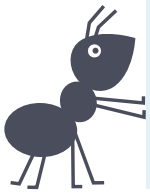
10 | Working Paper 2024

KonsortSWD
NFDI4Society

Introducing Open Data Format

A Platform-Independent, Non-Proprietary, Metadata-Enriched, Multilingual Data Format and its Implementation in R and Stata

Xiaoyao Han, Tom Hartl, Knut Wenzig



November 2024

www.konsortswd.de

Introducing Open Data Format

A Platform-Independent, Non-Proprietary, Metadata-Enriched, Multilingual Data Format and its Implementation in R and Stata

Xiaoyao Han[1], Tom Hartl[1], Knut Wenzig[1]

November 2024

https://doi.org/10.5281/zenodo.14215268

[1] SOEP at DIW Berlin

Abstract

This paper introduces the Open Data Format (ODF), a new, non-proprietary, multilingual, metadata enriched, and zip-compressed data format that meets the FAIR Guiding Principles for scientific data management and stewardship. The data format is specified as a CSV file with the raw data and an XML file containing the metadata both compressed into a zip file with the .zip extension. Data files can be enriched with multilingual metadata following the forthcoming DDI Codebook 2.6 standard. The paper also introduces software packages for R (opendataformat) and Stata (opendf) that provide import and export filters and enable data users to work with ODF data files in the respective environment.

Acknowledgment

We gratefully acknowledge the reviewers Andreas Daniel and Tobias Koberg for their invaluable feedback and insightful comments, which significantly enhanced the quality of this manuscript. Any remaining errors are solely our responsibility. We would like to sincerely thank Adam Lederer for his careful proofreading.

Keywords: ODF, Open Data Format, Metadata, DDI Codebook, Multilingual, opendataformat, opendf, R-Package, Stata Package

Table of Contents

1 Introduction

Open Science represents a new approach to the scientific process based on cooperative work and new ways of diffusing knowledge by using digital technologies and new collaborative tools. The European Commission has made Open Science the center of its research policy through comprehensive policies that emphasize transparency, collaboration, and accessibility in research and science (European Commission, 2015). In this context, calls for facilitating wider access and reuse of research data have rapidly gained traction in various disciplines. In 2016, the 'FAIR Guiding Principles for scientific data management and stewardship' were published in Scientific Data (Wilkinson et al., 2016). These principles provide guidelines to improve the Findability, Accessibility, Interoperability, and Reuse of digital assets.

The current practice of research data centers (RDCs) in providing tabular data often falls short of adhering to the FAIR principles. RDCs in Social, Behavioral, and Economic Sciences (SBE) often use the CSV format or proprietary data formats like Stata (.dta) and SPSS (.sav). While CSV lacks the possibility of metadata enrichment, proprietary data formats have drawbacks in openness and cross-platform compatibility. This illustrates the need for a new, open data format that complies with the FAIR principles for data management.

In this paper, we introduce the "Open Data Format," a new non-proprietary, multilingual, metadata enriched, and zip-compressed data format. With the Open Data Format, data producers can provide scientific use files in a data format that meets the FAIR principles. The open data format is platform-independent and open-source. Data files can be enriched with multilingual metadata structured in the upcoming DDI Codebook 2.6 standard.

In the following section, we first explain the Open Data Format (ODF) and its metadata schema, then introduce software packages in R and Stata that enable data users to work with ODF files. The paper starts by summarizing current practices of research data centers in SBE in providing tabular data and compliance issues with FAIR principles. We then discuss the requirements for a potential standardized and open data format. The third section elaborates on the metadata specification of the Open Data Format, providing a detailed description of its components. In the fourth section, we present software packages for R (*opendataformat*) and Stata (*opendf*) that enable data users to import and export the open data format and to retrieve metadata. Finally, we conclude with a summary of our work and the potential of the Open Data Format for RDCs in SBE.

2 Background

In 2016, the 'FAIR Guiding Principles for scientific data management and stewardship' were published due to the need to improve infrastructure and to support the reuse of scholarly data. The aim was to develop guidelines for data management that not just support knowledge discovery and innovation but also improve data and knowledge integration. They formulate

four foundational principles for sustainable data management: Findability, Accessibility, Interoperability, and Reusability. (Wilkinson et al., 2016).

2.1 Current Practice in SBE Sciences

Current practices in data management show substantial deficits in meeting the FAIR principles. Two key aspects of the FAIR principles that are frequently violated are the provision of extensive metadata and the use of open and nonproprietary data formats. Research data centers have the role of providing data for research purposes. They have the task of preparing data and making it available to data users in appropriate formats, as well as providing documentation and metadata.

