AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Choose an item.

Project

Assignment Title:	Data Science Final Project		
Assignment No:	Click here to enter text.	Date of Submission:	25 May 2025
Course Title:	Introduction to Data Science		
Course Code:	Click here to enter text.	Section:	E
Semester:	Spring 2024-25	Course Teacher:	Abdus Salam

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of their arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.: 9

No	Name	ID	Program	Signature
1	SABIHA IQBAL ETU	22-47227-1	BSc [CSE]	
2	MUKSHIT SAFI OWASI	22-47251-1	BSc [CSE]	
3	MAHMUDUS SAMI MAAHI	22-46446-1	BSc [CSE]	
4	MD. JAHIRUL ISLAM NAHID	22-47212-1	BSc [CSE]	

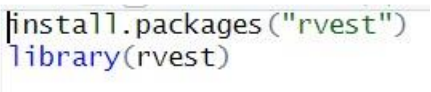
Faculty use only		
FACULTY COMMENTS	Marks Obtained	
	Total Marks	

Introduction:

In this project, we focused on extracting and processing news data from *The Daily Star*, one of Bangladesh's leading English-language news portals. We selected five diverse categories—**Business**, **Sports**, **Entertainment**, **Youth**, and **Life & Living**—and scraped **100 news articles from each**, resulting in a dataset of **500 articles**. For each article, we collected the **category**, **news link**, **title**, **description**, and **publication time**. The primary goal was to apply a comprehensive text preprocessing pipeline to the **description** field, preparing the data for further natural language processing tasks. The steps included **emoji handling** (replacing emojis with text), **text cleaning** (converting to lowercase, removing punctuation, numbers, and extra spaces), **tokenization**, **stopword removal**, **stemming and lemmatization**, and **basic spell correction**. The cleaned and processed text was stored in a new column named 'processed_description'. This workflow demonstrates how web scraping and text preprocessing can be combined to build a structured, clean dataset suitable for content analysis.

Extract news from the News portal:

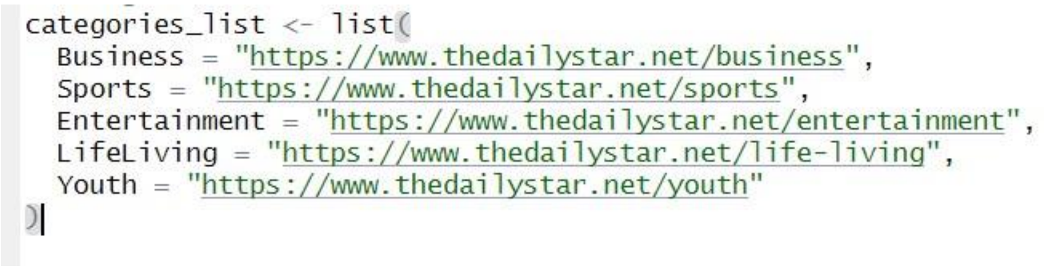
Required Library:



```
install.packages("rvest")
library(rvest)
```

The command is install.packages("rvest") installs the rvest package in R, which is used for web scraping, and library(rvest) loads the package so we can use its functions to extract data from web pages.

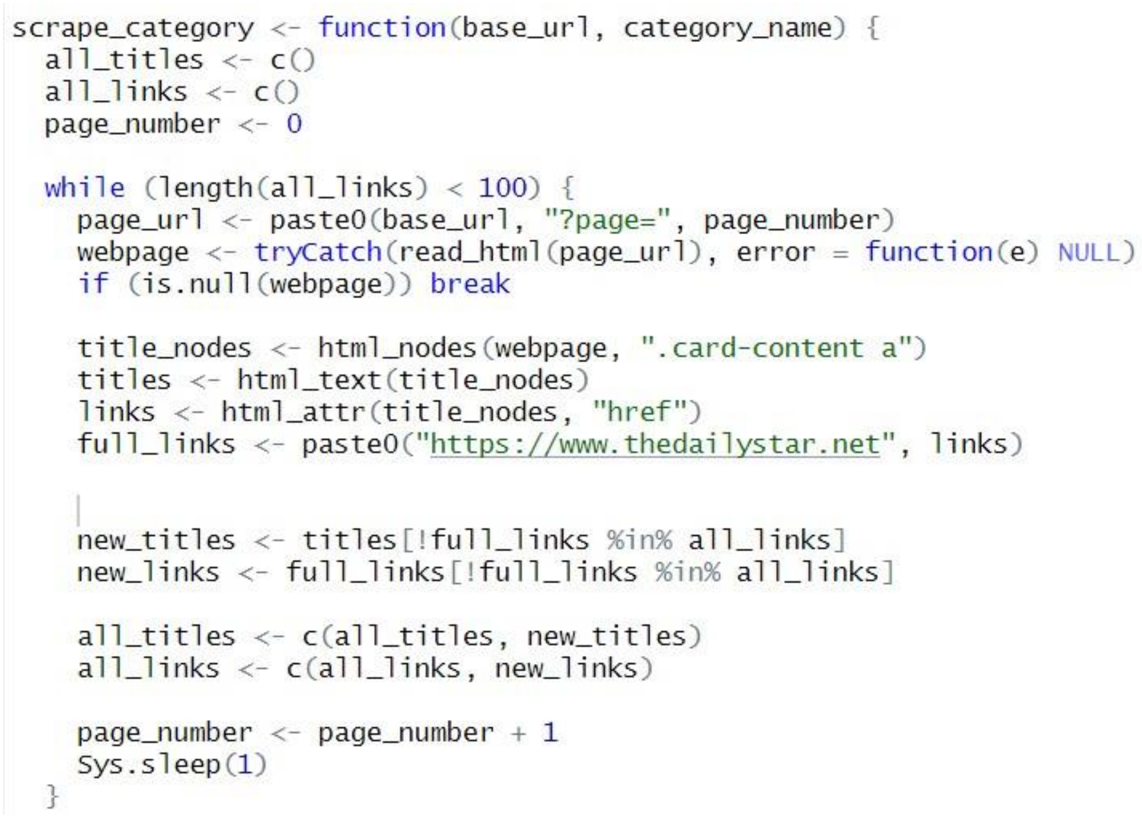
Select Categories and their URL:



```
categories_list <- list(
  Business = "https://www.thedailystar.net/business",
  Sports = "https://www.thedailystar.net/sports",
  Entertainment = "https://www.thedailystar.net/entertainment",
  LifeLiving = "https://www.thedailystar.net/life-living",
  Youth = "https://www.thedailystar.net/youth"
)
```

This R code defines a named list called categories_list that maps each news category to its corresponding URL on *The Daily Star* website. Each element of the list has a category name (e.g., *Business*, *Sports*, etc.) as the key, and the URL of that category's news page as the value**.** This list will be useful for looping through each category to scrape news articles from their respective pages.

