elephant_in_the_room

Basic Operations before stating the analysis

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
```

Loading the data-set sourced from Retraction Watch on 24-Dec-2023

```r
data<- read.csv("retraction_watch.csv")
```

Doing some wrangling of the data

```r
# 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()
```

Creating two separate subsets now - one for business related, and one for non business

```r
# 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
```

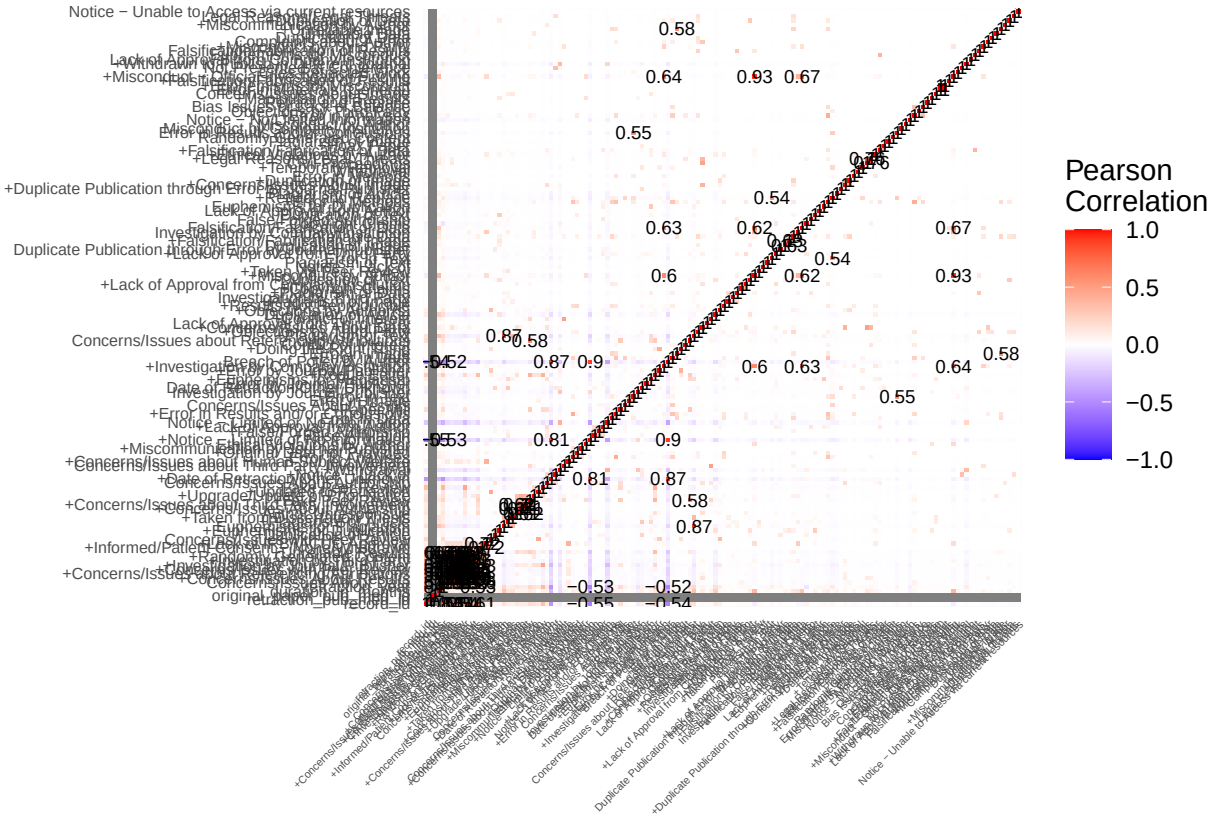
Creating a set of correlation matrices

For the non business data set

```r
# Step 1: Split the 'reason' field into individual reasons and remove the '+' prefix
reasons_list <- strsplit(gsub("^\\+", "", data_business$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) {
  data_business[[reason]] <- sapply(reasons_list, function(x) as.integer(reason %in% x))
}
```

```r
# Select only dummy variables (and any other numeric variables you want to include)
numeric_data <- data_business[, sapply(data_business, 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
```

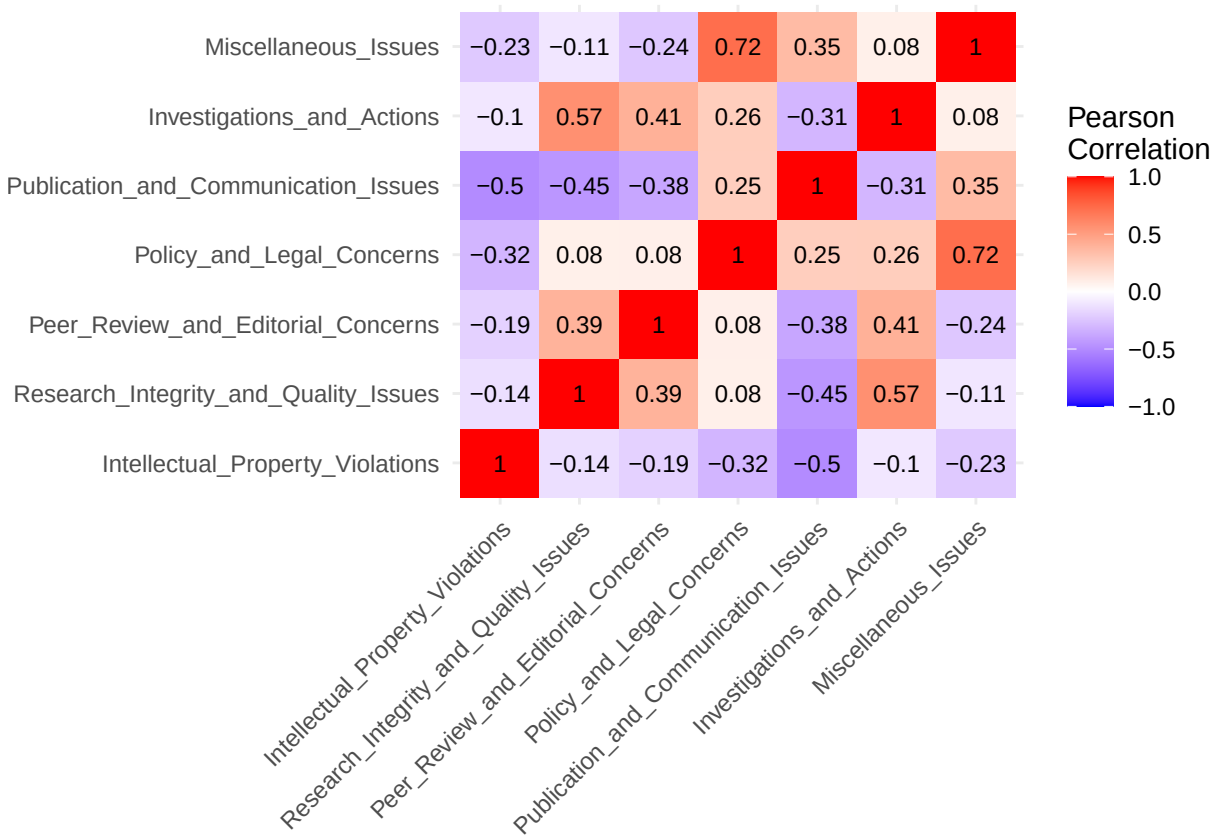
```
## Warning: Removed 546 rows containing missing values (`geom_text()`).
```



Condensing the co-plot with my own logic, for the business dataset

```r
# 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)
}

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

# Ensure these columns exist in your data_business 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 data_business
if(all(selected_columns %in% names(data_business))) {
    # Select only the specified dummy variable columns
    dummy_data <- data_business[, 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())
```

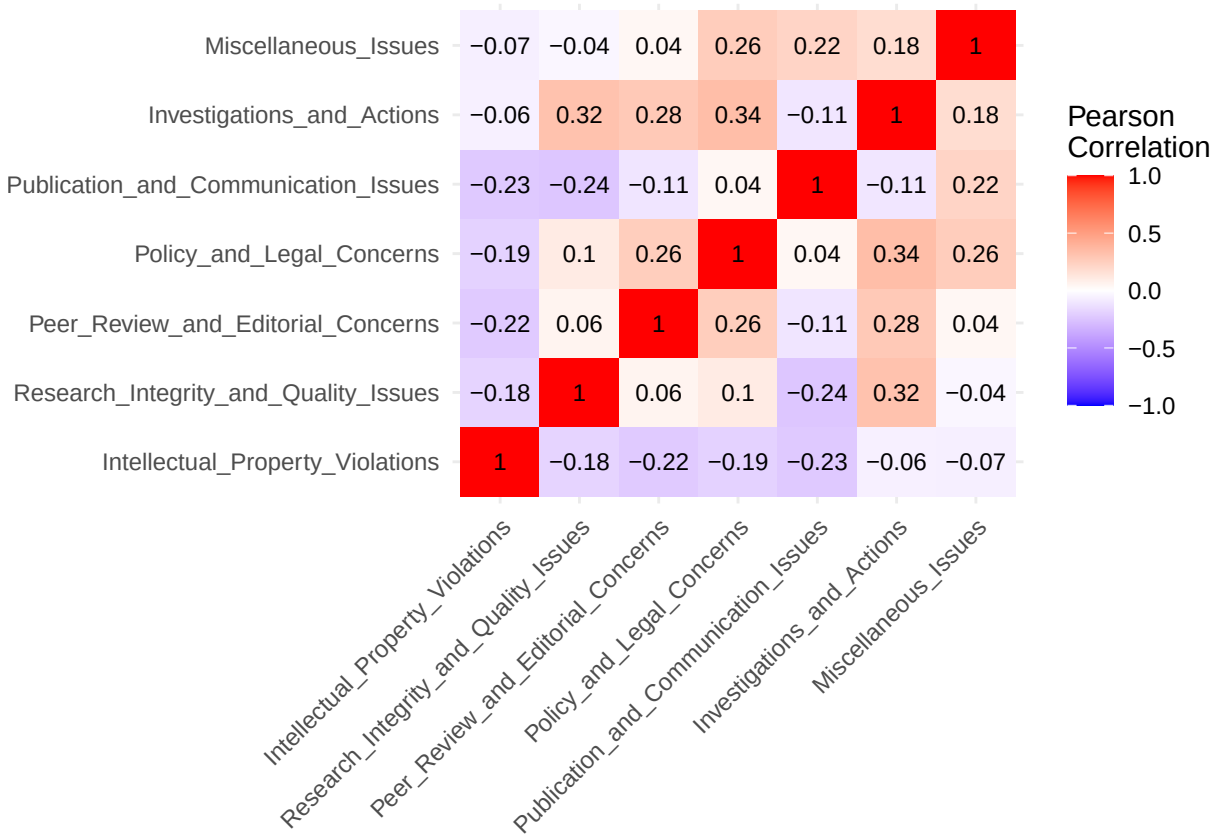
```
} else {
    stop("Not all specified columns exist in the dataframe.")
}
```



Condensing the co-plot with my own logic, for the non-business dataset

```
# Apply the function to the data_nonbusiness dataframe
data_nonbusiness <- create_dummy_vars(data_nonbusiness, patterns)

# Ensure these columns exist in your data_nonbusiness 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 data_nonbusiness
if(all(selected_columns %in% names(data_nonbusiness))) {
    # Select only the specified dummy variable columns
    dummy_data <- data_nonbusiness[, 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.")
}
```

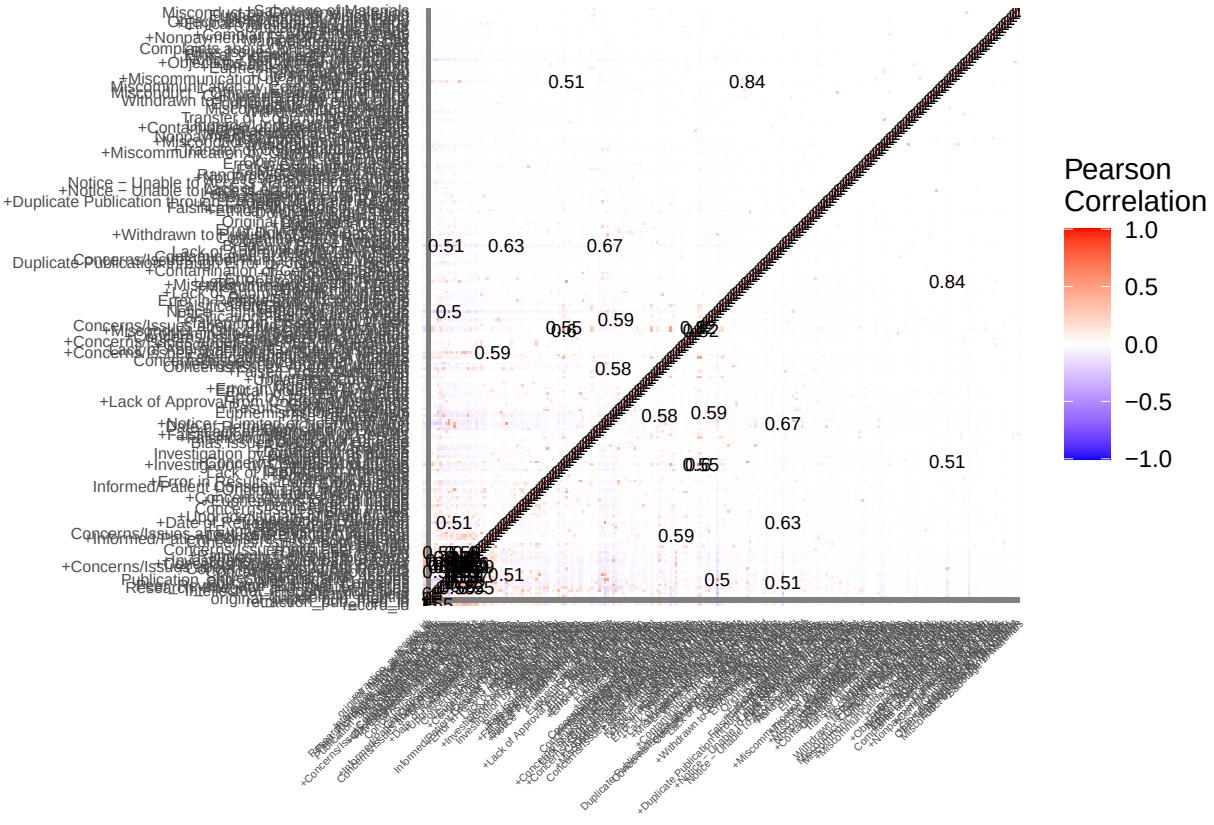


For the business dataset

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

