German Weekly Magazines From 2020 to 2023

Case Study in the Summer 2023

Inhaltsverzeichnis

Introduction **1**

Part 1: Data Exploration **2**

Part 2: Exploratory Data Analysis **2**

Part 3: Topic Identification **3**

Appendix A: Editing Notes, Grading, and Submission **4**

A.1 Editing Notes 4
A.2 Grading 4
A.3 Submission 5

Appendix B: Data **6**

B.1 Source 6
B.2 Data Files 6
B.3 Data Dictionary 6
B.4 Loading the Data 7

Introduction

This case study comprises three parts. In the first part, you will utilize your R programming skills to explore the new data sets guided by questions that are not related to the article's content.

You proceed in part two with exploratory data analysis to address specific content-related questions, as well as to gather evidence to support or dispute provided hypotheses.

In the last part, you are presented with an open-ended question. Your task here is to devise and execute a solution plan to answer this question.

Part 1: Data Exploration

1. Assess the data quality and perform the following checks:

    a. Are there any duplicate articles?
    b. For what percentage of articles do we have any information about the author?
    c. What is the complete rate and value range for the columns `section`, `subsection` and `ressort`? Does it differ between magazines?

2. How many issues does the data set contain for each magazine?
3. What is the mean title length per magazine? Visualize the result as a bar chart!
4. Which Spiegel category has the longest articles on average? Consider only categories with at least 100 articles since 2020! Visualize the result appropriately!
5. How many pages does each issue of each magazine contain? What is the distribution of the number of pages per issue for each magazine?

Part 2: Exploratory Data Analysis

1. What are the three persons most often named for each magazine?

    a. Overall
    b. Per year

    Visualize the result appropriately.

2. What are the most often named organizations from Osnabrück across all magazines? Create a suitable visualization for the top 10!
3. Which countries are mentioned most frequently in articles related to the Covid-19 pandemic in each magazine?
4. Explore the frequency of references to German political parties in the three magazines. Create a visualization that provides a good overview, emphasizing potential differences between the magazines.
5. Conduct an analysis to support or contradict the following hypothesis: Articles in the *Ausland* category in Spiegel magazine mention Annalena Baerbock, the German Foreign Minister, more frequently than articles in other categories. Support your findings through suitable visualizations.

Part 3: Topic Identification

1. Vegetarians and Vegans

Employ a deductive approach to pinpoint articles across all three magazines that revolve around vegan or vegetarian lifestyles (pro or contra). Once you've identified this subset of articles, work on discovering subtopics within them. Determine the frequency of each subtopic in the context of each magazine. Create visualizations to communicate your findings.

2. Unearthing Scientific and Technological Breakthroughs

Imagine you have been in a deep sleep for the past three years and you've just awoken. Eager to catch up on the scientific and technological advancements that have occurred during your absence, you realize all you have is the data from this case study: a collection of articles from Stern, Spiegel, and FOCUS since 2020.

Use your skills in R and text analytics to uncover at least three significant scientific or technological breakthroughs that have been discussed widely across multiple articles. Perform a data-driven (inductive) topic identification based on the articles and provided additional meta data. Apply visualization techniques to highlight when the breakthroughs you found happened and were subsequently discussed in the magazines.

Appendix A: Editing Notes, Grading, and Submission

A.1 Editing Notes

- Create visualizations when they are useful to better convey the result. For some exercises, a visualization is explicitly required.
- If your solution comprises multiple steps, add comments to all steps to outline your solution with a verbal description.
- Add comments where appropriate for better readability of your R-code. Do not comment trivial statements.
- You may only use R and the packages we introduced in this course to solve the case study. Ask for permission if you want to use any other R-packages. The list of allowed packages includes, but is not necessarily limited to:
  - `tidyverse` and all contained packages
  - `janitor`
  - `skimr`
- Create all visualizations with the `ggplot2` package only. You may use extensions to `ggplot2` to create specialized types of visualizations such as word clouds with `ggwordcloud`.

A.2 Grading

Consider the following hints when solving the case study to achieve your desired grade:

- Both what you present and how you present it matter: carefully consider which form of visualization to use and implement it accurately.
- Make it work, then make it nice! Don't get lost in details like axis formatting or color palettes before establishing the correct data foundation and selecting the appropriate visualization form. An unsuitable visualization format, even if it looks polished, won't help you. Think the other way around!
- Exploratory data analysis thrives on your creativity and perseverance. Don't settle for the first analysis you come up with; try multiple approaches. If you discover an interesting pattern, delve deeper. Although everyone has the same task, there will be many different ways to solve it.
- When in doubt about the task or information is missing, make assumptions and document them as comments! Make sure your assumptions are reasonable!