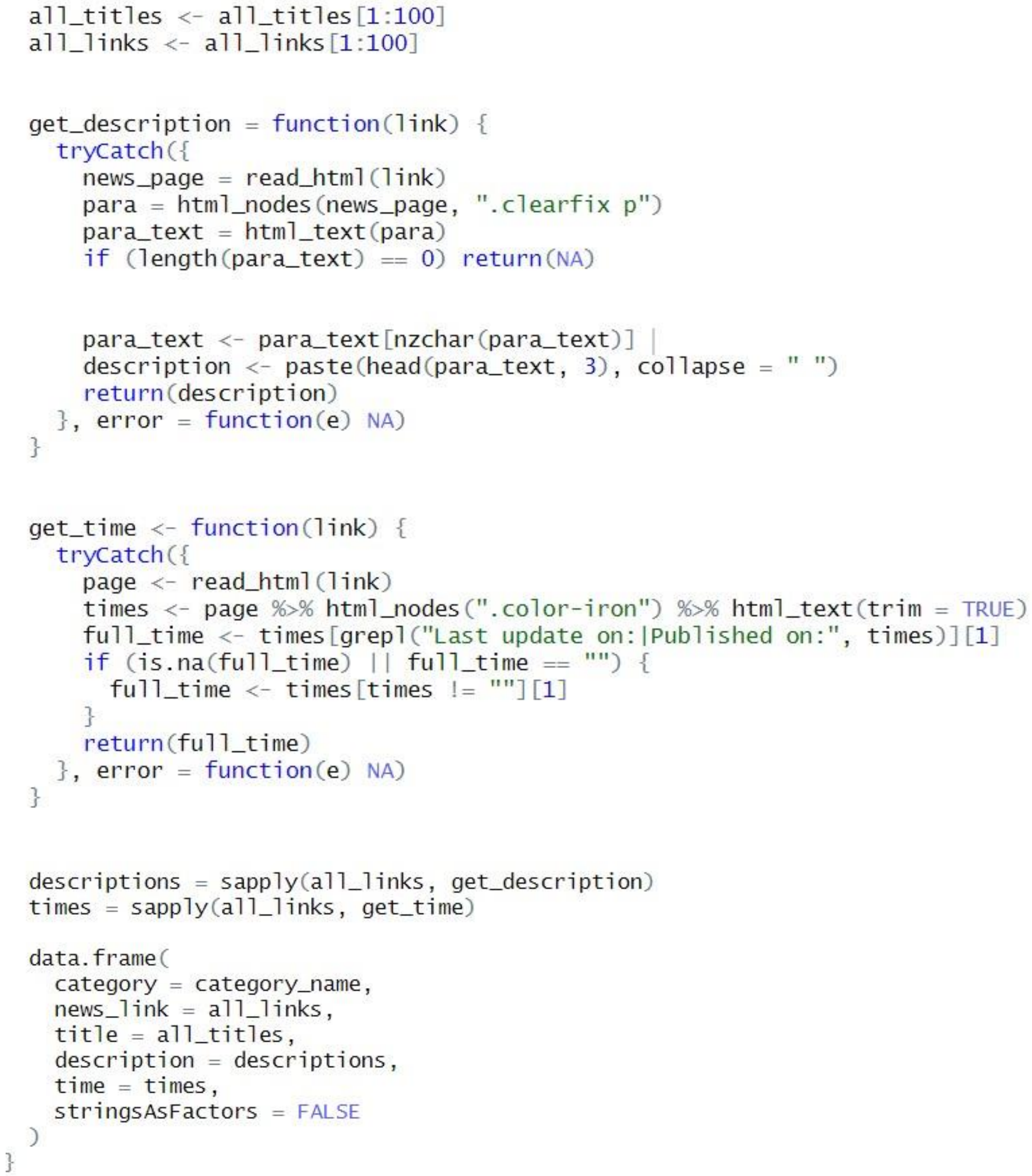
Function to Scrape News Articles from a Single Category



```
scrape_category <- function(base_url, category_name) {
  all_titles <- c()
  all_links <- c()
  page_number <- 0

  while (length(all_links) < 100) {
    page_url <- paste0(base_url, "?page=", page_number)
    webpage <- tryCatch(read_html(page_url), error = function(e) NULL)
    if (is.null(webpage)) break

    title_nodes <- html_nodes(webpage, ".card-content a")
    titles <- html_text(title_nodes)
    links <- html_attr(title_nodes, "href")
    full_links <- paste0("https://www.thedailystar.net", links)


    new_titles <- titles[!full_links %in% all_links]
    new_links <- full_links[!full_links %in% all_links]

    all_titles <- c(all_titles, new_titles)
    all_links <- c(all_links, new_links)

    page_number <- page_number + 1
    Sys.sleep(1)
  }
```

In this part of the project, we created a function named scrape_category to automate the process of scraping news article titles and their corresponding links from a specific category page on *The Daily Star* website. The function accepts two inputs: the category URL and the category name. In this web scraping function, we start by initializing empty vectors to store article titles and links. Using a while loop, we continuously request web pages by dynamically building the URL with an increasing page_number parameter, which allows us to navigate through multiple pages sequentially. For each retrieved webpage, we use html_nodes() to select specific HTML elements containing article links and titles based on their CSS selectors. Then, html_text() extracts the text content (titles), and html_attr() extracts the hyperlink URLs (href attributes). We combine the relative links with the base domain to form complete URLs. To avoid duplicates, we compare the new URLs (full_links) with the previously collected ones (all_links) using the %in% operator, selecting only those that are not already present. This filtered set of unique titles and links is appended to the cumulative collection. The loop repeats until we gather at least 100 unique articles or no more pages are available, ensuring a complete, non-redundant dataset.The output of this function is a clean and structured list of news titles and links for each selected category, which lays the foundation for collecting full news content and metadata in the next steps.

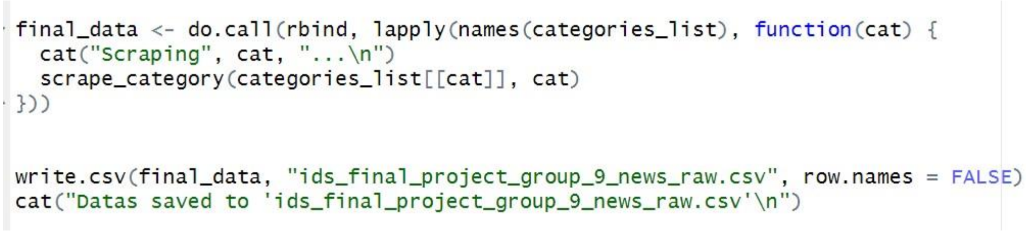
Extracting Descriptions and Publication Times for News Articles



```r
  all_titles <- all_titles[1:100]
  all_links <- all_links[1:100]


  get_description = function(link) {
    tryCatch({
      news_page = read_html(link)
      para = html_nodes(news_page, ".clearfix p")
      para_text = html_text(para)
      if (length(para_text) == 0) return(NA)


      para_text <- para_text[nzchar(para_text)]
      description <- paste(head(para_text, 3), collapse = " ")
      return(description)
    }, error = function(e) NA)
  }


  get_time <- function(link) {
    tryCatch({
      page <- read_html(link)
      times <- page %>% html_nodes(".color-iron") %>% html_text(trim = TRUE)
      full_time <- times[grepl("Last update on:|Published on:", times)][1]
      if (is.na(full_time) || full_time == "") {
        full_time <- times[times != ""][1]
      }
      return(full_time)
    }, error = function(e) NA)
  }


  descriptions = sapply(all_links, get_description)
  times = sapply(all_links, get_time)

  data.frame(
    category = category_name,
    news_link = all_links,
    title = all_titles,
    description = descriptions,
    time = times,
    stringsAsFactors = FALSE
  )
}
```

After collecting the first 100 unique article titles and links, we define two helper functions to extract additional content from each article’s webpage. The get_description function takes a news article URL and attempts to scrape the full text description of the article. It does this by reading the HTML content using read_html() and then searching for paragraph (<p>) elements within different possible CSS selectors (.clearfix p, .field--name-body p, and article p) to handle variations in webpage structure. The extracted paragraphs are combined into a single string representing the article's full description. If no paragraphs are found or an error occurs, the function returns NA. Similarly, the get_time function retrieves the publication or last update timestamp from the article page. It locates text elements with the CSS class .color-iron and filters for phrases containing "Last update on:" or "Published on:". If such a phrase is not found, it defaults to the first non-empty timestamp available. Both functions use tryCatch to gracefully handle any errors during scraping by returning NA. Finally, these functions are applied to all collected article links using sapply to generate vectors of descriptions and timestamps, which are combined along with titles, links, and category labels into a data frame. This structured dataset consolidates essential metadata for further analysis while maintaining robustness against inconsistent webpage formats.

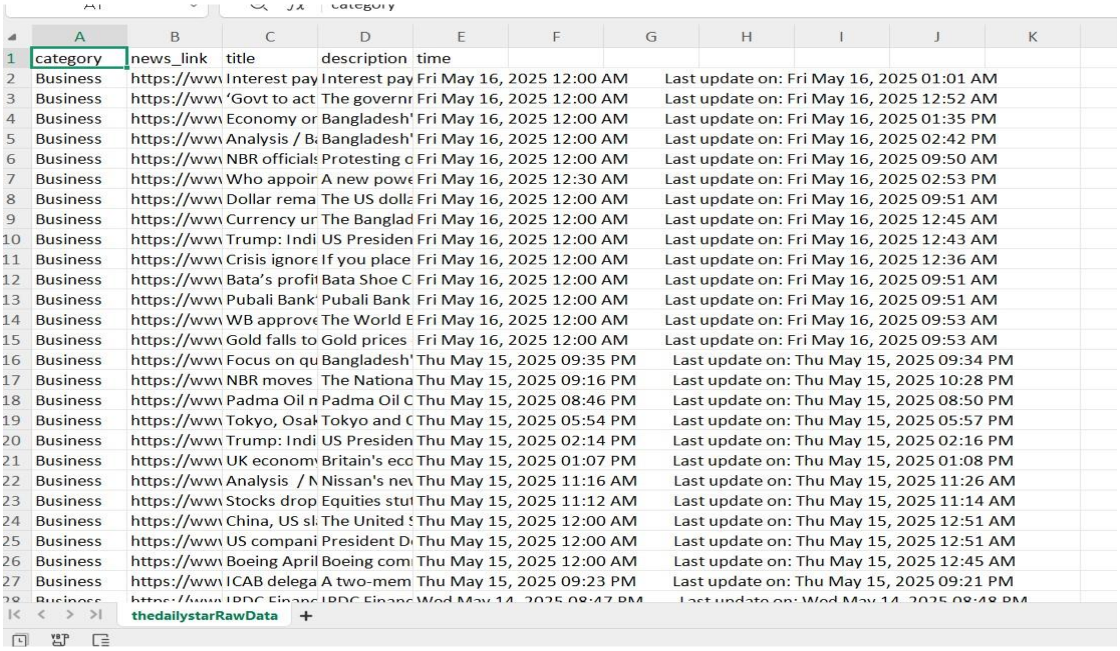
Scraping All Categories and Saving News Data



```
final_data <- do.call(rbind, lapply(names(categories_list), function(cat) {
  cat("Scraping", cat, "...\n")
  scrape_category(categories_list[[cat]], cat)
}))


write.csv(final_data, "ids_final_project_group_9_news_raw.csv", row.names = FALSE)
cat("Datas saved to 'ids_final_project_group_9_news_raw.csv'\n")
```

In this section, we automated the scraping process for all selected categories by using lapply() to loop through each category in the categories_list. For each category, the scrape_category function is called, and the results are combined into a single data frame using do.call(rbind, ...). This aggregated dataset, containing 500 news articles across five categories, is then saved to a CSV file named ids_final_project_group_9_news_raw.csv using write.csv(). A confirmation message is printed after successful completion.

