Load Packages

#install.packages(openxlsx)  
#install.packages("rgl")  
#install.packages("plotly")  
library(openxlsx)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(readxl)  
library(ggplot2)  
  
data <- read\_xls("C:/Users/KCVUSA1/Downloads/data\_extraction\_1.16.24.xls")  
  
colnames(data) <- tolower(colnames(data))  
data <- data %>% rename\_with(~str\_replace\_all(., " ", "\_"), everything())  
#str(data)  
#View(data)

Finding the Top 50 Article topics

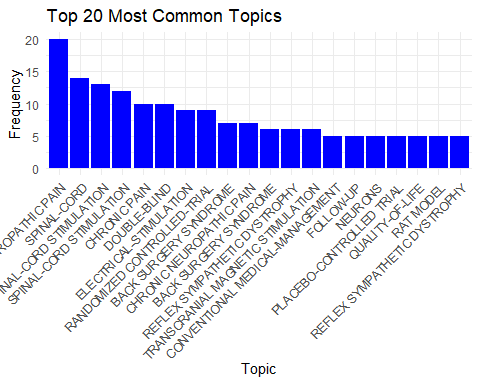
#Finding the Top 50 Article Topics  
topics\_column <- data$keywords\_plus  
  
# Concatenate all topics into a single string  
all\_topics <- paste(topics\_column, collapse = ';')  
  
# Split the string into a vector of individual topics  
all\_topics\_vector <- unlist(strsplit(all\_topics, ';'))  
  
# Create a data frame from the vector of topics  
topics\_df <- data.frame(topic = all\_topics\_vector)  
  
# Count the occurrences of each topic  
topic\_counts <- table(topics\_df$topic)  
  
# Order the topics by frequency in descending order  
sorted\_topics <- sort(topic\_counts, decreasing = TRUE)  
  
# Display the top 10 most common topics  
head(sorted\_topics, 50)

##   
## NEUROPATHIC PAIN SPINAL-CORD   
## 20 14   
## SPINAL-CORD STIMULATION SPINAL-CORD STIMULATION   
## 13 12   
## CHRONIC PAIN DOUBLE-BLIND   
## 10 10   
## ELECTRICAL-STIMULATION RANDOMIZED CONTROLLED-TRIAL   
## 9 9   
## BACK SURGERY SYNDROME CHRONIC NEUROPATHIC PAIN   
## 7 7   
## BACK SURGERY SYNDROME REFLEX SYMPATHETIC DYSTROPHY   
## 6 6   
## TRANSCRANIAL MAGNETIC STIMULATION CONVENTIONAL MEDICAL-MANAGEMENT   
## 6 5   
## FOLLOW-UP NEURONS   
## 5 5   
## PLACEBO-CONTROLLED TRIAL QUALITY-OF-LIFE   
## 5 5   
## RAT MODEL REFLEX SYMPATHETIC DYSTROPHY   
## 5 5   
## CHRONIC NONCANCER PAIN DIABETIC PERIPHERAL NEUROPATHY   
## 4 4   
## DORSAL COLUMN STIMULATION DORSAL HORN   
## 4 4   
## EXPERIENCE INTRACTABLE PAIN   
## 4 4   
## LOW-BACK-PAIN MECHANISMS   
## 4 4   
## MOTOR CORTEX STIMULATION RAT SPINAL-CORD   
## 4 4   
## SYNDROME TYPE-I THERAPY   
## 4 4   
## SPINAL-CORD-INJURY ACTIVATION   
## 4 3   
## BRAIN-STEM COST-EFFECTIVENESS   
## 3 3   
## DIRECT-CURRENT STIMULATION ELECTRICAL NERVE-STIMULATION   
## 3 3   
## INTENSITY INTRACTABLE ANGINA-PECTORIS   
## 3 3   
## INTRACTABLE DEAFFERENTATION PAIN LEFT PREFRONTAL RTMS   
## 3 3   
## LEG PAIN MANAGEMENT   
## 3 3   
## NEUROSTIMULATION THERAPIES PATHOLOGICAL PAIN   
## 3 3   
## PERIAQUEDUCTAL GRAY POSTHERPETIC NEURALGIA   
## 3 3   
## PSYCHOLOGICAL-FACTORS REFRACTORY ANGINA-PECTORIS   
## 3 3

T

Converting the top 50 article topics to a histogram distribution

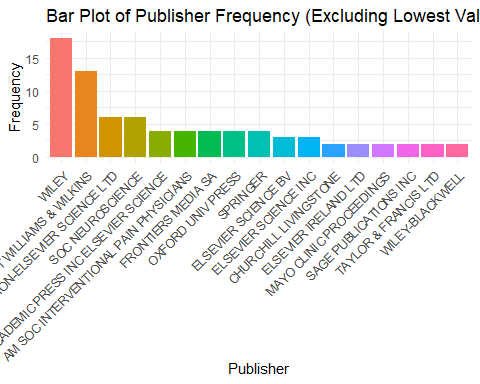
# Convert the topic\_counts to a data frame for plotting  
topics\_df <- data.frame(topic = names(topic\_counts), count = as.numeric(topic\_counts))  
  
# Sort the data frame by count in descending order  
topics\_df <- topics\_df[order(-topics\_df$count), ]  
  
# Select the first 50 topics  
topics\_df <- head(topics\_df, 20)  
  
# Create a histogram  
plot <- ggplot(topics\_df, aes(x = reorder(topic, -count), y = count)) +  
 geom\_bar(stat = "identity", fill = "blue") +  
 theme\_minimal() +  
 labs(title = "Top 20 Most Common Topics",  
 x = "Topic",  
 y = "Frequency") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
print(plot)



