCorrelation") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
              axis.title = element_blank())

} else {
    stop("Not all specified columns exist in the dataframe.")
}
```



Here are my key observations in this:

1. It looks like Investigations and Misc issues keep co-occuring
2. Policy and legal concerns seem to do so too.
3. Investigations seem to follow the Peer review concerns.
4. Investigations also seem to follow Research integrity issues.

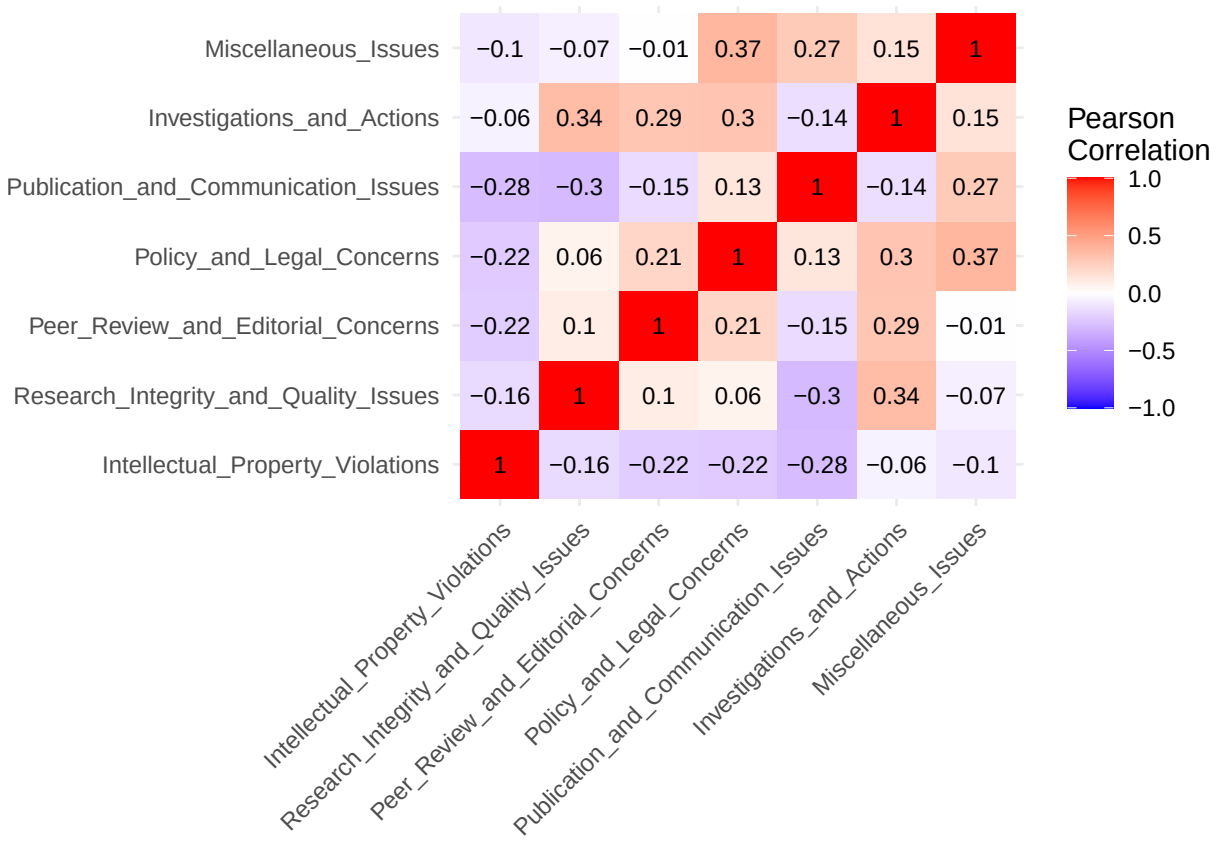
Now this begs the question, is there something different that's going on in the general set of papers?

Let me just check that out here

#trying to find out if there are correlations in the broader dataset of all retractions.

```
# Define the patterns for each category
patterns <- list(
  Intellectual_Property_Violations = c("Plagiarism", "Duplication", "Euphemisms for Plagiarism", "False
  Research_Integrity_and_Quality_Issues = c("Not Reproducible", "Unreliable Results", "Error in Text",
  Peer_Review_and_Editorial_Concerns = c("Fake Peer Review", "Rogue Editor", "Investigation by Journal/
  Policy_and_Legal_Concerns = c("Breach of Policy", "Issues about Referencing", "Legal Reasons", "Lack
  Publication_and_Communication_Issues = c("Withdrawal", "Limited or No Information", "Notice - Lack of
  Investigations_and_Actions = c("Investigation by Third Party", "Doing the Right Thing"),
  Miscellaneous_Issues = c("Date of Retraction", "Randomly Generated Content", "Original Data not Provi
)

# Function to create dummy variables for categories based on the presence of certain patterns
create_dummy_vars <- function(df, patterns) {
  for (category in names(patterns)) {
    pattern <- patterns[[category]]
    df[[category]] <- as.integer(sapply(df$reason, function(x) {
      any(sapply(pattern, function(y) str_detect(x, regex(y, ignore_case = TRUE))))
```

```
    }))
  }
  return(df)
}

our_interest_temp<- our_interest

# Apply the function to the data frame
our_interest <- create_dummy_vars(data, patterns)

# Ensure these columns exist in your our_interest dataframe
selected_columns <- c("Intellectual_Property_Violations", "Research_Integrity_and_Quality_Issues",
                      "Peer_Review_and_Editorial_Concerns", "Policy_and_Legal_Concerns",
                      "Publication_and_Communication_Issues", "Investigations_and_Actions",
                      "Miscellaneous_Issues")

# Check if all selected columns are present in our_interest
if(all(selected_columns %in% names(our_interest))) {
    # Select only the specified dummy variable columns
    dummy_data <- our_interest[, selected_columns]

    # Calculate the correlation matrix
    cor_matrix <- cor(dummy_data, use = "complete.obs") # use complete.obs to handle NA values

    # Melt the correlation matrix for visualization
    melted_cor_matrix <- reshape2::melt(cor_matrix)

    # Plot the correlation matrix with numbers
    ggplot(data = melted_cor_matrix, aes(x=Var1, y=Var2, fill=value)) +
        geom_tile() +
        geom_text(aes(label = round(value, 2)), color = "black", size = 3) +
        scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                             midpoint = 0, limit = c(-1,1), space = "Lab",
                             name="Pearson\nCorrelation") +
        theme_minimal() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1),
              axis.title = element_blank())

} else {
    stop("Not all specified columns exist in the dataframe.")
}
```




```
our_interest<- our_interest_temp
```

This presents the an interesting question. Are there two different stories here?

COntrasting management with other, non managmenet fields. Creating two separate subsets now - one for business related, and one for non business

```
# Define business-related subjects
business_subjects <- c("Business - Management",
                       "Business - Economics",
                       "Business - Marketing",
                       "Business - General",
                       "Business - Manufacturing",
                       "Business - Accounting")

# Create a single pattern string for matching
pattern <- paste(business_subjects, collapse = "|")

# This is the Business related subjects
data_business <- data %>%
  filter(str_detect(subject, pattern))

# This is the Non Business related subjects
data_nonbusiness <- data %>%
  filter(!str_detect(subject, pattern))

data_temp<- data
```

Trying to plot how retractions have been in Non Management disciplines

```r
# Prepare data
yearly_data <- data_nonbusiness %>%
  group_by(retraction_year) %>%
  summarize(average_duration = mean(duration_in_months, na.rm = TRUE),
            retraction_count = n())

# Normalize the RetractionCount for better visualization
max_duration <- max(yearly_data$average_duration, na.rm = TRUE)
max_count <- max(yearly_data$retraction_count, na.rm = TRUE)
yearly_data$NormalizedCount <- yearly_data$retraction_count / max_count * max_duration

retractions_nonmanagement<- ggplot(yearly_data, aes(x = retraction_year)) +
  geom_line(aes(y = average_duration, group = 1), color = "blue") +
  geom_point(aes(y = average_duration), color = "blue") +
  geom_bar(aes(y = NormalizedCount), stat = "identity", fill = "red", alpha = 0.5) +
    scale_x_continuous(limits = c(1990, NA)) +  # Limiting x-axis to start from 1970+

  scale_y_continuous(name = "Average Duration in Months",
                     sec.axis = sec_axis(~ . * max_count / max_duration,
                                         name = "Number of Retractions")) +
  labs(title = "Retractions over the Years: Duration and Count (Non Management Disciplines)",
       x = "Retraction Year") +
  theme_minimal()
```

Trying to plot how retractions have been in Management disciplines

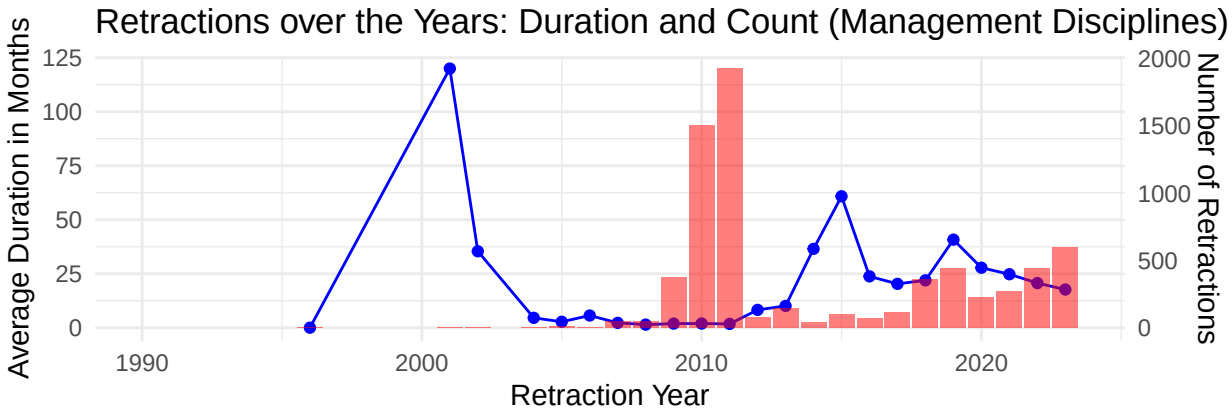
```r
# Prepare data
yearly_data <- data_business %>%
  group_by(retraction_year) %>%
  summarize(average_duration = mean(duration_in_months, na.rm = TRUE),
            retraction_count = n())