The output of extracting news from the news portal



	A	B	C	D	E
1	category	news_link	title	description	time
2	Business	https://www	Interest pay	Interest pay	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 01:01 AM
3	Business	https://www	'Govt to act	The governr	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 12:52 AM
4	Business	https://www	Economy or	Bangladesh'	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 01:35 PM
5	Business	https://www	Analysis / Ba	Bangladesh'	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 02:42 PM
6	Business	https://www	NBR officials	Protesting c	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:50 AM
7	Business	https://www	Who appoir	A new powe	Fri May 16, 2025 12:30 AM Last update on: Fri May 16, 2025 02:53 PM
8	Business	https://www	Dollar rema	The US dolla	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:51 AM
9	Business	https://www	Currency ur	The Banglad	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 12:45 AM
10	Business	https://www	Trump: Indi	US Presiden	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 12:43 AM
11	Business	https://www	Crisis ignore	If you place	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 12:36 AM
12	Business	https://www	Bata's profit	Bata Shoe C	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:51 AM
13	Business	https://www	Pubali Bank'	Pubali Bank	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:51 AM
14	Business	https://www	WB approve	The World E	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:53 AM
15	Business	https://www	Gold falls to	Gold prices	Fri May 16, 2025 12:00 AM Last update on: Fri May 16, 2025 09:53 AM
16	Business	https://www	Focus on qu	Bangladesh'	Thu May 15, 2025 09:35 PM Last update on: Thu May 15, 2025 09:34 PM
17	Business	https://www	NBR moves	The Nationa	Thu May 15, 2025 09:16 PM Last update on: Thu May 15, 2025 10:28 PM
18	Business	https://www	Padma Oil n	Padma Oil C	Thu May 15, 2025 08:46 PM Last update on: Thu May 15, 2025 08:50 PM
19	Business	https://www	Tokyo, Osak	Tokyo and C	Thu May 15, 2025 05:54 PM Last update on: Thu May 15, 2025 05:57 PM
20	Business	https://www	Trump: Indi	US Presiden	Thu May 15, 2025 02:14 PM Last update on: Thu May 15, 2025 02:16 PM
21	Business	https://www	UK econom	Britain's eco	Thu May 15, 2025 01:07 PM Last update on: Thu May 15, 2025 01:08 PM
22	Business	https://www	Analysis / N	Nissan's nev	Thu May 15, 2025 11:16 AM Last update on: Thu May 15, 2025 11:26 AM
23	Business	https://www	Stocks drop	Equities stut	Thu May 15, 2025 11:12 AM Last update on: Thu May 15, 2025 11:14 AM
24	Business	https://www	China, US sl	The United S	Thu May 15, 2025 12:00 AM Last update on: Thu May 15, 2025 12:51 AM
25	Business	https://www	US compani	President D	Thu May 15, 2025 12:00 AM Last update on: Thu May 15, 2025 12:51 AM
26	Business	https://www	Boeing April	Boeing com	Thu May 15, 2025 12:00 AM Last update on: Thu May 15, 2025 12:45 AM
27	Business	https://www	ICAB delega	A two-mem	Thu May 15, 2025 09:23 PM Last update on: Thu May 15, 2025 09:21 PM
28	Business	https://www	IPDC Financ	IPDC Financ	Wed May 14, 2025 08:47 PM Last update on: Wed May 14, 2025 08:48 PM

This is the output we get when extracting the information from the online news portal The Daily Star. We extract 100 news articles and save them (category, news link, title, time) in csv file.

Applying the text processing steps:

Why Preprocessing Is Important -

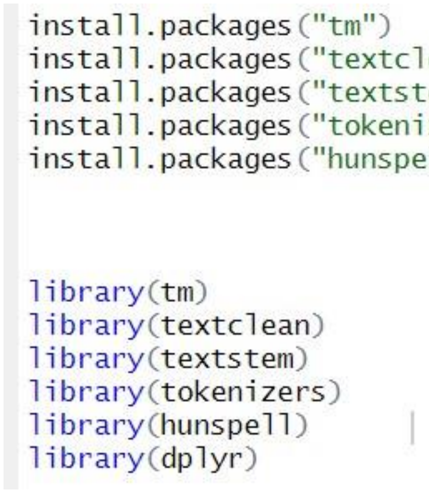
Raw text data is often messy and inconsistent. It contains contractions ("don't"), emojis, punctuation, HTML tags, numbers, spelling mistakes, and irrelevant words (stopwords) that can negatively affect the accuracy and performance of models or analyses. Preprocessing helps standardize the text, reduce noise, and highlight meaningful content.

The core objective of this work is to perform detailed text preprocessing on the news description field to prepare it for downstream tasks like content classification, sentiment analysis, or summarization. The following text processing steps were applied:

1. **Emoji Handling** – Replaced emojis with their textual descriptions.
2. **Text Cleaning** – Converted text to lowercase, removed punctuation, numbers, and extra whitespace.
3. **Tokenization** – Split text into individual tokens or words.
4. **Stopword Removal** – Eliminated common English stopwords that do not contribute much to meaning.
5. **Stemming and Lemmatization** – Reduced words to their root or base forms to normalize the vocabulary.
6. **Spell Checking** – Applied basic spell correction to improve textual quality, enforcing lowercase consistency.

The processed content was stored in a new column named `processed_description` and the complete dataset was exported to a CSV file for further analysis or modeling. This pipeline showcases the integration of web scraping with natural language preprocessing, establishing a foundation for robust content mining and NLP-based news analytics.

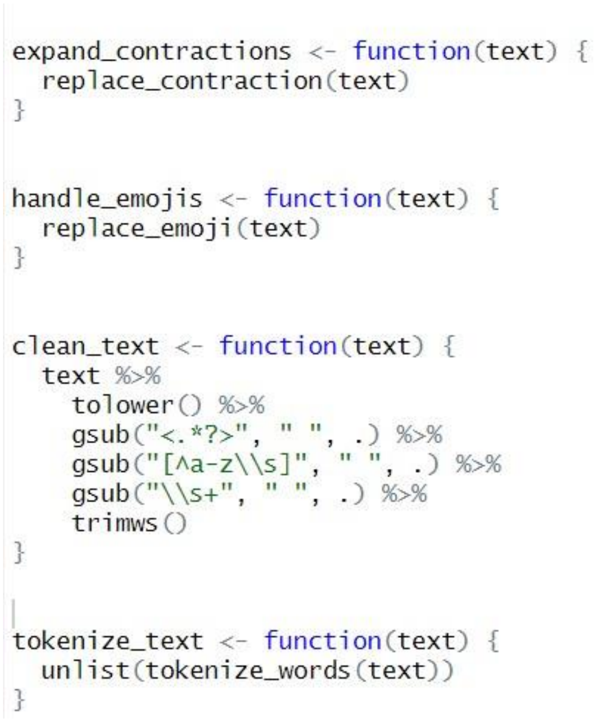
Required Libraries



```r
install.packages("tm")
install.packages("textclean")
install.packages("textstem")
install.packages("tokenizers")
install.packages("hunspell")



library(tm)
library(textclean)
library(textstem)
library(tokenizers)
library(hunspell)
library(dplyr)
```

These packages are used for text processing in R: **tm** handles text cleaning and management; textclean fixes messy or incorrect text; textstem reduces words to their base forms; tokenizers break text into words or sentences; and hunspell checks and corrects spelling errors. Together, they prepare text for analysis or modeling.

