INTRODUCTION

In this group project, as the Information Retrieval Specialist, the leading role is to analyze descriptive textual data from plant families using RStudio. The objective was to extract significant insights from natural language descriptions by processing the textual information and vectorizing it. This primarily involved working with an XML dataset, transforming it into a structured CSV file, and then further analysis was carried out in R Studio, applying various Natural Language Processing (NLP) methods.

To be more precise, the objectives are:

- Extract the FamilyName and Description fields from an XML dataset using XQuery in BaseX.
- Pre-process some textual data using techniques such as lowercase conversion, punctuation removal, stop-word removal, and lemmatization.
- Create a TF-IDF matrix to measure the importance of words used in the family descriptions.
- Compare the families with each other using cosine similarity, with a primary focus on Apocynaceae.
- Show the most similar families and top terms using bar plots and word clouds.

It has been shown that botanical families are most likely to be linguistically similar, as indicated by their descriptive texts, which provide insight into how descriptive text supports classification or the discovery of structured datasets.

TASK EXECUTION

1. **# Read the extracted CSV file and display the first few rows of the dataset**
   families <- read.csv("families.csv", header = FALSE, sep = ",", quote = "\"", fill = TRUE)
   colnames(families) <- c("FamilyName", "Description")
   head(families)

```
  FamilyName
1 FamilyName
2 Asteraceae
3  Liliaceae
4   Rosaceae
5 Orchidaceae
6   Fabaceae
                                                  Description
1                                                 Description
2                                    Known as the daisy family
3 The lily family consists of flowering plants like lilies and tulips
4                                  The rose family includes roses
5                               One of the largest plant families
6                                    Also known as the pea family
                                                          NA
1
2                          this group includes sunflowers
3        known for their showy flowers and ornamental value.
4                                                strawberries
5 orchids are prized for their unique and intricate flowers.
6                          it includes legumes like beans
                                                                NA
1
2                                                       daisies
3
4 and apples. They are vital for their fruit and ornamental flowers.
5
6                                                         peas
                                                            NA
1
2 and asters. They are distinguished by their composite flower heads.
3
4
5
6                                                   and lentils.
>
```

2. **# Load necessary libraries**
   library(readr)
   library(textstem)
   library(tm)
   library(stringr)
   library(wordcloud)
   library(tidytext)
   library(tidyverse)

```
> # Load necessary libraries
> library(readr)
> library(textstem)
> library(tm)
> library(stringr)
> library(wordcloud)
> library(tidytext)
> library(tidyverse)
> detach("package:RColorBrewer", unload = TRUE)
Error: package 'RColorBrewer' is required by 'wordcloud' so will not be detached
> library(RColorBrewer)
```

3.
```
# Create a corpus from the descriptions
corpus <- VCorpus(VectorSource(families$Description))
```

4.
```
# Function to perform text pre-processing
preprocess_text <- function(corpus) {
  corpus <- tm_map(corpus, content_transformer(tolower)) # Convert to lowercase
  corpus <- tm_map(corpus, removePunctuation) # Remove Punctuation
  corpus <- tm_map(corpus, removeWords, stopwords("english")) # Remove stop-words
  corpus <- tm_map(corpus, stripWhitespace) # Remove extra white space
  corpus <- tm_map(corpus, content_transformer(lemmatize_strings)) # Lemmatize text
  return(corpus)
}
```

5.
```
# This will apply all the cleaning steps above to the text data provided
corpus <- preprocess_text(corpus)

# Create the Document-Term Matrix and TF-IDF
tdm <- TermDocumentMatrix(corpus)
tdm_matrix <- as.matrix(tdm)
dtm_matrix <- t(tdm_matrix)

# Create TF-IDF weighted matrix manually
tf <- dtm_matrix / rowSums(dtm_matrix) # Term Frequency
idf <- log(nrow(dtm_matrix) / colSums(dtm_matrix > 0)) # Inverse Document Frequency
tfidf <- tf %*% diag(idf) # Multiply TF by IDF
tfidf_df <- as.data.frame(tfidf) # Convert to data frame
dim(tfidf_df) # Display dimensions of the TF-IDF data-frame
```

```
> # Create TF-IDF weighted matrix manually
> tf <- dtm_matrix / rowSums(dtm_matrix) # Term Frequency
> idf <- log(nrow(dtm_matrix) / colSums(dtm_matrix > 0)) # Inverse Document Frequency
> tfidf <- tf %*% diag(idf) # Multiply TF by IDF
> tfidf_df <- as.data.frame(tfidf) # Convert to data frame
> dim(tfidf_df) # Display dimensions of the TF-IDF data-frame
[1]  31 104
```

