

Bremen, May 2021

Alexander Polte  
Johanna Fischer

**“WeSISdadia” –  
A Formalized Work-  
Flow for Uploading  
and Documenting  
Data for WeSIS and  
its Implementation  
as an R Package**



Global Dynamics  
of Social Policy CRC 1342

Gefördert durch



Deutsche  
Forschungsgemeinschaft

**No. 9 · WeSIS — Technical papers**

**Alexander Polte, Johanna Fischer**

“WeSISdadia” – A Formalized Work-Flow for Uploading and Documenting Data for WeSIS and its Implementation as an R Package  
SFB 1342 Technical Paper Series, 9  
Bremen: SFB 1342, 2021

 in alphabetical order

Johanna Fischer  0000-0001-8889-9951



SFB 1342 Globale Entwicklungsdynamiken von Sozialpolitik /  
CRC 1342 Global Dynamics of Social Policy

Postadresse / Postaddress:  
Postfach 33 04 40, D - 28334 Bremen

Website:  
<https://www.socialpolicydynamics.de>

[DOI <https://doi.org/10.26092/elib/1609>]  
[ISSN 2700-0389]

Gefördert durch die Deutsche Forschungsgemeinschaft (DFG)  
Projektnummer 374666841 – SFB 1342

Alexander Polte  
Johanna Fischer

“WeSISdadia” – A Formalized  
Work-Flow for Uploading  
and Documenting Data for  
WeSIS and its Implementation  
as an R Packages



---

# “WESISDADIA” – A FORMALIZED WORK-FLOW FOR UPLOADING AND DOCUMENTING DATA FOR WESIS AND ITS IMPLEMENTATION AS AN R PACKAGE

---

Alexander Polte and Johanna Fischer<sup>1</sup>

## INDEX

1. INTRODUCTION . . . . .	4
2. BACKGROUND: WESIS DATA UPLOAD PROCESS . . . . .	5
2.1 WeSIS data specifications . . . . .	5
2.2 Data documentation on WeSISpedia . . . . .	6
3. WESISDADIA WORK FLOW . . . . .	7
3.1 Lookup file. . . . .	8
3.2 WeSISdadia – Data Upload. . . . .	10
3.3 WeSISdadia – Documentation . . . . .	12
4. SUMMARY . . . . .	13
APPENDIX. . . . .	15

---

<sup>1</sup> Author affiliations: Collaborative Research Centre 1342 and SOCIUM, University of Bremen and Bremen International Graduate School of Social Sciences (BIGSSS), University of Bremen and Jacobs University Bremen. Corresponding author: Alexander Polte, e-mail: [alexander.polte@uni-bremen.de](mailto:alexander.polte@uni-bremen.de)

---

# 1. INTRODUCTION

---

One of the central outcomes of the CRC 1342 is the co-creation of the Global Welfare State Information System (WeSIS), which is designed as an interactive web-based information system. In line with the analytical approach of the CRC, data in WeSIS covers three topics:

- 1) Data on social policies
- 2) Data on domestic characteristics
- 3) Data on interdependences between countries and between countries and international organizations

WeSIS is accompanied by WeSISpedia – a wiki website that serves as the documentation platform and “codebook” of WeSIS and data stored in WeSIS. For managing a variety of data types and formats within WeSIS’ database system, standards ranging from valid country identifiers over date time formats to file formats had to be agreed upon. Data that was newly collected within the CRC could be prepared to adhere to all requirements right from the beginning. However, existing datasets within the scientific community or from statistical data portals such as OECD.Stat, UNdata, the World Bank’s DataBank etc. hardly comply with WeSIS’ data specifications. Before populating WeSIS with “external” data, it is therefore (almost always) necessary to transform, reformat, or restructure the data and to add additional information, meta data and, of course, to document it. Consequently, the process of preparing, uploading and documenting data for WeSIS can turn into a very time consuming and sometimes cumbersome task for CRC researchers as the work is often repetitive (especially when dealing with large datasets from which many indicators shall be drawn) and a lot of different information sources have to be managed.