Text Preprocessing



```r
expand_contractions <- function(text) {
  replace_contraction(text)
}


handle_emojis <- function(text) {
  replace_emoji(text)
}


clean_text <- function(text) {
  text %>%
    tolower() %>%
    gsub("<.*?>", " ", .) %>%
    gsub("[^a-z\\s]", " ", .) %>%
    gsub("\\s+", " ", .) %>%
    trimws()
}


tokenize_text <- function(text) {
  unlist(tokenize_words(text))
}
```

We defined four specific text preprocessing functions:

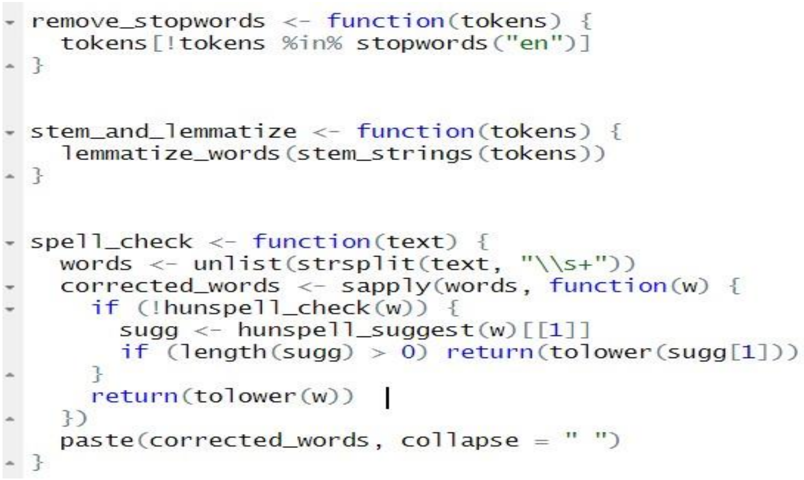
expand_contractions: Converts shortened forms like “don’t” to “do not” using `replace_contraction()`.

handle_emojis: Replaces emojis in the text with their descriptive words using `replace_emoji()`.

clean_text: Converts text to lowercase, removes HTML tags, non-letter characters, extra spaces, and trims whitespace.

tokenize_text: Splits the cleaned text into individual word tokens using `tokenize_words()`.

Text Preprocessing



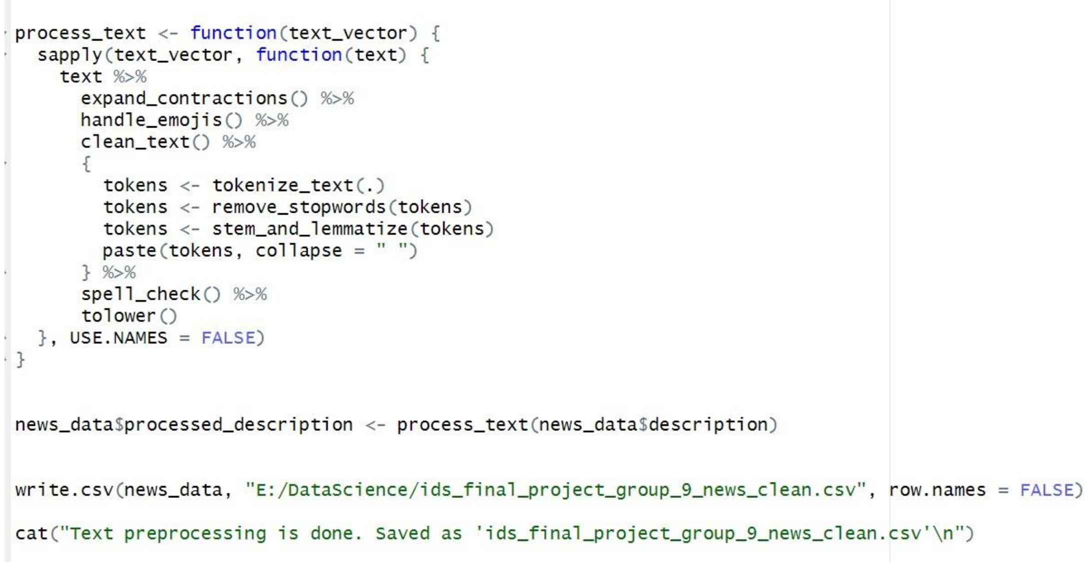
```r
remove_stopwords <- function(tokens) {
  tokens[!tokens %in% stopwords("en")]
}


stem_and_lemmatize <- function(tokens) {
  lemmatize_words(stem_strings(tokens))
}


spell_check <- function(text) {
  words <- unlist(strsplit(text, "\\s+"))
  corrected_words <- sapply(words, function(w) {
    if (!hunspell_check(w)) {
      sugg <- hunspell_suggest(w)[[1]]
      if (length(sugg) > 0) return(tolower(sugg[1]))
    }
    return(tolower(w))
  })
  paste(corrected_words, collapse = " ")
}
```

Here's what each function does specifically:

remove_stopwords: Filters out English stopwords from the token vector using `stopwords("en")`.

stem_and_lemmatize: First stems tokens with `stem_strings()`, then lemmatizes them using `lemmatize_words()` to get their base forms.

spell_check: Splits the text into words, checks each word's spelling with `hunspell_check()`, replaces misspelled words with the first suggested correction from `hunspell_suggest()`, and returns the corrected lowercase text.



```r
process_text <- function(text_vector) {
  sapply(text_vector, function(text) {
    text %>%
      expand_contractions() %>%
      handle_emojis() %>%
      clean_text() %>%
      {
        tokens <- tokenize_text(.)
        tokens <- remove_stopwords(tokens)
        tokens <- stem_and_lemmatize(tokens)
        paste(tokens, collapse = " ")
      } %>%
      spell_check() %>%
      tolower()
  }, USE.NAMES = FALSE)
}


news_data$processed_description <- process_text(news_data$description)


write.csv(news_data, "E:/DataScience/ids_final_project_group_9_news_clean.csv", row.names = FALSE)

cat("Text preprocessing is done. Saved as 'ids_final_project_group_9_news_clean.csv'\n")
```

This code defines a main function `process_text` that applies a full text preprocessing pipeline to each element of a text vector. For each text entry, it:

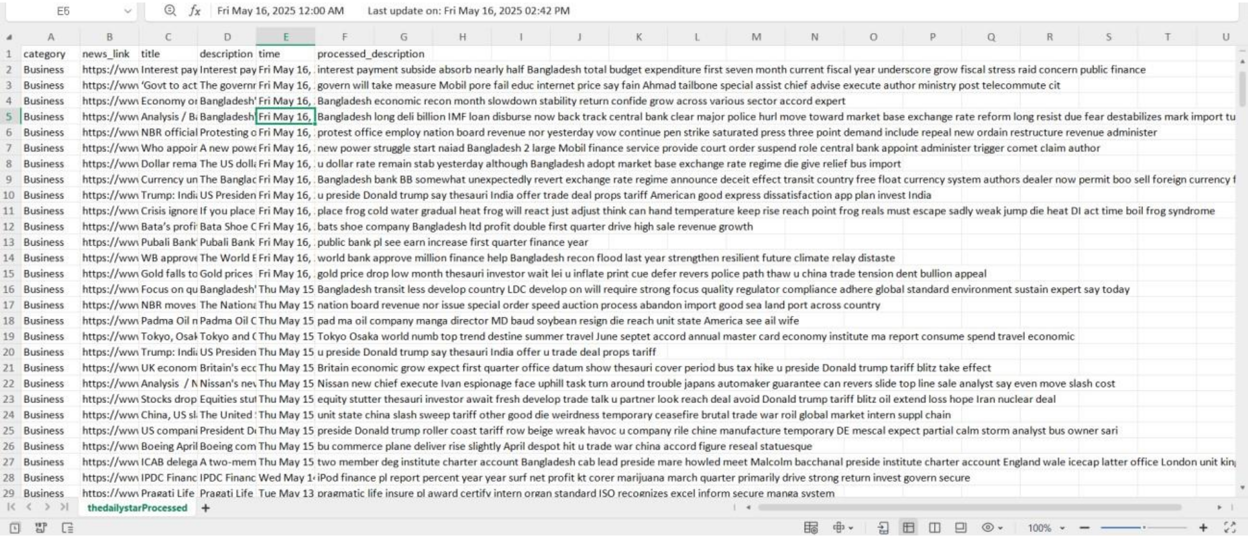
1. Expands contractions (e.g., "don't" → "do not").
2. Replace emojis with descriptive words.
3. Cleans the text by lowercasing, removing HTML tags, punctuation, numbers, and extra spaces.
4. Tokenize the cleaned text into words.
5. Removes stopwords from the tokens.
6. Applies stemming and lemmatization to reduce words to their base forms.

7. Join the processed tokens back into a single string.
8. Performs spell checking and correction.
9. Converts the final output to lowercase for consistency.

Then, it applies this function specifically to the `description` column of the `news_data` dataframe, storing the cleaned text in a new column `processed_description`.