```
## Warning: Removed 866 rows containing missing values (`geom_text()`).
```



Trying to plot how retractions have been in Non Management disciplines

```
# 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

```
# 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()
```

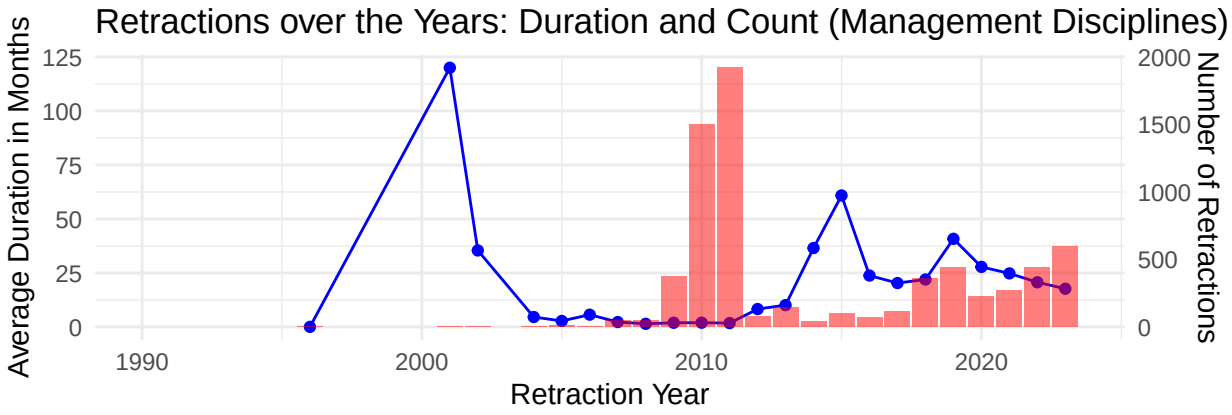
Displaying the plot

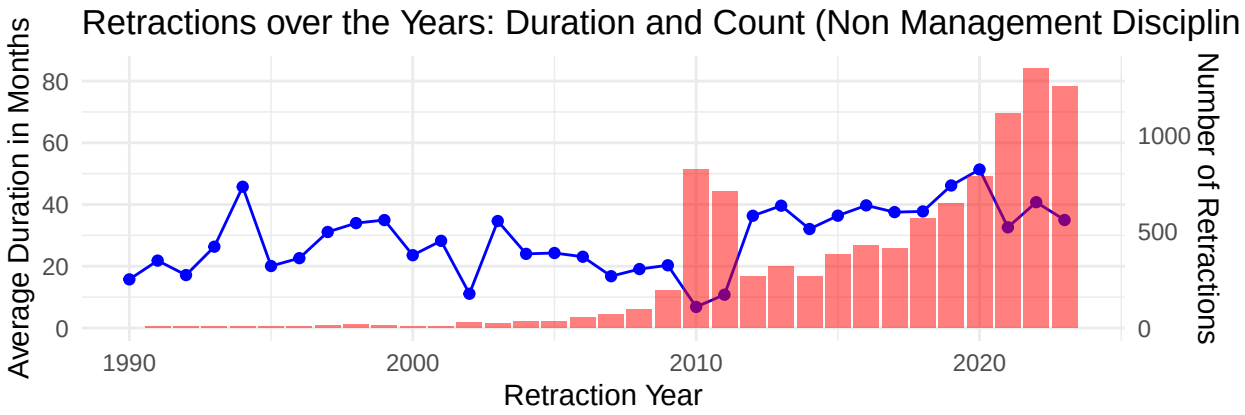
```
grid.arrange(retractions_management, retractions_nonmanagement, ncol = 1)
```

```
## 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()`).
```

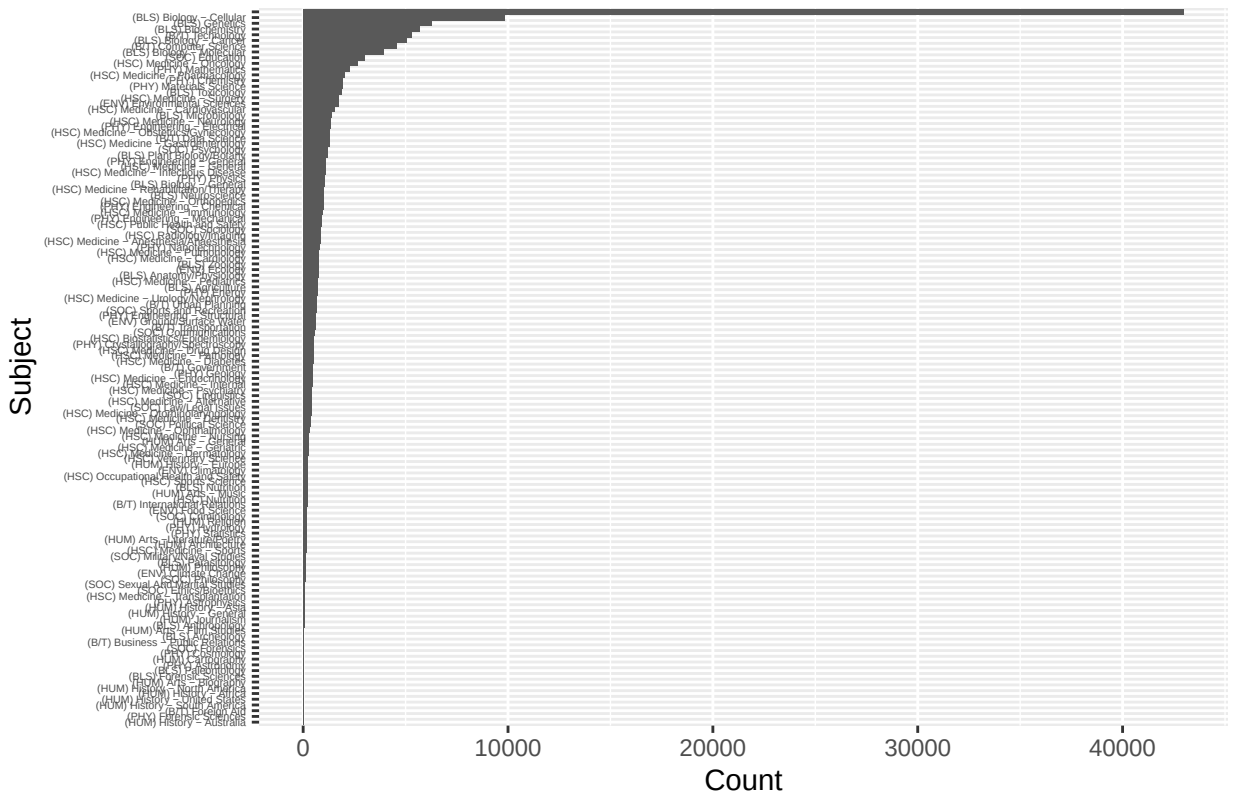




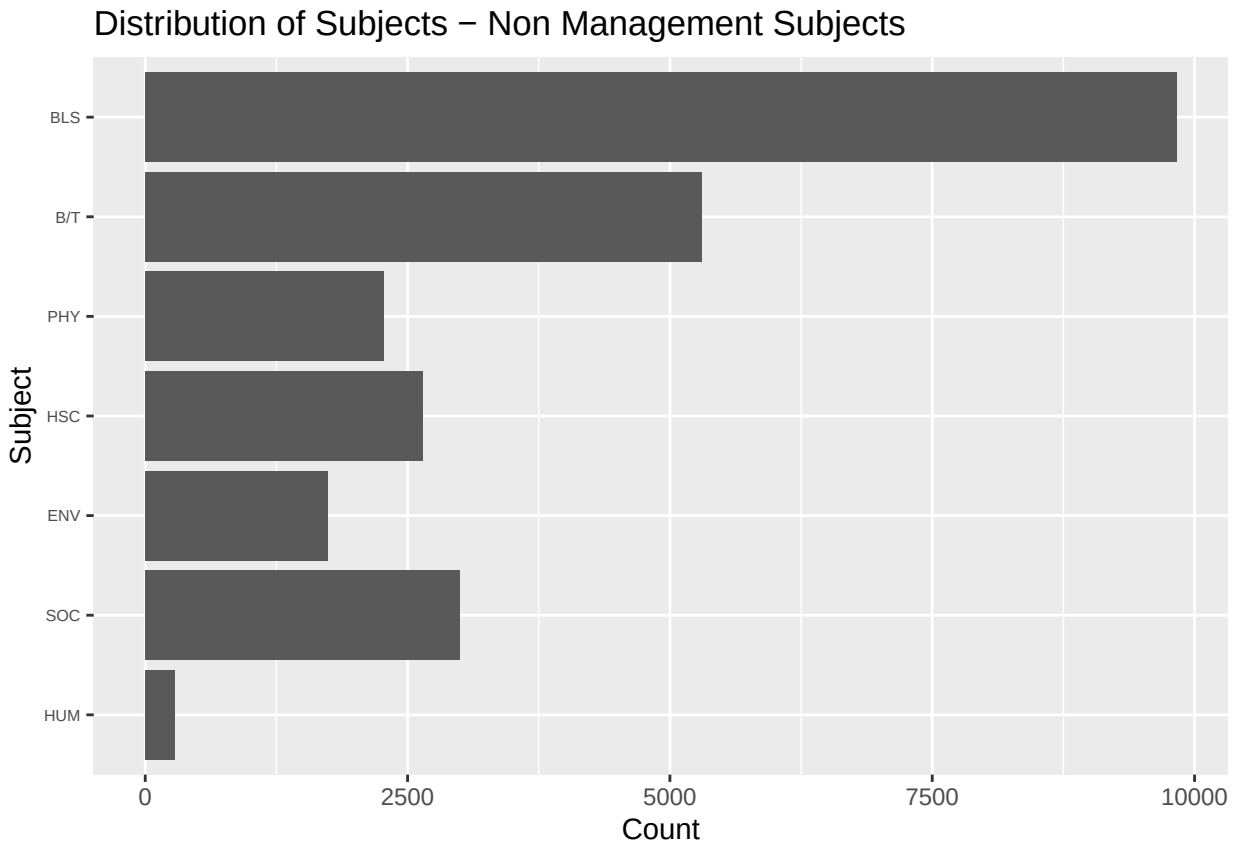
Trying to understand the distribution of subjects now.

Distribution of Subjects - Non Management Subjects

```r
# 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")
```

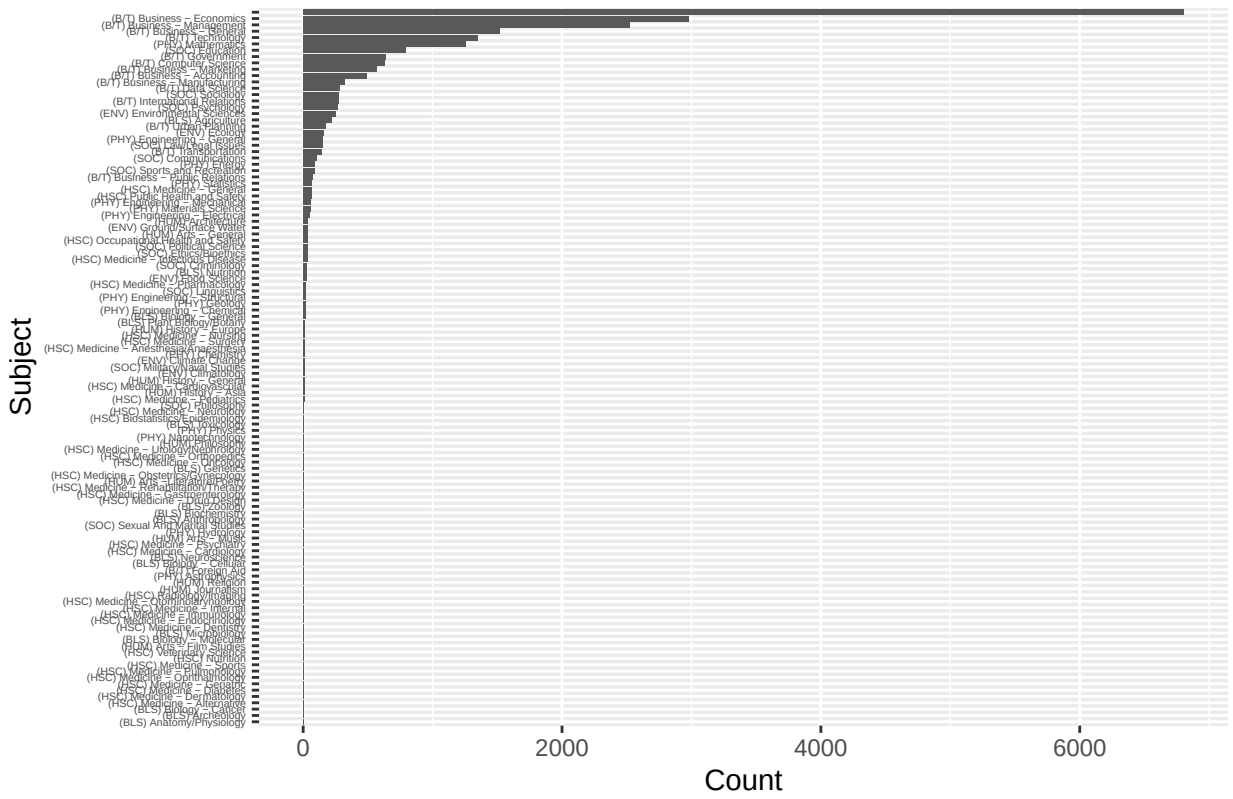



```r
# Rename the subject to the broader theme that's written in the brackets
subject_count <- subject_count %>%
  mutate(subject = str_match(subject, "\\(([^)]+)\\)")[,2])

# 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")
```

