# PE08: Text Visualization and Sentiment Analysis

## Overview

This exercise explores text visualization using word clouds and sentiment analysis on social media data. You’ll create various word cloud visualizations and analyze tweet sentiments using NLTK.

### Prerequisites

pip install wordcloud matplotlib nltk  
python -m nltk.downloader twitter\_samples vader\_lexicon

## Exercise 1: Word Cloud Generation

### Basic Word Cloud Implementation

Create a simple word cloud visualization using sample text about data science. Your implementation should: - Use the WordCloud class from wordcloud package - Set appropriate dimensions (e.g., 800x800) - Configure basic parameters like background color and minimum font size - Display and save the visualization

Hint: Start with these imports:

from wordcloud import WordCloud, STOPWORDS   
import matplotlib.pyplot as plt

### Questions

1. Enhance the word cloud with custom stop words:
   * Create a function that accepts text and optional custom stop words
   * Combine custom stop words with STOPWORDS from wordcloud
   * Generate two word clouds (original and with custom stops) for comparison
   * Display both clouds side by side using matplotlib subplots

* Hint: Consider common words like ‘said’, ‘using’, ‘would’ for your custom stop words list

1. Generate a frequency-based word cloud:
   * Create a function that finds top N most frequent words (default 50)
   * Filter out stop words and short words
   * Use Counter from collections to track word frequencies
   * Scale word sizes based on frequency
   * Display both the word cloud and a frequency list

* Hint: Use regex pattern r’+ with re.findall() for word tokenization

## Exercise 2: Social Media Sentiment Analysis

### Basic Sentiment Implementation

Using NLTK’s twitter samples and VADER sentiment analyzer: - Initialize SentimentIntensityAnalyzer - Load positive and negative tweet samples - Analyze and display sample tweets with their sentiment scores

Hint: Use try-except when initializing SentimentIntensityAnalyzer to handle missing lexicon

### Question

1. Implement extreme sentiment analysis:
   * Create functions to:
     + Find most extreme positive and negative tweets
     + Sort by compound sentiment score
     + Display formatted results with scores
   * Create a histogram showing sentiment score distribution

* Hint: Store tweet analysis results in dictionaries with ‘text’, ‘compound’, ‘pos’, and ‘neg’ keys

### Technical Requirements

* Use Python 3.x
* Include proper error handling
* Follow good coding practices (documentation, meaningful variable names)
* Create clear visualizations with proper labels and titles

### Submission Guidelines

1. Jupyter notebook including:
   * Working code with proper documentation
   * Clear visualizations
   * Output demonstrating functionality
2. Written analysis for each question:
   * Implementation approach
   * Comparison of results
   * Discussion of visualization choices

### Notes

* For word clouds, consider using sample text about a specific topic (e.g., data science)
* Pay attention to matplotlib figure sizes and layout
* Consider adding color schemes to enhance visualizations
* Think about ways to make the sentiment analysis output more readable