Finally, it saves the updated dataframe with the processed text to a new CSV file at the specified location and prints a confirmation message indicating that preprocessing is complete.

Final Output after doing Preprocessing on the description column



E5 | Fri May 16, 2025 12:00 AM | Last update on: Fri May 16, 2025 02:42 PM

	A	B	C	D	E	F
1	category	news_link	title	description	time	processed_description
2	Business	https://www	Interest pay	Interest pay	Fri May 16,	interest payment subside absorb nearly half Bangladesh total budget expenditure first seven month current fiscal year underscore grow fiscal stress raid concern public finance
3	Business	https://www	'Govt to act	The governr	Fri May 16,	govern will take measure Mobil pore fail educ internet price say fain Ahmad tailbone special assist chief advise execute author ministry post telecommute cit
4	Business	https://www	Economy or	Bangladesh'	Fri May 16,	Bangladesh economic recon month slowdown stability return confide grow across various sector accord expert
5	Business	https://www	Analysis / Ba	Bangladesh'	Fri May 16,	Bangladesh long deli billion IMF loan disburse now back track central bank clear major police hurl move toward market base exchange rate reform long resist due fear destabilizes mark import tu
6	Business	https://www	NBR official	Protesting o	Fri May 16,	protest office employ nation board revenue nor yesterday vow continue pen strike saturated press three point demand include repeal new ordain restructure revenue administer
7	Business	https://www	Who appoir	A new powe	Fri May 16,	new power struggle start naiad Bangladesh 2 large Mobil finance service provide court order suspend role central bank appoint administer trigger comet claim author
8	Business	https://www	Dollar rema	The US doll	Fri May 16,	u dollar rate remain stab yesterday although Bangladesh adopt market base exchange rate regime die give relief bus import
9	Business	https://www	Currency un	The Banglac	Fri May 16,	Bangladesh bank BB somewhat unexpectedly revert exchange rate regime announce deceit effect transit country free float currency system authors dealer now permit boo sell foreign currency f
10	Business	https://www	Trump: India	US Presiden	Fri May 16,	u preside Donald trump say thesauri India offer trade deal props tariff American good express dissatisfaction app plan invest India
11	Business	https://www	Crisis ignore	If you place	Fri May 16,	place frog cold water gradual heat frog will react just adjust think can hand temperature keep rise reach point frog reals must escape sadly weak jump die heat DI act time boil frog syndrome
12	Business	https://www	Bata's profi	Bata Shoe C	Fri May 16,	bats shoe company Bangladesh ltd profit double first quarter drive high sale revenue growth
13	Business	https://www	Pubali Bank'	Pubali Bank	Fri May 16,	public bank pl see earn increase first quarter finance year
14	Business	https://www	WB approve	The World E	Fri May 16,	world bank approve million finance help Bangladesh recon flood last year strengthen resilient future climate relay distaste
15	Business	https://www	Gold falls to	Gold prices	Fri May 16,	gold price drop low month thesauri investor wait lei u inflate print cue defer revers police path thaw u china trade tension dent bullion appeal
16	Business	https://www	Focus on qu	Bangladesh'	Thu May 15	Bangladesh transit less develop country LDC develop on will require strong focus quality regulator compliance adhere global standard environment sustain expert say today
17	Business	https://www	NBR moves	The Nationa	Thu May 15	nation board revenue nor issue special order speed auction process abandon import good sea land port across country
18	Business	https://www	Padma Oil n	Padma Oil C	Thu May 15	pad ma oil company manga director MD baud soybean resign die reach unit state America see ail wife
19	Business	https://www	Tokyo, Osak	Tokyo and (	Thu May 15	Tokyo Osaka world numb top trend destine summer travel June septet accord annual master card economy institute ma report consume spend travel economic
20	Business	https://www	Trump: India	US Presiden	Thu May 15	u preside Donald trump say thesauri India offer u trade deal props tariff
21	Business	https://www	UK econom	Britain's ecc	Thu May 15	Britain economic grow expect first quarter office datum show thesauri cover period bus tax hike u preside Donald trump tariff blitz take effect
22	Business	https://www	Analysis / N	Nissan's nev	Thu May 15	Nissan new chief execute Ivan espionage face uphill task turn around trouble japans automaker guarantee can revers slide top line sale analyst say even move slash cost
23	Business	https://www	Stocks drop	Equities stul	Thu May 15	equity stutter thesauri investor await fresh develop trade talk u partner look reach deal avoid Donald trump tariff blitz oil extend loss hope Iran nuclear deal
24	Business	https://www	China, US sl	The United :	Thu May 15	unit state china slash sweep tariff other good die weirdness temporary ceasefire brutal trade war roil global market intern suppl chain
25	Business	https://www	US compani	President D	Thu May 15	preside Donald trump roller coast tariff row beige wreak havoc u company rile chine manufacture temporary DE mescal expect partial calm storm analyst bus owner sari
26	Business	https://www	Boeing April	Boeing com	Thu May 15	bu commerce plane deliver rise slightly April despot hit u trade war china accord figure reseal statuesque
27	Business	https://www	ICAB delega	A two-mem	Thu May 15	two member deg institute charter account Bangladesh cab lead preside mare howled meet Malcolm bacchanal preside institute charter account England wale icecap latter office London unit kin
28	Business	https://www	IPDC Financ	IPDC Financ	Wed May 1	iPod finance pl report percent year year surf net profit kt corer marijuana march quarter primarily drive strong return invest govern secure
29	Business	https://www	Pragati Life	Pragati Life	Tue May 13	pragmatic life insure pl award certify intern organ standard ISO recognizes excel inform secure manga system

Finally we get the clean text after doing the preprocessing and all other steps the description part is processed and saved in the csv file named ids_final_project_group_9_news_clean.csv

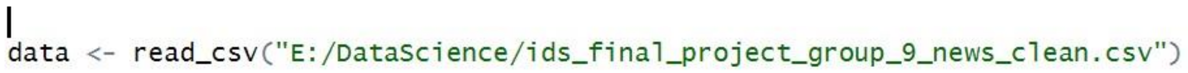
Applying Top Modelling :

Required Libraries

```
# Load libraries
library(tm)
library(topicmodels)
library(ggplot2)
library(tidytext)
library(dplyr)
library(readr)
library(tidyverse)
```

The tm package provides essential tools for text preprocessing such as cleaning, tokenization, and stopword removal. topicmodels is used to implement Latent Dirichlet Allocation (LDA), a popular algorithm for uncovering latent topics within a collection of documents. For data manipulation and transformation, the widely used dplyr and tidyverse packages offer a suite of functions that enable efficient handling and wrangling of data frames. The tidytext package integrates tidy data principles with text mining, allowing easy extraction and analysis of text features. To import external data files, readr is utilized for fast and reliable reading of CSV files. Finally, ggplot2 supports the creation of elegant and informative visualizations, such as plotting the top terms within each topic, helping to interpret and communicate the results clearly.

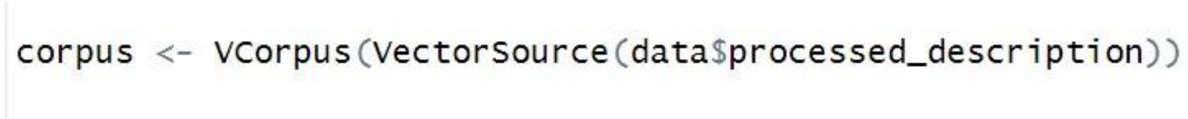
Loading the Preprocessed News Data



```
data <- read_csv("E:/DataScience/ids_final_project_group_9_news_clean.csv")
```

In this step, we load the cleaned dataset into R using the read_csv() function from the readr package. The file named ids_final_project_group_9_news_clean.csv contains the news articles that were collected and preprocessed in the earlier part of the project. This dataset includes multiple columns, but the one that is processed_description, which holds the cleaned version of the news content. This step is essential because it brings our prepared data into the current R environment, making it ready for further analysis. Without loading this file, we would not be able to perform topic modeling on the news descriptions. This step forms the foundation for all the tasks that follow in the topic modeling part of the project.

