# Data Science Lab 3: Text Visualisations with R

## Part 1: Generative AI with Chat GPT

### Introduction

ChatGPT is built using an organisation called OpenAI's proprietary series of generative pre-trained transformer (GPT) models and is fine-tuned for conversational applications using a combination of supervised learning and reinforcement learning from human feedback. If you do not already have an account, go to this page and sign up for a free account:

https://chatgpt.com/

Many people will already have an account with chat GPT, if you do, then sign in. Once you have signed in you can ask the chat prompt a question, for this example i have asked the following question: “will scotland ever be independent?”

ChatGPT replied with the following answer:

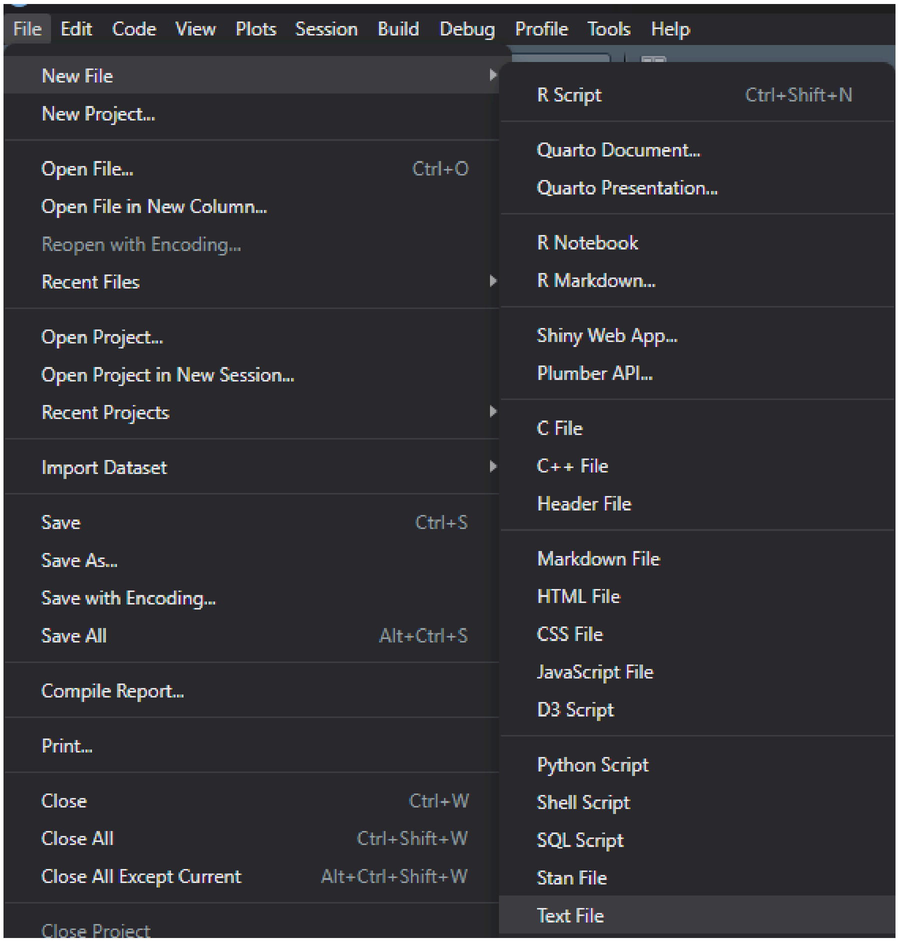
> “The future of Scottish independence is uncertain and depends on political, legal, and social factors that are constantly evolving. There are a few key elements that will shape the path forward:
>
> Public Support: Support for independence has fluctuated over time. The 2014 Scottish independence referendum resulted in 55% voting to remain part of the United Kingdom. However, since then, political changes, such as Brexit (which Scotland opposed), have led to renewed interest in independence. Polls show varying levels of support, and any future referendum will depend heavily on where public opinion stands.
>
> Political Will: The Scottish National Party (SNP) has been the main driving force behind independence, and it continues to push for another referendum. The UK government, however, must agree to legally allow another vote. As of now, the UK government has resisted these calls, arguing that the 2014 referendum was a "once in a generation" decision.
>
> Economic Concerns: Independence raises significant questions about Scotland's economy, including issues related to trade, currency, and public spending. A clear and convincing economic plan would likely be crucial to winning over undecided voters.