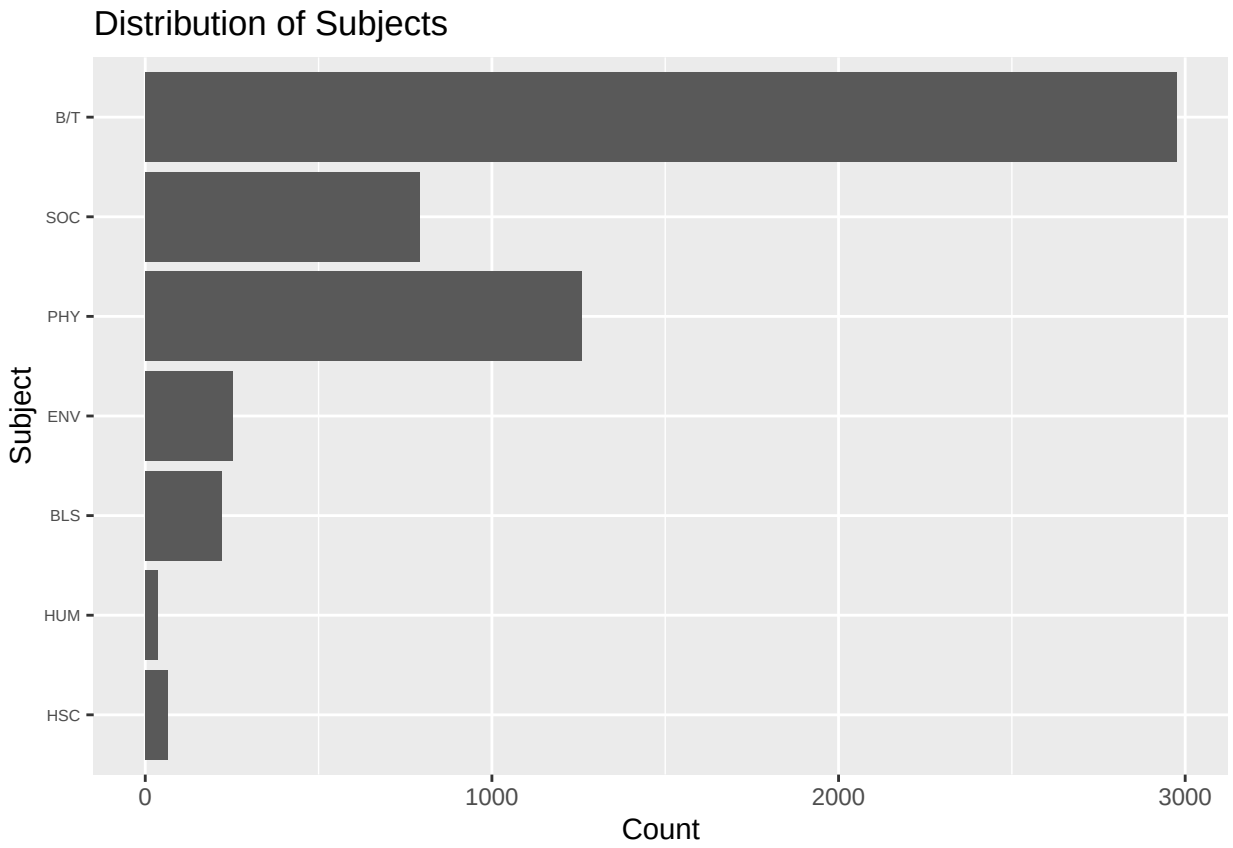


Distribution of Subjects - Management Subjects

```
# 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")
```



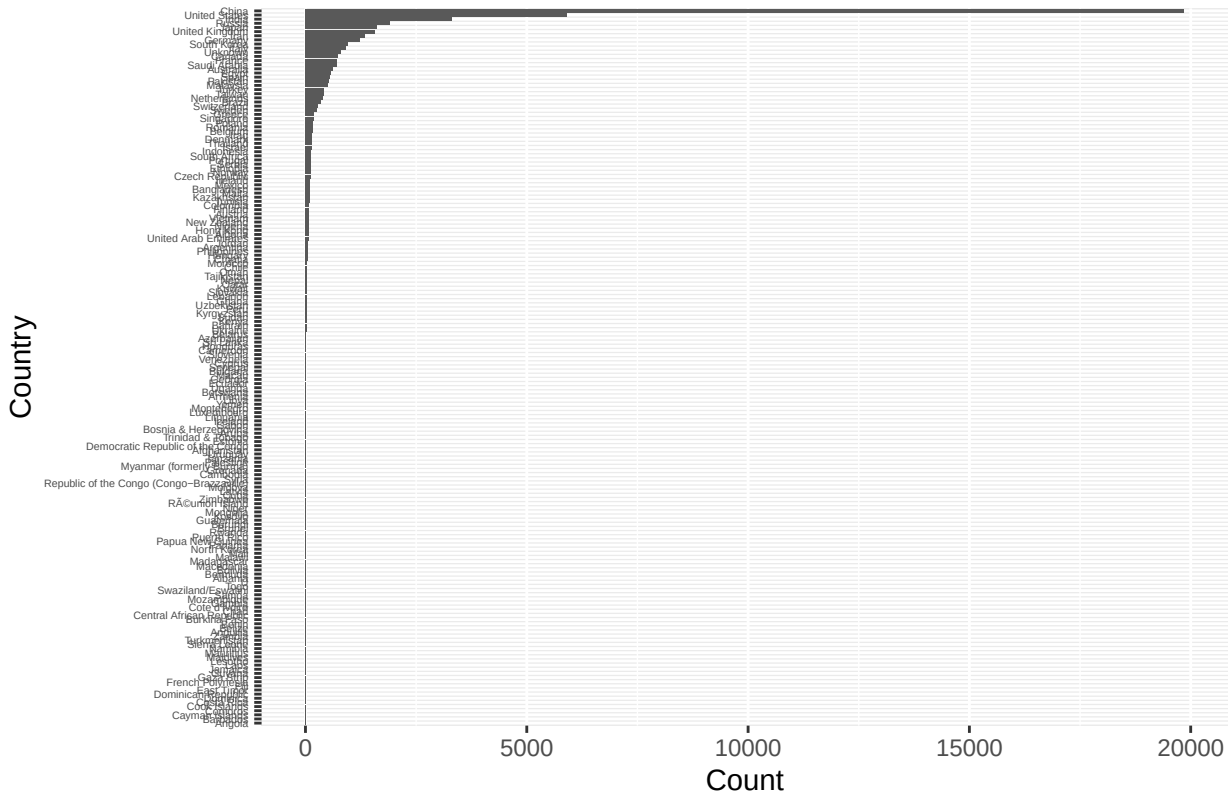

```r
# Rename the subject to the broader theme that's written in the brackets
subject_count <- subject_count %>%
  mutate(subject = str_match(subject, "\\(([^)]+)\\)")[,2])

# 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")
```



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")
```

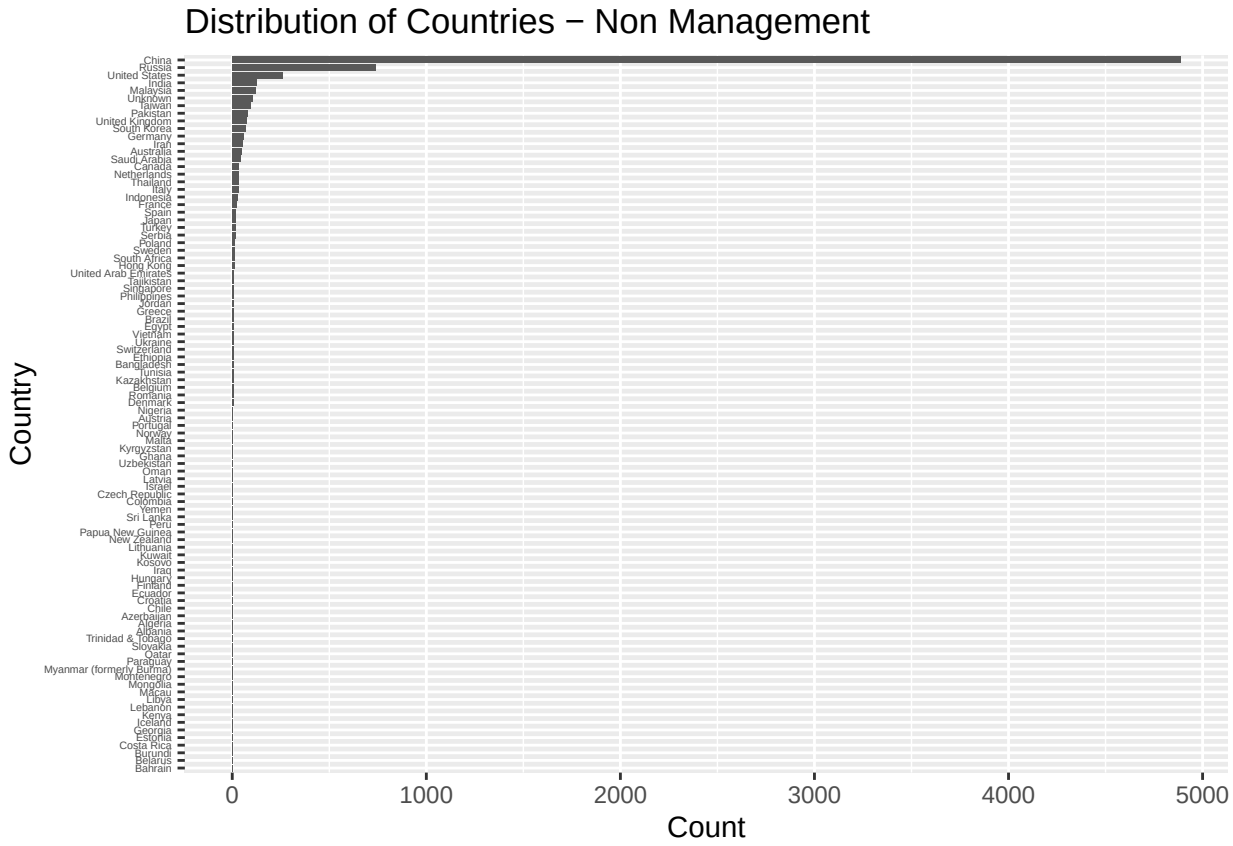



```
length(unique(country_count$country))
```

```
## [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")
```




```
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

```
# 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.