- Readable code is better than code that is hard to understand (use the pipe operator, break long expressions into multiple lines, indent the code, add comments where appropriate).
- Reproducibility is highly valued in science. Ensure that your R-scripts can be rendered 1:1 on another computer without additional effort. Lengthy setting of working directories or manually creating subfolders and copying files will decrease your score. Add all your custom data to the designated `data` folder.
- Copying code and getting inspired by others is okay as long as the copied code is understood and can be explained, and the source is indicated as a comment. This includes large language models like ChatGPT.

A.3 Submission

Follow the instructions below to get started working on the case study and submit your final results:

- Download the provided R-project template containing all R-scripts and the necessary data to work on the case study.
- The R-project template contains a total of four scripts in the `template` folder:
  - `load_data.R`
  - `part_1_data_exploration.R`
  - `part_2_exploratory_data_analysis.R`
  - `part_3_topic_identification.R`
- Use the R-script templates in the `templates` folder to document your solutions for each part of the case study. Do not include any other R-scripts in your final submission other than the ones listed above, and do not rename the files.
- You can use as many R-scripts as you need while you work on the case study. But be sure delete all others before submission.
- In addition to the lisetd R-scripts, you must submit all custom data you used in your solution. This includes any external data as well as dictionaries you created. Add the files to the `data` folder, for example as CSV- or XLSX-files.
- Before submission, delete the data files provided with the case study from the `data` folder. This will reduce the file size of the ZIP-archive you are going to create next for your final submission.
- Finally, create a ZIP-file from the folder `case_study_ss_23` and submit the file via ILIAS before the submission deadline.

Appendix B: Data

B.1 Source

The data was obtained from the WISO database (https://www.wiso-net.de/), using the University's subscription to the service. Python and the Beautiful Soup module were utilized for the extraction process. The named entities and part-of-speech tags were extracted with spaCy using the large German model (de_core_news_lg).

B.2 Data Files

The data set consists of the following files.

Tabelle 1: Data set overview

File name	Description
<magazine>_articles.rds	Articles and meta data for <magazine>
<magazine>_ner.csv.gz	Named entities for articles in <magazine>
<magazine>_pos.csv.gz	Part-of-Speech tags for articles in <magazine>

B.3 Data Dictionary

The following table contains a data dictionary for the data set:

Tabelle 2: Data dictionary

Data Set	Column Name	Description
<magazine>_articles.rds	doc_id	A unique ID for each article
	date	The article's publication date
	medium_code	A short code for the magazine
	title	The article's title
	source	Short text for the magazine's issue
	author_text	Extracted name or initials of author
	section	Structural unit specific to magazine
	subesection	Structural unit specific to magazine
	ressort	Structural unit specific to magazine
	text	The article's text body
	wiso_permalink	Link to the article on WISO website
<magazine>_ner.csv.gz	doc_id	The article's unique ID
	entity_text	The entity's original text

Data Set	Column Name	Description
	`entity_label`	The label according to spaCy's schema
`<magazine>_pos.csv.gz`	`doc_id`	The article's unique ID
	`token_text`	The token's original text
	`token_pos`	The part-of-speech tag for the token
	`token_lemma`	The lemma form of the token
	`is_stop`	Weather the token is a stop word

B.4 Loading the Data

Please use the following lines to load each data set. You can find the code in the script file `templates/load_data.R`:

```r
# Articles data
focus <- readRDS("data/focus_articles.rds")
stern <- readRDS("data/stern_articles.rds")
spiegel <- readRDS("data/spiegel_articles.rds")

# Merge all articles into one (if needed)
all <- bind_rows(focus, stern, spiegel)

# POS data
focus_pos <- read_csv("data/focus_pos.csv.gz")
stern_pos <- read_csv("data/stern_pos.csv.gz")
spiegel_pos <- read_csv("data/spiegel_pos.csv.gz")

# Merge all POS data into one (if needed)
all_pos <- bind_rows(focus_pos, stern_pos, spiegel_pos)

# NER data
focus_ner <- read_csv("data/focus_ner.csv.gz")
stern_ner <- read_csv("data/stern_ner.csv.gz")
spiegel_ner <- read_csv("data/spiegel_ner.csv.gz")

# Merge all NER data into one (if needed)
# Merge all articles into one (if needed)
all_ner <- bind_rows(focus_ner, stern_ner, spiegel_ner)
```