> International Context: If Scotland were to become independent, it would seek to rejoin the European Union, as many Scots opposed Brexit. However, the process of rejoining the EU could be complicated and lengthy.
>
> While it's impossible to predict with certainty whether Scotland will become independent, the issue remains central to Scottish politics, and another referendum could occur in the future if political and public conditions align.”

We can use this text to generate a data visualisation.

## Importing Data into R

1.1 Copy the answer text from ChatGPT above and paste it into a text file. This can be done with RStudio, in RStudio choose File > New File > Text File:



Then paste in the answer text from ChatGPT. Then Choose File Save and save the File as: `answer_gpt.txt`.

1.2 Now let’s create a new R script to work on the file. In R-Studio choose File > New File > R Script. Save the file as `text_script`.

1.3 Let's see if R can read the file, add the following to line 1 of the R script created in step 1.2 and then click on Run to run the code:

```
print(answer_gpt.txt)
```

1.4 This should fail with an error message, something like the following:

```
> print(answer_gpt.txt)
```

```
Error: object 'answer_gpt.txt' not found
```

1.5 We can't just print files out, we need to get R to read the file first, to do this we need to create a variable. To do this we are going to install the tm package:

```
install.packages("tm")
```

```
library(tm)
```

In this case we will call the variable “`my_file`” then we can pass the file to the `data.table` function. Inside the brackets we add the file name and say that the separator is a new line character, that's what the \n is for.

```
my_file = data.table::fread("answer_gpt.txt", sep = "\n")
```

```
print(my_file)
```

1.6 Hopefully this will print out the contents of the file, the next step is to create a variable called txt and pass my_file to the `PlainTextDocument` function. Then using `termFreq`, we can count the number of times each word appears in the file. Then we can sort these and get the top 10 using the `head` function.

```
txt <- PlainTextDocument(VectorSource(my_file))
x = termFreq(txt, control = list(tolower = T, removeNumbers = T,
removePunctuation = T))
head(sort(x, decreasing = TRUE), 10)
```

1.7 Now that we have worked through an example, return to ChatGPT and ask another question on the topic of your choice, then follow through the steps to generate the top 10 most common words in the answer.

# Part 2: R Wordclouds

## Install Libraries - Planning

2.1 Install the devtools package first, run these one at a time to give R a chance to install the packages, you may need to save your work and restart R studio during this process.

```
# Create R WordCloud Script - install packages
install.packages("devtools") # for dev tools
install.packages("tm")  # for text mining
install.packages("SnowballC") # for text stemming
install.packages("wordcloud") # word-cloud generator
install.packages("wordcloud2") # word-cloud generator
install.packages("RColorBrewer") # color palettes
# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("wordcloud2")
library("RColorBrewer")
```

## Obtain and Load a Text File - Data Acquisition / Pre-Processing

2.2 Go to the Moodle page and download the scotland.txt file, this is a text file made by copying the entire Wikipedia entry for Scotland into a text file. We can use this to generate a wordcloud about Scotland, place this in the same directory as your R script then run the following:

```
text <- readLines("C:\\UserName\\Documents\\NPA_DS\\scotland.txt")
docs <- Corpus(VectorSource(text))
# Convert the text to lowercase
docs <- tm_map(docs, content_transformer(tolower))
# Remove numbers
docs <- tm_map(docs, removeNumbers)
```

## Clean and Prepare the Data

2.3 Run the following to clean the data - the should give the top ten words at the end, stopwords are best to eliminate:

```
# specify your to remove stop words as a character vector
docs <- tm_map(docs, removeWords, c("the", "this", "said", "like", "now",
"went","come"))
```

2.4 The next commands will perform more data cleaning:

```
# Remove punctuation
docs <- tm_map(docs, removePunctuation)
```

```
# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
# Text stemming
docs <- tm_map(docs, stemDocument)
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
set.seed(1234)
```

## Data Analysis - Visualisation - Present

2.5 Here are three visualisation examples, use one of these to generate output, more options can be found here. The first example creates a bar chart of the most common words, then a wordcloud that demonstrates the default text which is about oil producing countries. Then two examples with our text using the wordcloud and then wordcloud2 packages.

