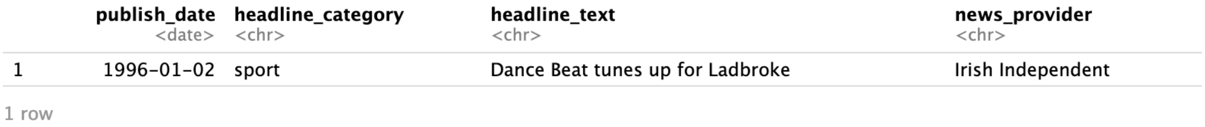
Question 1

```
## Load the required library
library(dplyr)
library(ggplot2)
#Read the CSV file into a DataFrame
irish_df <- read.csv("ireland_news.csv")
#Filter the data for articles from the "Irish Independent" news provider
irish_independent_df <- irish_df %>%
  filter(news_provider == "Irish Independent")
#Convert the "publish_date" column to date format
irish_independent_df$publish_date <- as.Date(irish_independent_df$publish_date, format = "%A,
%dth of %B, %Y")
#Remove NA values
irish_independent_df <- na.omit(irish_independent_df)
#Sort the DataFrame by the "publish_date" column in ascending order
sorted_irish_independent_df <- irish_independent_df %>%
  arrange(publish_date)
#Display the earliest and latest articles from the "Irish Independent"
earliest_article <- head(sorted_irish_independent_df, 1)
latest_article <- tail(sorted_irish_independent_df, 1)
#Print the earliest article from Irish Independent
print("Earliest article from Irish Independent:")
print(earliest_article[c("publish_date", "headline_category", "headline_text", "news_provider")])
#Print the Latest article from Irish Independent
print("\nLatest article from Irish Independent:")
print(latest_article[c("publish_date", "headline_category", "headline_text", "news_provider")])
# Display the last 5 records of the sorted DataFrame
print("\nLast 5 records of the sorted data:")
print(tail(sorted_irish_independent_df, 5))
```

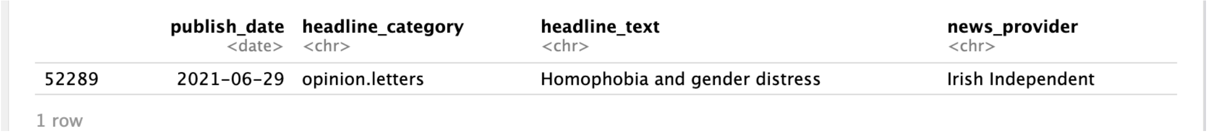
- **Output**

- **Print the earliest article from Irish Independent**



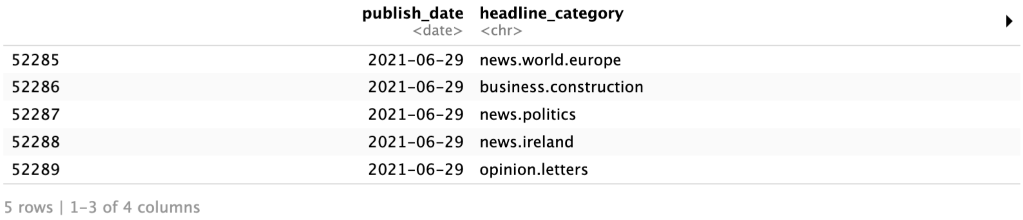
```
     publish_date  headline_category  headline_text                   news_provider
     <date>        <chr>              <chr>                           <chr>
1    1996-01-02    sport              Dance Beat tunes up for Ladbroke  Irish Independent
1 row
```

- **Print the Latest article from Irish Independent**



```
        publish_date  headline_category  headline_text                  news_provider
        <date>        <chr>              <chr>                          <chr>
52289   2021-06-29    opinion.letters    Homophobia and gender distress  Irish Independent
1 row
```