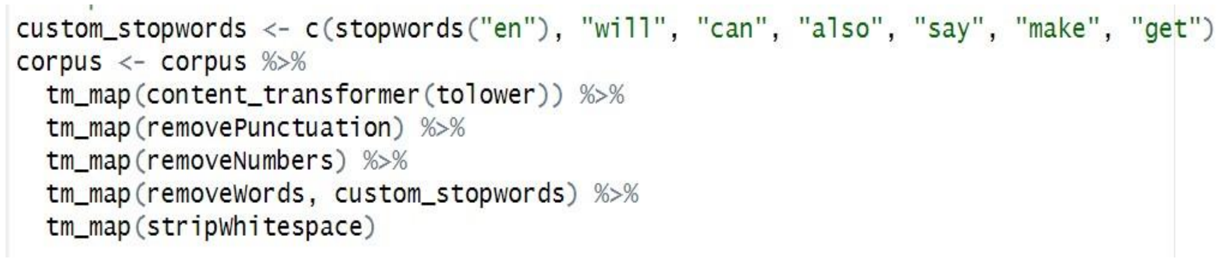
Creating a Text Corpus



```
corpus <- VCorpus(VectorSource(data$processed_description))
```

In this part, we create a text corpus using the VCorpus() function from the tm package. A corpus is a structured collection of text documents, and in this case, it is built from the processed_description column of our dataset, which contains the cleaned news article descriptions. We use VectorSource() to tell R that the source of the text is a vector (a column) from our data frame. Creating a corpus is an important step in this project because it prepares the text for further processing and analysis. Once we create the corpus, we can apply various text cleaning functions and later convert it into a document-term matrix, which is needed for topic modeling.

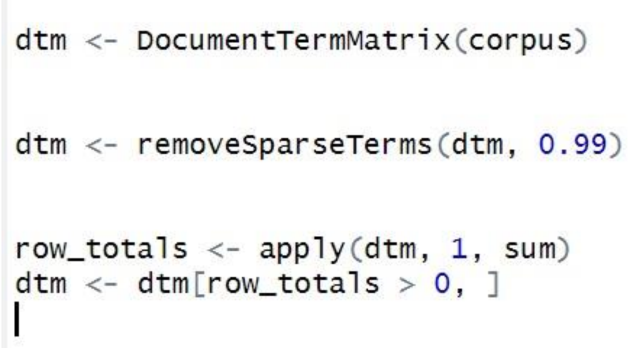
Cleaning the Text Again



```
custom_stopwords <- c(stopwords("en"), "will", "can", "also", "say", "make", "get")
corpus <- corpus %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(removeWords, custom_stopwords) %>%
  tm_map(stripWhitespace)
```

Here, to prepare the text data for topic modeling, we perform a series of preprocessing steps on the corpus. First, we define a set of custom stopwords by extending the standard English stopword list with commonly used but uninformative words like "will", "can", "also", "say", "make", and "get". These are removed from the text to reduce noise. Using the tm_map() function from the tm package, we then apply a sequence of transformations to each document in the corpus. The text is converted to lowercase with tolower() to maintain consistency. All punctuation and numeric characters are removed using removePunctuation and removeNumbers, respectively. The removeWords function eliminates the custom stopwords, and finally, stripWhitespace cleans up any extra spaces left behind. These steps help standardize the text and focus on the most meaningful terms, improving the quality of the subsequent topic modeling.

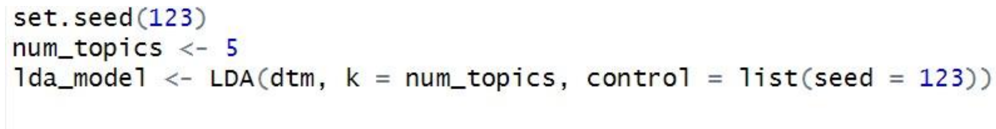
Document-Term Matrix (DTM)



```
dtm <- DocumentTermMatrix(corpus)


dtm <- removeSparseTerms(dtm, 0.99)


row_totals <- apply(dtm, 1, sum)
dtm <- dtm[row_totals > 0, ]
```

In this part, we convert the cleaned corpus into a Document-Term Matrix (DTM) using the DocumentTermMatrix() function. A DTM is a structured table where each row represents a document, and each column represents a unique word. The values in the matrix show how often each word appears in each document. This format is essential for topic modeling, as it allows algorithms like LDA to analyze word usage patterns across documents. After creating the DTM, we remove sparse terms using removeSparseTerms(dtm, 0.99). This means we keep only the words that appear in at least 1% of the documents, which helps remove rare or irrelevant terms. Then we check the row totals using apply(dtm, 1, sum) and remove any rows where the total is zero, which indicates documents that have no remaining words after filtering. This step ensures our DTM is clean, compact, and ready for topic modeling.

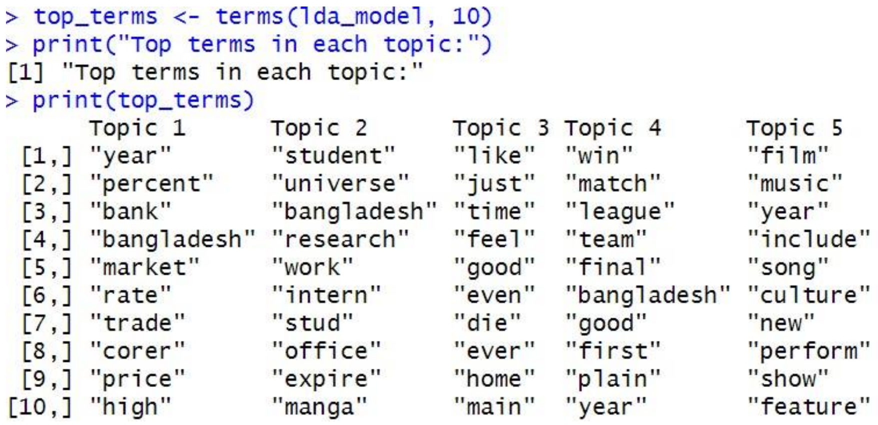
The LDA Topic Model



```
set.seed(123)
num_topics <- 5
lda_model <- LDA(dtm, k = num_topics, control = list(seed = 123))
```

To uncover the hidden thematic structure within the cleaned text data, we apply (LDA), a probabilistic topic modeling technique. We first set a random seed using set.seed(123) to ensure that the results are reproducible. The number of topics to be discovered is defined with num_topics <- 5, meaning we expect the dataset to contain five distinct topics. The LDA() function from the topicmodels package is then used to fit the model to our document-term matrix (dtm). The control parameter specifies the seed value again for internal randomness during model fitting. This model outputs the probability distribution of words within each topic and topics within each document, which helps us interpret and label the underlying themes in the news dataset.

Extracting Top Terms and Plotting



```
> top_terms <- terms(lda_model, 10)
> print("Top terms in each topic:")
[1] "Top terms in each topic:"
> print(top_terms)
      Topic 1      Topic 2      Topic 3 Topic 4      Topic 5  
 [1,] "year"       "student"    "like"  "win"        "film"   
 [2,] "percent"    "universe"   "just"  "match"      "music"  
 [3,] "bank"       "bangladesh" "time"  "league"     "year"   
 [4,] "bangladesh" "research"   "feel"  "team"       "include"
 [5,] "market"     "work"       "good"  "final"      "song"   
 [6,] "rate"       "intern"     "even"  "bangladesh" "culture"
 [7,] "trade"      "stud"       "die"   "good"       "new"    
 [8,] "corer"      "office"     "ever"  "first"      "perform"
 [9,] "price"      "expire"     "home"  "plain"      "show"   
[10,] "high"       "manga"      "main"  "year"       "feature"
```

Topic 1: Economy/Finance

Top Words: year, percent, bank, bangladesh, market, rate, trade, corner, price, high

This topic revolves around economic or financial matters. Terms like “bank,” “rate,” “price,” and “market” indicate monetary or trade discussions. "Percent" and "high" suggest data-oriented news such as inflation, interest rates, or financial performance.

Topic 2: Education/Academia

Top Words: student, universe, bangladesh, research, work, intern, stud, office, expire, manga

This topic focuses on educational or academic content. Words like “student,” “research,” “intern,” and “universe” (possibly misrecognized for “university”) point to institutional learning and academic activity. “Manga” could suggest student-related interests or extracurricular domains.

Topic 3: Lifestyle

Top Words: like, just, time, feel, good, even, die, ever, ever, home, main

This topic seems to involve personal narratives, opinions, or social commentary. Words like “feel,” “just,” “good,” “die,” and “home” suggest expressive, emotive content, perhaps related to human interest stories, blogs, or social issues.

Topic 4: Sports

Top Words: win, match, league, team, final, bangladesh, good, first, plain, year

This topic clearly centers on sports. Words like “match,” “team,” “win,” “final,” and “league” point to competitive sports coverage, possibly focused on cricket or football given the mention of “Bangladesh.”