6.

```
# Analyze Sparsity (calculate and print output)
total_elements <- nrow(tfidf_df) * ncol(tfidf_df)
non_zero_elements <- sum(tfidf_df != 0)
sparsity <- 1 - (non_zero_elements / total_elements)
cat("Sparsity of the TF-IDF matrix:", round(sparsity * 100, 2), "%\n")
```

```
> # Analyze Sparsity (calculate and print output)
> total_elements <- nrow(tfidf_df) * ncol(tfidf_df)
> non_zero_elements <- sum(tfidf_df != 0)
> sparsity <- 1 - (non_zero_elements / total_elements)
> cat("Sparsity of the TF-IDF matrix:", round(sparsity * 100, 2), "%\n")
Sparsity of the TF-IDF matrix: 93.58 %
```

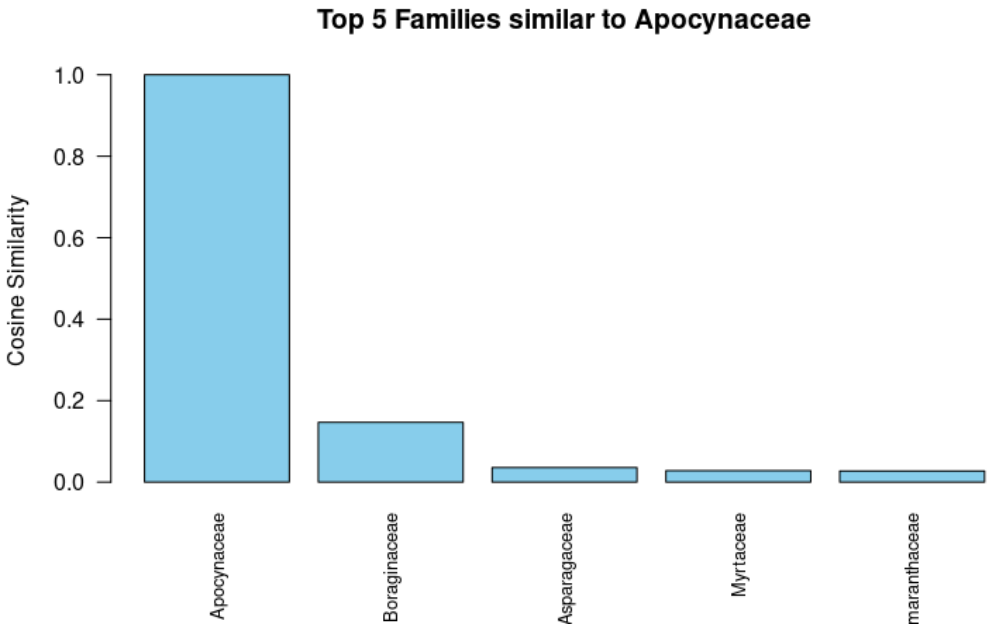
7.

```
# Get the vector for Apocynaceae
apocynaceae_index <- which(families$FamilyName == "Apocynaceae")
apocynaceae_vector<- as.numeric(tfidf_df[apocynaceae_index, ])

# Compute cosine similarity with all other families and combine with family name
similarities <- apply(tfidf_df, 1, function(x) cosine(apocynaceae_vector, as.numeric(x)))
similarity_df <- data.frame(FamilyName = families$FamilyName, cosinesimilarity = similarities)

# Sort and show top similar families (example: top 5)
similar_families <- similarity_df[order(-similarity_df$cosinesimilarity), ]
head(similar_families, 5)
```

```
> # Get the vector for Apocynaceae
> apocynaceae_index <- which(families$FamilyName == "Apocynaceae")
> apocynaceae_vector<- as.numeric(tfidf_df[apocynaceae_index, ])
> # Compute cosine similarity with all other families and combine with family name
> similarities <- apply(tfidf_df, 1, function(x) cosine(apocynaceae_vector, as.numeric(x)))
> similarity_df <- data.frame(FamilyName = families$FamilyName, cosinesimilarity = similarities)
> # Sort and show top similar families (example: top 5)
> similar_families <- similarity_df[order(-similarity_df$cosinesimilarity), ]
> head(similar_families, 5)
      FamilyName cosinesimilarity
26   Apocynaceae       1.00000000
21  Boraginaceae       0.14676750
24  Asparagaceae       0.03587406
13     Myrtaceae       0.02811857
16 Amaranthaceae       0.02725981
```