# Normalize the RetractionCount for better visualization
max_duration <- max(yearly_data$average_duration, na.rm = TRUE)
max_count <- max(yearly_data$retraction_count, na.rm = TRUE)
yearly_data$NormalizedCount <- yearly_data$retraction_count / max_count * max_duration

retractions_management<- ggplot(yearly_data, aes(x = retraction_year)) +
  geom_line(aes(y = average_duration, group = 1), color = "blue") +
  geom_point(aes(y = average_duration), color = "blue") +
  geom_bar(aes(y = NormalizedCount), stat = "identity", fill = "red", alpha = 0.5) +
    scale_x_continuous(limits = c(1990, NA)) +  # Limiting x-axis to start from 1970+

  scale_y_continuous(name = "Average Duration in Months",
                     sec.axis = sec_axis(~ . * max_count / max_duration,
                                         name = "Number of Retractions")) +
  labs(title = "Retractions over the Years: Duration and Count (Management Disciplines)",
       x = "Retraction Year") +
  theme_minimal()
```

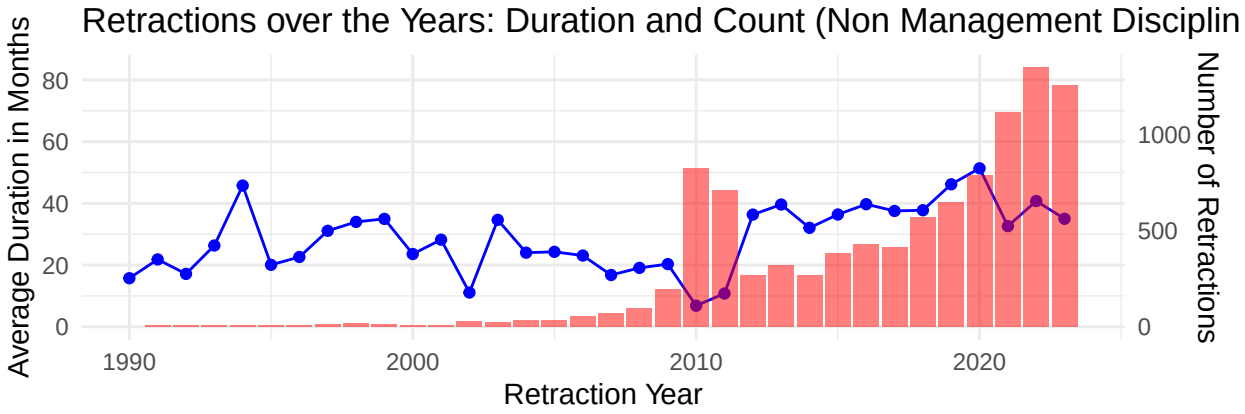
Displaying the plot

```r
grid.arrange(retractions_management, retractions_nonmanagement, ncol = 1)
```

```
## Warning: Removed 2 rows containing missing values (`position_stack()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_line()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).
## Warning: Removed 23 rows containing missing values (`position_stack()`).
## Warning: Removed 23 rows containing missing values (`geom_line()`).
## Warning: Removed 23 rows containing missing values (`geom_point()`).
## Warning: Removed 1 rows containing missing values (`geom_bar()`).
```

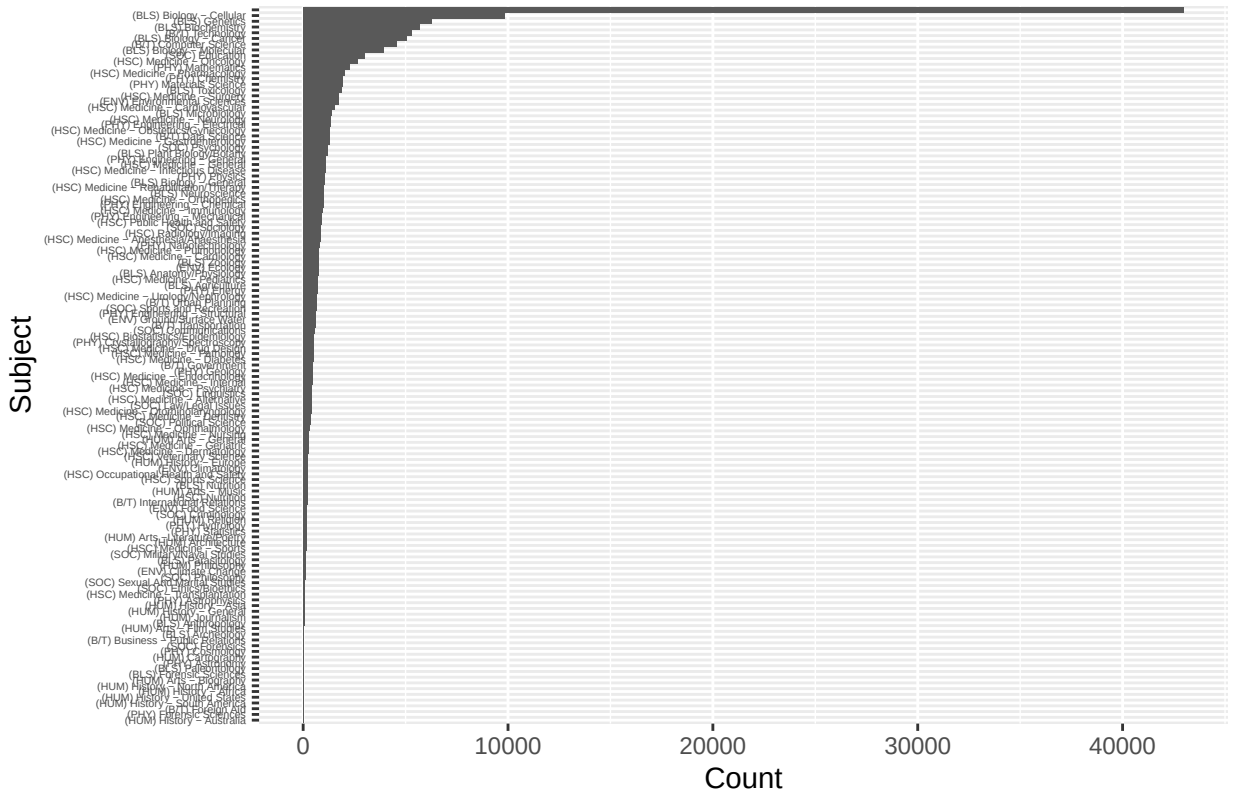




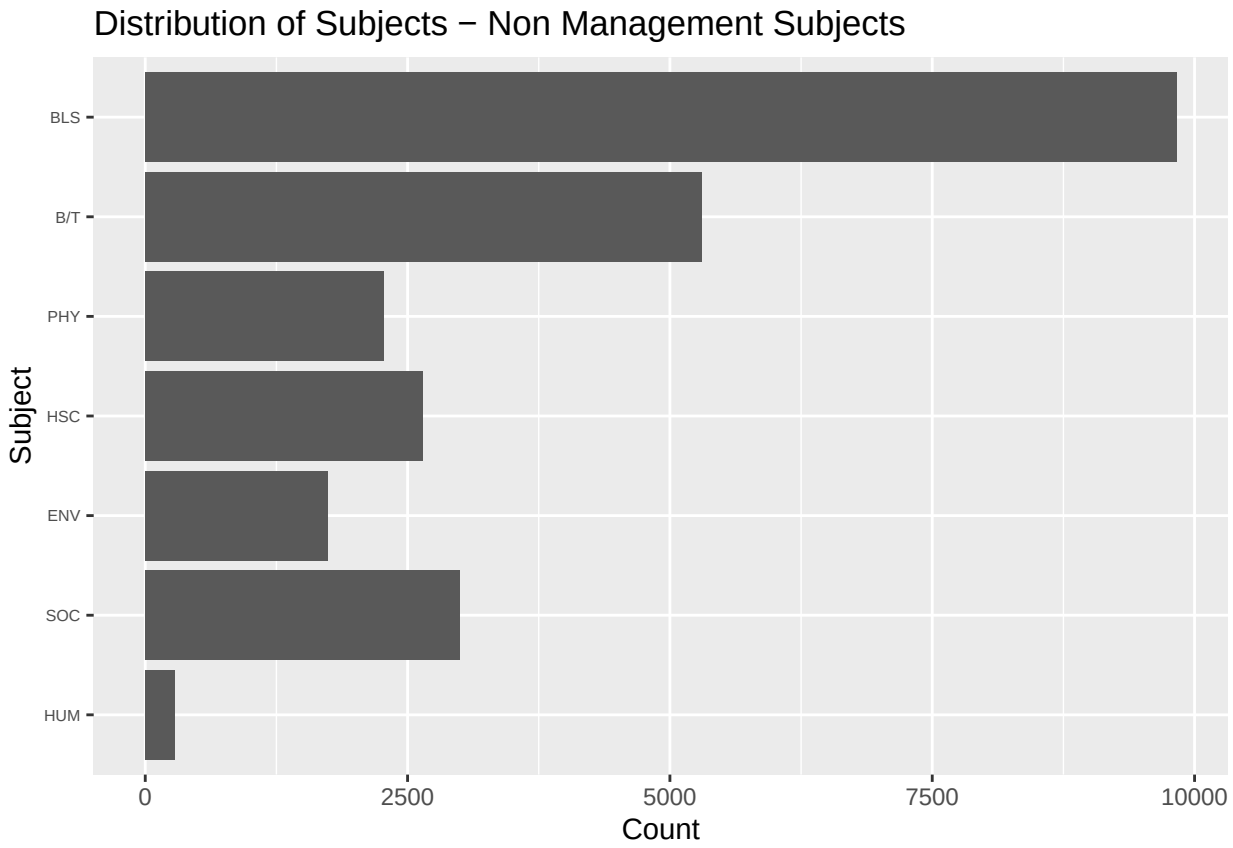
Trying to understand the distribution of subjects now.

Distribution of Subjects - Non Management Subjects

