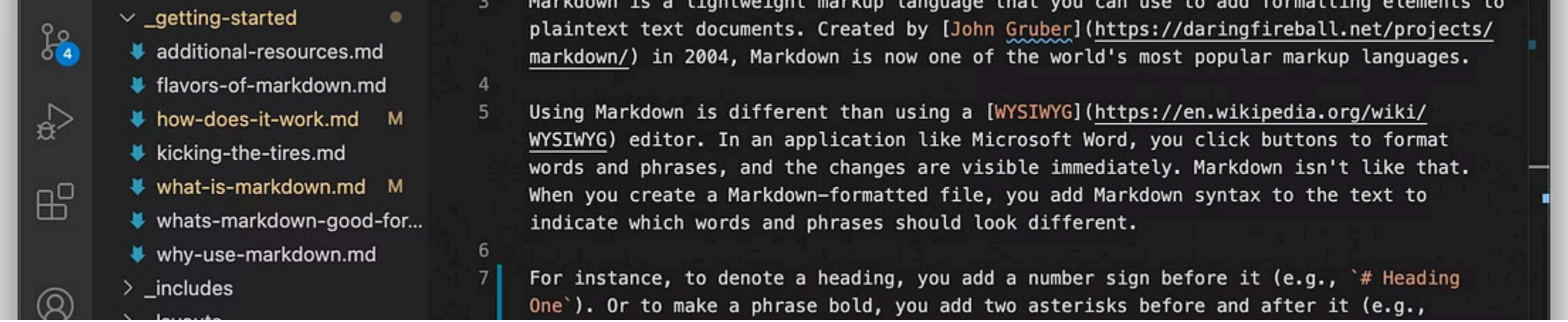

# Khaled walid mohamed ghalwash id : 2205018

# Building a Second Word Predictor Using a Bigram Model

This presentation walks through the creation of a second word predictor based on bigram frequency analysis. Using a sample text corpus, we preprocess the data, tokenize words, remove stopwords, and apply stemming to prepare for analysis. We then explore word frequency, TF-IDF calculations, and bigram extraction to build predictive models.

Finally, we implement a Shiny app interface that allows users to input a first word and receive predicted second words, supported by visualizations of top bigrams and predictions.




# Text Preprocessing and Tokenization

### Lowercasing and Cleaning

Text is converted to lowercase, punctuation and numbers are removed to standardize the corpus.

### Tokenization

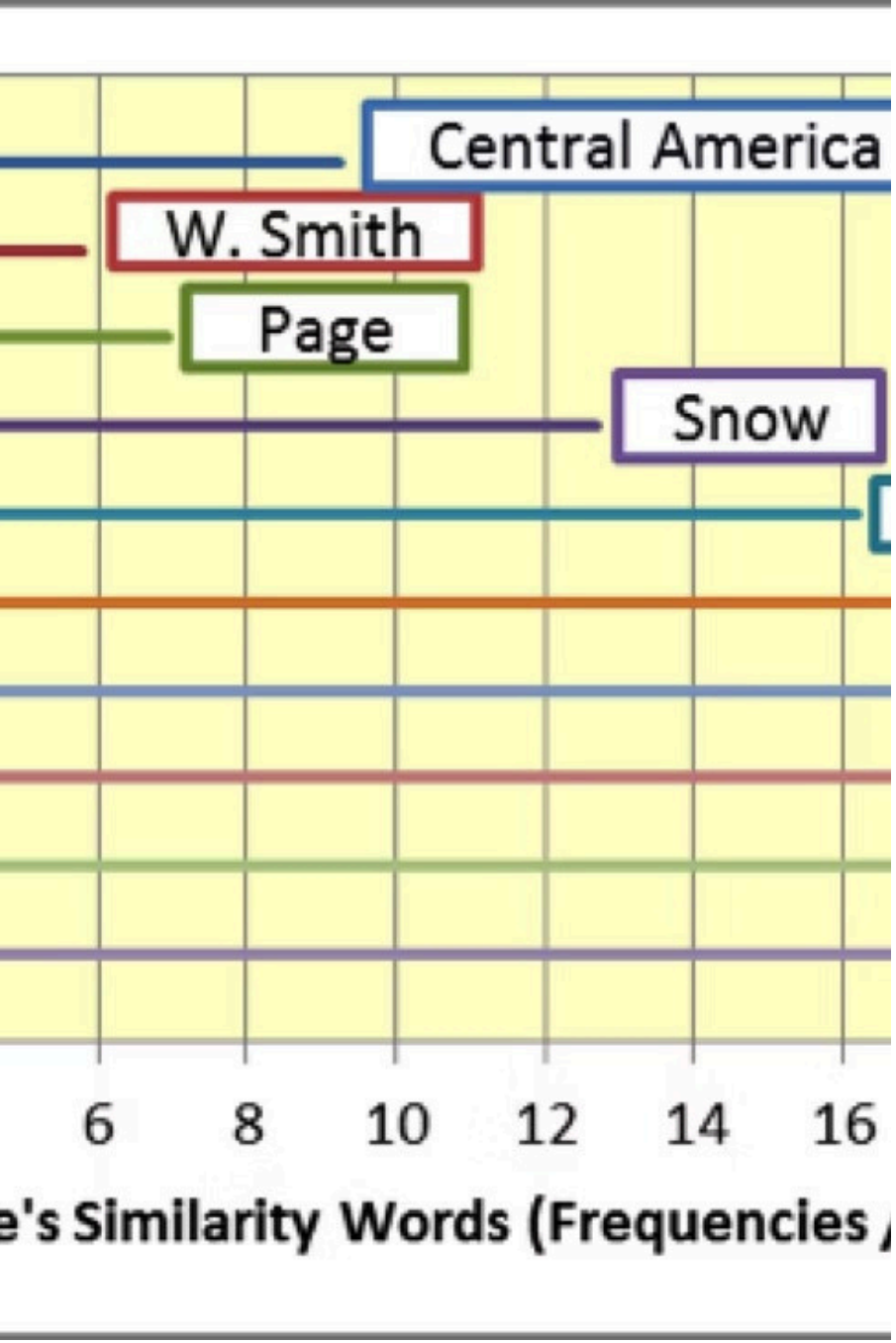
The cleaned text is split into individual words (tokens) for analysis.