```
# 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)
```

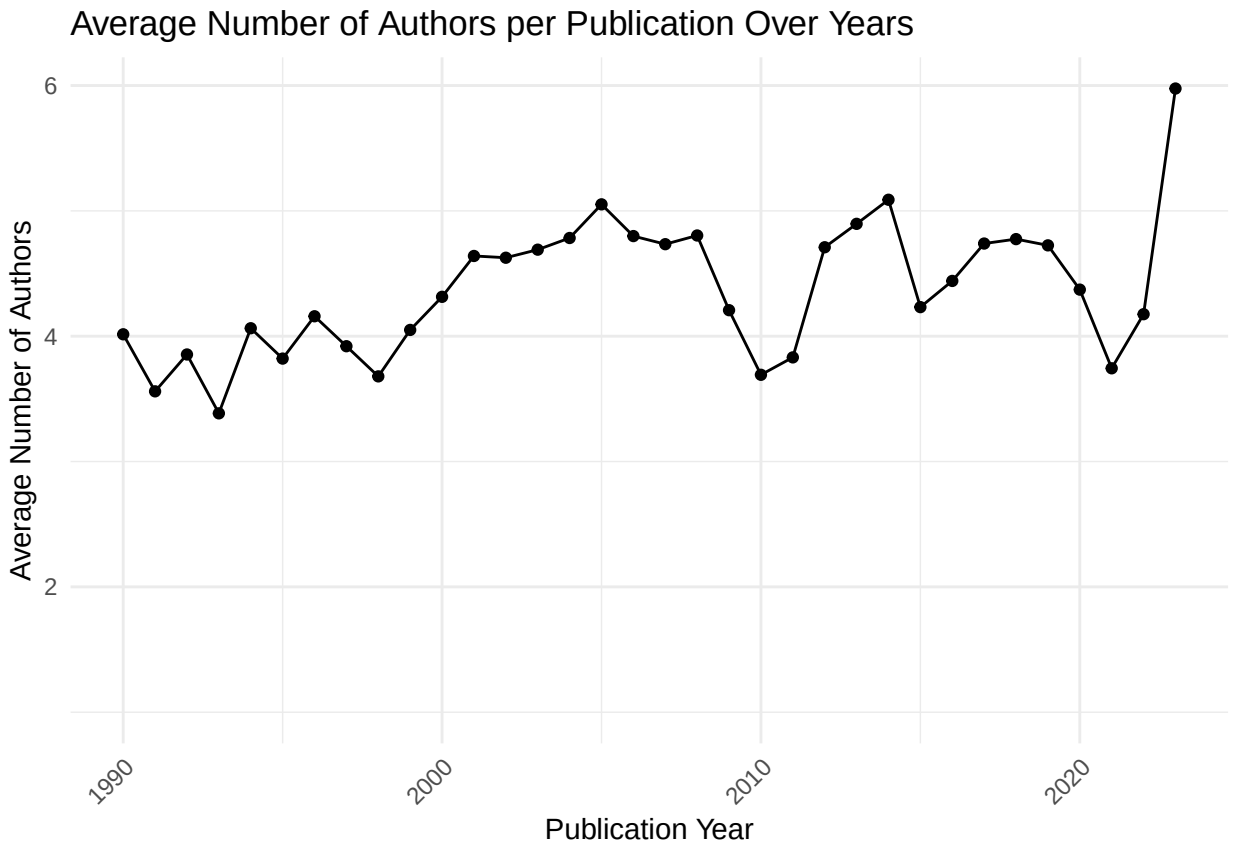
```
## [1] 2.850535
```

```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()`).
```

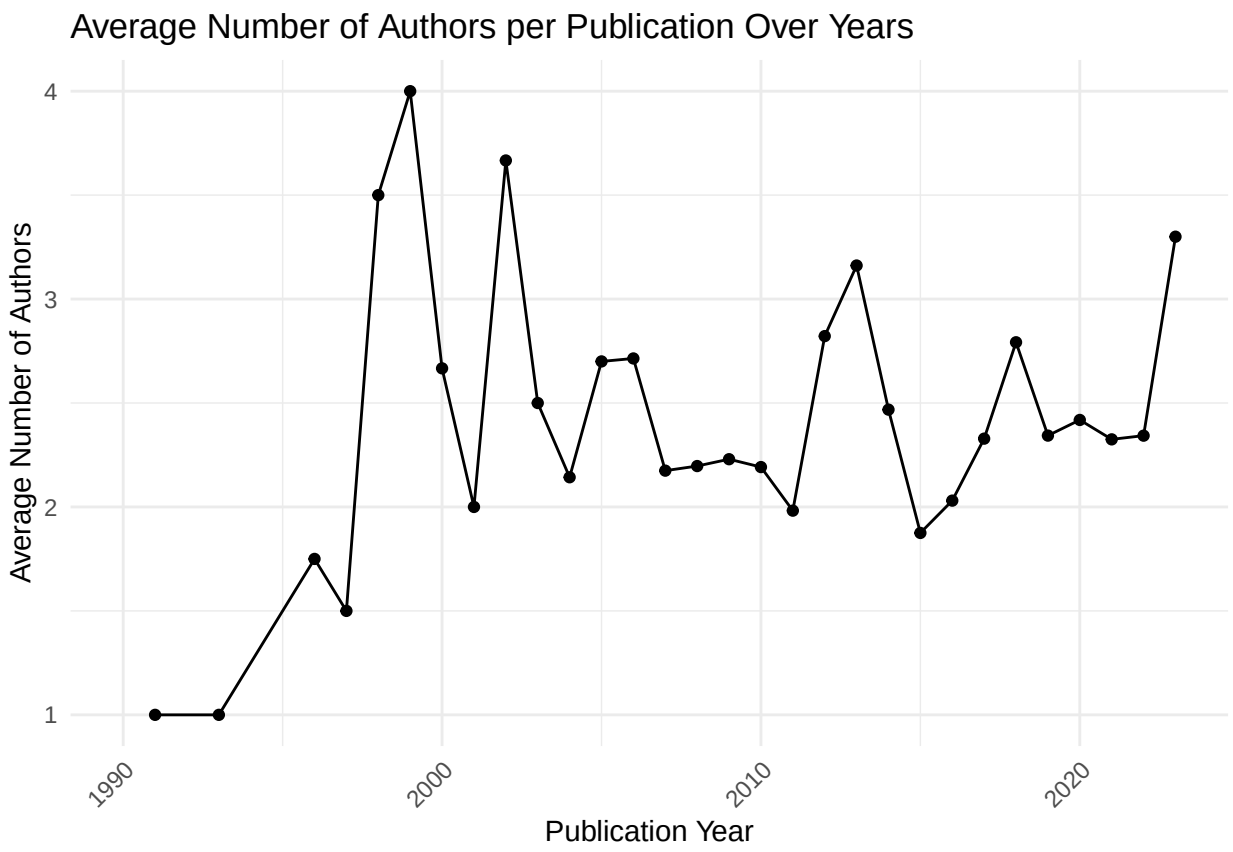



```
# Step 1: Add a column with the number of authors per publication
data_business <- data_business %>%
  mutate(
    num_authors = sapply(strsplit(as.character(author), ";"), length),
    publication_year = year(original_paper_date)
  )

# 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"
  )
```

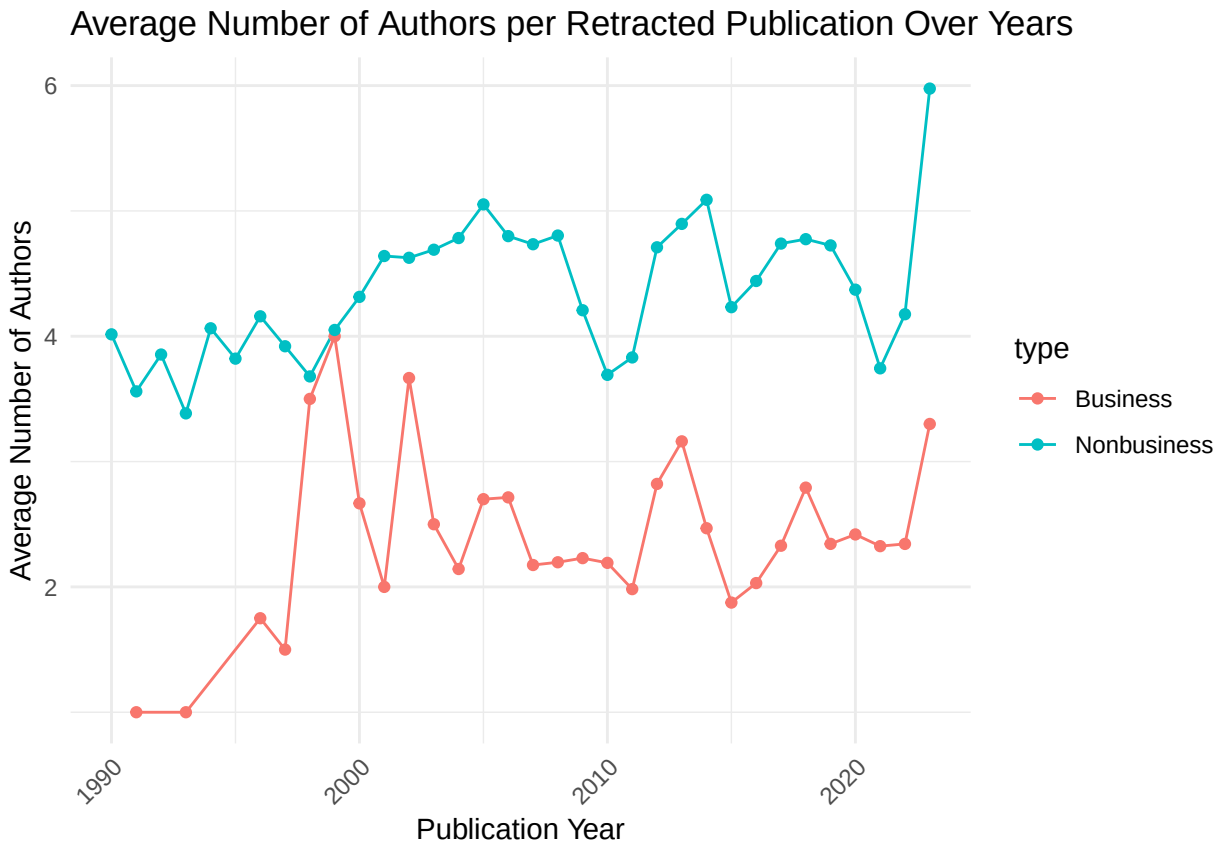
```r
# Combine the datasets
combined_data <- bind_rows(data_nonbusiness, data_business)

# 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()`).
```



Doing something similar for number of institutions

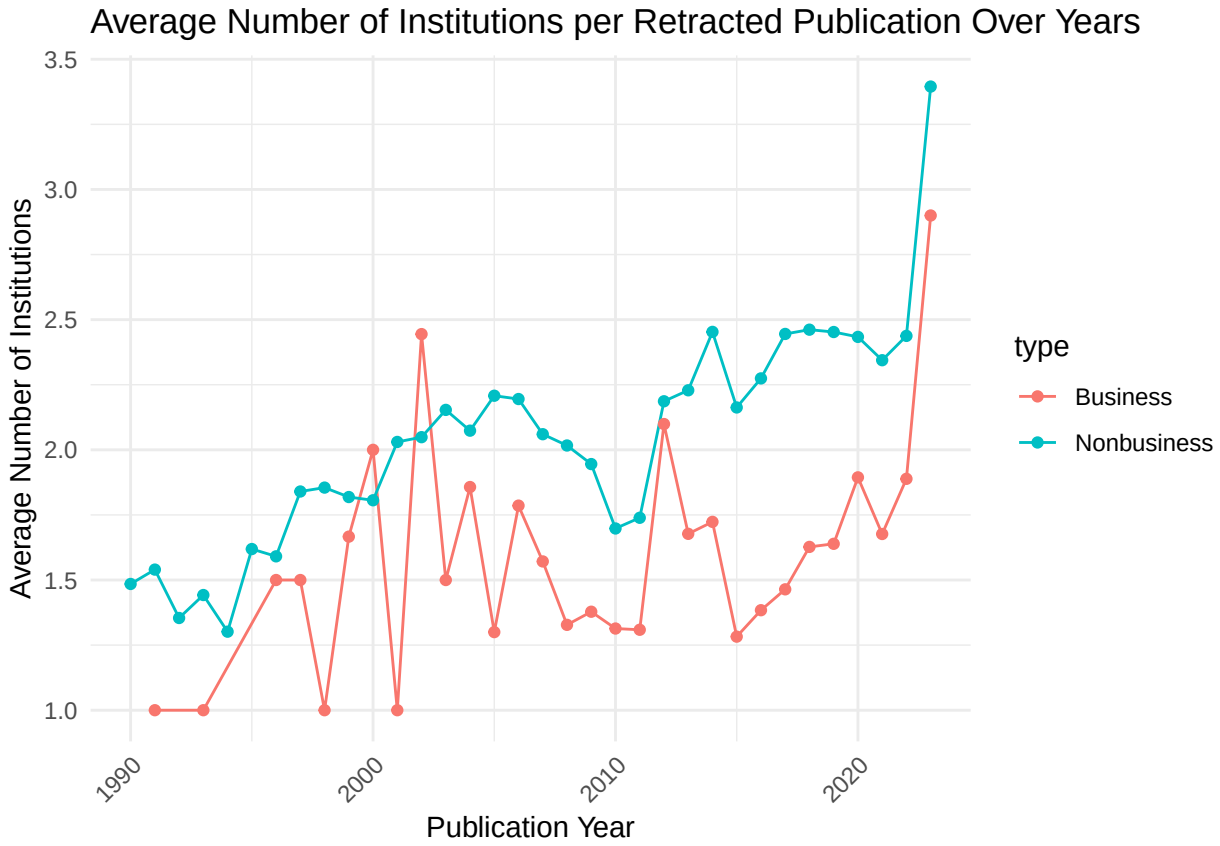
```
# Prepare nonbusiness data with the number of institutions
data_nonbusiness <- data_nonbusiness %>%
  mutate(
    num_institutions = sapply(strsplit(as.character(institution), ";"), length),  # Count institutions
    publication_year = year(original_paper_date),
    type = "Nonbusiness"
  )

# Prepare business data with the number of institutions
data_business <- data_business %>%
  mutate(
    num_institutions = sapply(strsplit(as.character(institution), ";"), length),  # Count institutions
    publication_year = year(original_paper_date),
    type = "Business"
  )

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

# Group by publication year and type, then calculate the average number of institutions
average_institutions_per_year_type <- combined_data %>%
  group_by(publication_year, type) %>%
  summarise(average_institutions = mean(num_institutions, na.rm = TRUE))
```

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

```r
# Plotting the average number of institutions per year for both types
ggplot(average_institutions_per_year_type, aes(x = publication_year, y = average_institutions, color = 
  geom_line() +  # Line plot
  geom_point() + # Adding points to each year
  theme_minimal() +
  labs(
    title = "Average Number of Institutions per Retracted Publication Over Years",
    x = "Publication Year",
    y = "Average Number of Institutions"
  ) +
  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_institutions_per_year_type$publication_year))) # Limi
```

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

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




```r
# Prepare nonbusiness data with the number of countries
data_nonbusiness <- data_nonbusiness %>%
  mutate(
    num_countries = sapply(strsplit(as.character(country), ";"), length),  # Count countries
    publication_year = year(original_paper_date),
    type = "Nonbusiness"
  )