# Save the plot as a PNG file  
ggsave("top\_20\_topics\_histogram.png", plot, width = 20, height = 6, units = "in")

Histogram of the most common Publishers

# Use table function to get counts  
publisher\_counts <- table(data$publisher)  
  
# Convert the table to a data frame  
publisher\_counts\_df <- as.data.frame(publisher\_counts)  
  
# Rename the columns for clarity  
colnames(publisher\_counts\_df) <- c("Publisher", "Frequency")  
  
# Set a threshold for excluding the lowest values  
threshold <- 2 # Using the value 2 excludes publishers who've only published 1 paper  
  
# Filter data to exclude the lowest values  
filtered\_data <- subset(publisher\_counts\_df, Frequency >= threshold)  
  
# Reorder levels based on frequency  
filtered\_data$Publisher <- factor(filtered\_data$Publisher, levels = names(sort(table(data$publisher), decreasing = TRUE)))  
  
# Create a regular bar plot with excluded lowest values  
plot3 <- ggplot(filtered\_data, aes(x = Publisher, y = Frequency, fill = Publisher)) +  
 geom\_bar(stat = "identity") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1),  
 legend.position = "none") +  
 labs(title = "Bar Plot of Publisher Frequency (Excluding Lowest Values)",   
 x = "Publisher", y = "Frequency")  
print(plot3)



#Save the plot as a PNG file  
ggsave("Publisher Frequency.jpg", plot3, width = 20, height = 6, units = "in")

Article Type Distribution

#What type of articles are there?  
unique\_article\_types <- unique(data$document\_type)  
print(unique\_article\_types)

## [1] "Article" "Review"

#How many articles and review are there?  
article <- "Journal"  
  
# Use table function to get counts  
article\_type\_counts <- table(data$document\_type)  
print(article\_type\_counts)

##   
## Article Review   
## 57 43

Which article title appears most frequently?

title\_counts <- table(data$article\_title)  
  
# Find the most common title  
most\_common\_title <- names(title\_counts)[which.max(title\_counts)]  
  
# Display the most common title  
print(most\_common\_title)

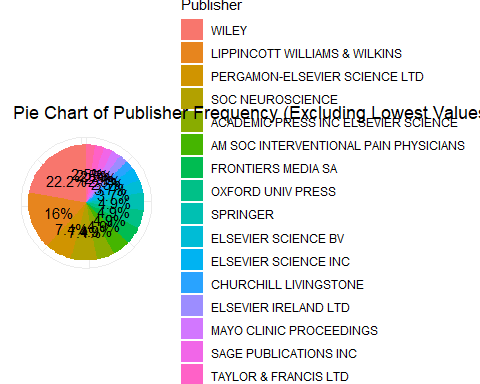
## [1] "Complex regional pain syndrome"

#Display the articles  
associated\_articles <- data[data$article\_title == most\_common\_title, ]  
print(associated\_articles)

## # A tibble: 2 × 72  
## publication\_type authors book\_authors book\_editors book\_group\_authors  
## <chr> <chr> <lgl> <lgl> <lgl>   
## 1 J Bruehl, S NA NA NA   
## 2 J Rho, RH; Brewer… NA NA NA   
## # ℹ 67 more variables: author\_full\_names <chr>, book\_author\_full\_names <lgl>,  
## # group\_authors <lgl>, article\_title <chr>, source\_title <chr>,  
## # book\_series\_title <lgl>, book\_series\_subtitle <lgl>, language <chr>,  
## # document\_type <chr>, conference\_title <lgl>, conference\_date <lgl>,  
## # conference\_location <lgl>, conference\_sponsor <lgl>, conference\_host <lgl>,  
## # author\_keywords <chr>, keywords\_plus <chr>, abstract <chr>,  
## # addresses <chr>, affiliations <chr>, reprint\_addresses <chr>, …

PieChart

# Use table function to get counts  
publisher\_counts <- table(data$publisher)  
  
# Convert the table to a data frame  
publisher\_counts\_df <- as.data.frame(publisher\_counts)  
  
# Rename the columns for clarity  
colnames(publisher\_counts\_df) <- c("Publisher", "Frequency")  
  
# Set a threshold for excluding the lowest values  
threshold <- 2   
  
# Filter data to exclude the lowest values  
filtered\_data <- subset(publisher\_counts\_df, Frequency >= threshold)  
  
# Reorder levels based on frequency  
filtered\_data$Publisher <- factor(filtered\_data$Publisher, levels = names(sort(table(data$publisher), decreasing = TRUE)))  
  
# Calculate percentages  
filtered\_data$Percentage <- filtered\_data$Frequency / sum(filtered\_data$Frequency) \* 100  
  
# Create a pie chart with distinct colors and percentage labels  
plot\_pie <- ggplot(filtered\_data, aes(x = "", y = Frequency, fill = Publisher)) +  
 geom\_bar(stat = "identity", width = 1) +  
 coord\_polar(theta = "y") + # Convert to pie chart  
 theme\_minimal() +  
 theme(axis.text.x = element\_blank(), # Remove x-axis text  
 ) +  
 labs(title = "Pie Chart of Publisher Frequency (Excluding Lowest Values)",   
 x = NULL, y = NULL) +  
 geom\_text(aes(label = paste0(round(Percentage, 1), "%")), position = position\_stack(vjust = 0.5))  
  
print(plot\_pie)



ggsave("pie\_chart.png", plot\_pie, width = 8, height = 8, units = "in", dpi = 300)