8.

```
> # Bar plot creation of the top 5 families
> top5 <- head(similar_families, 5)
> barplot(top5$cosinesimilarity,
+         names.arg = top5$FamilyName,
+         main = "Top 5 Families similar to Apocynaceae",
+         ylab = "Cosine Similarity",
+         las = 2,
+         col = "skyblue",
+         cex.names = 0.8)
```



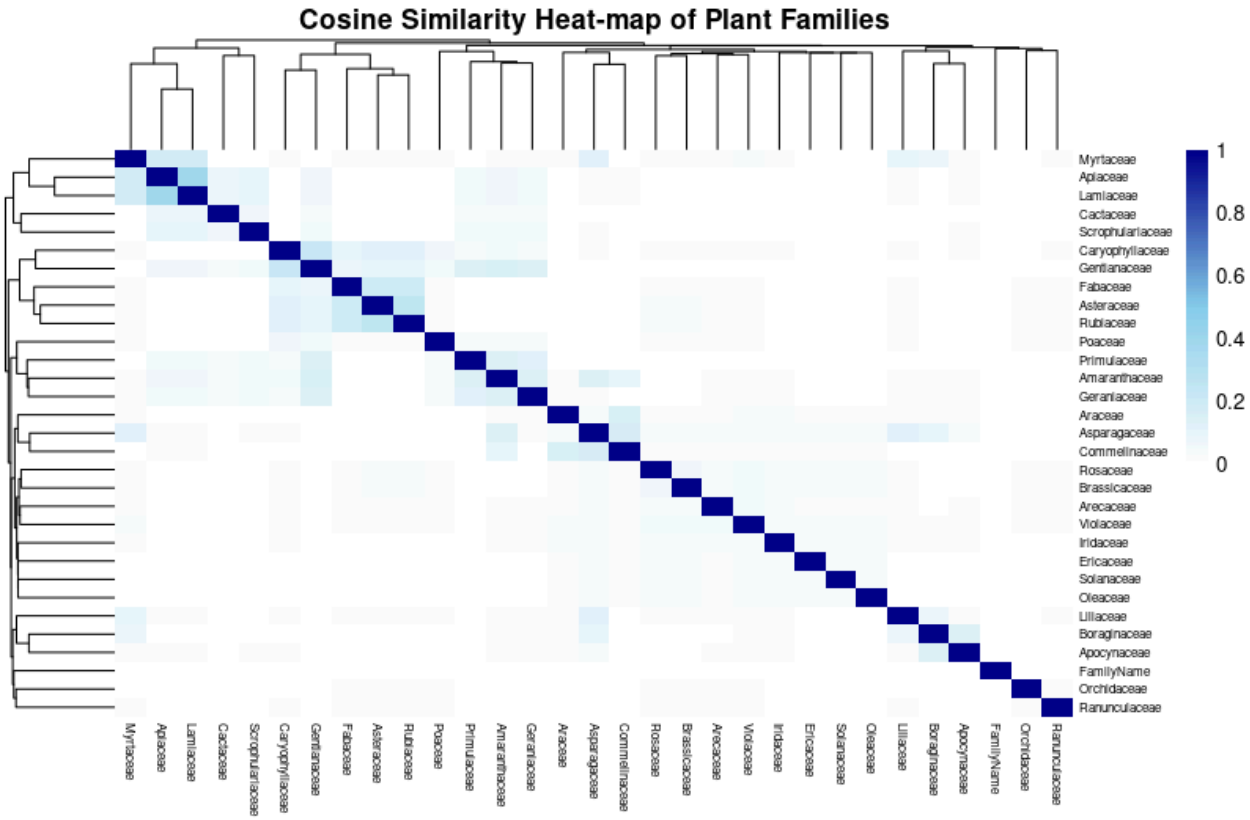
9.

```
# Get top 10 TF-IDF terms for Apocynaceae
terms <- colnames(tfidf_df)
apocynaceae_terms <- as.numeric(tfidf_df[apocynaceae_index, ])
names(apocynaceae_terms) <- terms
top10_terms <- sort(apocynaceae_terms, decreasing = TRUE)[1:10]
print(top10_terms)
```

```
> # Get top 10 TF-IDF terms for Apocynaceae
> terms <- colnames(tfidf_df)
> apocynaceae_terms <- as.numeric(tfidf_df[apocynaceae_index, ])
> names(apocynaceae_terms) <- terms
> top10_terms <- sort(apocynaceae_terms, decreasing = TRUE)[1:10]
> print(top10_terms)
    dogbane   milkweed. periwinkles     include         and      plants      family     adapted      agave.
 0.49056960  0.49056960  0.49056960  0.39154857  0.13558294  0.10370529  0.01454038  0.00000000  0.00000000
agriculture
 0.00000000
```