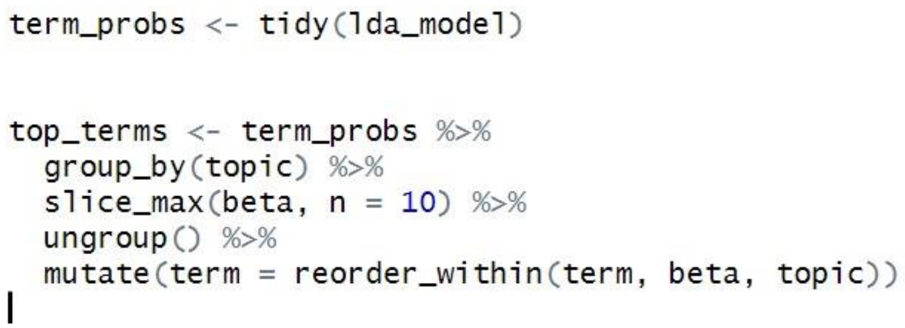
Topic 5: Entertainment

Top Words: film, music, year, include, song, culture, new, perform, show, feature

This topic focuses on the entertainment and cultural sector. Terms like “film,” “music,” “song,” “show,” and “perform” indicate content related to movies, music, and performing arts. “Culture” and “feature” reinforce the artistic and creative context.

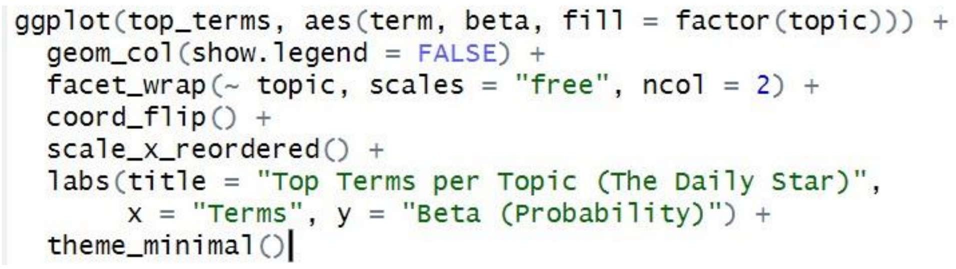
This section extracts and displays the most representative words for each topic generated by the LDA model. The function terms(lda_model, 10) is used to retrieve the top 10 terms (words) that have the highest probability of belonging to each topic in the model. These terms are considered the most relevant keywords that characterize the content of each topic. By examining these top terms, we can gain insight into the themes or subjects that the model has discovered within the text data. After extracting these terms, the code prints a message "Top terms in each topic:" to inform the actual lists of the top words for each of the five topics. This output is crucial for interpreting and labeling the topics in a meaningful way, as it provides a summary of the key vocabulary associated with each topic.

(BETA)



```
term_probs <- tidy(lda_model)


top_terms <- term_probs %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  ungroup() %>%
  mutate(term = reorder_within(term, beta, topic))
```

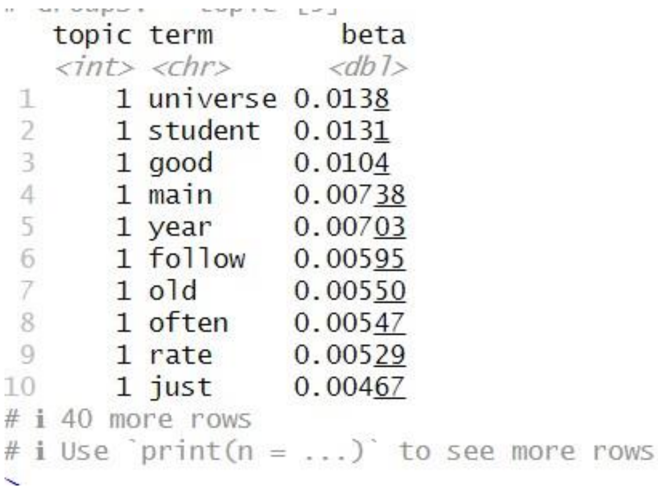
In this section, we extract the most relevant terms from the trained LDA topic model using the tidy() function from the tidytext package, which converts the model output into a tidy data frame. Each row in this tidy format contains a term, its corresponding topic, and the beta value (the probability of that term being associated with that topic). We then group this data by topic using group_by(topic) and use slice_max(beta, n = 10) to select the top 10 terms with the highest probability (beta) for each topic, which helps us identify the most representative words per topic. The ungroup() function removes the grouping structure, and finally, mutate(term = reorder_within(term, beta, topic)) reorders terms within each topic so they can be properly plotted using ggplot2. This process prepares the data for visual representation and aids in interpreting and labeling the topics based on their most prominent terms.



```
ggplot(top_terms, aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol = 2) +
  coord_flip() +
  scale_x_reordered() +
  labs(title = "Top Terms per Topic (The Daily Star)",
       x = "Terms", y = "Beta (Probability)") +
  theme_minimal()
```

This creates a bar plot using **ggplot2** to visualize the top terms associated with each topic in the LDA model. The ggplot() function takes the top_terms data (prepared earlier) and maps the term to the x-axis, beta values (term probabilities) to the y-axis, and fills the bars based on the topic number. The geom_col() function draws the bars, while show.legend = FALSE hides the legend for cleaner visualization. facet_wrap(~ topic, scales = "free", ncol = 2) creates small multiples (facets) for each topic, allowing a side-by-side comparison of top terms across topics, with free scales to better fit varying term probabilities. coord_flip() flips the axes for horizontal bars, making long term labels easier to read. scale_x_reordered() ensures terms are correctly ordered within each facet. The labs() function adds a descriptive title and axis labels, and theme_minimal() applies a clean and minimalistic style. This visualization helps in quickly identifying which terms are most representative of each topic and assists in interpreting topic meanings more intuitively.

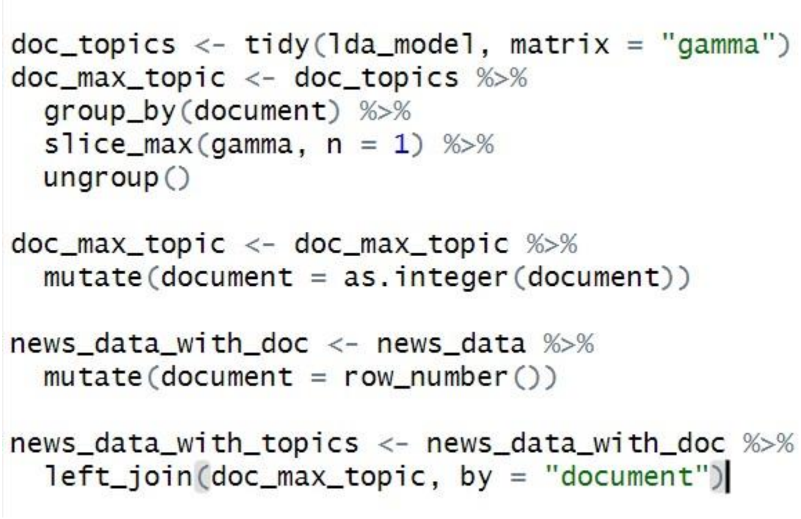
Output:



```
# Groups:   topic [5]
   topic term         beta
   <int> <chr>       <dbl>
 1     1 universe 0.0138
 2     1 student  0.0131
 3     1 good     0.0104
 4     1 main     0.00738
 5     1 year     0.00703
 6     1 follow   0.00595
 7     1 old      0.00550
 8     1 often    0.00547
 9     1 rate     0.00529
10     1 just     0.00467
# ℹ 40 more rows
# ℹ Use `print(n = ...)` to see more rows
```

The term "universe" has the highest beta value (0.0138), indicating that it's the most representative word in Topic 1. This is followed by "student", "good", and "main", with slightly lower probabilities. Words like "year", "old", and "rate" also appear, suggesting that this topic may relate to education, youth, or academic life, possibly even something like student life in universities or educational quality/rating. The beta values gradually decrease down the list, showing diminishing relevance of each word to the topic. These top terms give insight into the thematic focus of Topic 1 and are essential for labeling or interpreting the topic meaningfully.