To ease the work and relieve researchers (at least partly) from the workload when preparing data for WeSIS, we created “WeSISdadia” – a dedicated R package<sup>2</sup>. The source code of the package is publicly available on GitLab<sup>3</sup> together with instructions for the package installation<sup>4</sup>, and a practical example and tutorial. A complete list and description of all ‘WeSISdadia’ functions is available in the ‘WeSISdadia’ package documentation.

In this Technical Paper, we introduce WeSISdadia and describe a formalized semi-automatic workflow building on the functionality of the package. The process covers:

- 1) the preparation of data prior to the upload to WeSIS, and
- 2) the creation of a data documentation on WeSISpedia.

Core parts of the workflow are implemented in WeSISdadia which provides a set of R functions within the R environment for aligning data with WeSIS standards. It supports with automation of repetitive data coding and formatting tasks, and integrates several data processing

---

2 The development of the WeSISdadia package prototype as well as the proposed workflow is the result of collaborative efforts from many researchers within the CRC 1342. It rests on the work of everybody who was and is involved in the development and documentation of the WeSIS ecosystem which was most prominently conducted by the members of the A01 project. Special thanks goes to Achim Schmidt and Gabriela de Carvalho from the A04 project for their help in designing the workflow and testing the functionalities of the WeSISdadia. We also thank Helen Seitzer and Fabian Besche-Truthe from the A06 project for their valuable code contribution, and Nils Düpont from project A01 whose suggestions and comments significantly improved this technical paper.

3 <https://gitlab.bremen-social-sciences.de/apolte/wesisdadia>

4 Installation by R command:  
remotes::install\_gitlab(repo = “apolte/wesisdadia”, host = “https://gitlab.bremen-social-sciences.de”, dependencies = TRUE)

and management steps<sup>5</sup>. The workflow is the result of collaborative efforts from researchers of the CRC project A04 “Global Developments in Health Care Systems and Long-term Care as a New Social Risk” who developed and tested this approach.

The remainder of the paper is structured as follows. In the next section, we describe WeSIS’ data upload and documentation requirements and shortly discuss their implications for preparing and (pre-) processing (external) data. We then describe our workflow focusing on the key elements – the so-called ‘lookup file’ and the functions that are implemented in WeSISdadia. We conclude with a summary, discussion and outlook of the proposed workflow and the WeSISdadia package.

---

## 2. BACKGROUND: WeSIS DATA UPLOAD PROCESS

---

Setting up WeSIS and the underlying database system requires standards and rules that need to be followed. As with every large database, however, WeSIS’ specifications impose practical constraints on the data upload processes. We will describe these conditions first before discussing how to approach them efficiently.

---

### 2.1 WeSIS data specifications

---

In order to add data to WeSIS it is crucial to upload a data file to WeSIS which complies to the WeSIS data specifications. Importantly, data files that do not conform to the defaults will be rejected by the automatic WeSIS data validation process, resulting in a failed data upload to WeSIS<sup>6</sup>. The reasons for having such a data validation prior to the import of data into the database are manifold e.g. to ensure the integrity of the database, to reduce error or to have a consistent representation of the data. All specifications are described in WeSISpedia. They can be divided into two classes of requirements concerning the data upload file.

First, some specifications determine which compulsory information has to be provided and in which structure. These rules usually stem directly from the WeSIS upload file templates for monadic and dyadic data<sup>7</sup>. While some of the mandatory information can be more or less directly derived from the original “external” data (e.g. country, year, value, unit, scale), some information has to be added by the researcher because it either directly concerns the structure, content or functionality of WeSIS or WeSISpedia (e.g. technical variable names, label, category, data quality rating) or is simply not part of the original dataset (e.g. data source, publication date).

Second, there are several standards, conventions and coding and formatting rules. We are not going into detail here<sup>8</sup>, but want to highlight some specifications because they are particularly relevant for pre-processing “external” data.

---

5 In addition, the WeSISdadia package provides functions, which may help researchers to process WeSIS data for their analysis. Since these are not part of the core WeSIS data upload process, they will not be addressed in this paper.