- **Display the last 5 records of the sorted DataFrame**



	publish_date <date>	headline_category <chr>
52285	2021-06-29	news.world.europe
52286	2021-06-29	business.construction
52287	2021-06-29	news.politics
52288	2021-06-29	news.ireland
52289	2021-06-29	opinion.letters

5 rows | 1-3 of 4 columns

- **Explanation**

In this question we will check the latest, earliest and last record from the dataset. So from the code we can see that first I have loaded libraries which are necessary for running to get questions that are dplyr and ggplot2. After that I inserted the CSV file using the reading data command that is “ireland_news.csv” into a dataframe named irish_df. After the file is read, we need to filter the irish independent articles that includes only articles from the “Irish Independent” news providers and also i have assigned a new dataframe called “irish_independent_df”. Since I need to get the published date as per output given below in the output section. I have specified the format as ("%A, %dth of %B, %Y") and converted the publish_date column in irish_independent_df in this format.

I made sure to remove the NA values from the dataframe so that I can get the earliest date. Latest date and last record properly as per instruction given. Then simply sorted the published date.

After sorting the publish date, then we will be identifying the earliest and latest article and then print the result.

Overall, this code tells how to filter, process and present information about the articles from the “Irish Independent” news provider which includes the earliest and latest articles as well as last 5 records by the publish date which I have sorted earlier.

Question 2A

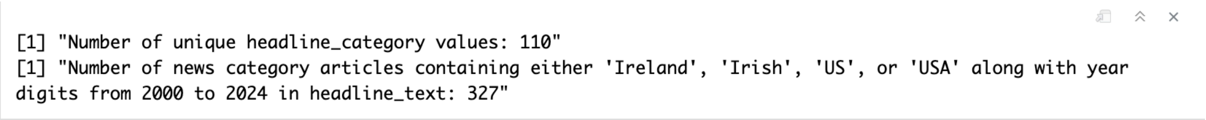
```
# Convert the headline_category values to lowercase
irish_df$headline_category <- tolower(irish_df$headline_category)

# Count the number of unique headline_category values
num_unique_categories <- irish_df %>%
  distinct(headline_category) %>%
  nrow()
#Print the Number of unique headline category values
print(paste("Number of unique headline_category values:", num_unique_categories))
```

#Question 2B

```
# Define keywords and year range
keywords <- c("Ireland", "Irish", "US", "USA")
years <- 2000:2024
```

```
# Function to check if a headline contains any of the keywords and a year
contains_keyword_and_year <- function(headline, keywords, years) {
  any_keyword <- any(grepl(paste(keywords, collapse = "|"), headline, ignore.case = TRUE))
  any_year <- any(grepl(paste(years, collapse = "|"), headline))
  return(any_keyword && any_year)
}

# Filter news category articles containing either of the keywords and a year in the headline_text
matching_articles <- irish_df %>%
  filter(headline_category == "news") %>%
  filter(sapply(headline_text, contains_keyword_and_year, keywords, years)) %>%
  na.omit()

# Count the number of matching articles
num_matching_articles <- nrow(matching_articles)
print(paste("Number of news category articles containing either 'Ireland', 'Irish', 'US', or 'USA' along with year digits from 2000 to 2024 in headline_text:", num_matching_articles))
```

- **Output**



```
[1] "Number of unique headline_category values: 110"
[1] "Number of news category articles containing either 'Ireland', 'Irish', 'US', or 'USA' along with year
digits from 2000 to 2024 in headline_text: 327"
```

- **Explanation**

Question 2A

From this code we can see that we are trying to convert the values in the “headline_category” column in the data frame that is “irish_df” to lower case using the ‘tolower()’ command. This command is usually for capitalization of the headline categories. Then after that calculate the number of unique values in the headline_category column using the ‘distinct()’ function to remove the duplicate values and ‘nrow()’ function to count the number of rows. After that print function is used for getting the unique headline_category value which is 110 as the output given below.