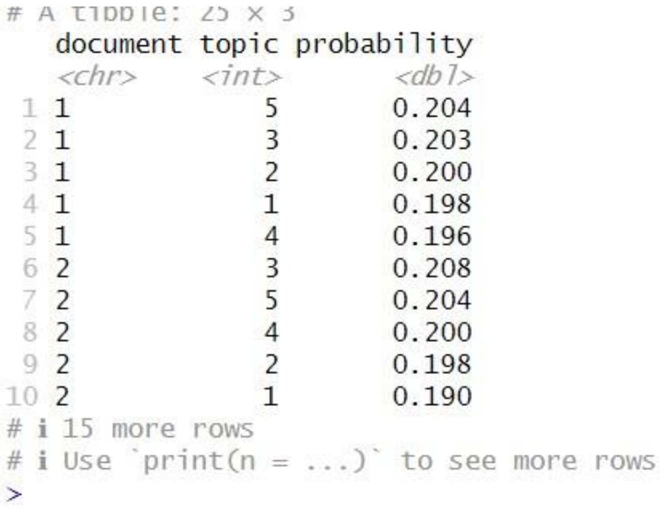
Mapping LDA-Derived Topics to Individual Documents



```
doc_topics <- tidy(lda_model, matrix = "gamma")
doc_max_topic <- doc_topics %>%
  group_by(document) %>%
  slice_max(gamma, n = 1) %>%
  ungroup()

doc_max_topic <- doc_max_topic %>%
  mutate(document = as.integer(document))

news_data_with_doc <- news_data %>%
  mutate(document = row_number())

news_data_with_topics <- news_data_with_doc %>%
  left_join(doc_max_topic, by = "document")
```

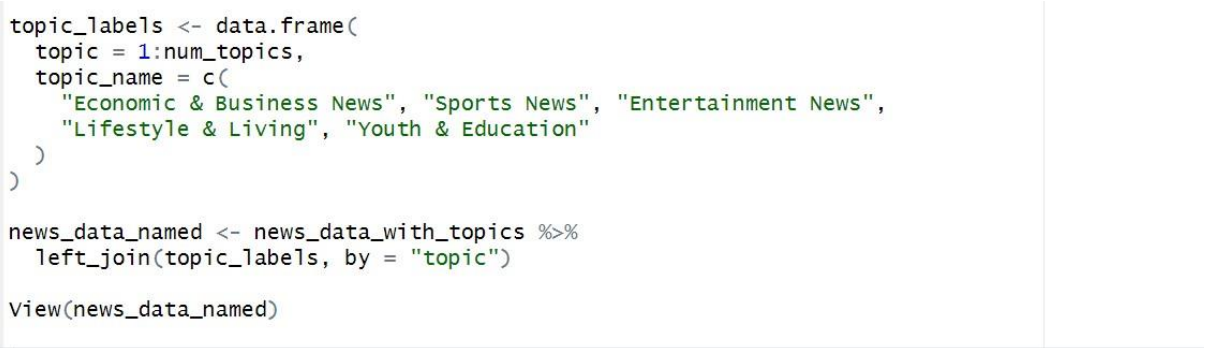
After fitting the LDA topic model, we extracted the document-topic probabilities (gamma matrix) using the tidy() function, which transformed the model output into a tidy dataframe with each document's probability distribution over topics. We then grouped this dataframe by document and selected the topic with the highest gamma value for each document to identify its dominant topic. Since the document IDs were character type, we converted them to integers to ensure compatibility for joining. We also added a numeric document identifier to the original dataset using mutate() with row_number(), which allowed us to merge the dominant topic information back into the original news data via a left_join() on the document ID. This integration enriched each news article with its most representative topic, facilitating meaningful analysis and interpretation of the topic structure within the corpus.

(GAMMA)



```
# A tibble: 25 × 3
   document topic probability
   <chr>    <int>       <dbl>
 1 1            5       0.204
 2 1            3       0.203
 3 1            2       0.200
 4 1            1       0.198
 5 1            4       0.196
 6 2            3       0.208
 7 2            5       0.204
 8 2            4       0.200
 9 2            2       0.198
10 2            1       0.190
# ℹ 15 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Document 1 has the highest probability with Topic 5 (20.4%), closely followed by Topic 3 (20.3%), and so on. This means Document 1 likely discusses themes from multiple topics almost equally, with a slight emphasis on Topic 5.• Document 2 is most associated with Topic 3 (20.8%), followed by Topic 5 (20.4%), suggesting that its content is primarily related to those topics. The probabilities across each document add up to approximately 1 (or 100%), confirming that the model distributes topic proportions across all topics for each document.

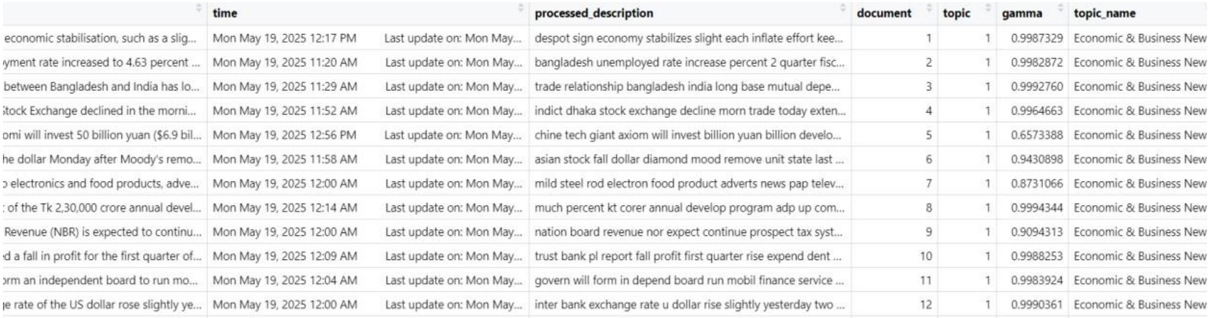


```
topic_labels <- data.frame(
  topic = 1:num_topics,
  topic_name = c(
    "Economic & Business News", "Sports News", "Entertainment News",
    "Lifestyle & Living", "Youth & Education"
  )
)

news_data_named <- news_data_with_topics %>%
  left_join(topic_labels, by = "topic")

View(news_data_named)
```

To provide meaningful interpretations to the topics generated by the LDA model, we first created a lookup table (topic_labels) as a data frame that maps each numeric topic ID to a descriptive topic name based on our prior interpretation of the top terms per topic. This table contained two columns: topic, which holds the topic numbers, and topic_name, which assigns a human-readable label to each topic. We then merged this mapping with the dataset containing the dominant topic assignments for each document (news_data_with_topics) using a left_join()

on the topic column. This join operation appended the corresponding topic names to each news article based on its assigned dominant topic, effectively enriching the dataset with clear, interpretable topic labels. Finally, the enriched dataset (news_data_named) was viewed for inspection, facilitating easier understanding and presentation of topic distributions within the corpus.

Output:



	time		processed_description	document	topic	gamma	topic_name
economic stabilisation, such as a slig...	Mon May 19, 2025 12:17 PM	Last update on: Mon May...	despot sign economy stabilizes slight each inflate effort kee...	1	1	0.9987329	Economic & Business New
yment rate increased to 4.63 percent ...	Mon May 19, 2025 11:20 AM	Last update on: Mon May...	bangladesh unemployed rate increase percent 2 quarter fisc...	2	1	0.9982872	Economic & Business New
between Bangladesh and India has lo...	Mon May 19, 2025 11:29 AM	Last update on: Mon May...	trade relationship bangladesh india long base mutual depe...	3	1	0.9992760	Economic & Business New
Stock Exchange declined in the morni...	Mon May 19, 2025 11:52 AM	Last update on: Mon May...	indict dhaka stock exchange decline morn trade today exten...	4	1	0.9964663	Economic & Business New
omi will invest 50 billion yuan ($6.9 bil...	Mon May 19, 2025 12:56 PM	Last update on: Mon May...	chine tech giant axiom will invest billion yuan billion develo...	5	1	0.6573388	Economic & Business New
he dollar Monday after Moody's remo...	Mon May 19, 2025 11:58 AM	Last update on: Mon May...	asian stock fall dollar diamond mood remove unit state last ...	6	1	0.9430898	Economic & Business New
electronics and food products, adve...	Mon May 19, 2025 12:00 AM	Last update on: Mon May...	mild steel rod electron food product adverts news pap telev...	7	1	0.8731066	Economic & Business New
of the Tk 2,30,000 crore annual devel...	Mon May 19, 2025 12:14 AM	Last update on: Mon May...	much percent kt corer annual develop program adp up com...	8	1	0.9994344	Economic & Business New
Revenue (NBR) is expected to continu...	Mon May 19, 2025 12:00 AM	Last update on: Mon May...	nation board revenue nor expect continue prospect tax syst...	9	1	0.9094313	Economic & Business New
d a fall in profit for the first quarter of...	Mon May 19, 2025 12:09 AM	Last update on: Mon May...	trust bank pl report fall profit first quarter rise expend dent ...	10	1	0.9988253	Economic & Business New
orm an independent board to run mo...	Mon May 19, 2025 12:04 AM	Last update on: Mon May...	govern will form in depend board run mobil finance service ...	11	1	0.9983924	Economic & Business New
e rate of the US dollar rose slightly ye...	Mon May 19, 2025 12:00 AM	Last update on: Mon May...	inter bank exchange rate u dollar rise slightly yesterday two ...	12	1	0.9990361	Economic & Business New

The output table is a comprehensive data frame where each row represents a news article. It contains the original article attributes, including the processed_description column, which holds the cleaned and preprocessed text used for topic modeling. Additional columns include document, a unique numeric identifier assigned to each article for linking with model outputs; topic, indicating the numeric ID of the dominant topic assigned based on the highest gamma value; and gamma, representing the probability that the article belongs to that dominant topic. Finally, the topic_name column provides a descriptive label for the topic, enabling clear interpretation of the main themes. This enriched dataset integrates the raw data with topic modeling results, facilitating thematic analysis across the corpus.