### Stopword Removal

Common words like "the" and "and" are removed to focus on meaningful terms.

### Word Stemming

Words are reduced to their root forms to unify variations (e.g., "jumps" to "jump").




# Analyzing Word Frequencies

### Frequency Table

Counting occurrences of each stemmed word reveals the most common terms.

### Top 10 Words

The most frequent words provide insight into the corpus themes and content.

### Visualization

A horizontal bar plot displays the top 10 frequent words for easy interpretation.

# TF-IDF Calculation for Word Importance

### Term Frequency-Inverse Document Frequency

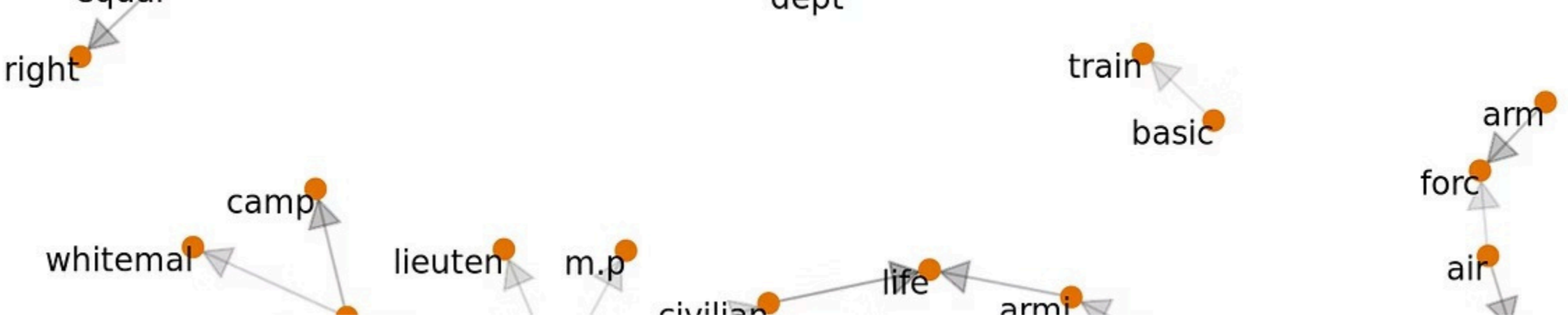
TF-IDF measures how important a word is to a document relative to the cor pus.

### Application

Calculating TF-IDF helps identify distinctive words that characterize each document.

### Top TF-IDF Words

The highest TF-IDF scores highlight the most unique and informative terms.




# Bigram Extraction and Frequency Analysis

### Bigram Tokenization

Text is split into pairs of consecutive words to capture word relationships.

### Frequency Counting

Counting bigrams reveals common word pairs and patterns in the corpus.

### Top Bigrams

The most frequent bigrams provide the foundation for predictive modeling.

# Shiny App UI Design

### Background and Styling

The app features a visually appealing background image with transparent panels for readability.

### User Input

Users enter a first word to trigger prediction of the second word using bigram data.

