**Potrzeba klienta**

System do analizy częstości tekstu, która pozwoli określić, jakie słowa i tematy dominują w pliku tekstowym (tu przemówienia Bidena z 2021 i 2024), a także wskazać różnice w sposobie przekazu i priorytetach prezydenta w danym wystąpieniu. System ma wizualizować wyniki.

Etapy procesu tworzenia systemu informatycznego

1. Planowanie

Celem analizy jest **identyfikacja i porównanie najczęściej występujących słów w przemówieniu** (tu prezydenta Joe Bidena). Analiza pozwoli określić **dominujące tematy oraz potencjalne podobieństwa/różnice** dla każdego przemówienia.

1. Analiza (wymagań)

Proces obejmie **ekstrakcję tekstu, jego wstępne przetworzenie oraz analizę częstości występowania słów**, a także wizualizację wyników w formie **chmur słów oraz wykresów słupkowych**.

1. Projektowanie

Przygotowanie metod analizy tekstu i wizualizacji wyników

1. **Wczytanie tekstu przemówień** – import pliku tekstowego zawierającego przemówienie.
2. **Przetwarzanie tekstu** – oczyszczenie danych, usunięcie znaków interpunkcyjnych i konwersja tekstu do postaci tokenów.
3. **Usunięcie stop słów** – eliminacja słów o wysokiej częstości, ale niskiej wartości analitycznej (np. „i”, „oraz”, „dla”).
4. **Analiza częstości słów** – identyfikacja i porównanie najczęściej występujących terminów w obu przemówieniach.
5. **Wizualizacja wyników**:
   * **Wykresy słupkowe** – przedstawienie najczęściej używanych słów i ich liczebności.
   * **Chmury słów** – graficzne zobrazowanie częstości słów, gdzie większa czcionka oznacza wyższe występowanie.
   * **Porównanie wyników** – zestawienie najważniejszych różnic między przemówieniami pod kątem słownictwa i tematów.
6. Implementacja -wytworzenie kodu systemu

Zadanie: rozsypanka

Rozwiązanie (kod) nie jest uporządkowane. Zidentyfikuj kolejność i uporządkuj proces implementacji i kodoania tworzenia systemu informatycznego, a następnie uruchom kod i wykonaj analizy dla obu plików tekstowych (przemówień Bidena z 2021 i 2024). Czy priorytety wykryte w obu przemówieniach są podobne czy różnią się?

# Dodanie różnych palet kolorystycznych

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, min.freq = 4, colors = brewer.pal(9,"Blues"))

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, min.freq = 4, colors = brewer.pal(9,"Reds"))

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, min.freq = 4, colors = brewer.pal(9,"Greens"))

frequent\_terms <- freq\_terms(text)

frequent\_terms

frequent\_terms <- freq\_terms(text, stopwords = Top200Words)

plot(frequent\_terms)

# Utwórz chmurę słów

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ)

# Opcje chmury słów

?wordcloud

# Zmiana wartości min.freq i max.words w celu wyświetlenia mniejszej/większej liczby słów.

# min.freq: słowa o częstości poniżej tej wartości nie będą wyświetlane

# max.words: maksymalna liczba słów do wyświetlenia

# Wczytaj dane tekstowe

# Wczytaj plik tekstowy z lokalnego dysku

text <- readLines(file.choose())

text

# Ograniczenie liczby słów w chmurze poprzez określenie minimalnej częstości

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, min.freq = 4)

# Ograniczenie liczby słów w chmurze poprzez określenie maksymalnej liczby słów

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, max.words = 5)

# Optymalizacja i dostosowanie wyników

# Dodanie koloru do chmury słów dla lepszej wizualizacji

# Dodanie koloru

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, min.freq = 4, colors = brewer.pal(8,"Dark2"))

# Dodanie koloru

wordcloud(frequent\_terms$WORD, frequent\_terms$FREQ, max.words = 5, colors = brewer.pal(8,"Accent"))

?brewer.pal

brewer.pal.info

# Tworzenie chmury słów za pomocą pakietu wordcloud

install.packages("wordcloud")

library(wordcloud)