Clio Infra Website:
Creation, Use and Maintenance.

Michail Moatsos*

October 29, 2021

Abstract

This document contains everything one needs to know about how the new Clio Infra website is created and how to recreate it from scratch if necessary. All the requirements are detailed, while all materials and software used are open source. The main part of the procedure is build in R-scripts. Those scripts are used to fill-in manually constructed html templates. The result is a set of static pages for the web server. The R code is fed with both data and metadata that are available at the Clio Infra dataverse repository. Some additional metadata are drawn from other online open sources, which are detailed in the document. The html layout is made using Bootstrap. The demo front end is reachable here, and the official here. Some pending development and debugging tickets for the website and the service improvement are available on Github. And the Github directory for the elements necessary to rebuild the website from scratch are available here.

*Researcher at the Economic and Social History Group, Department of History and Art History, Utrecht University, e-mail: m.moatsos@uu.nl

Contents

1 Introduction

Various considerations made it necessary to re-create a more focused version of the Clio Infra website, and the most resource efficient way appeared to be a rather simple R script that would create the html pages using the Clio Infra data available at its dedicated Dataverse repository. This approach is described in detail here so any Clio Infra responsible, with entry-level undedrstanding of R, to be able to update the content when necessary.

1.1 Built using Bootstrap

The website is developed using Bootstrap version 3.3.7. However, some of the css files have been modified for additional functionalities (such as the back-to-the-top button). Thus, one needs to use the Bootstrap files as they are found on the server or at the Github repository for smooth functioning. For the exact html templates developed and used see section 2.

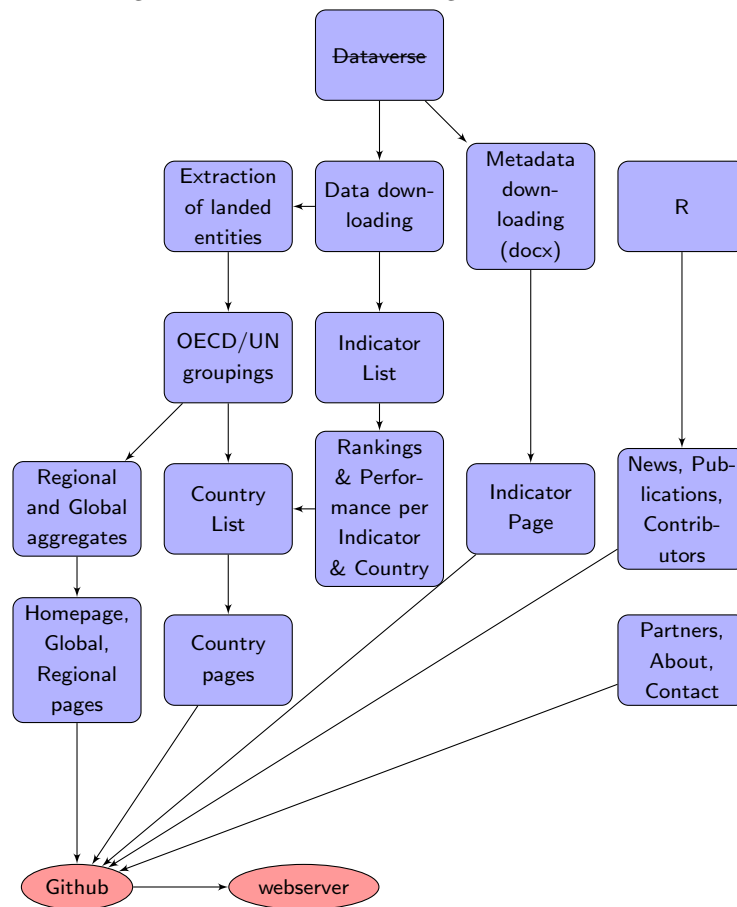## 1.2 Workflow

Figure 1: Workflow for an R generated website

Figure 1 shows the workflow of website creation via the various R scripts and the available

3

data. The Dataverse block is striked-through to underline the fact that in the current version those files need to be fetched manually in an appropriate local folder, in order for the R scripts to access and process them.