Within the German National Research Data Infrastructure[1] (Nationale Forschungsdaten-infrastruktur, NFDI), there is the Consortium for the Social, Behavioural, Educational, and Economic Sciences in the (KonsortSWD)[2] network, which focuses on developing the research data infrastructure in these disciplines. As of November 2024, a total of 41 Research Data Centers (RDC) are part of KonsortSWD, of which at least 33 RDCs provide tabular data for quantitative analysis. We were interested in the formats in which they provide their data. For this purpose, we analyzed the websites and data portals. For 29 of 33 websites, we obtained information about the available data formats (Figure 1).

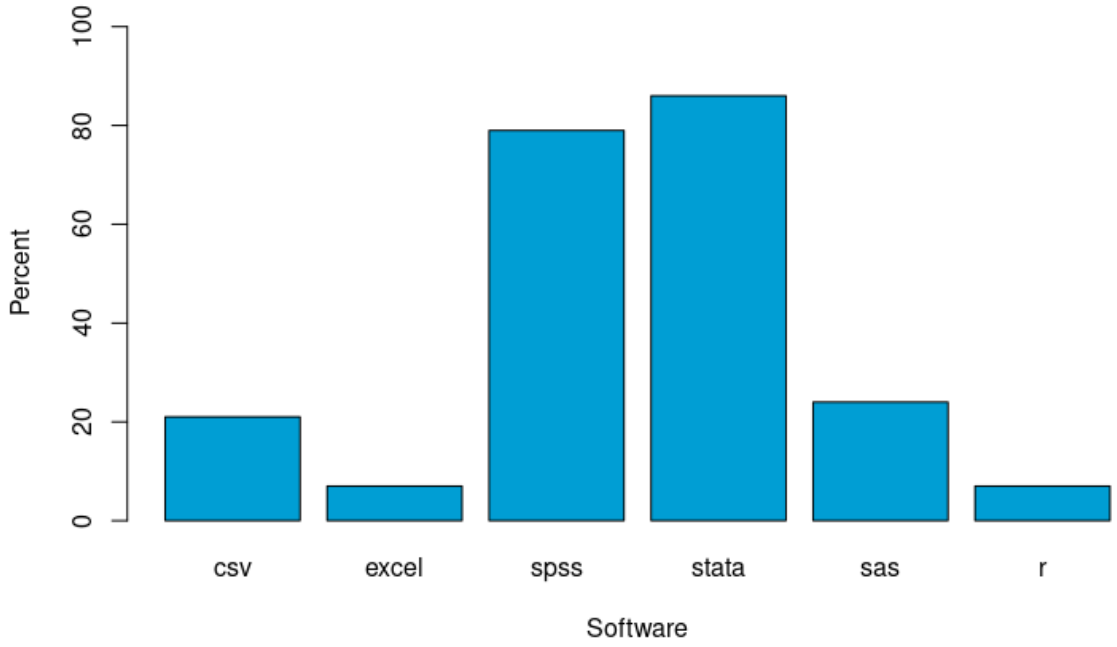


Figure 1: Data Formats provided by RDCs within KonsortSWD (self-collected data)

As shown in Figure 1, the most frequently provided data formats are those for the SPSS, Stata, and SAS software programs. These formats are proprietary and have limited cross-platform compatibility. Stata files (.dta), for example, can be enriched with additional information in the characteristics (see chapter 12.8, StataCorp., 2023), but this feature is not supported in other

1 https://www.nfdi.de/?lang=en
2 https://www.konsortswd.de/en/

software platforms. Therefore, metadata stored in the characteristics of a dta-file is lost when loaded, for example, in R using the haven package. Similar problems regarding cross-platform compatibility occur with R data files. CSV files, which could be considered a more open file format, are offered far less frequently and lack the possibility to embed metadata in the data file. Finally, the proprietary Excel data format is rarely offered.

An interesting example of a data provider that follows a more open and FAIR practice is IPUMS,[3] which offers survey and census data from all over the world. IPUMS offers a service that includes a data file in DAT format combined with so-called command files. For R, the data file and the metadata file are downloaded along with an R script. This script loads a package specifically developed for the IPUMS data portal, "ipumsR," which is used to load the data into R and label it using the metadata (XML file). The offering from IPUMS comes very close to an open data format, as the labeled datasets within each software are likely produced on the same basis: a data file (DAT) and a metadata file (XML).