Question 2B

First we will be defining the keywords and year range which includes Ireland, Irish, USA and US and then we will put a separate command which is from the range of year (2000 to 2024) which are already defined. Then we will put a contains_keyword_and_year function to define to check if the headlines in the dataset contains any kind of keyword and a year that we have previously defined in the function before. Then we will be using ‘grepl()’ for searching the exact match of keywords and years for the headline text.

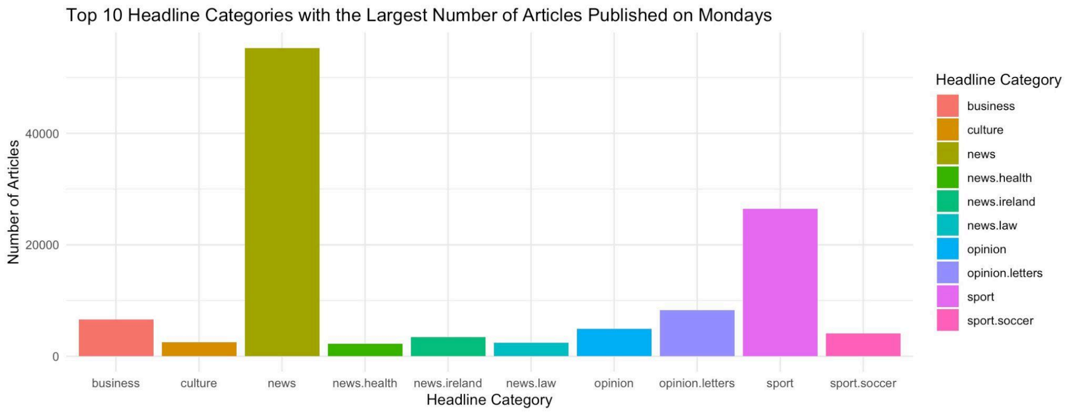
To filter matching articles we will be using the data frame that we defined in the previous step then only include articles which have the ‘news’ category that contains keywords that I have defined in the code with year in the headline text. Then we will be using ‘filter()’ to be used twice in this code to filter the news category which is to filter the news category articles and then filter articles based on the keyword and year function. Then count the number of articles which are matching articles.

Overall, we can see that this code analyses the headline categories and contains specific keywords and years.

Question 3

```
# Convert the "publish_date" column to date format
irish_df$publish_date <- as.Date(irish_df$publish_date, format = "%A, %dth of %B, %Y")

# Filter articles published on Mondays
monday_articles <- irish_df %>%
  filter(weekdays(publish_date) == "Monday")

# Count the number of articles for each headline category
article_count <- monday_articles %>%
  group_by(headline_category) %>%
  summarise(num_articles = n()) %>%
  arrange(desc(num_articles))

# Select the top 10 headline categories with the largest number of articles
top_10_categories <- head(article_count, 10)

# Plot using bar graph - headline category vs number of articles
ggplot(top_10_categories, aes(x = headline_category, y = num_articles, fill = headline_category)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 10 Headline Categories with the Largest Number of Articles Published on
Mondays",
       x = "Headline Category",
       y = "Number of Articles") +
  scale_fill_discrete(name = "Headline Category") +
  theme_minimal()
```

- **Output**



- **Explanation**

As before we had converted the publish_date column format in our first question. Then we will filter articles published on Monday using the function combined with ‘weekday(publish_date) ==”Monday”. Then after that calculate the number of articles for each headline category which are published on Mondays. Then after that use the groupby function to group data by the headline_category column and then summarise the count of the articles. Then we will be selecting the top 10 headlines categories using the head function.

