Extracting Information from the YouTube API

Overview

In this exercise, you'll use a Python script and the YouTube API to compile a list of videos along with some metadata. You will save the results to a CSV file and load the file in your R environment for exploratory analysis.

In a following exercise, you'll use the video list and extract the audio track from and transcribe the spoken words using a text-to-speech model like Whisper v3. With this body of data, you can analyze the video content using R and the Tidyverse.

This exercise assumes you already completed the previous exercise on getting started with the HPC cluster. If not, go back and complete that exercise first.

Task 1: Become familiar with the provided Jupyter notebook

In this exercise, you'll need to run code snippets that would go beyond the scope of this course if you had to write them from scratch. Therefore, I prepared a Jupyter notebook with helpful code snippets for you. You can find the notebook in the code repository you cloned in the previous exercise.

Your first task in this exercise is to start a Jupyter Lab server on the HPC cluster and open the notebook `youube_api.ipynb`. The notebook is structured into six sections using markdown cells. In each section, you find the code for a specific task. Your job is to review the code cells in each section, get a basic understanding what each cell does, and try to execute each cell. Make sure work your way through the notebook from top to bottom, as the code in later cells may depend on the results of earlier cells.

> Note
>
> In the first section of the notebook, you need to set the constant `API_KEY` and assign a valid YouTube Data API key. In the context of this course, I will provide you with a valid key that you can use. Make sure you never share this key with others or commit it to a public repository.

Task 2: Compile a list of YouTube videos

The notebook contains code snippets with two options to compile a list of YouTube videos. You can either search for videos by a specific keyword or retrieve videos from a specific channel. Depending on your research question, either one of these options might be more suitable.

1. Come up with a hypothetical research question that involves analyzing YouTube videos and their content. Write down the research question in a new cell of the notebook.
2. Based on your research question, decide what kinds of videos might help you answer it. You can choose to search for videos by a specific keyword and use a subset of the result. Or you can retrieve videos from a specific channel. Explain your decision in a new cell of the notebook.
3. Adjust the code in the notebook to compile a list of videos based on your decision. Run the code and save the results to a CSV file.

Attention: Take a close look at the functions that retrieve the video data. For example, there might be a filter on a specific language. Is the filter useful for your research question? If not, adjust the filter accordingly.

Task 3: Perform exploratory analysis on video list in R

After you compiled the list of YouTube videos, you can download the CSV file to your local computer. From there, load the file into a data frame with R and the Tidyverse. Take a look at the available columns in the data set and perform an exploratory analysis. Visualize whenever possible.

Task 4: Improve your video list based on the exploratory analysis

With the insights from the exploratory analysis, you can go back and refine your compilation of the video list. For example, you might find that the videos you retrieved are not suitable for your research question. In this case, you can adjust the search criteria and compile a new list of videos. Or, you can add more videos, for example by including the videos from a another channel or additional search terms.

Repeat the cycle several times until you are satisfied with the video list. When that's the case, you are ready to move on to the next exercise.