Reading .csv files into your R script in an R project directly from Google Drive

Han Olff

2024-08-08

Why you need this sometimes

Frequently, we read datafiles directly with read_csv("filename") from the published link of the dataset within a Google Sheets database. The big advantage of is that you then keep all your related datafiles together, plus their metadata.
But this does always work well. For example the .csv files may change all the time because they come from a bioinformatics pipeline, or because they are very big, too big for Google Sheets database (a Google Sheets table cannot hold more than 10 millions cells, i.e. the product of the number of rows and columns filled with data)
In these cases is better to read .csv files that are on your google drive directly into an R script. It is not a good idea to syncing them first to your local drive with Drive for Desktop, and then read them with read_csv(filename) into your script. As we generally work with R projects, such a project should allow cloning between different computers, and version management with Git. This means that the data in the script should always be read from the Google Drive of the data owner, not requiring first downloading or syncing the datafiles first.
Only when the project is completed and published, the .csv files should be "frozen" in a data archive. Until then, they should remain dynamic in a single location of the person managing the data.

Setting up your R project

A we work with R projects with library version management through the library renv, first set this up:

```
library(renv)       # for locking the versions of libraries in your project
renv::activate()    # Activate the project (also showing if your project needs Git synchronizing)
renv::restore()     # restore the correct versions of the libraries for this project
```

```
## - The library is already synchronized with the lockfile.
```

```
library(tidyverse) # for readr etc
library(here)      # for correct handling of relative file paths
```

Authenticating your access to Google Drive

Enable read-only access for Tidyverse libraries (include the library googledrive) for your Google Drives

```
read_only_scope <- "https://www.googleapis.com/auth/drive.readonly"
googledrive::drive_auth(scopes = read_only_scope)
```

This includes access to folders and drives that are not yours but that you have access to. When doing this the first time after opening the R project, switch to to your browser, choose your google account, and check the box "read your Google Drive files". Once you have authenticated a particular account (indicated by the email adress) you can select this in the console window of RSTudio direct next time in the script when you run googledrive::drive_auth(scopes = read_only_scope) without going to your browser. This is because a file named .httr-oauth containing the authentication is written in your working directory

Set the path to your local data folder

In the rest of this explanation, you learn how to get your .csv datafiles from Google Drive folders that you have access too, store them locally and read them in your script. For this, first set set your local working directory for data files, not anywhere, but specifically as the "data" subfolder of your local github project. Create this folder directly in the root folder of your project if it does not yet exist. Make sure then to the following line to your .gitignore file that you find (or create) in the root folder of your R project: data/ This ensures that these downloaded datafiles are ignored by Git. So you do not have to commit changes for them all the time. They do not need to be kept in sync by Git as they are in your Google Drive anyway.

```
data_dir<-here::here("data")
```

Now your local working directory is changed to the subfolder data of your Github project root folder

Downloading CSV files to your data folder and reading them in your script

We will use the googledrive::drive_download() function to download csv files from Google Drive. For this, you need to know the file ID of the CSV file you want to download. The file ID is a unique identifier for the file in Google Drive, which you can find in the file's URL (web link to the file). This file ID can be found by:
- opening the google drive at drive.google.com
- navigating to the .csv file, right-click on the file and select "copy link"
- pasting the link in your R script and extract the ID

For example, of the link https://drive.google.com/file/d/1_k4LBOo_yJlzQ_yYlLtzCY3xRS6_0LkG/view?usp=drive_link the ID of the file is 1_k4LBOo_yJlzQ_yYlLtzCY3xRS6_0LkG

Download and read this google drive file EVI2001_2023.csv with file ID 13DVqqDp_SpwshIPk3YIIh6iagkjTUF7B This file is on the drive GSME_Data as "anyone with the link can view". it will first be downloaded to the 'data' subfolder of your local github project, and is then read from there in R so that you can analyse it.

```
file_id<- "16kDi6DHH2moJ6Y7_APMk3cU64nuy_04S"  # ID of the file file EVI2001_2023.csv on Google Drive
file_info<-googledrive::drive_get(googledrive::as_id(file_id)) # get metadata
file_name<-file_info$name # extract the file name
file_name_short<-sub("\\..*", "", file_name)  # cut off the .csv part of the name
destination_path<-file.path(data_dir,file_name)  # make the folder+name path
googledrive::drive_download(file_info, path=destination_path, overwrite = TRUE) # download the file
filename<-readr::read_csv(destination_path, show_col_types = FALSE) #read the file in your script
assign(file_name_short,readr::read_csv(destination_path, show_col_types = FALSE)) #read the file in you
```

The same procedure can be used for a file that is on any google drive with only access by specific people. Just change the file_id. You can only read files with IDs that you are authorized to view on Google Drive.

Final remarks

Using this procedure, what you do is: - You store the .csv datafiles of your project in a central place on a Google Drive. This can be a shared drive, or a personal drive. It is better if you use a shared drive, as the files stay there if the google account of the person who puts the data there would disappear, e.g., expire.
- You read the files through their Google file ID, not by their name. If you rename a file on your google drive, the ID remains the same.
- In the script, you read the files from any google drive, and store them locally in the "data" subfolder of your local Github project folder. Note that this Github folder should be outside your local synchronised G drives, as otherwise you get problems with double synchronization. Because you download them in the script, they are always up to date when your online files change. Also, they can this way be used by multiple people who work jointly on the GitHub project.
- This replaces a workflow where you use Drive for Desktop to sync your Google Drive files to a local folder, and then read your files into R from there. The disadvantage of this is that another person looking at / working on the same Github project should all the time adjust all the links to the file based on their local Drive for Desktop path.
- This workflow is based on the stronger concept that the data are only in one place, and different people can read them from there, similar as we use in the published links from Google Sheets
- Note that when you use this workflow and keep your data in .csv, it is still very important to document your dataset in the Star Schedule database format, with lists of tables, variables etc. in the Google Sheets database concept that we use. Only particular big .csv files can then be kept outside the .gsheet file, but include their information in the list of tables, documentation of variable names etc.