1.3 Overview of items generated by the scripts

1. Indicators per country in separate xlsx with long format, created by running CountryHTML_JSON.R, having filename structure:
   CountryName_IndicatorName_TerritorialRef_BorderStart_2012_CCode_XXX.xlsx
   for example: Zambia_LabourersRealWage_TerritorialRef_1964_2012_CCode_894.xlsx
   Exported at folder: /IndicatorsPerCountry.

2. All indicators for a given country in one xlsx with wide and long format, created by running CountryHTML_JSON.R, having filename structure:
   CountryName_AllIndicatorsAvailable_TerritorialRef_BorderStart_2012_CCode _XXX.xlsx
   for example: BosniaandHerzegovina_AllIndicatorsAvailable_TerritorialRef_1992_2012_CCode _70.xlsx
   Exported at folder: /CountryData.

3. (a) A "broad" xlsx per indicator using all "2012 border" countries of the Clio Infra dataset, and (b) a "compact" xlsx per indicator with rows only for countries with available data. Both are created by using IndicatorHTML.R, with both wide and long formats.
   Exported at folder: /data.

4. One html page for each indicator
   Exported at folder: /IndicatorPagesWithMenus.

5. One html page for each country
   Exported at folder: /CountryPagesWithMenus.

6. index.html, indexOECD.html, Partners.html, News_Publications.html located at the same folder as the R scripts.

7. The html files that actually need to be uploaded on the server are located in the folder "ToUpload".

8. An xlsx file containing all indicators for historical entities at "DataAtHistoricalBorders.xlsx".
   Exported at folder: /data.

WARNING: if at any point the xlsx files fail to export in the r scripts replace write.xls with write.xls2. Do note that most of them have been already replaced.

2 Requirements in tools and data

The website is created by using a set of templates and some scripts written in R. The templates are written in html and make use of the css functionalities of Bootstrap.

Those HTML templates are:

- IndexTemplate.html

- IndicatorsTemplate.html

- CountryTemplateJSON.html and CountryTemplateNoMoreVisualsJSON.html

- NewsPublicationsTemplate.html

- PartnersTemplate.html

The R scripts are (in strict order of execution):

- ReadData.R

- IndicatorHTML.R

- CountryHTML_JSON.R

- HistoricalBordersData.R

- AddMenusToIndicatorAndHome.R

- AddCountryMenu.R (only run this after running AddMenusToIndicatorAndHome.R)

- Contributors.R this needs to run only once to create ContributorsList.xlsx; so if there is no new contributor there is also no need to run this script again

- NewsAndPublications.R

- PartnersPage.R

- FooterSubstitution.R

- some supportive scripts that only need to be called from the other scripts are (this obviously means that *you don't need to call them directly*): IndicatorHTMLSubstitutions.R, LongToClio.R, citations.R, MakeTheGGplot.R, f_IndicatorMenu.R, and f_IndicatorMenu2.R.

Some main texts and external images used in the process are:

- AboutClioInfra.txt

- logo of various participating entities (found in images folder under www)

To run the aforementioned R scripts you need to install the following R libraries: *readxl, xlsx, ggplot2, jsonlite, tidyr, RefManageR, plyr, WriteXLS, stringr, foreign, countrycode.* The scripts execute successfully with R version 3.6.3 (2020-02-29); last successful run on September 6th, 2021.

Files related to Bootstrap that need to be copied to the server www folder are all placed in the www folder of the Github repository. This part is excecuted automatically by the updating script that fetches the files from Github and places them in the production server at IISH.

A few additional data files are required, which are not included in the dataverse repository, but they are included in the Github repository:

- "CIA-Factbook-Countries with notes for their independence status.xls", constructed with information found in: `https://www.cia.gov/library/publications/the-world-factbook/fields/2088.html#af`

- "statcan - countries list and codes from statistics canada.xls", constructed with information found in: `http://www.statcan.gc.ca/eng/subjects/standard/sccai/2011/scountry-desc`

- "UN Countries or areas, codes and abbreviations.xls", constructed with information found in: `http://unstats.un.org/unsd/methods/m49/m49alpha.htm`

- "oecdregions.csv" file containing the OECD split of the world intro regions. Provided by Auke Rijpma.