# Prepare business data with the number of countries
```

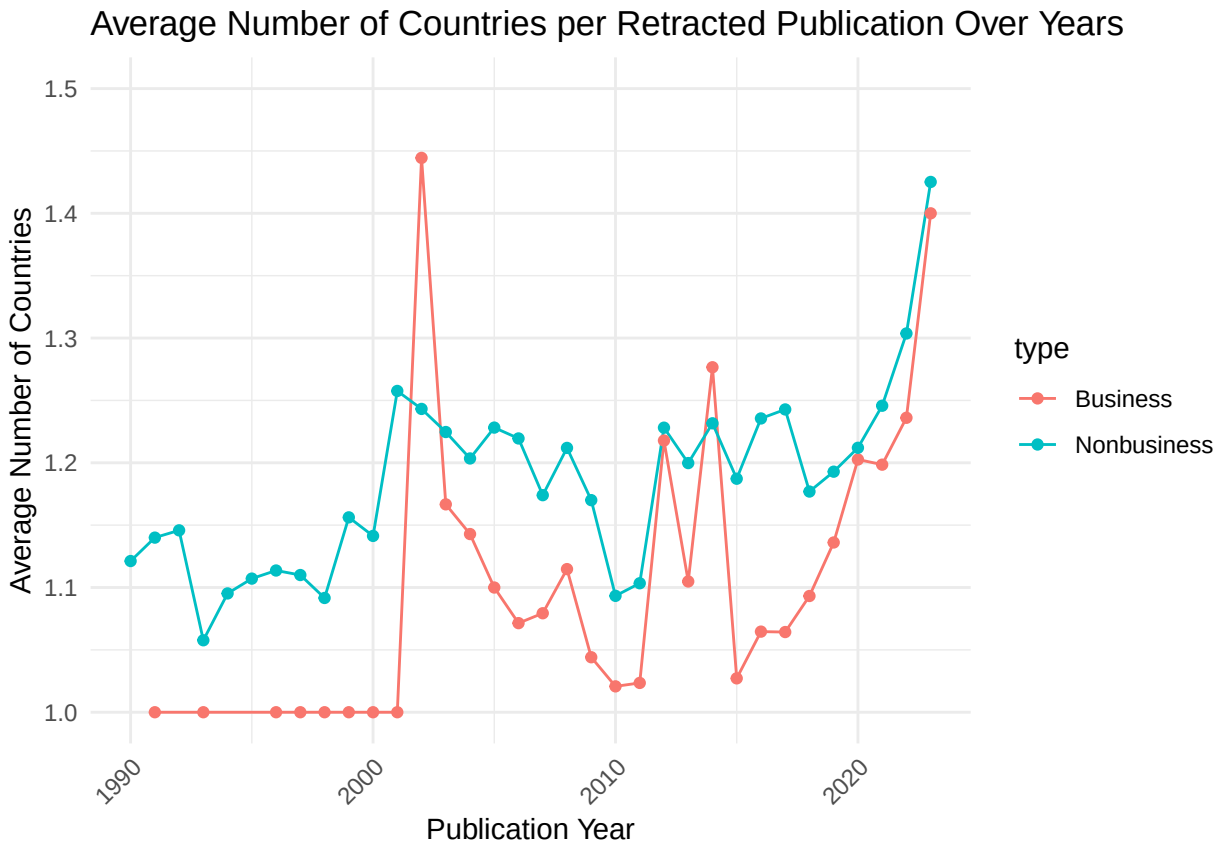
```r
data_business <- data_business %>%
  mutate(
    num_countries = sapply(strsplit(as.character(country), ";"), length),  # Count countries
    publication_year = year(original_paper_date),
    type = "Business"
  )

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

# Group by publication year and type, then calculate the average number of countries
average_countries_per_year_type <- combined_data %>%
  group_by(publication_year, type) %>%
  summarise(average_countries = mean(num_countries, na.rm = TRUE))
```

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

```r
# Plotting the average number of countries per year for both types
ggplot(average_countries_per_year_type, aes(x = publication_year, y = average_countries, color = type)) +
  geom_line() +  # Line plot
  geom_point() + # Adding points to each year
  theme_minimal() +
  labs(
    title = "Average Number of Countries per Retracted Publication Over Years",
    x = "Publication Year",
    y = "Average Number of Countries"
  ) +
  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_countries_per_year_type$publication_year))) +
    scale_y_continuous(limits = c(1, 1.5))
```

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

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



What is pending: 1. Eigen vector centrality 2. Cor plots for reasons? 3. Can I put the retractions data on a map to show where the retractions have been coming from? 4. Time series clustering?

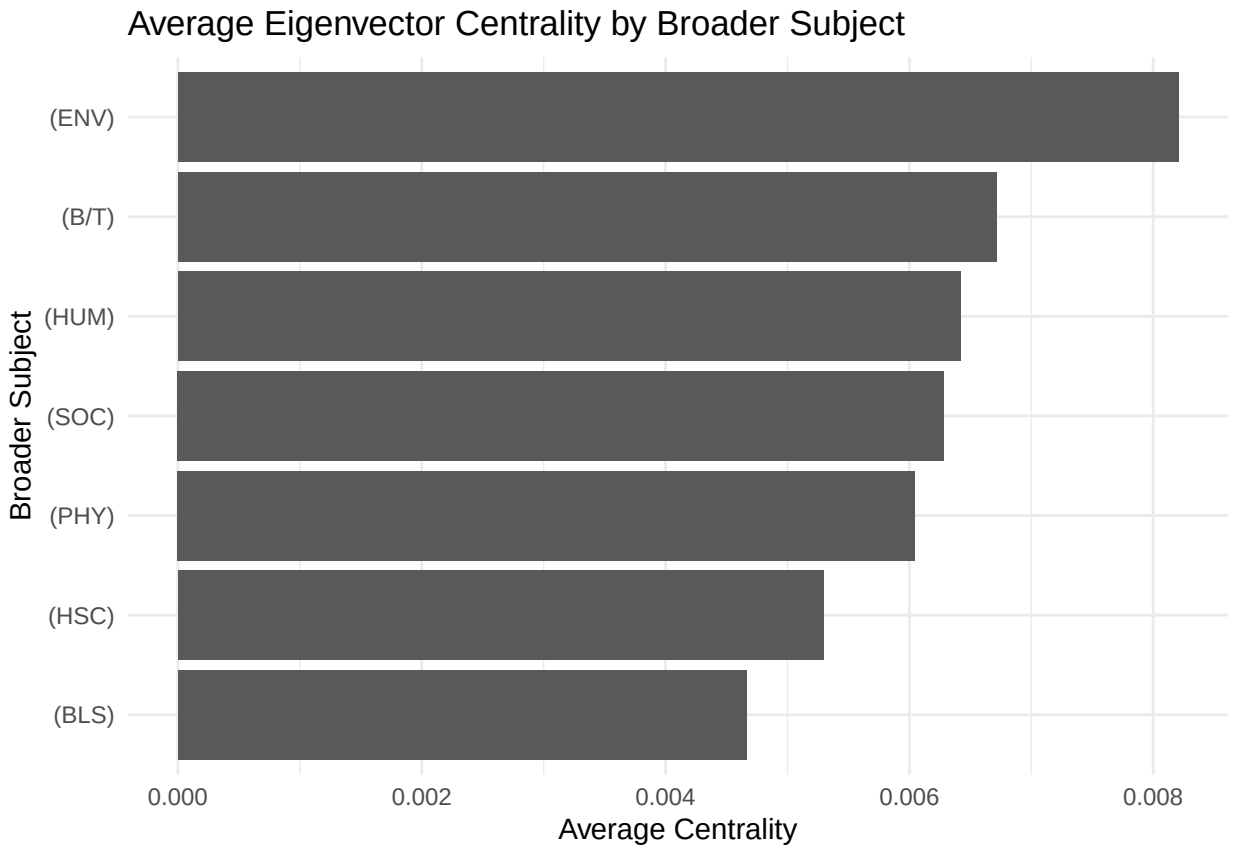
```
# Separate the authors and associate them with broader subjects and years
author_data <- data_temp %>%
  mutate(publication_year = year(original_paper_date),
         broader_subject = str_extract(subject, "\\(([^)]+)\\)")) %>%
  select(author, broader_subject, publication_year) %>%
  separate_rows(author, sep = ";\\s*") %>%
  filter(broader_subject != "" & !is.na(broader_subject))

# Assuming each row in author_data is unique to an article, if not, ensure uniqueness
author_data <- author_data %>% distinct()
```

```
# Create a data frame for edges (co-authorships)
edges <- author_data %>%
  group_by(publication_year) %>%
  summarise(authors = list(unique(author))) %>%
  filter(length(authors) > 1) %>%
  unnest(authors) %>%
  combn(2, simplify = FALSE) %>%
  lapply(function(pair) data.frame(from = pair[1], to = pair[2])) %>%
  bind_rows()

# Create a graph from the edge list
author_graph <- graph_from_data_frame(edges, directed = FALSE)
```

```r
eig_centrality <- eigen_centrality(author_graph)$vector
names(eig_centrality) <- V(author_graph)$name
```

```r
centrality_subjects <- author_data %>%
  distinct(author, broader_subject) %>%
  mutate(eig_centrality = eig_centrality[author])

# Aggregating centrality scores by broader subject
subject_centrality <- centrality_subjects %>%
  group_by(broader_subject) %>%
  summarise(avg_centrality = mean(eig_centrality, na.rm = TRUE))
```

```r
library(ggplot2)

ggplot(subject_centrality, aes(x = reorder(broader_subject, avg_centrality), y = avg_centrality)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Eigenvector Centrality by Broader Subject", x = "Broader Subject", y = "Average
  coord_flip()  # For horizontal bars
```



Trying something random with time dimension

```r
author_data <- data_temp %>%
  mutate(
    publication_year = year(original_paper_date),
    broader_subject = str_extract(subject, "\\(([^)]+)\\)")
  ) %>%
  select(record_id, author, broader_subject, publication_year) %>%
```

```
  separate_rows(author, sep = ";\\s*") %>%
  filter(broader_subject != "" & !is.na(broader_subject))


# Initialize a list to store centrality data for each year
yearly_centrality_data <- list()

unique_years <- unique(author_data$publication_year)

for(year in unique_years) {
    year_data <- author_data %>% filter(publication_year == year)

    edges <- year_data %>%
        group_by(record_id) %>%
        summarise(authors = list(unique(author))) %>%
        filter(length(authors) > 1) %>%
        unnest(authors) %>%
        combn(2, simplify = FALSE) %>%
        lapply(function(pair) data.frame(from = pair[1], to = pair[2])) %>%
        bind_rows()

    graph <- graph_from_data_frame(edges, directed = FALSE)

    centrality <- eigen_centrality(graph)$vector
    names(centrality) <- V(graph)$name

    yearly_centrality_data[[as.character(year)]] <- centrality
}
```

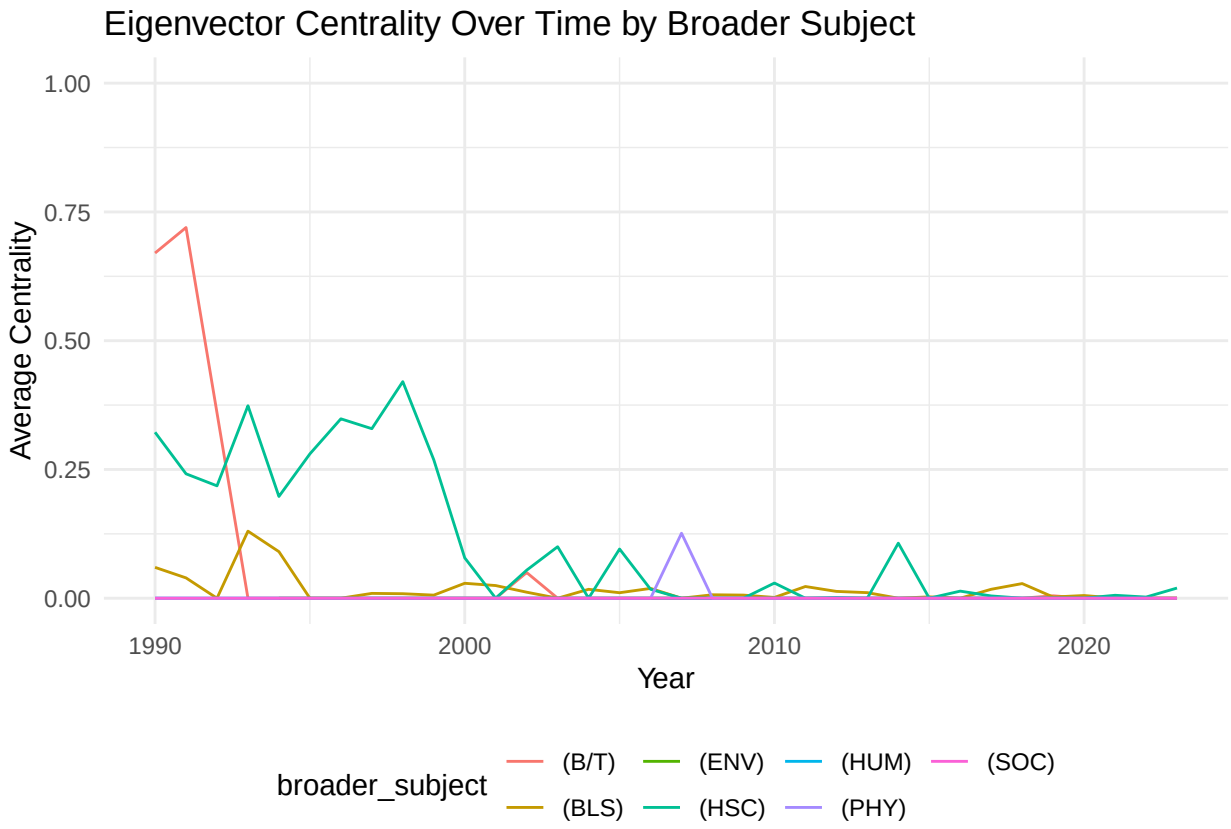
```
centrality_subject_year <- author_data %>%
    mutate(eig_centrality = map2(publication_year, author, ~yearly_centrality_data[[as.character(.x)]][
    unnest(eig_centrality)
```