```
# Separate the subjects into individual rows
data_subjects <- data_nonbusiness %>%
                 separate_rows(subject, sep = ";\\s*")

# Count the occurrences of each subject
subject_count <- data_subjects %>%
                 count(subject, sort = TRUE)

# Create a horizontal bar chart
ggplot(subject_count, aes(y = reorder(subject, n), x = n)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme(axis.text.y = element_text(size = 4)) +
    labs(y = "Subject", x = "Count", title = "Distribution of Subjects - Non Management")
```

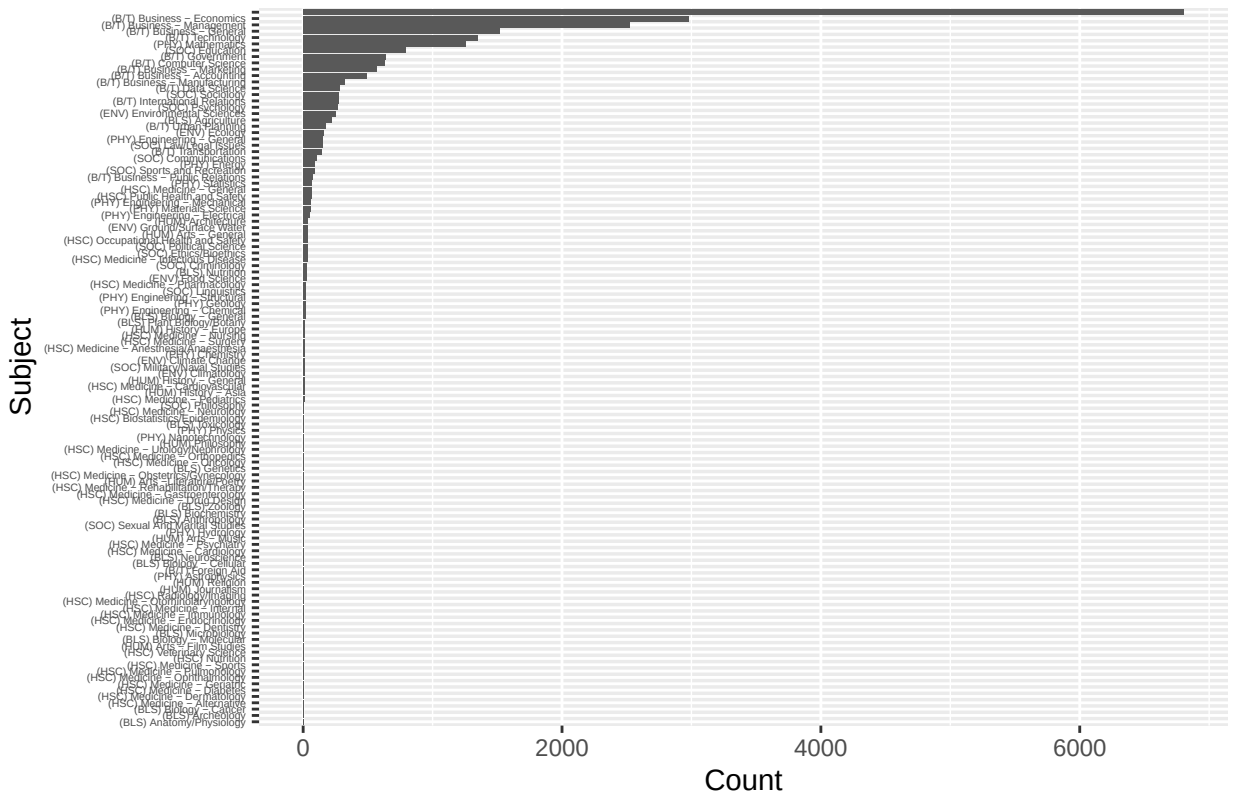



```r
# Rename the subject to the broader theme that's written in the brackets
subject_count <- subject_count %>%
  mutate(subject = str_match(subject, "\\(([^)]+)\\)")[,2])

# Remove NA values that might have been introduced if there were subjects without brackets
subject_count <- subject_count %>%
  filter(!is.na(subject))

# Create a horizontal bar chart
ggplot(subject_count, aes(y = reorder(subject, n), x = n)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.y = element_text(size = 6)) +
  labs(y = "Subject", x = "Count", title = "Distribution of Subjects - Non Management Subjects")
```

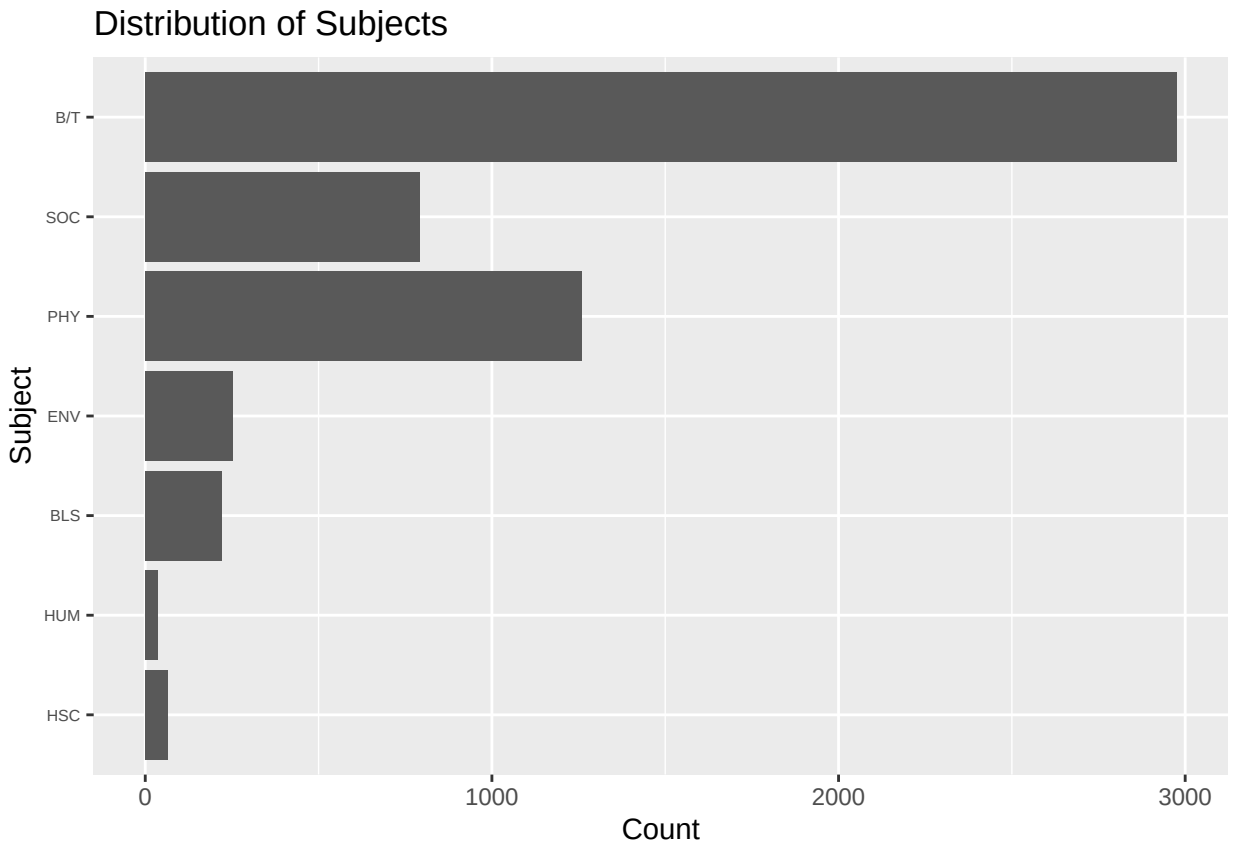


Distribution of Subjects - Management Subjects