3   Processing

As already mentioned, the data (and metadata) files need to be manually placed in the appropriate folders so that R scripts can locate them (see below; for example, the data and the metadata files from the data contributors must be placed in "historical.all.standardized.2" folder that is located in the same folder as the scripts that create the website; the main folder). However, in future versions of the scripts the data files will be fetched automatically from the dataverse.

3.1   Metadata files

For ease of processing and until (and if) the metadata files on dataverse are updated with new ones, the pre-edited metadata files found on this Github repository must be fed to the R scripts instead.

Because the R script requires the metadata in txt format **follow the instructions** found in ConvertDOCStoTXT.xls to convert the docx files to txt. For convenience the txt files are also provided in the Github repository.

NOTE: the names of the contributors in the metadata file need to be seperated with "and" or a comma.

3.2 The website templates

For creating the website, all the HTML templates (listed in section 1) need to be placed in the same folder where the R scripts are located.

3.3 Exporting data

The data exporting takes place in

```
CountryHTML_JSON.R
```

and can be found right below the comment within that script:

```
### Now exporting indicator and country files
```

Change the value of ExportData flag to export the files or not. If no change takes place in the data, then to considerably save time set this variable to FALSE.

4 Other pages

4.1 News & Publications page

This page is created fully manually. When new items are added then they have to be written in HTML code within the News_Publications.html page. **NB: I need to specify how articles are selected in the footer.**

4.2 Partners page

Some preprocessing to extract the list of contributors is necessary. This is done by Contributors.R script. This needs to run only once to create ContributorsList.xlsx. After that only calling the PartnersPage.R is enough to create the Partners HTML page. Then of course you need to run the FooterSubstitution.R script to properly set the footer, as indicated in the sequence of script execution above.

5 Adding new countries and indicators

This section explains how to update the files and scripts when one needs to add one or more indicators or one or more countries.

For new indicators, or the update of old ones, the initial step is to upload the xlsx with the data and the docx with the data on the Clio Infra Dataverse: `https://datasets.iisg.amsterdam/dataverse/clioinfra/` (you also need the citation details from Dataverse as described below).

First step is to place the new data and metadata files in the proper folders (doc, txt and xlsx should be in the folder historical.all.standardized.2). Remember: the first cell of the xls should have the full name of the indicator and the second row is empty.

In case of a brand new indicator (i.e. not an update of a previously available indicator) also convert the metadata doc file that the creator of the dataset has filled to a txt file. To do so see the instructions listed in the ConvertDOCStoTXT.xls (or use simple copy paste to a txt file). Those commands are intended for use in Ubuntu. If you are not using such a system, then you will have to do this manually using for example OpenOffice under Windows. Also make sure that the data are given in the proper xls(x) layout (see file "ClioLayout.xlsx"). If you have received them in a non-Clio layout, then you have to convert them manually. There is a script to convert the "How was life? 2.0" data to the Clio Layout that can be expanded accordingly ("ConvertToClioInfraLayout.R"). Second Reminder: After the conversion you need to add two lines on top and in the first cell of the first line place the name of the indicator.

If the metadata are not given in the format required (see current txt files for the format required) then the process will not conclude and an error will occur. Thus, some manual editing of the metadata file to achieve conformity might be necessary. Currently the metadata template I'm using is slightly different than the one used previously. Thus I had to downgrade the latest template to achieve conformity with the old one. This means that at some point I need to update the procedure that will allow for the new template to be used.

Before proceeding to the next step some modifications are necessary. Go to section starting with the comment: "ADD WebName and WebCategory" of the *ReadData.R* script and add those two variables (ClioMetaData$WebName and ClioMetaData$WebCategory) for the new dataset you want to include (the first being the name shown on the website for the new dataset, and the second representing the category that this new dataset belongs to). Example:

```
ClioMetaData$WebName[which(ClioMetaData$title=="Composite Measure of
Wellbeing")] <- "Composite Measure of Wellbeing"
ClioMetaData$WebCategory[which(ClioMetaData$title=="Composite Measure of
Wellbeing")] <- "Demography"
```

For ease of use keep the same WebName as it is given in the original xlsx top cell, or change that named to a one that is common among those variables. If you do not do so, naming consistency problems arise among the various tables in the scripts.