```
# Aggregate centrality scores
subject_year_centrality <- centrality_subject_year %>%
    group_by(broader_subject, publication_year) %>%
    summarise(avg_centrality = mean(eig_centrality, na.rm = TRUE))
```

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

```
ggplot(subject_year_centrality, aes(x = publication_year, y = avg_centrality, group = broader_subject,
    geom_line() +
    scale_x_continuous(limits = c(1990, max(subject_year_centrality$publication_year))) +  # Set x-axis
    theme_minimal() +
    labs(title = "Eigenvector Centrality Over Time by Broader Subject",
         x = "Year",
         y = "Average Centrality") +
    theme(legend.position = "bottom")
```

```
## Warning: Removed 90 rows containing missing values (`geom_line()`).
```



Questions to think about 3. Are there some areas/subjects more prone to retractions?

1. Are there retractions in higher quality journals? Is this problem just there in low quality journals?

Finding a solution to this now:

Getting the ABDC dataset

```
abdc<- read.csv("abdc.csv")

#Filtering to latest abdc rankings only from 2022
abdc<- abdc %>%
  filter(
    year == 2022
  )
```

```
# Merging the two datasets
data_business_temp <- inner_join(data_business, abdc)
```

```
## Joining with `by = join_by(journal)`
```

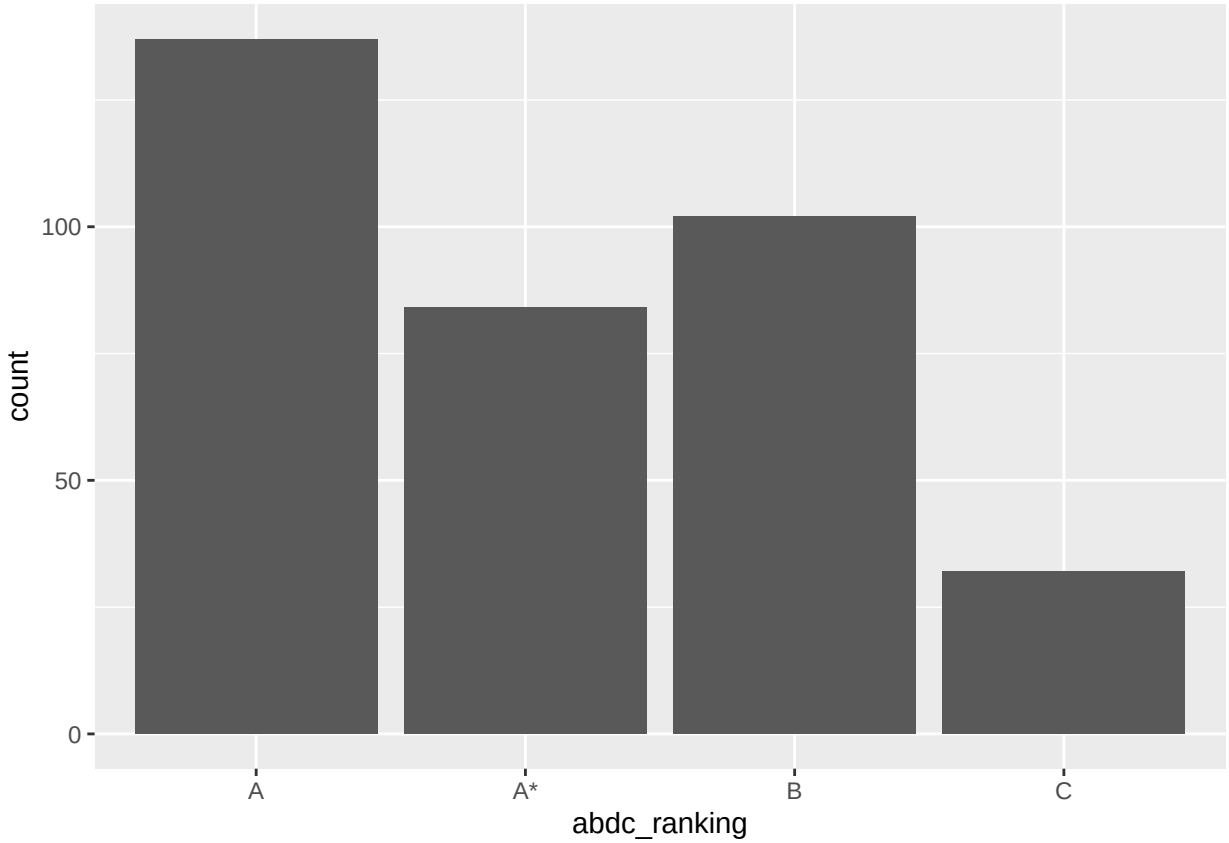
```
#getting the journals that are not listed- complementart set
journals_not_listed <- data_business$journal[!data_business$journal %in% data_business_temp$journal]
journals_not_listed_df <- data.frame(journal = journals_not_listed)
```

Checking distribution

```
# Plot the histogram of reasons
ggplot(data_business_temp, aes(x = abdc_ranking)) +
  geom_histogram(stat = "count")
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
```

```
## `binwidth`, `bins`, and `pad`
```



It appears that a lot of A*s still get retractions. It is fully possible that the A*s and As have a much better policy to police the retractions.

Just looking at reasons for the retractions

```
library(dplyr)

# Identify the reason columns
reason_columns <- grep("^\\+", names(data_business_temp), value = TRUE)

# Summarize the data to get counts of each reason for each abdc_ranking
top_reasons <- data_business_temp %>%
  select(abdc_ranking, all_of(reason_columns)) %>%
  gather(key = "reason", value = "count", -abdc_ranking) %>%
  group_by(abdc_ranking, reason) %>%
  summarise(total = sum(count)) %>%
  arrange(abdc_ranking, desc(total)) %>%
  group_by(abdc_ranking) %>%
  filter(rank(-total) <= 5)  # Select top 5 reasons for each abdc_ranking
```

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

```
# View the results
print(top_reasons)
```

```
## # A tibble: 18 x 3
## # Groups:   abdc_ranking [4]
```

```
##    abdc_ranking reason                                        total
##    <chr>        <chr>                                         <int>
##  1 A            +Investigation by Journal/Publisher              36
##  2 A            +Misconduct by Author                            25
##  3 A            +Investigation by Company/Institution            24
##  4 A            +Misconduct - Official Investigation/Finding     23
##  5 A            +Fake Peer Review                                14
##  6 A*           +Investigation by Company/Institution            21
##  7 A*           +Misconduct - Official Investigation/Finding     18
##  8 A*           +Investigation by Journal/Publisher              17
##  9 A*           +Misconduct by Author                            17
## 10 A*           +Unreliable Results                              17
## 11 B            +Fake Peer Review                                38
## 12 B            +Unreliable Results                              36
## 13 B            +Rogue Editor                                    35
## 14 B            +Plagiarism of Article                           13
## 15 B            +Withdrawal                                      12
## 16 C            +Duplication of Article                           5
## 17 C            +Plagiarism of Article                            5
## 18 C            +Euphemisms for Duplication                       2
```

These results are interesting and team needs to see

One reason we are seeing a lot of journals not listed I think is because a lot of them are engineering ones that also list as management. There are also those that are not journal articles but conference papers, and other scholarly worlds.

2. Are inter disciplinary works prone to more retractions

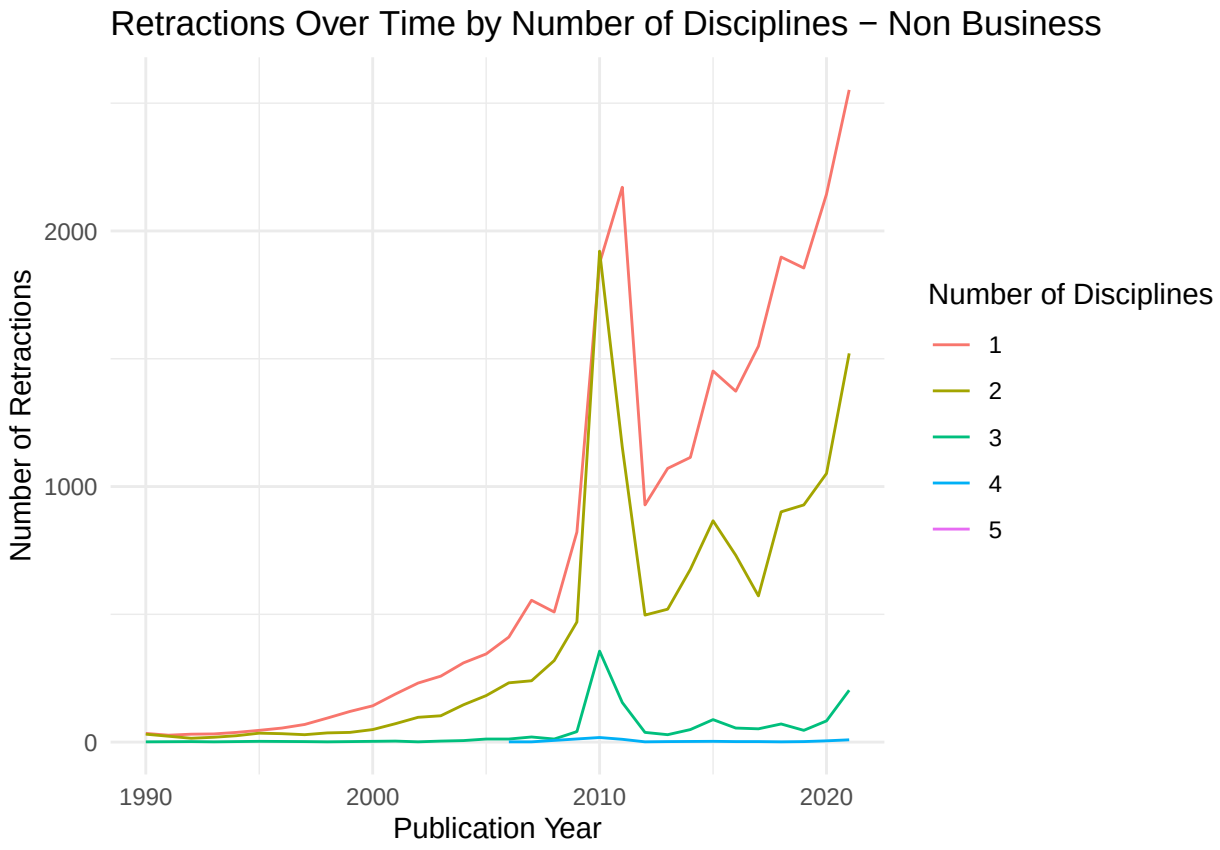
```r
# Step 1: Add a column with the number of disciplines per publication

data_nonbusiness <- data_nonbusiness %>%
  mutate(
    num_subjects = sapply(strsplit(as.character(subject), ";"), length),
    num_unique_disciplines = sapply(
      strsplit(as.character(subject), ";"),
      function(subjects) {
        # Extract disciplines
        disciplines <- unlist(lapply(subjects, function(x) str_extract_all(x, "\\(([^()]*)\\)")))
        # Flatten the list and remove NA
        disciplines <- unlist(disciplines)
        disciplines <- disciplines[!is.na(disciplines)]
        # Count unique disciplines
        length(unique(disciplines))
      }
    )
  )

# mirroring
data_business <- data_business %>%
  mutate(
    num_subjects = sapply(strsplit(as.character(subject), ";"), length),
    num_unique_disciplines = sapply(
      strsplit(as.character(subject), ";"),
      function(subjects) {
        # Extract disciplines
```

```
        disciplines <- unlist(lapply(subjects, function(x) str_extract_all(x, "\\(([^()]*)\\)")))
        # Flatten the list and remove NA
        disciplines <- unlist(disciplines)
        disciplines <- disciplines[!is.na(disciplines)]
        # Count unique disciplines
        length(unique(disciplines))
      }
    )
  )
```

Retractions Over Time by Number of Disciplines - Non Business