6 More detailed information about WeSIS data validation in the File upload guide on WeSISpedia: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File\\_upload\\_guide](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File_upload_guide)

7 WeSIS data specifications are available at WeSISpedia: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File\\_formats](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File_formats)

8 WeSIS coding rules are available at WeSISpedia: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/Coding\\_rules](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/Coding_rules)

An important WeSIS coding rule concerns numerical entity identifiers and the corresponding naming of countries through time. WeSIS' country coding<sup>9</sup> is based on the coding standard proposed by the Correlates of War project (COW), and thus employs a three-digit numeric code (`cow_code`) to identify (historical) state entities and assigns a standardized name to it (`country_name`<sup>10</sup>). However, the WeSIS `cow_code`-`country_name`-pairs deviate from the original specification of COW in some cases as documented in WeSISpedia.

Other notable coding conventions concern, for example, the symbolic representation of dates, the decimal separator for numeric values, or the column separator for upload file in \*.csv format<sup>11</sup>.

---

## 2.2 Data documentation on WeSISpedia

---

The second part/task of preparing data for WeSIS is the documentation of data/indicators in WeSISpedia. To this end, WeSISpedia employs two types of standardized wiki pages to organize and document WeSIS data: 'indicator tables' and 'indicator pages'.

As of May 2021, there are 16 different indicator tables listed on the main page of WeSISpedia. In line with the analytical approach of the CRC 1342, each indicator table is assigned to a specific policy area and is clustered into one of three groups of variables – social policies (Y), domestic conditions (X1), and interdependencies and relations (X2).

Indicator tables are by convention 'wikitable's' with seven pre-defined columns<sup>12</sup>. Each row lists an indicator with some additional information. At this point, especially three kinds of information (respectively columns) are of interest, because they are either evaluated by the data validation during the WeSIS file upload (Technical name\*, Scale\*) or are functional by linking to the respective indicator page on WeSISpedia (Indicator name\*).

The second type of WeSISpedia wiki pages are indicator pages. The purpose of an indicator page is to provide detailed information about a single indicator. Thus, for each indicator uploaded to WeSIS, a corresponding indicator page has to be created and populated in WeSISpedia. The structure of an indicator pages is highly standardized and explicitly defined by a WeSISpedia wiki page template encompassing fields for (conceptually) describing the indicator, documenting coding rules, sources, bibliographic information and additional meta data.<sup>13</sup> The content for each category, however, is mainly unstandardized<sup>14</sup>, providing space and flexibility to thoroughly record the data collection and coding procedures, for instance.

With both the data specifications and the documentation standards in mind, it is easy to see the (additional) work that is required when preparing data for WeSIS. For this reason, we tested and set up a formalized workflow and created WeSISdadia to reduce the workload – especially when gathering a large number of indicators from external sources.

---

9 WeSIS' country coding standard is described and discussed in more detail at WeSISpedia: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/Country\\_codes](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/Country_codes)

10 WeSIS data templates require the country name in the 'country\_name' column, while the WeSIS country coding standard defines the country name in the 'entity\_name' column.

11 More detailed information about the validation can be found in the file upload guide on WeSISpedia: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File\\_upload\\_guide](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File_upload_guide)

12 ! Indicator name\* ! Subcategory I ! Subcategory II ! Subcategory III ! Technical name\* ! Scale\* ! Short description

13 <https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/Template:Indicator>