```
# Separate the subjects into individual rows
data_subjects <- data_business %>%
                 separate_rows(subject, sep = ";\\s*")

# Count the occurrences of each subject
subject_count <- data_subjects %>%
                 count(subject, sort = TRUE)

# Create a horizontal bar chart
ggplot(subject_count, aes(y = reorder(subject, n), x = n)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme(axis.text.y = element_text(size = 4)) +
    labs(y = "Subject", x = "Count", title = "Distribution of Subjects - Management")
```



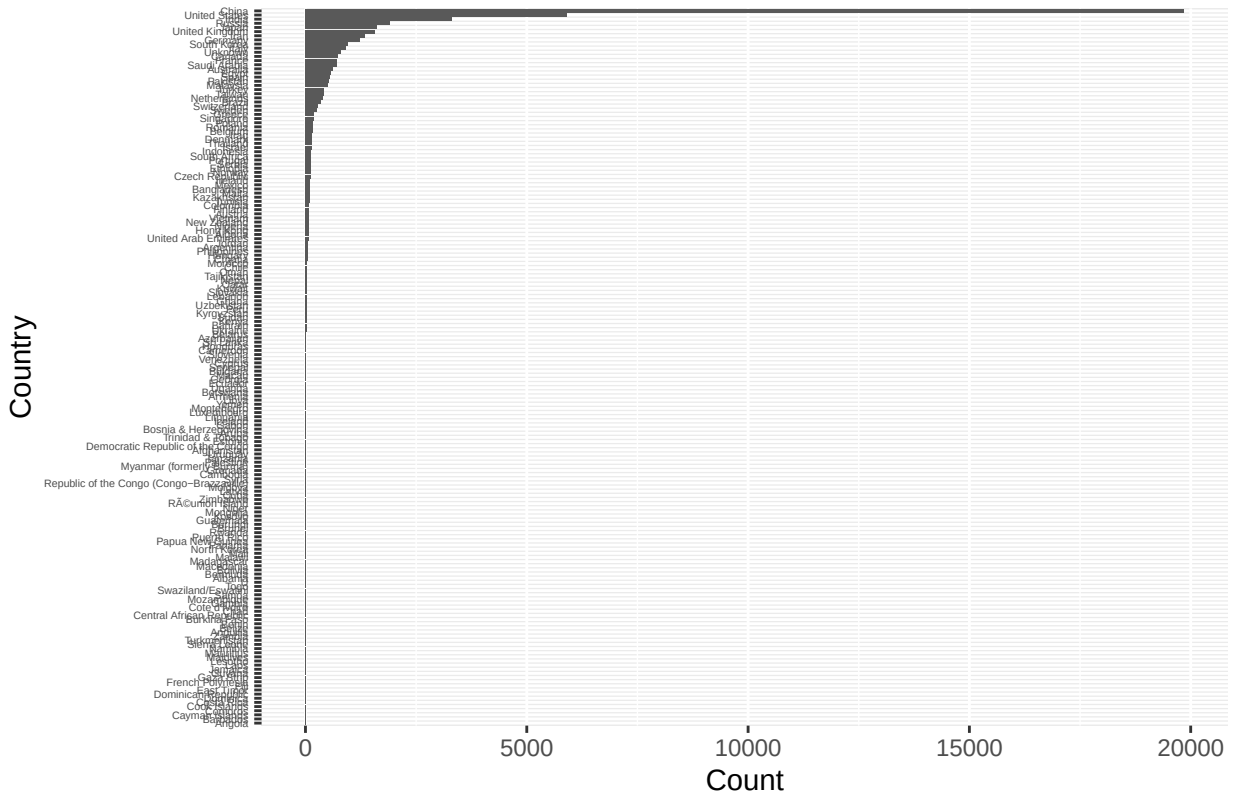

```
# Rename the subject to the broader theme that's written in the brackets
subject_count <- subject_count %>%
  mutate(subject = str_match(subject, "\\(([^)]+)\\)")[,2])

# Remove NA values that might have been introduced if there were subjects without brackets
subject_count <- subject_count %>%
  filter(!is.na(subject))

# Create a horizontal bar chart
ggplot(subject_count, aes(y = reorder(subject, n), x = n)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.y = element_text(size = 6)) +
  labs(y = "Subject", x = "Count", title = "Distribution of Subjects")
```



Note: You see that there are other non management areas also linked here. That's because a lot of the papers have also been listed as science, environment and sociology, etc.

Trying to understand the distribution of countries now

```
# Separate the subjects into individual rows
data_country <- data_nonbusiness %>%
                separate_rows(country, sep = ";\\s*")

# Count the occurrences of each subject
country_count <- data_country %>%
                count(country, sort = TRUE)

# Create a horizontal bar chart
ggplot(country_count, aes(y = reorder(country, n), x = n)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme(axis.text.y = element_text(size = 4)) +
    labs(y = "Country", x = "Count", title = "Distribution of Countries - Non Management")
```

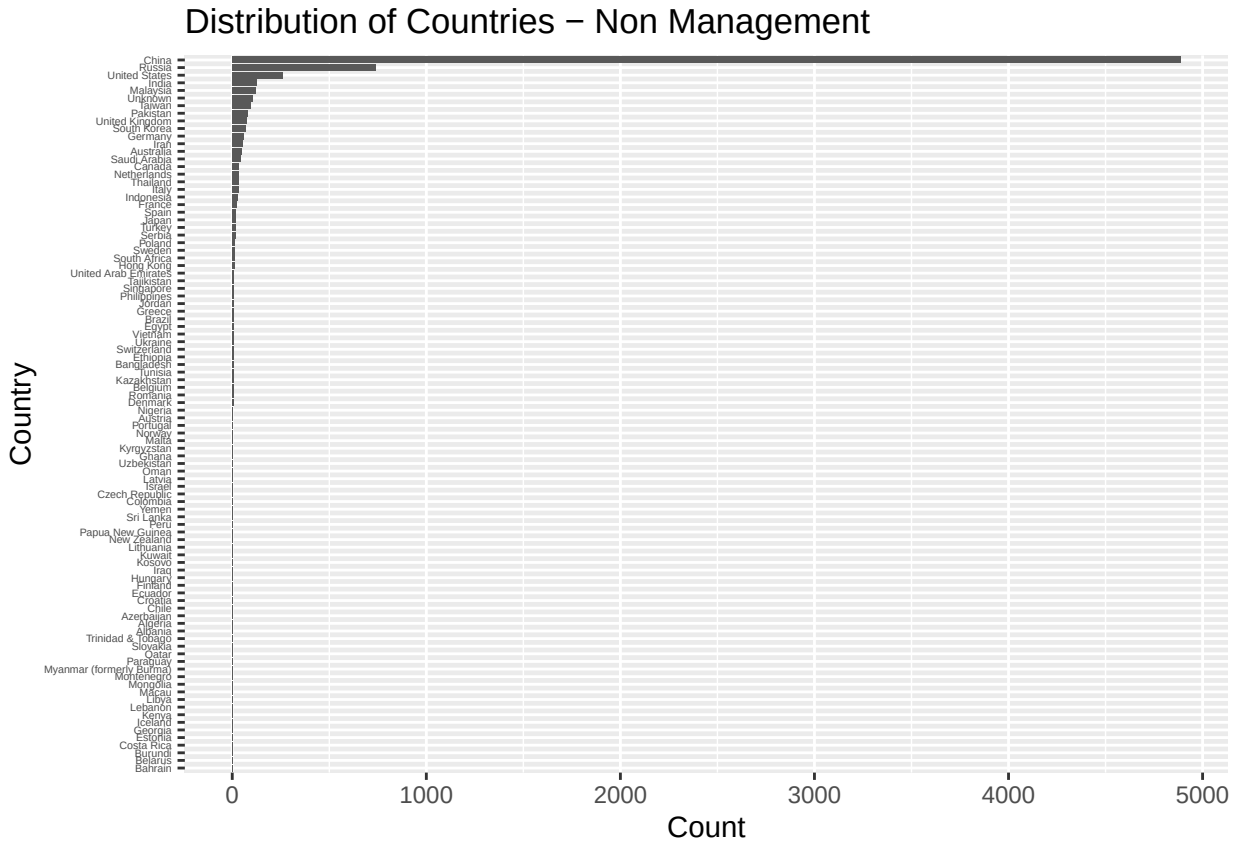



```
length(unique(country_count$country))
```

```
## [1] 173
```

There appear to be authors from 173 countries participating here.

```
# Separate the subjects into individual rows
data_country <- data_business %>%
                separate_rows(country, sep = ";\\s*")

# Count the occurrences of each subject
country_count <- data_country %>%
                count(country, sort = TRUE)

# Create a horizontal bar chart
ggplot(country_count, aes(y = reorder(country, n), x = n)) +
    geom_bar(stat = "identity", position = "dodge") +
    theme(axis.text.y = element_text(size = 4)) +
    labs(y = "Country", x = "Count", title = "Distribution of Countries - Non Management")
```




```r
length(unique(country_count$country))
```

```
## [1] 94
```

There appear to be authors from 94 countries participating here.

Trying to understand the distribution of institutions now

```r
# Separate the subjects into individual rows
data_institution <- data_nonbusiness %>%
                  separate_rows(institution, sep = ";\\s*")

# Count the occurrences of each subject
institution_count <- data_institution %>%
                  count(institution, sort = TRUE)

length(unique(institution_count$institution))
```

```
## [1] 77610
```

There appear to be authors from 77610 institutions participated here.

