Exercise 1: Data Transformation and Visualization

The WordLens project

Prof. Dr. Nicolas Meseth

OVERVIEW

To develop proficiency in natural language processing (NLP), it is essential to first update your general data exploration skills. For this reason, your initial exercise will not involve working with unstructured data. Instead, you must demonstrate your ability to use **the five fundamental data transformation techniques** and employ **the basics of data visualization** with structured data first.

Here are the relevant lessons from the online script. Make sure you study them carefully, including the provided literature, in order to solve this exercise:

- 1 - Working Environment
- 2 - R and the Tidyverse
- 3 - Data Loading
- 4 - Data Transformation
- 5 - Data Visualization

Good Luck!

YOUR TASKS

To pass this exercise, you must complete the following tasks and submit your results via ILIAS. You find the submission details below.

1. Install the Working Environment

In this task, install the necessary software for the development environment on your computer and make sure everything works properly. You find all instructions and download links in the online script. As a reminder, here is a list of the software you need to install:

- A current version of the R programming language (>= 4.2.3)
- The latest version of RStudio Desktop
- Git

To test your working environment, open a new terminal in RStudio and checkout the GitHub code repository for the WordLens project. Install all required R-libraries and run the first script `1_why_data_transformation.R` from top to bottom. If everything works properly, the script should produce two different visualizations.

2. Transforming Tweets

In this task, you must apply your data transformation skills to modify the `tweets` data set in various ways. The goal is always to create a view on the data that gives answers to questions or that allows visualizing the data afterward.

1. Create a transformation to get only the original tweets from Anna-Lena Baerbock!
2. On average, which user gets the most likes (favorites) for their original tweets?
3. Calculate the percentage of retweets vs. original tweets for each user in the data set!
4. Create a new Tibble with the months of 2022 in one and the number of tweets in another column! Group the data by user!
5. On average, which users in the data set add the most hash tags to their tweets?
6. What is the average tweet length for each user in the data set?
7. Create a top 10 list showing who writes the longest tweets on average! Consider only original tweets!
8. What hour of the day is the most active regarding the number of tweets? Create a tibble sorted by the hour of day and a second column with the average number of tweets!

3. Exploring Tweets

In this task, you must combine your data transformation and data visualization skills to create a plot for each of the following questions:

1. Is there an association between the number of retweets and the number of likes (favorites) a tweet gets?
2. Visualize the development of the total number of tweets per month by political party (SPD, Grüne, FDP)! What do you observe, and how could you explain this phenomenon?
3. Create a suitable visualization for the last question from the previous exercise to uncover patterns in the use of Twitter over the hours of the day!
4. Extend on the previous visualization and create a separate plot for each political party (SPD, Grüne, FDP) in one visualization (hint: `facet_wrap` might help) to highlight potential differences in usage patterns across the political parties.

SUBMISSION

For your first exercise, please submit:

- One or more photos (.jpg or .png) of your assembled hardware kit.
- A screenshot (.jpg or .png) from your computer in which the `smoke_test.py` is opened in Visual Studio Code, the code is running without errors, and the terminal shows the line "Please hit enter to exit".

Submit all files via the corresponding exercise in ILIAS.

MORE EXERCISES TO PRACTICE

4. Simple Transformations

If you want to practice your data transformation (and partly visualization) skills with simpler exercises, try the following ones. You don't have to submit these. Solutions will be provided.

1. Filter the data set for tweets that are not retweets (i.e., `is_retweet` is FALSE).
2. Extract the top 10 tweets with the highest `retweet_count`.
3. Create a new column called `is_reply`, where the value is `TRUE` if `in_reply_to_screen_name` is not `NA` and `FALSE` otherwise
4. Filter the data set for tweets with at least one hash tag. Sort the tweets by number of hash tags in them.
5. Filter the data set for tweets in the English language (`lang == "en"`) and containing at least one URL.
6. *Unnest* the `urls` column. Then, count the frequency of each unique domain (e.g., "twitter.com").
7. Create a new column called `created_day` which extracts the date (without time) from the `created_at column`. Then, count the number of tweets per day and visualize the result as a time series.
8. Identify tweets that mention other users. *Unnest* the `user_mentions` column and calculate the number of times each user is mentioned. Then, find the top 10 most mentioned users.
9. For each user, calculate the proportion of their tweets that are retweets, replies, and quotes. Then, visualize the results as a stacked bar chart, where the x-axis represents users and the y-axis represents the proportion of each type of tweet.

MORE QUESTIONS TO PRACTICE

Try to answer the following questions to practice your understanding of the topics around the WordLens project. The questions are optional and not part of the submission.

- What are advantages of a Tibble over the classic R data frame?
- What are the benefits of using the pipe operator `|>` when working with data in R? How does it improve code readability and organization?

- In the context of data visualization, what is the Grammar of Graphics, and how does ggplot2 implement it? How does this framework help users create complex and customizable visualizations?
- Research and explain the concept of "laziness" in data manipulation with dplyr. How do "lazy" operations improve efficiency when working with large datasets?