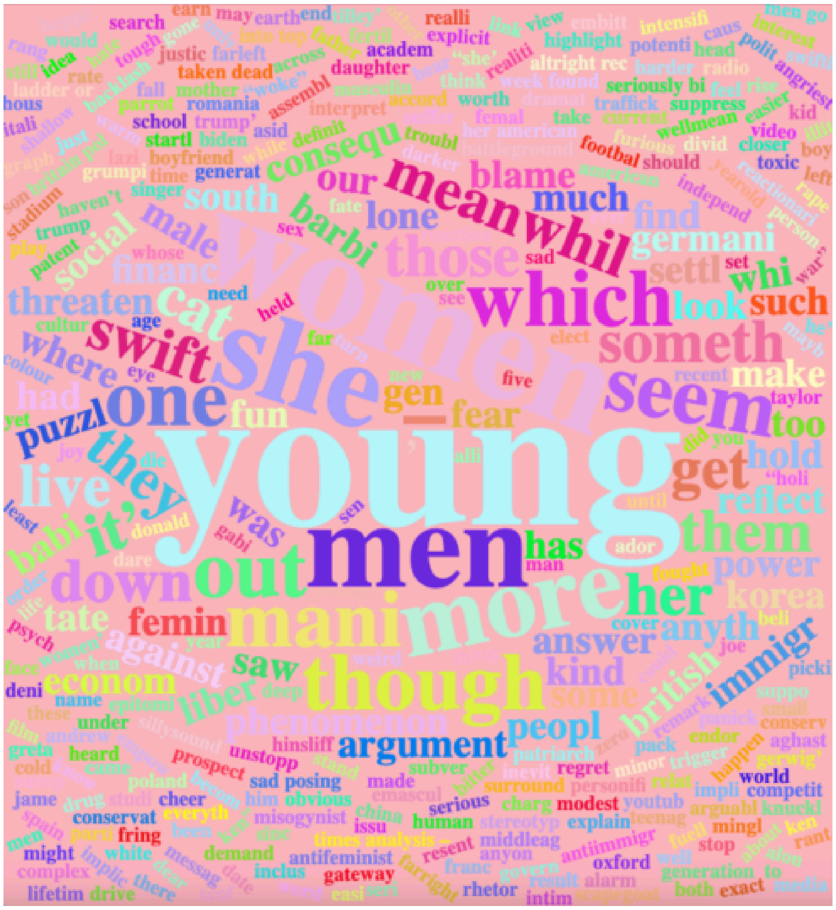
```
findFreqTerms(dtm, lowfreq = 4)
barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,
        col ="lightgreen", main ="Most frequent words",
        ylab = "Word frequencies")
# Example 1 - Wordcloud 2 with demo word collection
wordcloud2(demoFreq, color = "random-light", backgroundColor = "grey")

# Example 2 - Basic cloud with our words
wordcloud(words = d$word, rot.per=.5, max.words = 200,
          min.freq = 5, scale=c(4,.5))

# Example 3 - Improved cloud withe colour palette
wordcloud(words = d$word, freq = d$freq, min.freq = 15,
          max.words=50, random.order=FALSE, rot.per=0.5,
          colors=brewer.pal(9, "Dark2"))
```

## Example of wordcloud created

To give you an idea, here is an example of a student word cloud from a former student about an article about the Barbie movie and the social reaction to it.



## E-portfolio tasks

1. Find an article online that was written by a human (to the best of your knowledge). Then use this to create a wordcloud, by running the text from the article into a text file and using the R package to generate the wordcloud.
2. Add an explanation of why you chose the text that you have chosen, what are the most common words, was this expected?
3. Ask ChatGPT to write an article on the same topic as you found, then use this to create a second word cloud.
4. Are there any obvious differences between the human written article and the ChatGPT generated article, or do you think these are similar? Write a paragraph giving your opinion of the differences or similarities between the two articles.