```r
# Separate the subjects into individual rows
data_institution <- data_business %>%
                  separate_rows(institution, sep = ";\\s*")

# Count the occurrences of each subject
institution_count <- data_institution %>%
                  count(institution, sort = TRUE)
```

```r
length(unique(institution_count$institution))
```

```
## [1] 8240
```

There appear to be authors from 8240 institutions participated here.

Checking on repeat offenders

```r
# Separate the subjects into individual rows
data_author <- data_nonbusiness %>%
                separate_rows(author, sep = ";\\s*")

# Count the occurrences of each subject
author_count <- data_author %>%
                count(author, sort = TRUE)

author_count_nonbusiness <-author_count

length(unique(author_count$author))
```

```
## [1] 122573
```

There are about 1,22,573 unique authors in the non business dataset.

```r
length(unique(author_count$author))/length(data_nonbusiness$record_id)
```

```
## [1] 2.850535
```

There have on average been, 2.85 authors per paper.

```r
length(unique(author_count$author))/length(data_nonbusiness$record_id)
```

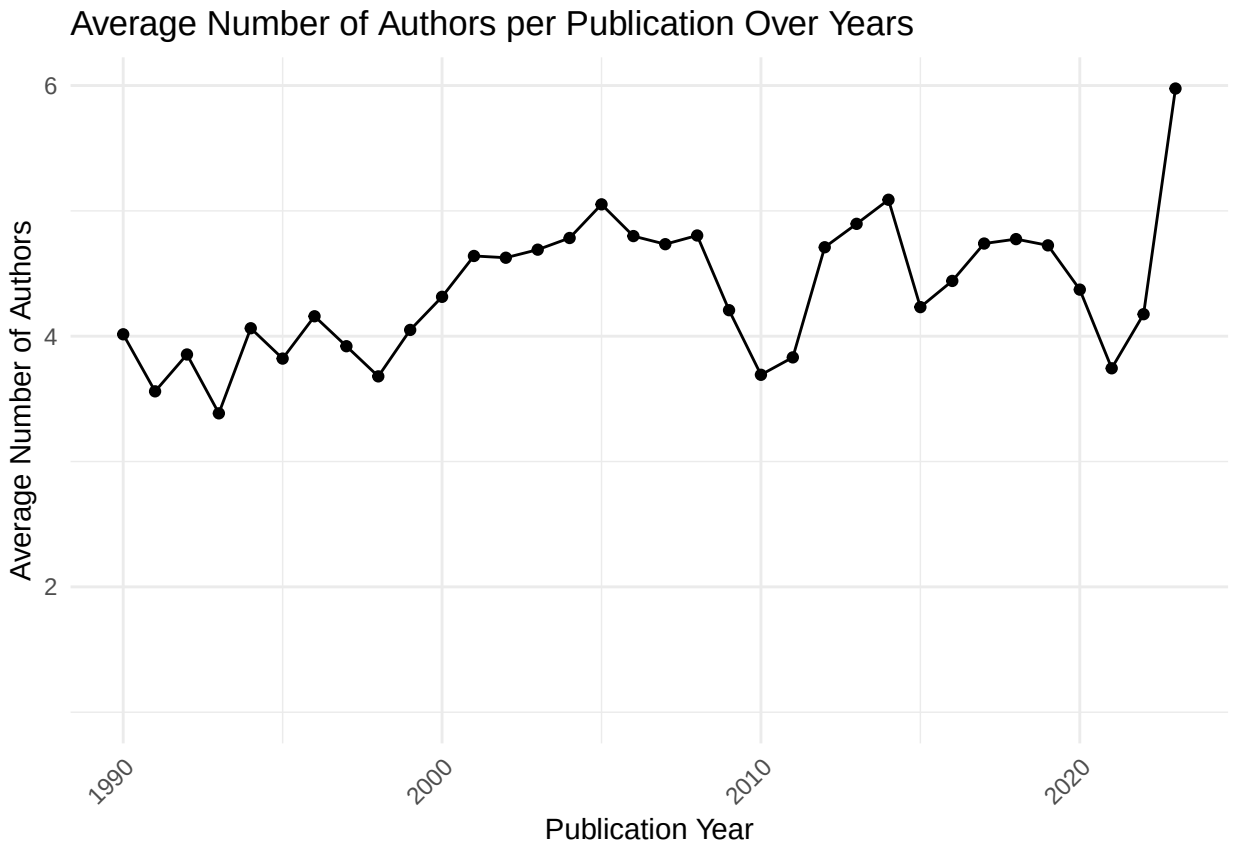
```
## [1] 2.850535
```

```r
# Step 1: Add a column with the number of authors per publication
data_nonbusiness <- data_nonbusiness %>%
  mutate(
    num_authors = sapply(strsplit(as.character(author), ";"), length),
    publication_year = year(original_paper_date)
  )

# Step 2: Group by publication year and calculate the average number of authors
average_authors_per_year <- data_nonbusiness %>%
  group_by(publication_year) %>%
  summarise(average_authors = mean(num_authors, na.rm = TRUE))

# Plotting the average number of authors per year
ggplot(average_authors_per_year, aes(x = publication_year, y = average_authors)) +
  geom_line() +  # Line plot
  geom_point() + # Adding points to each year
  theme_minimal() +
  labs(
    title = "Average Number of Authors per Publication Over Years",
    x = "Publication Year",
    y = "Average Number of Authors"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Adjusting x-axis labels for better readability
```

```
  )+
    scale_x_continuous(limits = c(1990, max(average_authors_per_year$publication_year))) # Limiting x-a
```

```
## Warning: Removed 42 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 42 rows containing missing values (`geom_point()`).
```

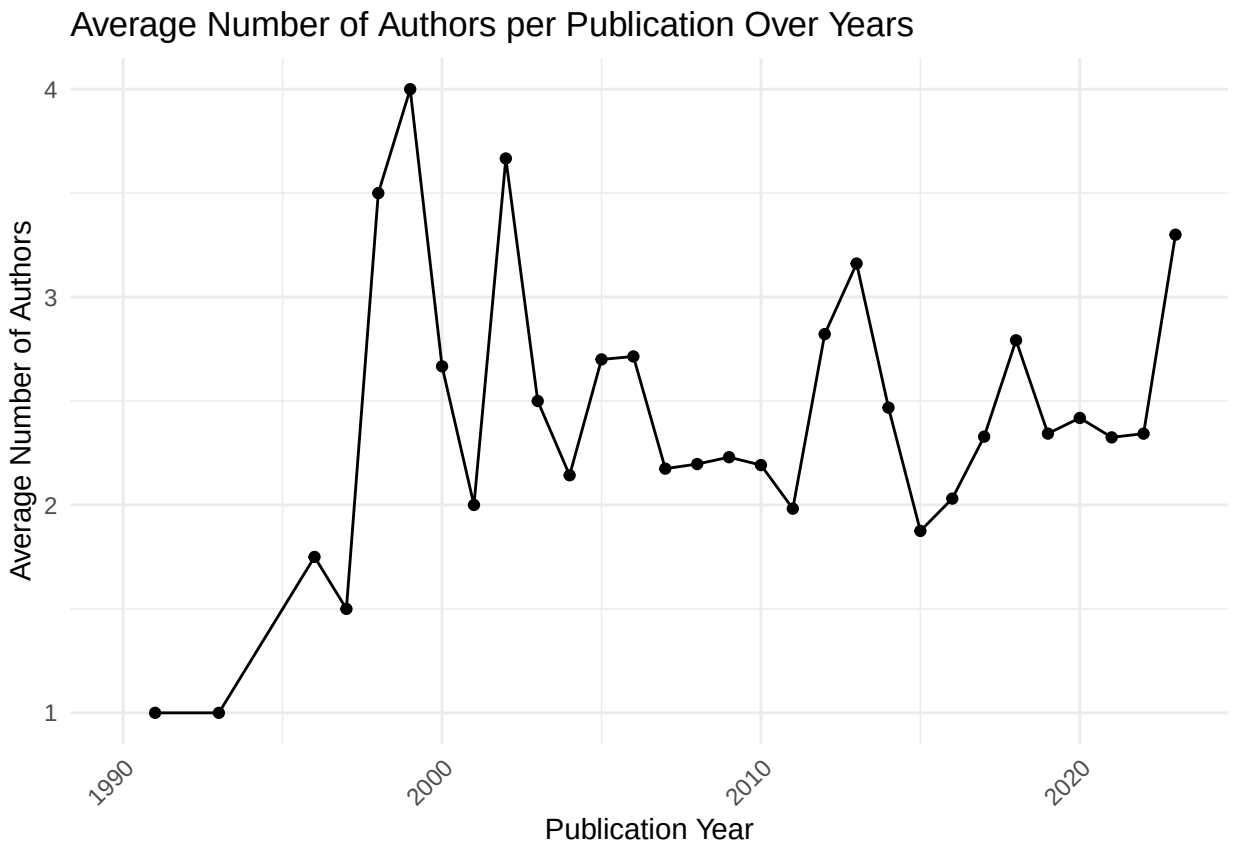