10.

```
# Heat-map of Cosine Similarity between all families
library(text2vec) # Before computing below, install text2vec since sim2 is from there
similarity_matrix <- sim2(as.matrix(tfidf_df), method = "cosine", norm = "l2") # Compute pairwise cosine similarities
similarity_matrix <- round(similarity_matrix, 2) # round the number for output
rownames(similarity_matrix) <- families$FamilyName # Add row names
colnames(similarity_matrix) <- families$FamilyName # Add column names
library(pheatmap) # Load visualization library
pheatmap(similarity_matrix,
         main = "Cosine Similarity Heat-map of Plant Families",
         fontsize_row = 6,
         fontsize_col = 6,
         color = colorRampPalette(c("White", "skyblue", "darkblue"))(100),
         border_color = NA)
```



Assumptions and Challenges

The R package version compatibility became one of the main hurdles in this project. Initially, I attempted to continue working on RStudio Desktop on my laptop, which had R version 4.1. As a result, I encountered difficulties installing and using the required packages, including rmarkdown, knitr, and others, for the final rendering of the report into HTML format. Despite various attempts to install the dependencies, I was unable to knit the R Markdown file on that particular setup. I opted for Posit Cloud, which runs the latest version of RStudio (version 4.4.3). This environment already contained the tools required for the analyses in this project, allowing me to install all packages without encountering version conflicts. It streamlined the entire process, and I generated the required HTML report.

Additionally, I assumed that the family descriptions extracted from the XML file were clean and properly formatted, which was crucial for accurately calculating TF-IDF scores and cosine similarity metrics. I have also taken the stance that lemmatization should suffice to reduce terms to their root forms without stemming.

Conclusion

In this project, I demonstrated how text mining techniques can be applied to botanical family descriptions to reveal semantic relationships. Proceeding with the extracted dataset, I used multiple text preprocessing techniques, including lowercase conversion, punctuation removal, stop-word filtering, and lemmatization, to prepare the data for meaningful analysis.

Thus, by constructing a TF-IDF matrix, I was able to evaluate the importance of terms across families quantitatively. The resulting TF-IDF matrix was of size 31 × 10^4 with a sparsity of 93.58%, which is characteristic of text data, where all documents do not share most terms.

From this matrix, I then took the 10 highest TF-IDF terms of the Apocynaceae family - these being the words that, for this corpus, are most unique and descriptive to it. It also included computing cosine similarity between Apocynaceae and other families to identify the most similar ones (such as Boraginaceae, Asparagaceae, and Myrtaceae). Then it visualized the top 5 using a bar plot and all pairwise similarities using a heatmap.

This would provide further evidence of how non-natural language-derived descriptions can indeed be converted into numerical representational forms for quantitative similarity analysis, revealing even descriptive, non-structured relationships.