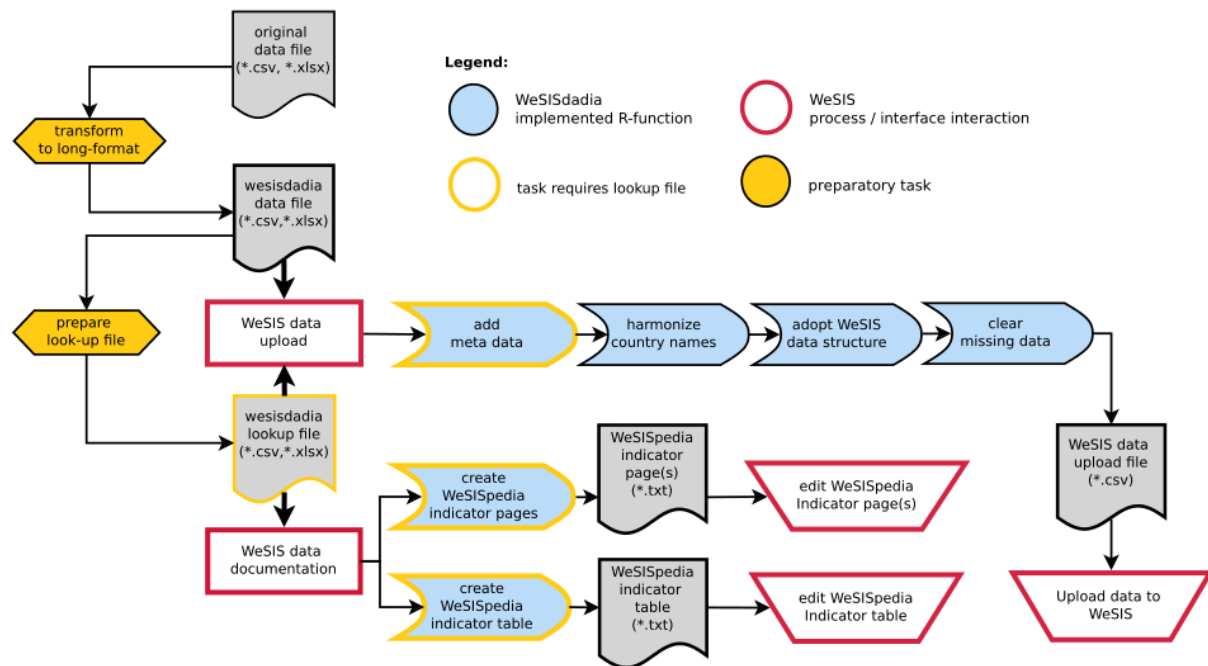
14 Except for the "Quick info box".



### 3. WEISDADIA WORK FLOW

This section introduces our workflow and the WeSISdadia package for R which supports (CRC 1342) researchers to efficiently manage data and complete the various (preparatory) tasks that are connected with the data upload to WeSIS. Figure 1 provides an overview of the different steps of this workflow, how they depend on each other, and what kind of files are required as input or will be created as output.

Figure 1: Flowchart - WeSISdadia work flow



As figure 1 shows, there are several tasks which can be performed in a single step by the use of WeSISdadia functions, for example:

- » implement different WeSIS data specifications and templates,
- » take over WeSIS specific data formatting and transformation tasks,
- » allow to efficiently organize and add metadata where necessary,
- » create standardized WeSIS data upload files and WeSISpedia indicator pages.

While some of the package functionalities can be used independently, making them widely applicable, others are rather specific to the WeSIS data upload procedure by integrating a key element of our proposed workflow – the so-called lookup file. The next section outlines the structure and function of the look-up file before we move on to describe the functions of the WeSISdadia R package.

To ease understanding our description assume an (external) dataset in a “typical” time-series–cross-section format where one row equals one country-year observation and data is stored in columns aka “variables”. From this original file several variables/columns shall be extracted and be prepared for WeSIS. Since the assumed dataset is represented in wide format it first has to be transformed to a tabular in long format, which features one column for variable values and (at least) one column for variable names. The result of this transformation will then serve as the reference data for the lookup file.

## 3.1 Lookup file

To unleash the functionality of WeSISdadia, a so-called ‘lookup file’ is required which contains meta data for each indicator. The look-up file consists of several sections which connect to different steps of the data documentation and formatting. The relevance of each column for either data formatting and/or data documentation is indicated by a mark in the respective column in table 1. In the following, we will start by describing the lookup file columns for data formatting, and continue with information on data documentation.