```
# Step 1: Add a column with the number of authors per publication
data_business <- data_business %>%
  mutate(
    num_authors = sapply(strsplit(as.character(author), ";"), length),
    publication_year = year(original_paper_date)
  )

# Step 2: Group by publication year and calculate the average number of authors
average_authors_per_year <- data_business %>%
  group_by(publication_year) %>%
  summarise(average_authors = mean(num_authors, na.rm = TRUE))

# Plotting the average number of authors per year
ggplot(average_authors_per_year, aes(x = publication_year, y = average_authors)) +
  geom_line() +  # Line plot
  geom_point() + # Adding points to each year
  theme_minimal() +
  labs(
    title = "Average Number of Authors per Publication Over Years",
    x = "Publication Year",
    y = "Average Number of Authors"
```

```
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Adjusting x-axis labels for better readability
  )+
    scale_x_continuous(limits = c(1990, max(average_authors_per_year$publication_year))) # Limiting x-a
```

```
## Warning: Removed 2 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



Preparing a joint plot

```
# Prepare nonbusiness data
data_nonbusiness <- data_nonbusiness %>%
  mutate(
    num_authors = sapply(strsplit(as.character(author), ";"), length),
    publication_year = year(original_paper_date),
    type = "Nonbusiness"
  )

# Prepare business data
data_business <- data_business %>%
  mutate(
    num_authors = sapply(strsplit(as.character(author), ";"), length),
    publication_year = year(original_paper_date),
    type = "Business"
  )
```

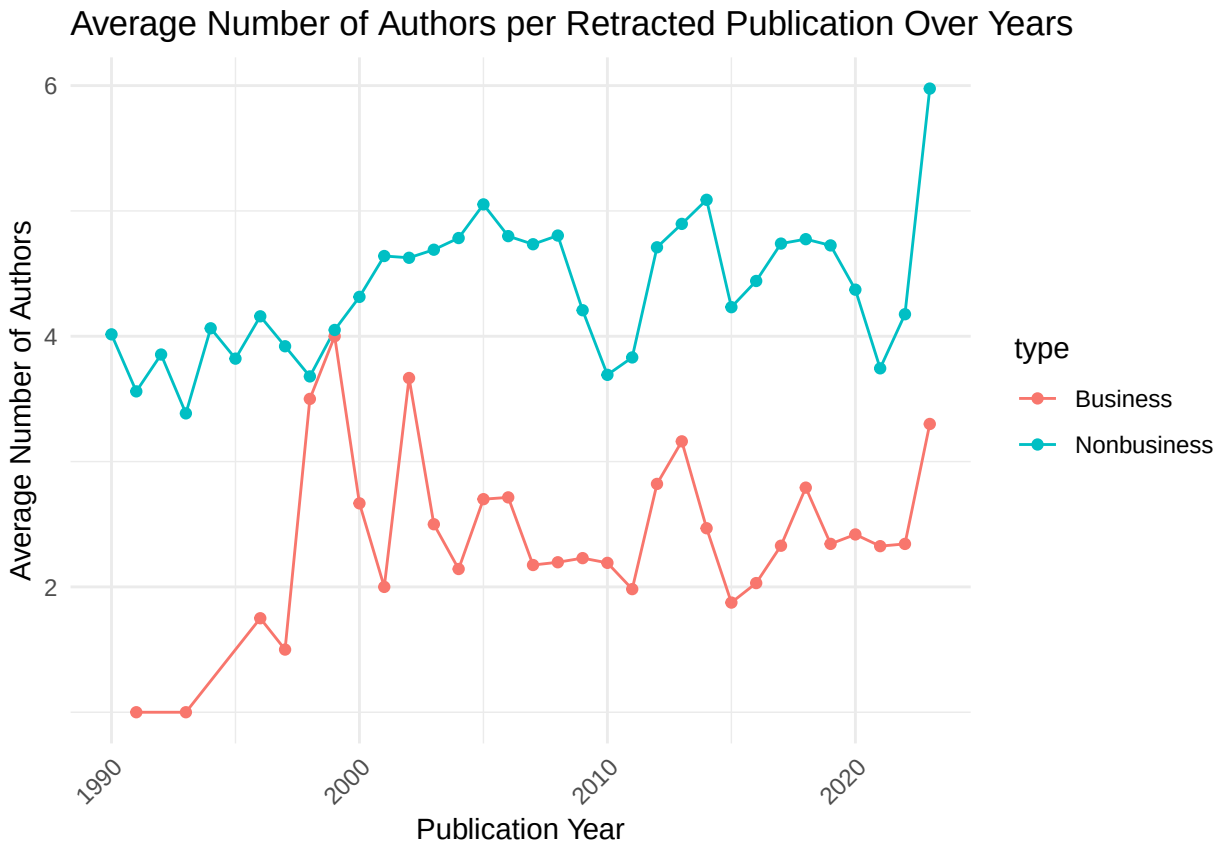
```r
# Combine the datasets
combined_data <- bind_rows(data_nonbusiness, data_business)

# Group by publication year and type, then calculate the average number of authors
average_authors_per_year_type <- combined_data %>%
  group_by(publication_year, type) %>%
  summarise(average_authors = mean(num_authors, na.rm = TRUE))
```

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

```r
# Plotting the average number of authors per year for both types
ggplot(average_authors_per_year_type, aes(x = publication_year, y = average_authors, color = type)) +
  geom_line() +  # Line plot
  geom_point() + # Adding points to each year
  theme_minimal() +
  labs(
    title = "Average Number of Authors per Retracted Publication Over Years",
    x = "Publication Year",
    y = "Average Number of Authors"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1) # Adjusting x-axis labels for better readability
  )+
  scale_x_continuous(limits = c(1990, max(average_authors_per_year_type$publication_year))) #
```

```
## Warning: Removed 44 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 44 rows containing missing values (`geom_point()`).
```