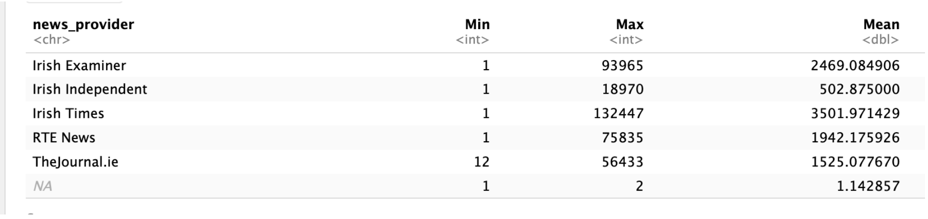
After that we will be using the ggplot to create a bar graph . From the below screenshot we have got the output for the top 10 headlines categories with a large number of articles that are published on monday, it can be seen that news is the highest category which is published on monday.

Overall, this code talks more about filtering, analysing and visualising the distribution of articles across the headline categories that are mostly published on mondays.

Question 4

```
# Compute the total number of articles for each headline category and news provider
total_article <- irish_df %>%
  group_by(headline_category, news_provider) %>%
  summarise(total_articles = n()) %>%
  ungroup()
# Compute and display the statistical information (Min, Max, and Mean) of the total number of articles for each news provider
statistical_info <- total_article %>%
  group_by(news_provider) %>%
  summarise(Min = min(total_articles),
           Max = max(total_articles),
           Mean = mean(total_articles))
# Print the statistical information
print(statistical_info)
```

- **Output**



news_provider	Min	Max	Mean
<chr>	<int>	<int>	<dbl>
Irish Examiner	1	93965	2469.084906
Irish Independent	1	18970	502.875000
Irish Times	1	132447	3501.971429
RTE News	1	75835	1942.175926
TheJournal.ie	12	56433	1525.077670
NA	1	2	1.142857

- **Explanation**

As we can see in the code that first we need to compute the total number of articles for each combination of headline category and news provider which is there in the dataset. This can be used by the grouping the data by headline category and new provider using group_by() function then summarising the grouped data by counting the number of articles and then storing those values through adding new columns such total articles using summaries but then after doing that we will be using the ungroup function to remove the grouping so that operation are affected when you are grouping.

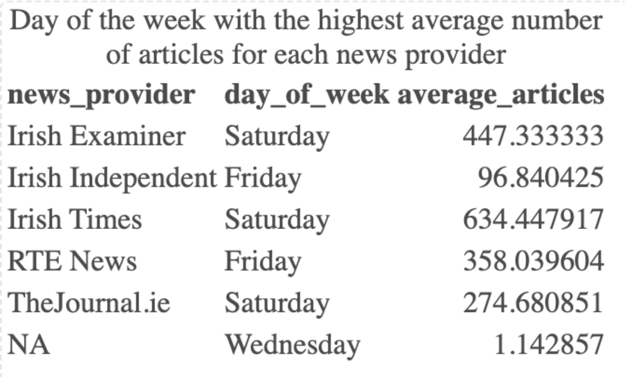
After that we will be computing the statistical information which is the maximum, minimum and mean function for finding the total number of articles for each new provider. Then after printing the statistical information for each news provider using the print function.

Overall from this code we can get good insights and information about articles across different headline categories and news providers. From that analysis we can say that Irish times is highest when it comes to distribution of articles across different headline categories and news providers.

Question 5

```
library(knitr)
# Compute the total number of articles for each headline category, news provider, and day of the week
total_articles <- irish_df %>%
  group_by(headline_category, news_provider, day_of_week) %>%
  summarise(total_articles = n())
# Compute the average number of articles for each news provider and day of the week
average_article <- total_articles %>%
  group_by(news_provider, day_of_week) %>%
  summarise(average_articles = mean(total_articles))
# Find the day of the week with the highest average number of articles for each provider
max_avg_day <- average_article %>%
  group_by(news_provider) %>%
  filter(average_articles == max(average_articles)) %>%
  arrange(news_provider)
# Display the results in tabular format
kable(max_avg_day, caption = "Day of the week with the highest average number of articles for each news provider")
```

- **Output**



Day of the week with the highest average number of articles for each news provider

news_provider	day_of_week	average_articles
Irish Examiner	Saturday	447.333333
Irish Independent	Friday	96.840425
Irish Times	Saturday	634.447917
RTE News	Friday	358.039604
TheJournal.ie	Saturday	274.680851
NA	Wednesday	1.142857

- **Explanation**

This code will talk about the total number of articles for each combination of headline category, news provider and day of the week. For this i have used the group by function and added headline category, news provider and day of weeks and then used for counting the number of articles using summarise() command. After that we will be using the average number of articles for each category of the news provider and day of the week and then calculate the mean values of the total article.