**Table 1:** Lookup file template for data transformation

Columns lookup file	Description	Data format	Documentation
<b>A Information on data files for data formatting:</b>			
data.file.name	Filename of file in which the data is stored (e.g. WHO_GHED_data)	x	
identifier	Original variable name in data file if unique or combination of name parts and units to arrive at unique identifier (e.g. name1 + name2 + unit)	x	
column.position.identifier	Numbers of columns in original data files where the identifier information used to construct the unique identifier is stored; in same order as identifier and separated by semicolon (e.g. 3;1;2)	x	
<b>B Mandatory fields WeSIS long-format template</b>	For guidelines see WeSIS long-format template		
technical_variable_name	Technical name	x	x
unit		x	
scale		x	x
source		x	
label	Equals also ‘indicator name’	x	x
category		x	x
publication_date		x	
data_quality		x	
data_quality_confidence		x	
<b>C Optional fields WeSIS long-format</b>			
URL.copyrights		x	
URL.indicator		x	
<b>D Information for WeSISpedia indicator table (wt)</b>	For guidelines see WeSISpedia		
wt.subcategoryI	Subcategory I		x
wt.subcategoryII	Subcategory II		x
wt.subcategoryIII	Subcategory III		x
wt.short.description	Short description		x
<b>E Information for WeSISpedia indicator page (wp)</b>	For guidelines see WeSISpedia indicator template		
wp.datatype	Data type		x
wp.valuelabels	Value labels		x
wp.relatedindicators	Related indicators		x
wp.description	Description		x

Columns lookup file	Description	Data format	Documentation
wp.codingrules	Coding rules		x
wp.citation	Citation		x
wp.relatedpublications	Related publications		x
wp.projectmanagers	Project manager(s)		x
wp.datarelease	Data release		x
wp.revisions	Revisions		x
wp.sources	Sources		x

Section A in table 1 contains key information for formatting data files according to the WeSIS format as specified in the WeSIS data template. ‘data.file.name’ refers to the file (e.g. csv, excel) in which the data for the respective indicator is stored. The ‘identifier’ and ‘column.position.identifier’ are important for extracting the indicators of interest from the data file. The former spells out the indicator name as used in the original data, e.g. “Care facilities [long-term care]”. In many cases, however, to identify a unique indicator from the original data file, a combination of information from several columns needs to be used, for example the original indicator name (“Domestic General Government Expenditure on Long-term care (health)”) plus a specification of the unit (e.g. “Million NCU”). This would then be specified as follows in the ‘identifier’ column: “Domestic General Government Expenditure on Long-term care (health)+Million NCU”. In turn, the ‘column.position.identifier’ specifies the column number(s) where (parts of) the indicator name can be found in the data file; if several columns are combined to identify the indicator, they are separated by a semicolon, e.g. 2;1. It is worth to note at this point that the order of column numbers indicated here needs to fit to the intended identifier composition. Section A is crucial to match the data from the original file with the specified indicators and the indicator metadata (e.g. technical variable name, type of scale, data quality) from the lookup file. Linking the original data file with the metadata specified in the look-up file is done with the WeSISdadia function *m2w\_lookup\_IndProperties()*.

For data formatting, information on any number of indicators from different data files can be combined in one lookup file. However, there are some limitations to be considered when using one lookup file for formatting several original data files. It only works when column names and column.position.identifiers in all data files are identical. For example, only two data files storing indicator identifier information in column 3 and 4 (i.e. column.position.identifier is 3;4) can be processed using one single lookup file. Otherwise, the look-up file needs to be split and re-specified.

To proceed, section B and C also contain information for data formatting. On the one hand, section B contains the mandatory columns of the WeSIS long format template used for uploading data to WeSIS. These columns simply need to be present for the upload. On the other hand, section C contains optional columns for the WeSIS data upload. When filling section B and C, the rules and standards as documented in the WeSIS Long Form Template apply<sup>15</sup>.

Turning to data documentation in WeSISpedia (cf. chapter 2.2), section B, D and E are relevant here. Section D, supplemented by some columns already present in section B (see table 1), contains the relevant information for populating the indicator table. Section E (also partly supplemented by information from section B) contains the information from which the WeSISpedia indicator pages are build. Consequently, these columns adhere to the templates

15 See also: [https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File\\_formats](https://seth.informatik.uni-bremen.de/wesis/wiki/index.php/File_formats)

outlined in WeSISpedia for creating indicator tables and pages. The Wiki syntax for the indicator pages can be automatically created from the lookup table using the WeSISdadia function *m2w\_WP\_fill\_IndicatorPages\_with\_lookupfile()*.

Once the lookup-file is prepared, one can start preparing the actual data from the (wide-formatted) original file before moving on to the documentation of the data in WeSISpedia. This is where WeSISdadia steps in and relieves the researcher of much of the tedious and repetitive tasks.