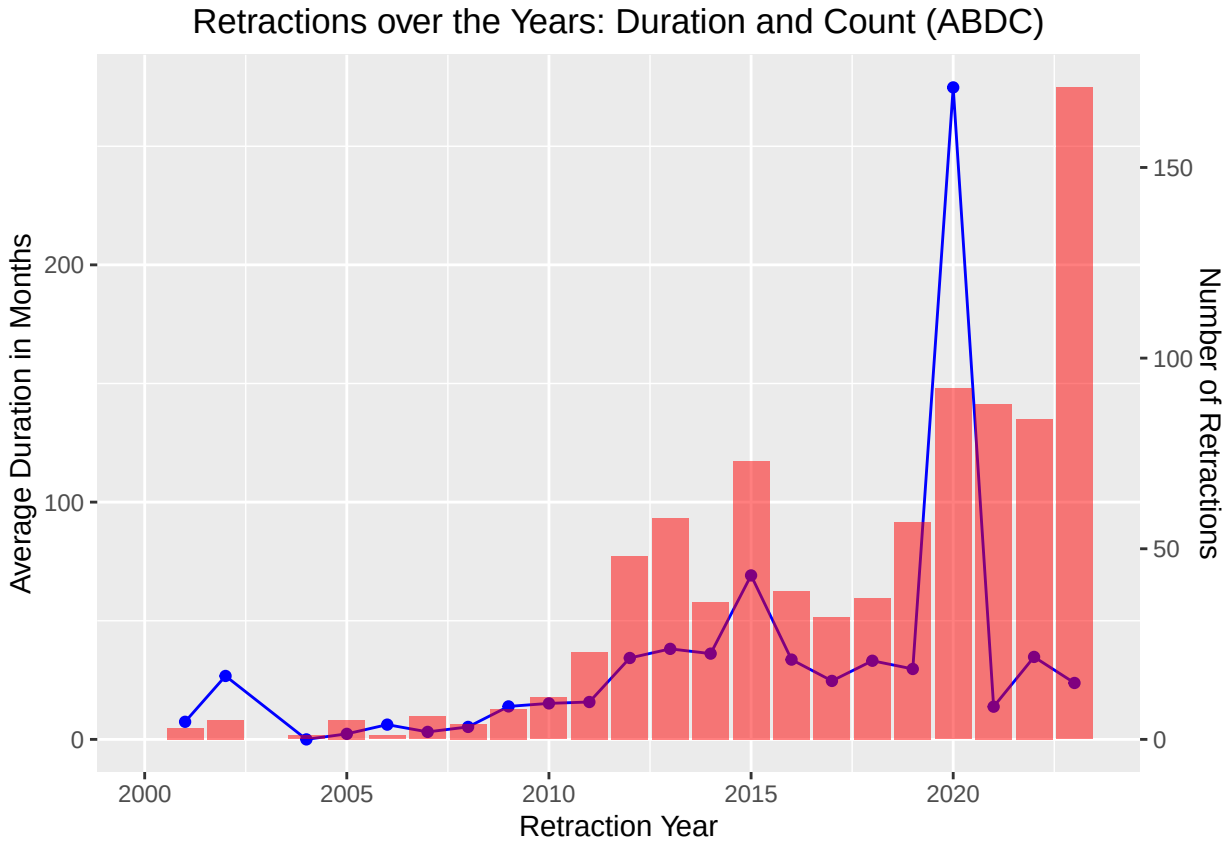


From this, it does look like management related publications (as classified by the authors) does not seem to have as many authors as non management related articles.

```
# Prepare data
yearly_data <- ret_int_abdc %>%
  group_by(retraction_year) %>%
  summarize(average_duration = mean(duration_in_months, na.rm = TRUE),
            retraction_count = n())

# Normalize the RetractionCount for better visualization
max_duration <- max(yearly_data$average_duration, na.rm = TRUE)
max_count <- max(yearly_data$retraction_count, na.rm = TRUE)
yearly_data$NormalizedCount <- yearly_data$retraction_count / max_count * max_duration

retractions_management<- ggplot(yearly_data, aes(x = retraction_year)) +
  geom_line(aes(y = average_duration, group = 1), color = "blue") +
  geom_point(aes(y = average_duration), color = "blue") +
  geom_bar(aes(y = NormalizedCount), stat = "identity", fill = "red", alpha = 0.5) +
    scale_x_continuous(limits = c(2000, NA)) +

  scale_y_continuous(name = "Average Duration in Months",
                     sec.axis = sec_axis(~ . * max_count / max_duration,
                                         name = "Number of Retractions")) +
  labs(title = "Retractions over the Years: Duration and Count (ABDC)",
       x = "Retraction Year") +
  theme(
    plot.title = element_text(hjust = 0.5),  # Center-align the title
    plot.subtitle = element_text(hjust = 0.5),  # Center-align the subtitle if you have one
```

```
    axis.title.x = element_text(hjust = 0.5)  # Center-align the x-axis label
  )


retractions_management
```

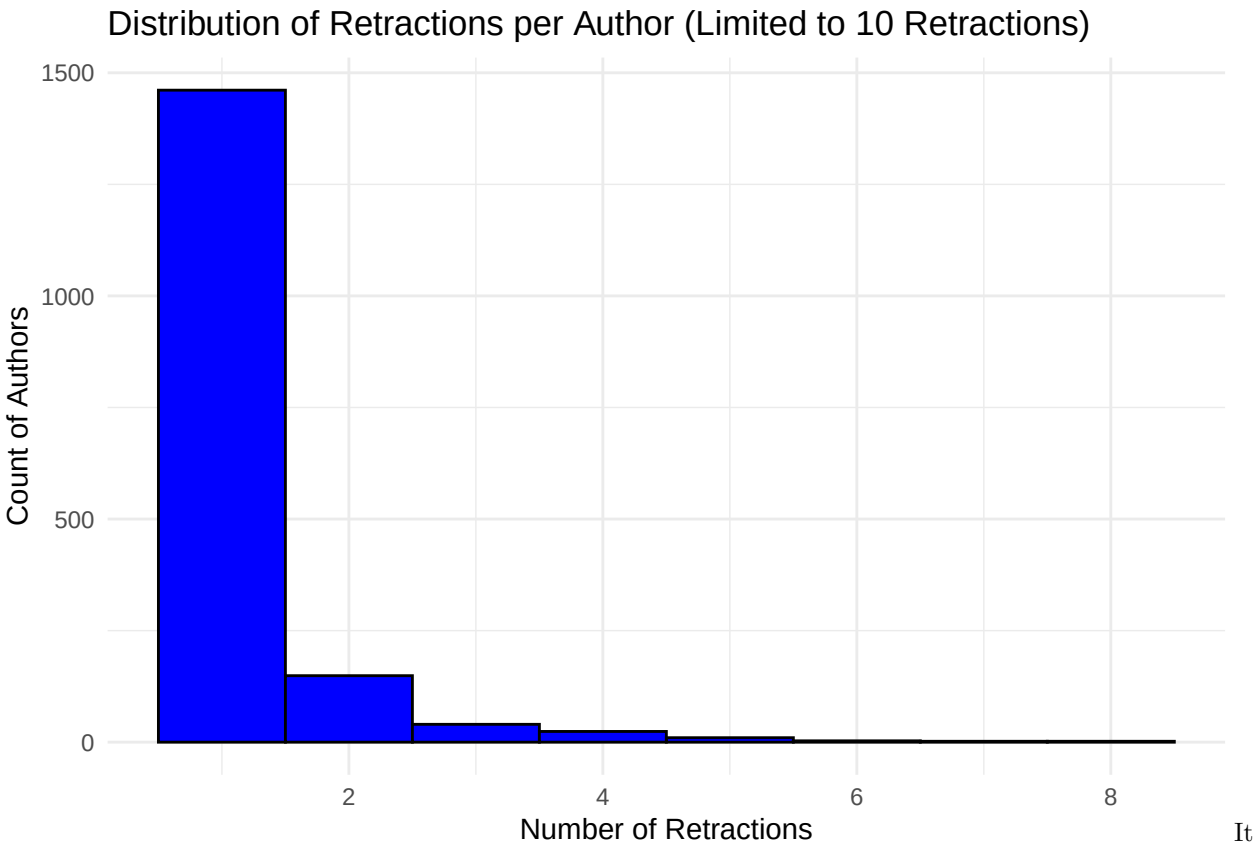
```
## Warning: Removed 1 rows containing missing values (`position_stack()`).
## Warning: Removed 1 row containing missing values (`geom_line()`).
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



From this data set, let's try to pull out the names of every author.. and see if there are 1. repeat offenders 2. How many total authors are there with retraction related issues

```
# Expand the dataset so each row represents a record-author combination
expanded_dataset <- ret_int_abdc %>%
  separate_rows(author, sep = ";") %>%
  mutate(author = trimws(author))  # Remove any leading/trailing whitespace

# Count the number of retractions for each author
author_retraction_counts <- expanded_dataset %>%
  group_by(author) %>%
  summarise(retraction_count = n()) %>%
  filter(author != "")  # Remove empty author entries if any


# Define a threshold for the maximum number of retractions per author
threshold = 10  # Adjust this number based on your data and needs
```

```r
# Filter the data to exclude outliers
filtered_author_retraction_counts <- author_retraction_counts %>%
  filter(retraction_count <= threshold)

ggplot(filtered_author_retraction_counts, aes(x = retraction_count)) +
geom_histogram(binwidth = 1, fill = "blue", color = "black") +
labs(title = paste("Distribution of Retractions per Author (Limited to", threshold, "Retractions)"),
x = "Number of Retractions",
y = "Count of Authors") +
theme_minimal()
```



It looks like serial retractions are not very commonplace. There are hardly 200 authors among 2229 who have multiple counts of retraction.

Let's try to create an author author relationship network.

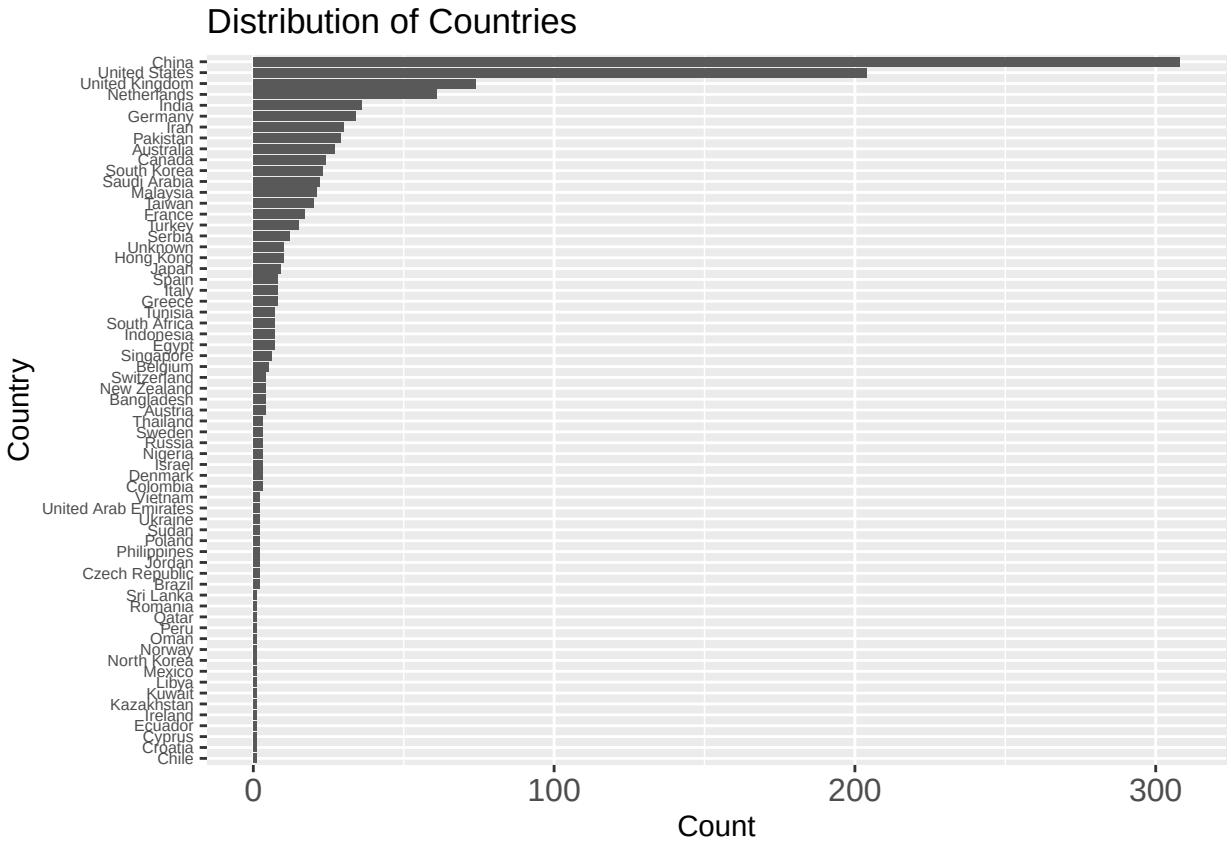
But that's not possible to do with the limited data that we have access to at the moment. We are going to have to explore who all the authors have linked up with in the past aswell.

One of the issues that becomes increasingly apparent is the nationality of the retractions. It appears that many of the retractions stem from nations like China.

Let me show you.