```
# Aggregate data
retractions_over_time <- data_nonbusiness %>%
  group_by(publication_year, num_unique_disciplines) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time, aes(x = publication_year, y = retractions, color = as.factor(num_unique_di
  geom_line() +
  labs(title = "Retractions Over Time by Number of Disciplines - Non Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Disciplines") +
  theme_minimal() +
    scale_x_continuous(limits = c(1990, 2021)) #
```

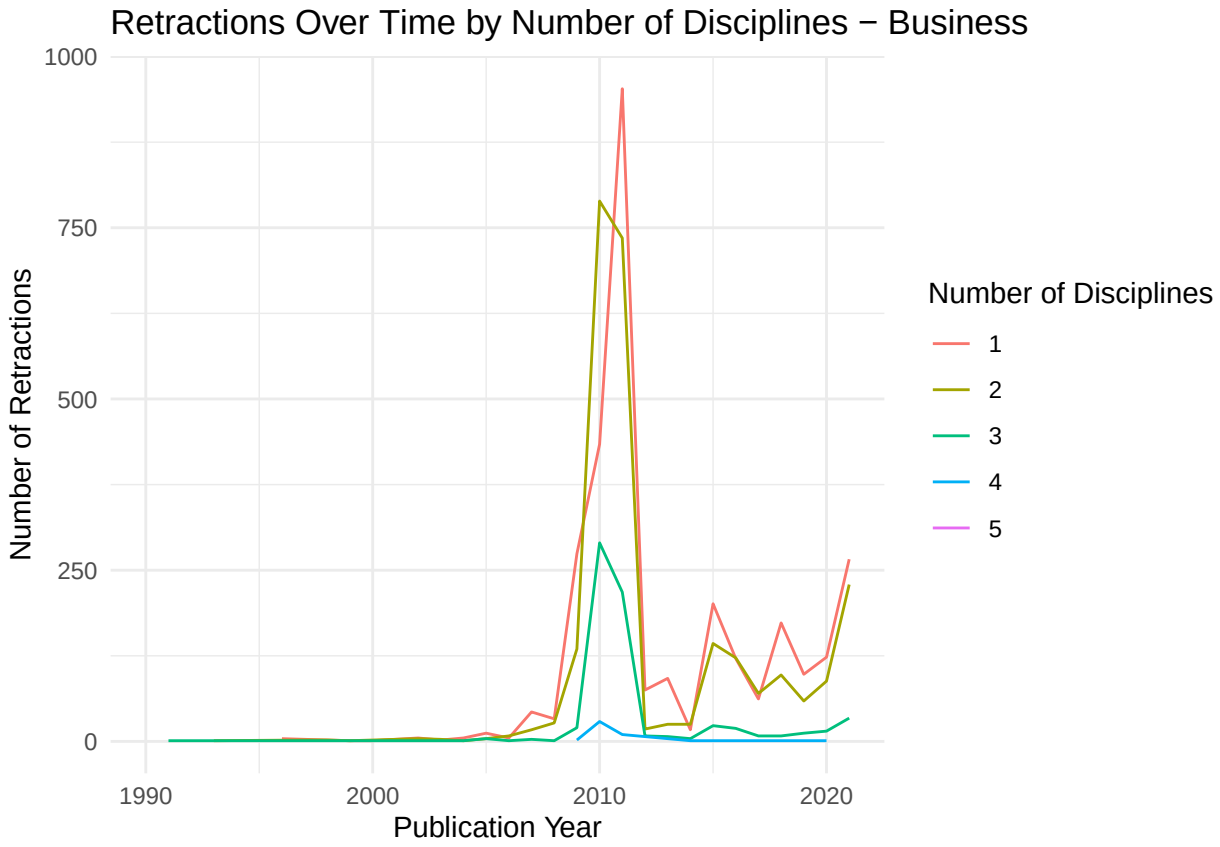
```
## Warning: Removed 80 rows containing missing values (`geom_line()`).
```



Retractions Over Time by Number of Disciplines - Business

```
retractions_over_time <- data_business %>%
  group_by(publication_year, num_unique_disciplines) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time, aes(x = publication_year, y = retractions, color = as.factor(num_unique_di
  geom_line() +
  labs(title = "Retractions Over Time by Number of Disciplines - Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Disciplines") +
  theme_minimal() +
    scale_x_continuous(limits = c(1990, 2021)) #
```

```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
```

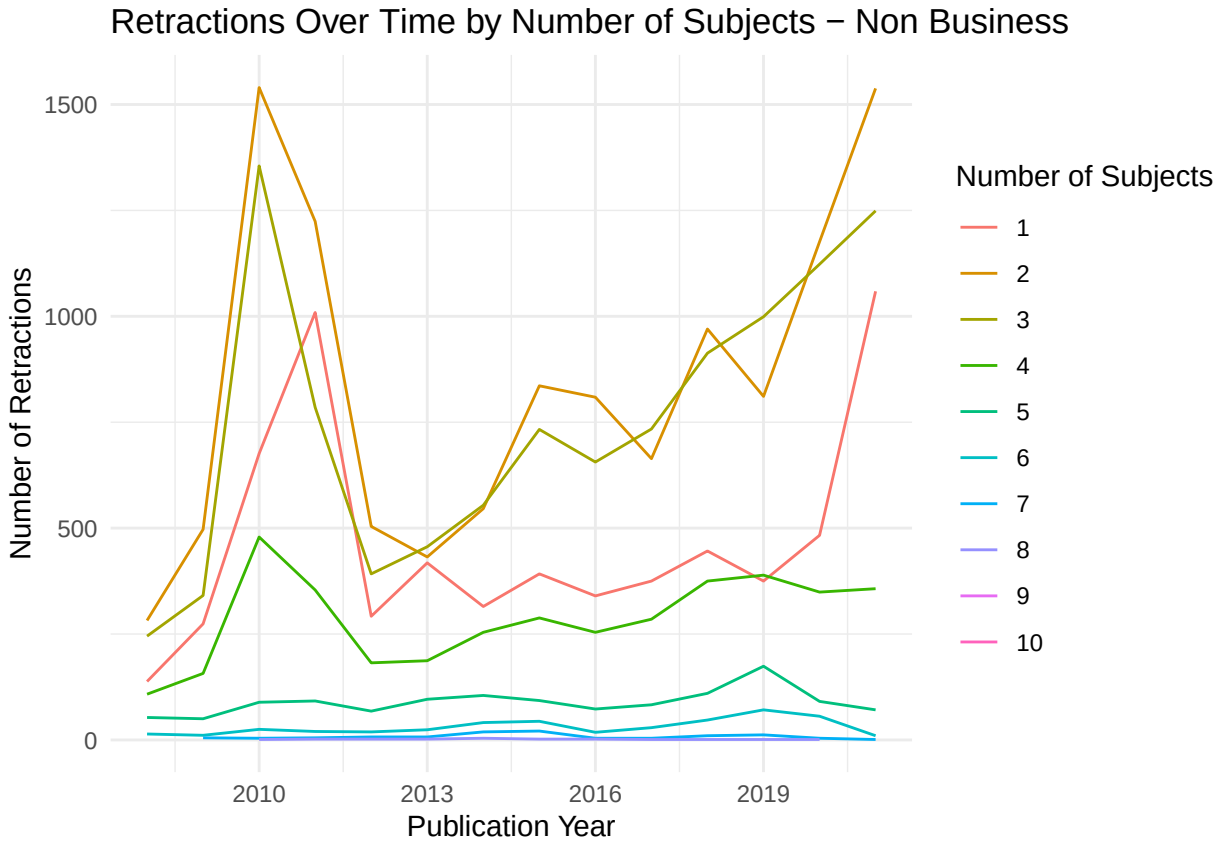


Retractions Over Time by Number of Subjects - Non Business

```
# Aggregate data
retractions_over_time <- data_nonbusiness %>%
  group_by(publication_year, num_subjects) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time, aes(x = publication_year, y = retractions, color = as.factor(num_subjects)
  geom_line() +
  labs(title = "Retractions Over Time by Number of Subjects - Non Business",
```

```
      x = "Publication Year",
      y = "Number of Retractions",
      color = "Number of Subjects") +
  theme_minimal() +
    scale_x_continuous(limits = c(2008, 2021)) #
```

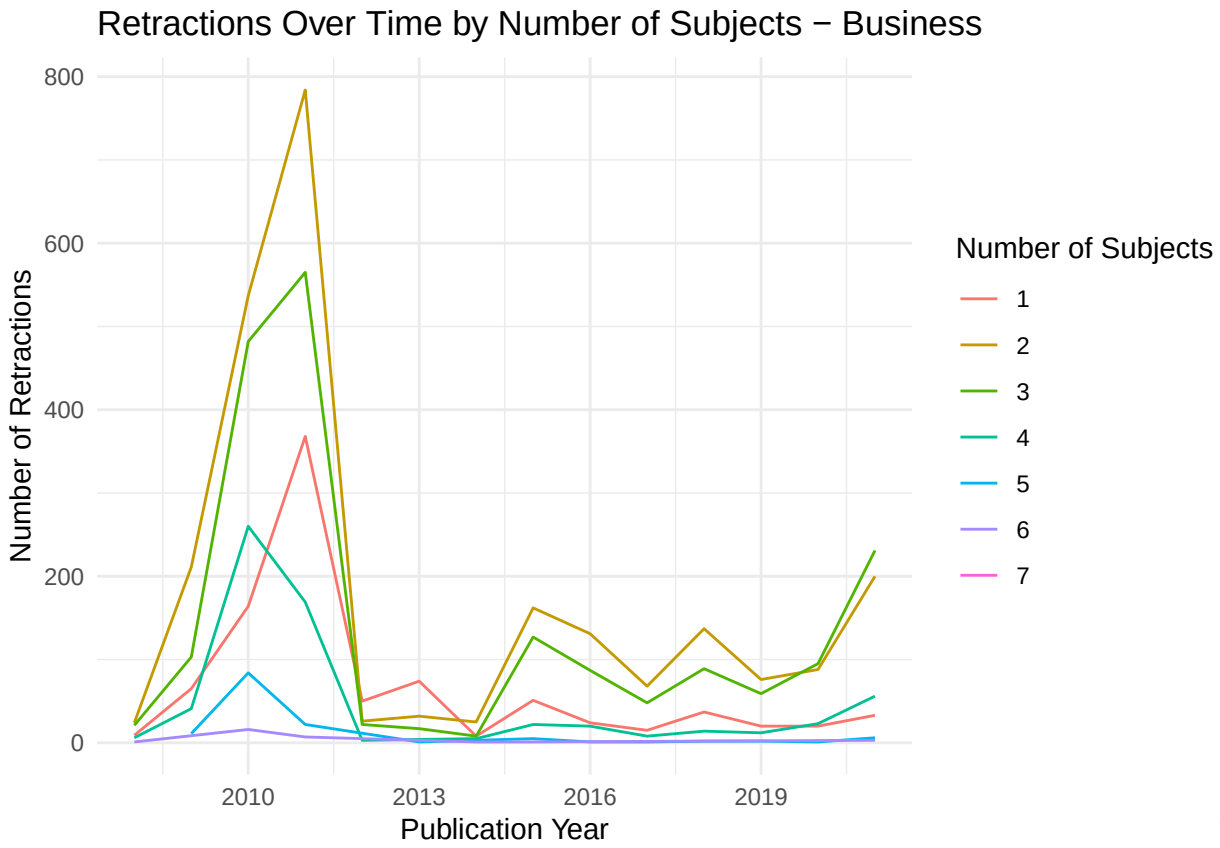
```
## Warning: Removed 237 rows containing missing values (`geom_line()`).
```



Retractions Over Time by Number of Subjects - Business

```
retractions_over_time <- data_business %>%
  group_by(publication_year, num_subjects) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time, aes(x = publication_year, y = retractions, color = as.factor(num_subjects)
  geom_line() +
  labs(title = "Retractions Over Time by Number of Subjects - Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Subjects") +
  theme_minimal() +
    scale_x_continuous(limits = c(2008, 2021)) #
```

```
## Warning: Removed 51 rows containing missing values (`geom_line()`).
```



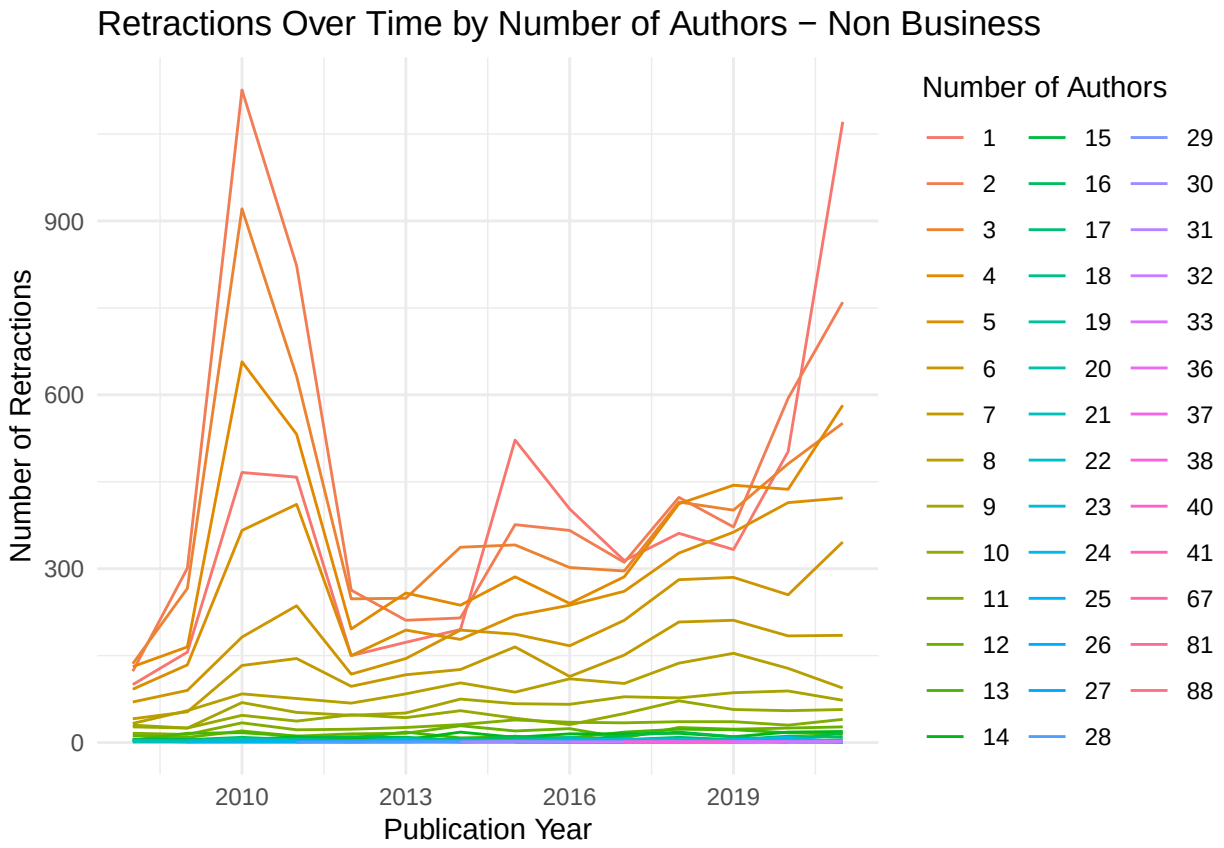
We can't really comment on this since we do not have the overall non retracted data. But I think we can safely say that number of papers listed in single category would be » than number of papers listed in multiple categories ..

so in % terms, the 2 cat thing would be more pronounced. Same with 3,4,5, etc.

Retractions Over Time by Number of Authors - Non Business