---

## 3.2 WeSISdadia – Data Upload

---

In this section we introduce and briefly describe several key functions of WeSISdadia package relevant to the WeSIS data upload workflow. Although WeSISdadia also contains functions which may help with other WeSIS and research related tasks or allow more flexible data handling, we will focus in this section on tools, that directly relate to a straight forward use in our proposed workflow. However, the package documentation lists and describes all available user level functions.

### 3.2.1 ADDING METADATA / INDICATOR PROPERTIES

→ *m2w\_lookup\_IndProperties(df, lookup\_file)*

As mentioned above WeSIS expects certain indicator meta data and a specific file and data format. Given a valid lookup file, the function *m2w\_lookup\_IndProperties()* adds this information to the data in R.

To do so, the function requires two input arguments:

- » *df*: which is the data.frame object in R containing the original data (in long-format),
- » *lookup\_file*: which represents the exact file path of the related lookup file.

The *m2w\_lookup\_IndProperties()* function then maps all additional indicator information which are contained in the lookup file to the data.frame object. As described previously in the lookup file section, the mapping is based on the entries in section A of the lookup file template. Applying the function, adds all indicator meta data that is mandatory for the WeSIS data upload file in columns which, in turn, are already named according to the WeSIS long form file template.

### 3.2.2 HARMONIZING COUNTRY CODES AND NAMES

→ *we sis\_harmonize\_countrynames(Cnames, try\_countryname, custom\_match, type)*

A tedious, but crucial and potentially work intensive step in preparing data is the harmonization of country codes and country names. Just as WeSIS defines its own conventions most data providers and international organizations apply their custom naming scheme (or are not standardized at all). Furthermore, there might be differences among data sources in the language they use or what country coding standards they apply.

Usually all relevant data sources have in common that they provide some form of actual country names. The function *we sis\_harmonize\_countrynames()* is a tool to transform these natural language country names into the respective country codes and the WeSIS country names, which then are returned as a data.frame object with the two columns 'country\_code' and 'country\_name'<sup>16</sup>. The transformation procedure follows two steps. As compulsory func-

---

<sup>16</sup> Column names can be adjusted to fit for the dyadic data template by using the 'type' arguments 'sender'

tion argument vector of country names has to be provided using the 'Cnames' argument of the function.

In a first step the `cow_code` for each original country name is identified by using the 'countrycode' package for R and its function `countrycode()`. This is a powerful tool that allows to "convert to and from 40+ different country coding schemes, and to 600+ variants of country names in different languages and formats"<sup>17</sup>. However, despite this overwhelming coverage of languages and codes, the 'countrycode' package fails to properly assign `cow_codes` in several instances. For these cases, a straight forward solution is to define pairs of country name and country code as 'custom\_match'<sup>18</sup> and provide them as a named vector to the `countrycode()` function.

By default the function `weasis_harmonize_countrynames()` employs `custom_match` pairs, which are based on the data and experience of the A04 project, so far. They are implemented in `WeSISdadia` and can be called by the function `weasis_custom_match_cname2cown()`. Alternatively, it is possible to pass an adjusted or individual `custom_match` vector to the `weasis_harmonize_countrynames()` function by using the optional 'custom\_match' argument.

After the country codes are mapped to each country name, their WeSIS country names are identified in the second step in line with the conventions documented on WeSISpedia's country codes page<sup>19</sup>.

Finally, `weasis_harmonize_countrynames()` returns a `data.frame` object with the two columns 'country\_code' and 'country\_name', which corresponds to the input vector of written-out country names.

Depending on the specific task, other optional arguments can be employed to customize the results returned by the function. The 'type' argument allows to adjust column names for dyadic data. A detailed description of arguments is available within the function documentation which can be called within R by calling `?weasis_harmonize_countrynames`.

### 3.2.3 ADOPT WE SIS DATA STRUCTURE

→ `m2w_format_data2weasis(df, type, year_col, value_col)`

Once indicator properties have been added to the data and country names were harmonized, the processed dataset should contain all necessary information for the WeSIS data upload. However, the proper data structure to fit WeSIS requirements will usually still be missing at this point. The structure can be implemented with a single line of code using the `WeSISdadia`'s function `m2w_format_data2weasis()`.