```r
# Ensure the 'country' column is a character vector if it is not already
our_interest <- ret_int_abdc %>%
  mutate(country = as.character(country))

# Assuming 'country' is a column where countries are separated by semicolons (and possibly followed by
```

```r
data_country <- our_interest %>%
  separate_rows(country, sep = ";\\s*") # sep regex matches a semicolon followed by any number of space

# Count the occurrences of each country
country_count <- data_country %>%
  count(country, sort = TRUE)

# Create a horizontal bar chart
ggplot(country_count, aes(x = reorder(country, n), y = n)) +
  geom_bar(stat = "identity") +
  coord_flip() +  # Flip the coordinates to make the bars horizontal
  theme(axis.text.x = element_text(size = 12),axis.text.y = element_text(size = 6)) +  # Adjust the siz
  labs(x = "Country", y = "Count", title = "Distribution of Countries")
```



it looks like China, US, UK, Netherlands and India are the key sources of retractions.

Can in be that this is a problem is located to within certain countries? or is this a truly corss cultural issue?

Let's try and doing a QAP correlation to find out.

Creating retraction-author-author network

```r
# Create a data frame where each row represents a single author for a given record_id
author_records <- our_interest %>%
  separate_rows(author, sep = ";") %>%
  distinct(record_id, author)

# Create a square matrix of authors, initialized to zero
unique_authors <- sort(unique(author_records$author))
```

```
coauthor_retraction_matrix <- matrix(0, nrow = length(unique_authors), ncol = length(unique_authors),
                        dimnames = list(unique_authors, unique_authors))

# Fill the matrix with 1's where authors share a record_id
for (i in seq_len(nrow(author_records))) {
  for (j in seq_len(nrow(author_records))) {
    if (author_records$record_id[i] == author_records$record_id[j]) {
      coauthor_retraction_matrix[author_records$author[i], author_records$author[j]] <- 1
    }
  }
}

# Remove self-loops if desired (authors compared with themselves)
diag(coauthor_retraction_matrix) <- 0
```

Now we have a dataset which has all the retraction author's retraction network.. But then, this is only the retracted papers, and not the rest of all the papers that they have worked on.

Building another matrix - this time to talk about the countries the authors are from.

```
# Splitting the author field as there could be multiple authors per record
author_country <- our_interest %>%
  separate_rows(author, sep = ";") %>%
  distinct(record_id, author, country)

# Creating a list of unique authors
unique_authors <- sort(unique(author_country$author))

# Creating an empty matrix to store the connections
author_country_matrix <- matrix(0, nrow = length(unique_authors), ncol = length(unique_authors))
rownames(author_country_matrix) <- unique_authors
colnames(author_country_matrix) <- unique_authors

# Filling the matrix
for (i in 1:length(unique_authors)) {
  for (j in 1:length(unique_authors)) {
    if (i != j) {
      # Check if authors i and j are associated with the same country
      common_countries <- intersect(author_country$country[author_country$author == unique_authors[i]],
                                    author_country$country[author_country$author == unique_authors[j]])
      if (length(common_countries) > 0) {
        author_country_matrix[i, j] <- 1
        author_country_matrix[j, i] <- 1 # Matrix is symmetric
      }
    }
  }
}

diag(author_country_matrix) <- 0
```

Trying to use asnipe and vegan together to run a QAP correlation.

```
library(asnipe)
library(vegan)
```

```
## Loading required package: permute
```

```
##
## Attaching package: 'permute'

## The following object is masked from 'package:igraph':
##
##     permute

## Loading required package: lattice

## This is vegan 2.6-4

##
## Attaching package: 'vegan'

## The following object is masked from 'package:igraph':
##
##     diversity
```

```
qap_result <- mantel(coauthor_retraction_matrix, author_country_matrix, permutations = 1000)
print(qap_result)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = coauthor_retraction_matrix, ydis = author_country_matrix,      permutations = 1000)
##
## Mantel statistic r: 0.1181
##       Significance: 0.000999
##
## Upper quantiles of permutations (null model):
##     90%     95%   97.5%     99%
## 0.00140 0.00183 0.00221 0.00291
## Permutation: free
## Number of permutations: 1000
```

This test suggests that there is a statistically significant, though relatively weak, positive correlation between the coauthor retraction matrix and the author country matrix. This means that there is some level of association between the patterns or relationships represented by the country relationship matrix and the retraction relationship matrix.

This is interesting..

Now, maybe there is a stronger relationship to detect if we check the institute level relations.

Creating an instituion relationship matrix.

```
# Splitting the author field as there could be multiple authors per record
author_institution <- our_interest %>%
  separate_rows(author, sep = ";") %>%
  distinct(record_id, author, institution)

# Creating a list of unique authors
unique_authors <- sort(unique(author_institution$author))

# Creating an empty matrix to store the connections
author_institution_matrix <- matrix(0, nrow = length(unique_authors), ncol = length(unique_authors))
rownames(author_institution_matrix) <- unique_authors
colnames(author_institution_matrix) <- unique_authors
```

```r
# Filling the matrix
for (i in 1:length(unique_authors)) {
  for (j in 1:length(unique_authors)) {
    if (i != j) {
      # Check if authors i and j are associated with the same institution
      common_countries <- intersect(author_institution$institution[author_institution$author == unique_a
                                    author_institution$institution[author_institution$author == unique_a
      if (length(common_countries) > 0) {
        author_institution_matrix[i, j] <- 1
        author_institution_matrix[j, i] <- 1 # Matrix is symmetric
      }
    }
  }
}

diag(author_institution_matrix) <- 0
```

Once again, Trying to run a QAP correlation.

```r
qap_result <- mantel(coauthor_retraction_matrix, author_institution_matrix, permutations = 1000)
print(qap_result)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = coauthor_retraction_matrix, ydis = author_institution_matrix,      permutations = 1000)
##
## Mantel statistic r: 0.9866
##       Significance: 0.000999
##
## Upper quantiles of permutations (null model):
##     90%     95%   97.5%     99%
## 0.00109 0.00149 0.00189 0.00229
## Permutation: free
## Number of permutations: 1000
```

This suggests a very strong positive correlation between the coauthor_retraction_matrix and the author_institution_matrix. The extremely low p-value indicates that this correlation is highly unlikely to have occurred by chance. This implies a significant association between the patterns or relationships represented in these two matrices, suggesting that the coauthorship retraction patterns are closely related to the institutions of the authors.

I think we are doing something wrong in this approach. The matrices are correlated because of how the data was sourced. I wonder why there was not a 1.00 correlation.

Creating author-publisher network

```r
# Splitting the author field as there could be multiple authors per record
author_split <- expanded_dataset %>%
  separate_rows(author, sep = ";") %>%
  distinct(record_id, author, publisher)

# Creating a list of unique authors
unique_authors <- sort(unique(author_split$author))
```

```r
# Creating an empty matrix to store the connections
author_publisher_matrix <- matrix(0, nrow = length(unique_authors), ncol = length(unique_authors))
rownames(author_publisher_matrix) <- unique_authors
colnames(author_publisher_matrix) <- unique_authors

# Filling the matrix
for (i in 1:length(unique_authors)) {
  for (j in 1:length(unique_authors)) {
    if (i != j) {
      # Check if authors i and j have published with the same publisher
      common_publishers <- intersect(author_split$publisher[author_split$author == unique_authors[i]],
                                     author_split$publisher[author_split$author == unique_authors[j]])
      if (length(common_publishers) > 0) {
        author_publisher_matrix[i, j] <- 1
        author_publisher_matrix[j, i] <- 1 # Matrix is symmetric
      }
    }
  }
}
```

Once again, Trying to run a QAP correlation. This time to see if the author-publisher relationship has some impact..

```r
qap_result <- mantel(coauthor_retraction_matrix, author_publisher_matrix, permutations = 1000)
print(qap_result)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = coauthor_retraction_matrix, ydis = author_publisher_matrix,      permutations = 1000)
##
## Mantel statistic r: 0.09124
##       Significance: 0.000999
##
## Upper quantiles of permutations (null model):
##     90%     95%   97.5%     99%
## 0.00121 0.00156 0.00201 0.00228
## Permutation: free
## Number of permutations: 1000
```

The test suggests that there is a statistically significant, albeit weak, positive correlation between the patterns of coauthorship retraction and author publisher associations represented in the two matrices. However, given the low correlation coefficient, the strength of this association is not very strong.