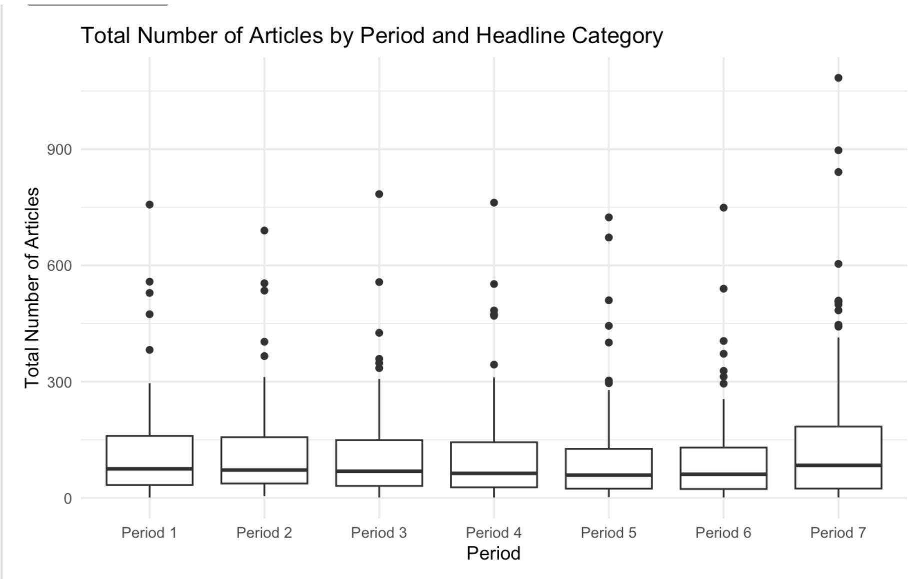
Then after finding the day with the highest average articles using each news provider. The question says to display the results in tabular format. From the table also we can see that the Irish Examiner has an average article that is around on saturday whereas the other news providers are also showing the average weekday distribution too.

Overall, this code tells about the insights about the distribution of articles across different weekdays for each news provider like it talks about the highest average number of articles for each provider.

Question 6

```
# Load the required libraries
library(lubridate)
library(dplyr)
library(ggplot2)

# Convert the "publish_date" column to date format
irish_df$publish_date <- as.Date(irish_df$publish_date, format = "%A, %dth of %B, %Y")

# Select the data for the years 2019 and 2020
df_2019_2020 <- irish_df %>%
  filter(year(publish_date) %in% c(2019, 2020))

# Add a new column named "Period" based on the publish_date values
df_2019_2020 <- df_2019_2020 %>%
  mutate(Period = cut(publish_date,
                 breaks = as.Date(c("2019-01-01", "2019-04-01", "2019-07-01", "2019-10-01",
"2020-01-01", "2020-04-01", "2020-07-01", "2021-01-01")),
                 labels = c("Period 1", "Period 2", "Period 3", "Period 4", "Period 5", "Period 6", "Period
7"),
                 include.lowest = TRUE))

# Compute the total number of articles by period and headline category for the top 10 headline
categories
top_10_categories <- df_2019_2020 %>%
  group_by(Period, headline_category) %>%
  summarise(total_articles = n()) %>%
  group_by(headline_category) %>%
  top_n(10, total_articles)

# Plotting Boxplot - Period and Total number of articles
ggplot(top_10_categories, aes(x = Period, y = total_articles)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Total Number of Articles by Period and Headline Category",
      x = "Period",
      y = "Total Number of Articles") +
  theme_minimal()
```

- **Output**



- **Explanation**

We will be loading required libraries such as lubridate, dplyr and ggplot2. Then we will be converting the date and select the data for 2019 and 2020 using the specified format which I did previous steps then we will be adding a new column that is period to the dataframe based on the publish date values then will be using mutate() function to create the period column where the values are categorised according to the breaks and labels.