It is important to note that the reformatting of the data is solely done by mapping the column names of the input `data.frame` (`df`) to the column names of the WeSIS data templates. Thus, the function requires the columns of the `df` argument to be already properly named. However, the outputs of the two previously described functions (`m2w_lookup_IndProperties()`, `weasis_harmonize_countrynames()`) fulfill this requirement. When they are applied prior to reprocessing the original data then only two columns of the dataset remain unassigned: the 'year' and the 'value' column. Therefore, the two arguments `year_col` and `value_col` have to be defined, by a character string, which indicates the column name of the columns

---

or 'receiver' (see. `WeSISdadia` documentation)

17 Cited from `countrycode` git: <https://github.com/vincentarelbundock/countrycode> (17.09.2020)

18 Further information at: [https://github.com/vincentarelbundock/countrycode#custom\\_match-override-default-values](https://github.com/vincentarelbundock/countrycode#custom_match-override-default-values) (17.09.2020)

19 [https://seth.informatik.uni-bremen.de/weasis/wiki/index.php/Country\\_codes](https://seth.informatik.uni-bremen.de/weasis/wiki/index.php/Country_codes)

where the corresponding information are stored in the input dataset. The default values of these arguments are the corresponding names in WeSIS data template ('year', 'value').

The optional type argument allows to define to which WeSIS data template the data has to be formatted. There are two options, which have to be passed as character objects: 'monadic' (the default value) or 'dyadic'.

#### 3.2.4 CLEAR MISSING DATA

→ *m2w\_helper\_removeNAs(data, desiredCols)*

Often it is useful, or even required, that the dataset does not contain missing values (NAs) in a set of columns. WeSISdadia's helper function can perform this task within a single line of code. It removes all rows of the input data.frame ('data'), which contain NA values in at least one of the specified columns. They are specified using in the second argument *desiredCols*. The argument takes a character or numeric vector indicating which columns to utilize.

As a possible shortcut one can enter WeSISdadia's function *m2w\_get\_WeSIS\_noNAcols()* to the *desiredCols* argument because it returns a vector of column names of the WeSIS data template for which WeSIS does not accept NA entries.

---

### 3.3 WeSISdadia – Documentation

---

Once the original data is formatted and/or restructured and thus in shape for the upload, the task of documenting the data in WeSISpedia still needs to be tackled. Given a complete lookup file, the documentation of the corresponding indicators on WeSISpedia is straight forward because the lookup file itself already contains all necessary information for the WeSISpedia indicator page as well as the WeSISpedia indicator table. WeSISdadia provides two functions which help to convert this (user generated) content into the proper wiki syntax (e.g. the WeSIS indicator page template).

#### 3.3.1 CREATE WESISPEDIA INDICATOR PAGE

→ *m2w\_WP\_fill\_IndicatorPages\_with\_lookupfile(lookup\_file, export\_dir, url\_by)*

With the function *m2w\_WP\_fill\_IndicatorPages\_with\_lookupfile()* it is possible to generate one or multiple WeSISpedia indicator pages. As the first argument *lookup\_file*, it requires the file-path of a lookup file and as the second argument *export\_dir* a local file directory where the results shall be exported to.

For each indicator of the lookup file there will be one indicator page created as a text file (\*.txt) in the export directory. Each of these files contains the content for each respective indicator page in wiki syntax and a wiki comment, suggesting a WeSISpedia url for this indicator page. By default, the URL is created by using the 'label' column of the lookup file. It however, can be adjusted using the optional *url\_by* argument (which is not recommended, though). To finally create a WeSISpedia indicator page it suffices to manually paste the whole text from an output file into the WeSISpedia wiki page editor, which can be easily accessed by opening the suggested URL.

#### 3.3.2 CREATE WESISPEDIA INDICATOR TABLE

WeSISdadia offers two functions, which support users to create or change entries of the WeSISpedia indicator table. While the first focuses on adding new indicators to the table, the latter aims to update the entire indicator table in one step.