Second step is that the *ReadData.R* script needs to be run to read the new data in.

Before going to the third step you need to upload the new indicator on dataverse to be able to get the citation info. After depositing the new indicator on dataverse you need to

download the three types of available citations to the local citation folder[1]. Be careful to follow the same filename structure as the other files in the Citations folder. Also update the entries in the "CitationsStatic.xls" file.

Moreover, you need to add the document filename of the new indicator in the "IndicatorsListWithDocFiles.xlsx", and appropriately in "IndicatorsGraphType.xlsx"; and in "IndicPriorityList.xlsx" set the priority in selecting the variable for country visualizations and the indicator name "translation", short name, short menu name and WebName.

Third step requires running the IndicatorHTML.R script.

Important for completing successfully the procedure: the consistent naming of the indicator in various locations is critical. The filename of the xlsx with the original data must end with "-historical.xslx". The prefix (e.g. CompositeWellbeingIndex) should be used as it is in the docx and the txt files. It is preferable that the prefix will also be used in the "ADD WebName and WebCategory" above. Very important: the WebName must be used in the Citations.xlsx list above.

Then follow and execute the remaining scipts in order of appearance in the R script list of section 2

Running ReadData.R will execute in about 5 minutes on a good laptop, and IndicatorHTML.R will do so in approximately 10 minutes (with all flags marked as True). CountryHTML_JSON.R takes about 45 minutes with all flags up.

In the section that begins with "# PARAMETERS:" in script CountryHTML_JSON.R you can set the parameters with which the data for the visualizations in the country HTML pages will be selected.

Note that when adding a historical entity a different process needs to be followed in script: AddMenusToIndicatorAndHome.R

Note if a new dataset has a contributor not in the list, then it should be added in ContributorsList.xlsx.

Critical: When data from new countries occur, then make sure that their entry in the xls has the same structure in terms of start and end years, otherwise only the set with the most recent ones will be kept, and if an entry from different data has both start and end years, but another only has end year, the latter will be removed. So always use (the same) start and end years. This is not implemented in the script that converts the data to the clio format, so BE CAREFUL! For Cooks, French Guiana, "Macau, China", United States Virgin Islands, Netherlands Antilles, Cayman Islands, Isle of Man, Jersey only end year (2012) is necessary. For State of Palestine use 1950 and 2012 respectively.

To avoid reproducing the historical data files set CreateHistoricalDataFiles to false in script: AddMenusToIndicatorAndHome.R

Manually change citations for Historical Gender Inequality Index: this means until the script is updated to do this automatically, in every total export of the data, the

---

[1]…/Clio Infra/Website/Citations

citation needs to be manually changed again to: "Dilli, Selin, Sarah G. Carmichael, and Auke Rijpma. 2019. 'Introducing the Historical Gender Equality Index'. Feminist Economics 25 (1): 31 - 57. https://doi.org/10.1080/13545701.2018.1442582"

6   Uploading to the server

6.1   Folder correspondence table

Table 1: Where on server to upload which items from which local folder.

Local Folder	Server Folder (under /var/www)
ToUpload (no subdirectories)	/
ToUpload/CountryPagesWithMenus	/Countries
ToUpload/IndicatorPagesWithMenus	/Indicators
CountryData	/docs
JSON	/json
IndicatorsPerCountry	/IndicatorsPerCountry
data	/data
html/graphs	/graphs
Citations	/Citations

See table 1. **Do note that the docx and xlsx files with the original data as deposited by the contributors must also be placed in the /docs folder of the server**

7   Appendix

7.1   Functions and Scripts

7.1.1   LongToClio.R

This function converts long format to clio infra format

7.2   MadToClio.R

This function converts the format used in the Maddison project, to that used in the clio-infra website and dataverse. This function was written when adding the Composite Measure of Wellbeing, and it is written around it, thus it is not a generic script yet.

7.2.1 Contributors.R

Exports the list of authors with their homepage and affiliation to ContributorsList.xlsx. The correct path needs to be provided.