After that, compute the total number of articles by period and headline category using the group by function and then select the top 10 headlines categories based on the total number of articles within each category. After all these functions are dome we will be generating a box plot using ggplot with the Period on the X axis and the total number of articles on the Y axis. Then added some additional formatting options such rotating X axis labels and adding a title and axis labels. From the box plot, we can see that period 7 has the highest number of articles, that is 1/7/20 - 30/9/20 had highest number of articles compared to other periods in the box plot.

Question 7

# Load the required libraries

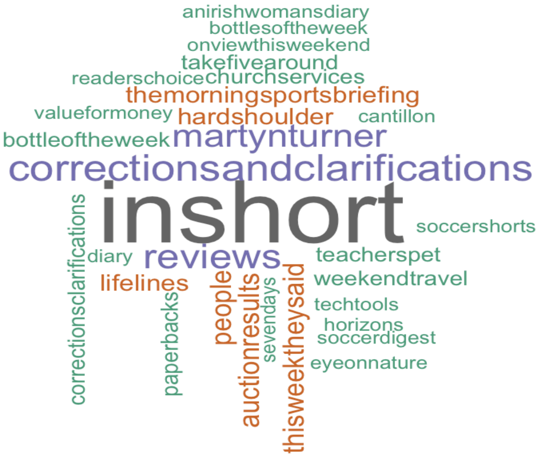
```
#library(dplyr)
#install.packages("tm")
#library(tm)
#install.packages("wordcloud")
#library(wordcloud)
#library(ggplot2)
```

# Read the CSV file into a DataFrame

```
irish_df <- read.csv("ireland_news.csv")
```

```
# Sample 1% of the data
set.seed(123)  # For reproducibility
sampled_irish_df <- irish_df %>%
  sample_frac(0.01)

# Perform text preprocessing on the sampled data
preprocess_text <- function(text) {
  # Convert text to lowercase
  text <- tolower(text)
  # Remove numbers and punctuation
  text <- gsub("[^a-zA-Z\\s]", "", text)
  # Remove stopwords
  text <- removeWords(text, stopwords("en"))
  # Remove extra white spaces
  text <- gsub("\\s+", " ", text)
  return(text)
}
sampled_irish_df$clean_text <- sapply(sampled_irish_df$headline_text, preprocess_text)

# Create a document-term matrix
corpus <- Corpus(VectorSource(sampled_irish_df$clean_text))
dtm <- DocumentTermMatrix(corpus)

# Display a portion of the document-term matrix
inspect(dtm[1:5, 1:10])

# Get word frequencies
word_freq <- colSums(as.matrix(dtm))

# Get top 10 most frequent words
top_words <- head(sort(word_freq, decreasing = TRUE), 10)

# Plot the top 10 most frequent words
word_freq_df <- data.frame(word = names(top_words), freq = top_words)
p <- ggplot(word_freq_df, aes(x = freq, y = word)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 10 Most Frequent Words",
       x = "Frequency",
       y = "Word")

# Generate a word cloud
wordcloud(names(word_freq), word_freq, max.words = 50, random.order = FALSE, colors =
brewer.pal(8, "Dark2"))
```

- **Output**



- **Explanation**

We will be installing required libraries like dyplr,Text mining, wordcloud and ggplot2. But since we don't have wordcloud packages since we will be installing the package.We will be using sampling data which samples 1% of the data from the Dataframe that is irish_df for further analysis. We will be using the 'sample_frac' function from the 'dplyr' package. After that we will be using text processing that is why we installed the text mining package so we will preprocess the sampled data by converting the text to lowercase and then removing numbers and punctuation, removing unnecessary stop words and then extra white spaces. Then we will be creating a document term matrix where we will preprocess text data which we did in text preprocessing in the previous step. Then the matrix represents the frequency of terms,that is words in each document i.e. headline text.