→ *m2w\_WP\_fill\_IndicatorTableRows\_with\_lookupfile(lookup\_df, export\_file\_name, export\_dir)*

If the indicator pages are in place, there is only one step left: To upload data of an indicator to WeSIS it is also necessary to list it in the corresponding WeSISpedia indicator table. This needs to be done before the actual data upload, because WeSIS requires the information for the upload validation. The function `m2w_WP_fill_IndicatorTableRows_with_lookupfile()` supports researchers to create new indicator table entries using the wiki table syntax. Similar to `m2w_WP_fill_IndicatorPages_with_lookupfile()`, it takes the `data.frame` of the lookup file or its file-path as main input argument, `lookup_df`. Based on the lookup file it then generates the new wiki table entries and saves them in an output file which can be specified by using the remaining arguments `export_file_name` and `export_dir`. One can then add the new entries to the indicator table by simply pasting the text of the output file at the top or bottom of the WeSISpedia indicator table. It is important to note that the URL created in the 'indicator name' column has to be the same as the URL used for the indicator page. By default the label column of the lookup file is used to populate it.

→ *m2w\_WP\_update\_IndicatorTable(lookup\_df, table\_url, user, password, export\_file)*

A more convenient way to populate the indicator table is provided by the function `m2w_WP_update_IndicatorTable()`, which updates a currently available indicator table at WeSISpedia with additional entries or changed information. Besides the obligatory lookup file, an indicator table has to be specified by its URL using the `table_url` argument. In order to retrieve the current state of the indicator table, the function also requires to pass the username and password to access WeSISpedia. An export file can be specified with the 'export\_file' argument. To finally create a WeSISpedia indicator table it suffices to manually paste the whole text from the output file into the WeSISpedia wiki page editor of the indicator table.

---

## 4. SUMMARY

---

In this paper we discuss the implications stemming from (technical) requirements when uploading data to WeSIS and writing the documenting in WeSISpedia. We proposed a formalized semi-automatic workflow and describe its implementation in the dedicated R-package WeSISdadia. Compared to an unformalized manual procedure, our approach substantially improves many aspects of pre-processing data for WeSIS, creating efficiency and maintainability while using synergies.

By introducing the lookup file the process revolves around a single, central structure for information and data management. Instead of manually adding or storing information related to each indicator in at least three different 'places' (WeSIS data upload file, WeSIS indicator page, WeSIS indicator table) and different formats (wiki mark up, \*.csv), the lookup file enables us to only use one single file in standard table format to edit and manage the information. This is especially useful when it comes to preparing and uploading data from large international data bases (e.g. World Health Organization, World Bank, Eurostat) that often contain a large number of social policy related indicators. It also allows to easily separate the different tasks: content creation/editing (lookup file), data transformation (R-Script & WeSISdadia) and upload to WeSIS (upload file, indicator page, indicator table), and thus to divide the labor concerned with data upload among different persons.

Apart from these organizational and practical benefits, the automation of several tasks substantially improves the efficiency in uploading data to WeSIS. Especially the automation relieves researchers from repetitive tasks related to formatting or coding data and meta data

for WeSIS. The WeSISdadia package helps CRC members in performing these tasks and thus saves valuable time. By summer 2021, in the A04 project already more than 3.000.000 data entries for more than 300 different indicators have been prepared, uploaded, and documented using the procedure<sup>20</sup>.

Another notable benefit of the WeSISdadia workflow is the improvement of maintainability. Once established, updates, error corrections or documentation improvements can be conducted within a view of minutes instead of repeating the whole manual preparation and upload process again. Thus, not only the risk of unsystematic human error during data preparation is decreased but also the effort needed to correct any mistakes. We argue that this can turn out to be an important asset, especially in the long-term perspective of building WeSIS, because most relevant external datasets are updated regularly and might have to be updated in WeSIS as well.

Last but not least, the WeSISdadia package fosters synergies among CRC researchers by providing a custom set of tools to process longitudinal global data on the one hand, and by integrating solutions or pre-processing steps from additional contributors. While the two functions `m2w_cow_exists_country_year()` or `m2w_format_introY2tscs()` are a good example for the former, the function `m2w_WP_fill_IndicatorTableRows_with_lookupfile()` is an example for the latter, because it was initially contributed by colleagues from another CRC-project.

---

20 Keeping in mind that this was achieved already during the development phase of 'WeSISdadia' and covers only monadic data for health care indicators, this can be regarded as an impressive result.



# APPENDIX

Figure 2: WeSISdadia - data processing scheme