```
# Aggregate data
retractions_over_time_authors <- data_nonbusiness %>%
  group_by(publication_year, num_authors) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time_authors, aes(x = publication_year, y = retractions, color = as.factor(num_a
  geom_line() +
  labs(title = "Retractions Over Time by Number of Authors - Non Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Authors") +
  theme_minimal() +
      scale_x_continuous(limits = c(2008, 2021)) #
```

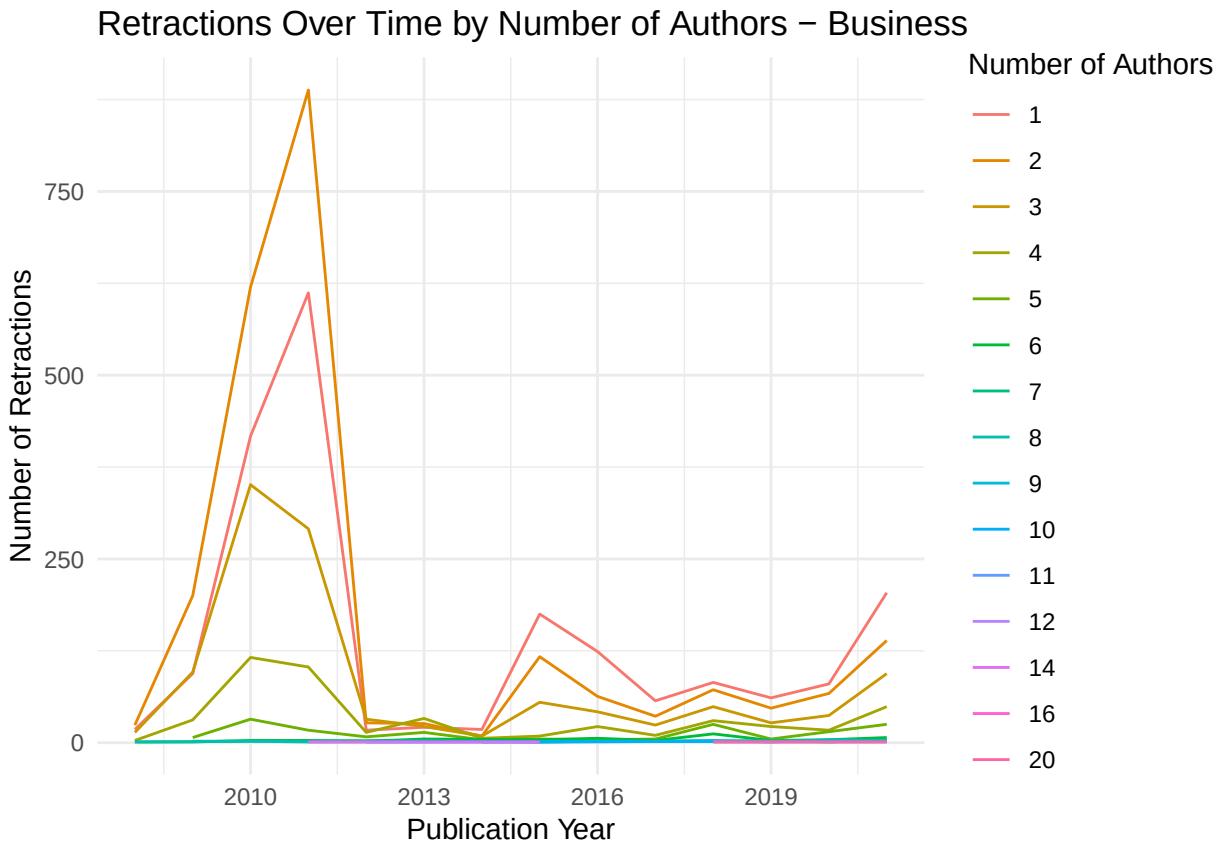
```
## Warning: Removed 403 rows containing missing values (`geom_line()`).
```



Retractions Over Time by Number of Authors - Business

```
# Aggregate data
retractions_over_time_authors <- data_business %>%
  group_by(publication_year, num_authors) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time_authors, aes(x = publication_year, y = retractions, color = as.factor(num_
  geom_line() +
  labs(title = "Retractions Over Time by Number of Authors - Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Authors") +
  theme_minimal() +
      scale_x_continuous(limits = c(2008, 2021)) #
```

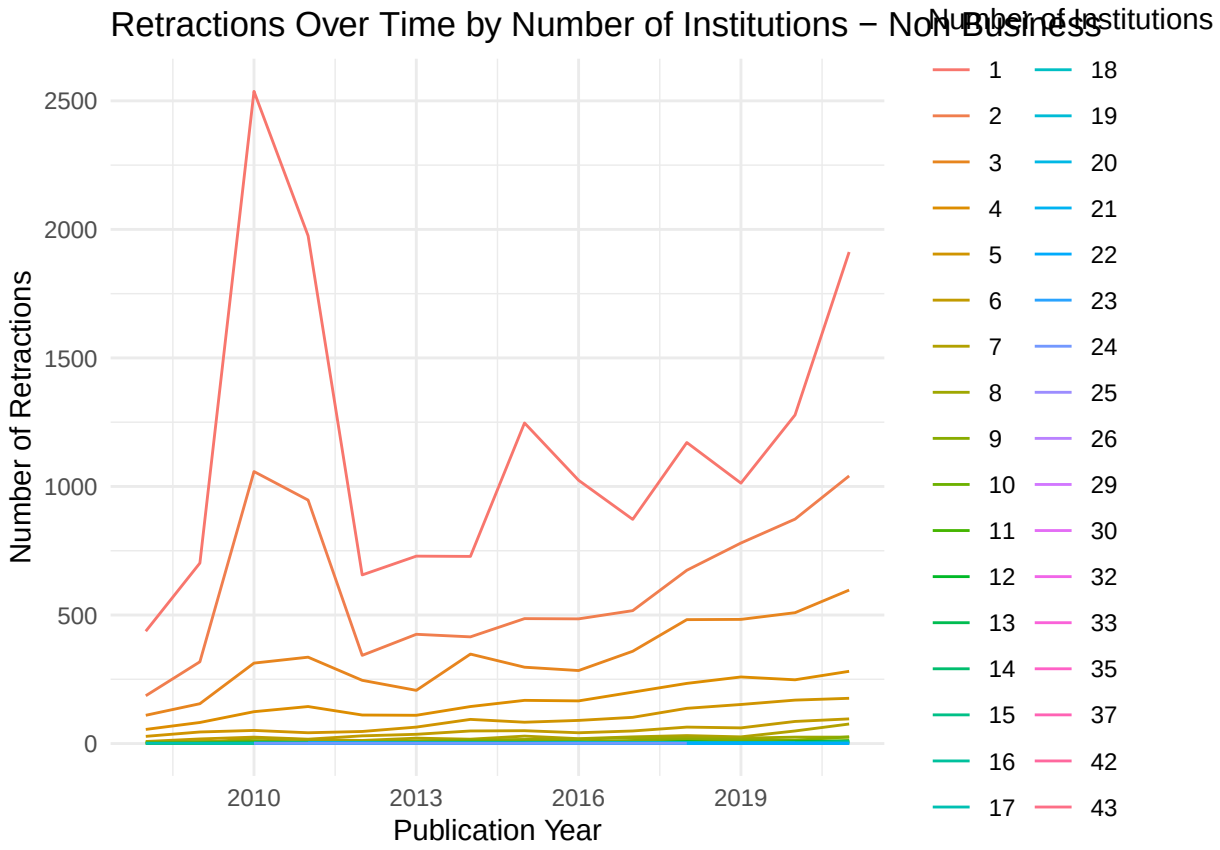
```
## Warning: Removed 60 rows containing missing values (`geom_line()`).
```



Retractions Over Time by Number of Institutions - Non Business

```
# Aggregate data
retractions_over_time_institutions <- data_nonbusiness %>%
  group_by(publication_year, num_institutions) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time_institutions, aes(x = publication_year, y = retractions, color = as.factor
  geom_line() +
  labs(title = "Retractions Over Time by Number of Institutions - Non Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Institutions") +
  theme_minimal() +
       scale_x_continuous(limits = c(2008, 2021)) #
```

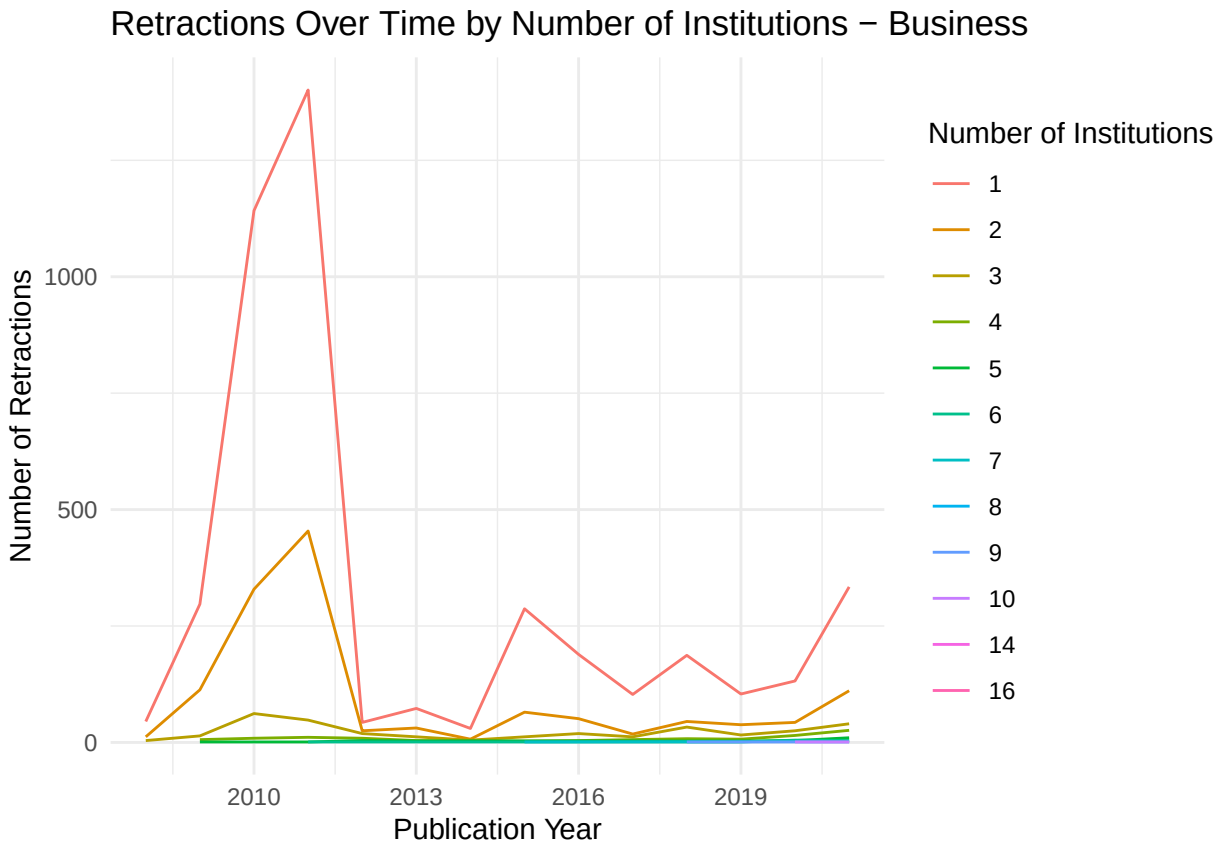
```
## Warning: Removed 221 rows containing missing values (`geom_line()`).
```



Retractions Over Time by Number of Institutions - Business

```
# Aggregate data
retractions_over_time_institutions <- data_business %>%
  group_by(publication_year, num_institutions) %>%
  summarise(retractions = n(), .groups = 'drop')

# Create the plot
ggplot(retractions_over_time_institutions, aes(x = publication_year, y = retractions, color = as.factor
  geom_line() +
  labs(title = "Retractions Over Time by Number of Institutions - Business",
       x = "Publication Year",
       y = "Number of Retractions",
       color = "Number of Institutions") +
  theme_minimal() +
       scale_x_continuous(limits = c(2008, 2021)) #
```

```
## Warning: Removed 47 rows containing missing values (`geom_line()`).
```


Retractions Over Time by Number of Institutions − Business
Number of Retractions
1000
500
0
2010
2013
2016
2019
Publication Year
Number of Institutions
1
2
3
4
5
6
7
8
9
10
14
16