After that we will do calculations of word frequencies for each word in the document term matrix across all documents and then get all top 10 most frequent words based on frequencies. Using these most frequent words we will be able to plot bar plots to show the frequency of the top 10 as well as generate word clouds.

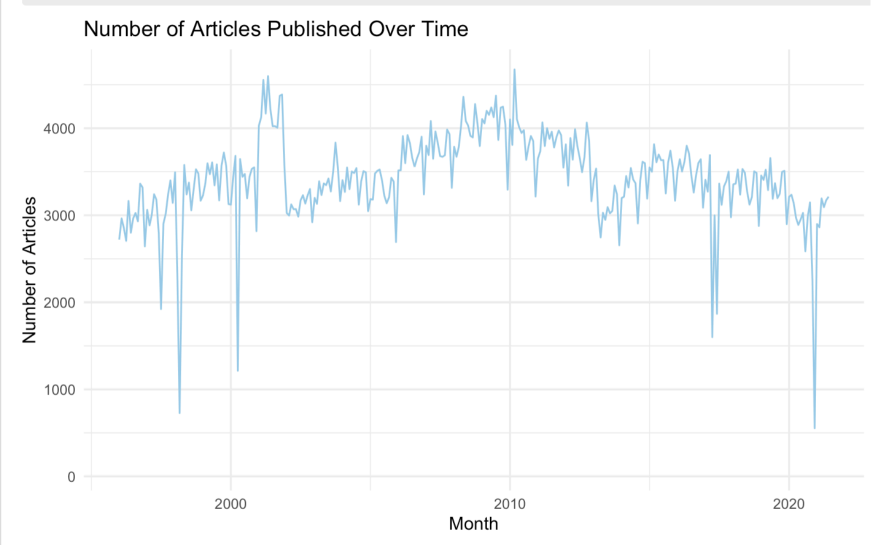
From the word cloud we can see the top 10 words that are used in the output given below.

Question 8

```
# Convert publish_date to Date format
irish_df$publish_date <- as.Date(irish_df$publish_date, format="%A, %dth of %B, %Y")

# Extract month and year from publish_date
irish_df$month_year <- format(irish_df$publish_date, "%Y-%m")

# Aggregate the number of articles published each month using group by function and
summarise function
article_counts <- irish_df %>%
  group_by(month_year) %>%
  summarize(Count = n())
```

```
# Convert month_year to Date format for plotting
article_counts$month_year <- as.Date(paste(article_counts$month_year, "-01", sep=""),
format="%Y-%m-%d")

# Plot line graph using Number of articles published over tim
ggplot(article_counts, aes(x = month_year, y = Count)) +
  geom_line(color = "skyblue") +
  labs(title = "Number of Articles Published Over Time",
       x = "Month",
       y = "Number of Articles") +
  theme_minimal()
```

- **Output**



- **Explanation**

In this code we will be converting publish date to date format and then extract month and year from the publish date which we did in the previous step.We will format the exact month and year in the "YYYY-MM" format. Then aggregate the number of articles published each month using the groupby function then group the data by the month_year column and then summarise the function for calculating the count of articles for each group.

Convert the month year to date format for the purpose for plotting using the date function which converts to the month_year column in the article_counts dataframe to date format. Then we will use the ggplot function to plot the time series of article counts. From the graph given below we can see that in the months of 2000 and 2010 the highest number of articles has been published over time and then by 2020 it has drastically changed.