### Output Display

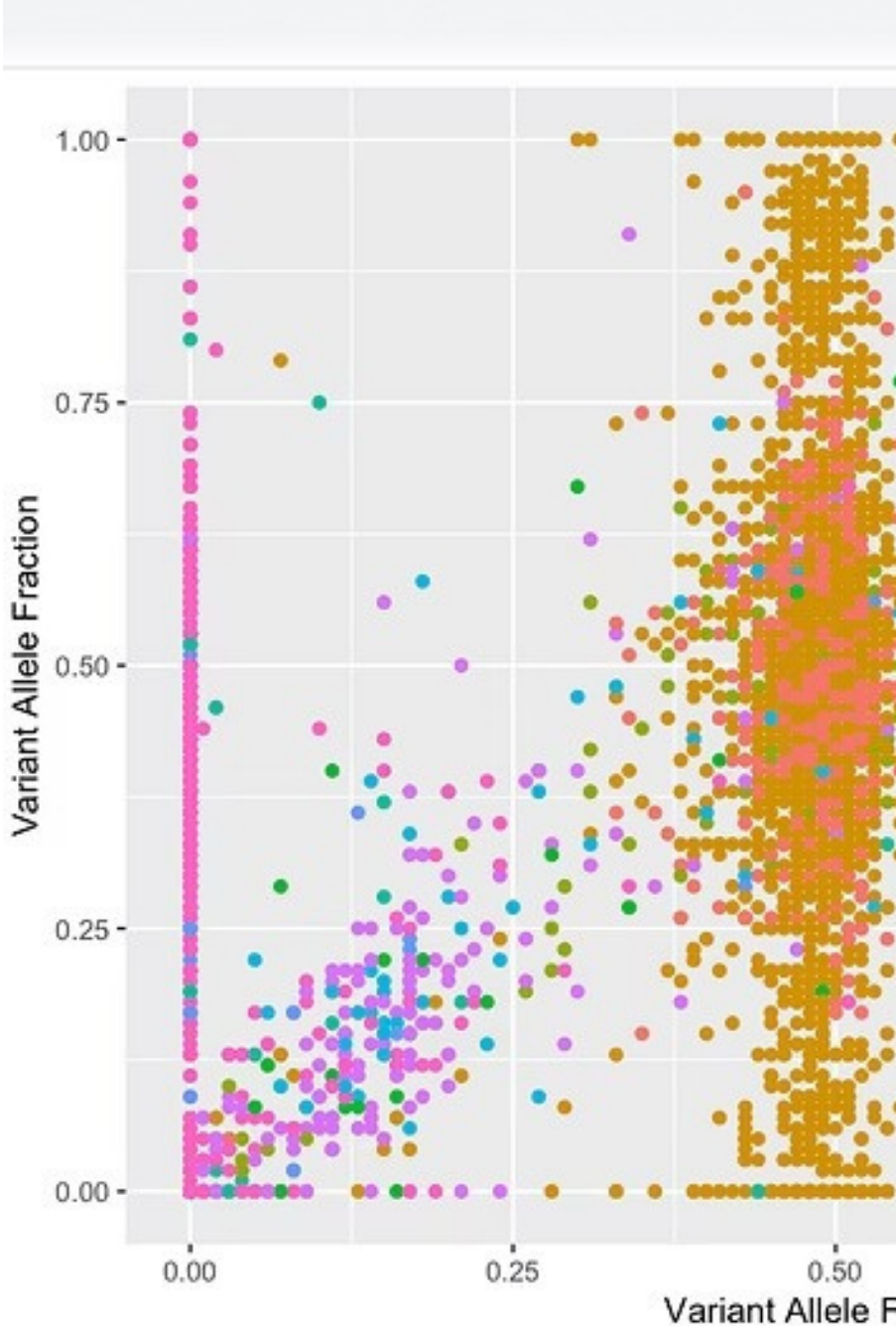
Predicted words and bigram frequency plots are displayed dynamically based on input.

# Server Logic and Prediction Workflow

**1** **Top Bigrams Plot**
Displays the overall top 10 bigrams in the corpus as a static plot.

**2** **Reactive Prediction**
When the user inputs a word, the server filters bigrams starting with that word and selects top predictions.

**3** **Dynamic Outputs**
Predicted second words are shown as text and a bar plot updates to visualize prediction frequencies.



# Summary and Next Steps

### Data Preparation
Cleaning, tokenization, stopword removal, and stemming are essential preprocessing steps.

### Modeling
Bigram frequency and TF-IDF analyses enable effective prediction of second words.

### Interactive Application
The Shiny app provides a user-friendly interface for exploring bigram-based predictions.

Future improvements could include expanding the corpus, refining prediction algorithms, and enhancing UI features.