Since the sole focus on the data provision practice of the RDC may neglect the existing needs of data users, we should also include the user side in our considerations. The German Socioeconomic Panel (SOEP) at DIW Berlin runs a regular user survey regarding the statistical software they use (SOEP User Survey, 2023). The results for the 2023 survey (Figure 2) show that SOEP data is mainly being analyzed from users using Stata and R. Furthermore, SPSS usage seems to flatten out at a low level, while the share of R users has been increasing constantly.

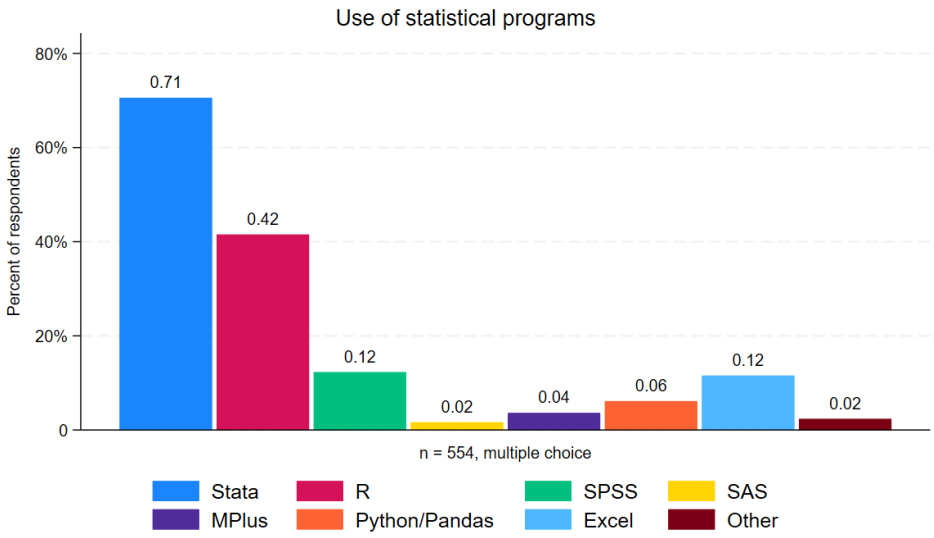


Figure 2: Statistical software used by SOEP users

The user preferences are also reflected in the data formats provided. In the past, SOEP datasets were only available in SPSS, Stata, and CSV formats. While Stata and SPSS users could work with native data files, users of other statistical software don't have this privilege. As a result, many R users had to work with CSV files, which lacked metadata, or with Stata files (.dta), which have significant interoperability problems. Availability of metadata stored in variable labels,

3 https://www.ipums.org/

value labels, or characteristics of the dta files is heavily dependent on the used import filter and software package in R. In addition, Stata, SPSS, and R have different data models for missing values, resulting in substantial information loss during data format conversion.

As shown above, tabular data is predominantly provided in proprietary formats with limited cross-platform compatibility or in data formats without embedded metadata. However, in accordance with the FAIR principles, research data should be Findable, Accessible, Interoperable, and Reusable, which includes exchangeability between different software. Thus, the convertibility issues across platforms of widely used data formats pose a significant barrier to achieving the FAIR principles. The fact that this is not a purely theoretical requirement is also reflected in the heterogeneous use of different software solutions described above. The limitations of the CSV-format regarding metadata enrichment and documentation and of other proprietary data formats regarding cross-platform compatibility necessitate the use of separate files, like PDFs, or online portals to provide metadata. Those metadata are often not machine-readable and unstructured. The current practice has further drawbacks. Provision of various metadata offerings and data formats requires additional work by RDCs. At the same time, to use this additional information, researchers must leave their familiar work environment, which can be inconvenient and potentially susceptible to errors.

These results show that current practices in data provision of RDCs in SBE sciences tend to violate the FAIR principles and at least partly neglect the needs of the scientific community. This illustrates the necessity for a new open and metadata-enriched data format.

2.2. Recent Work on Open Metadata Standards

While storing raw tabular data in CSV or similar data formats is relatively straightforward and common, there is no generally accepted practice when it comes to storing metadata. Several metadata standards have been developed in the last decades. For the social sciences, important standards include the DDI Metadata Standard, Dublin Core, and SDMX. Another interesting data and metadata format is the StatDataML, which combines data and metadata in one single XML file (Meyer et al., 2004).

Dublin Core is a simple metadata standard developed for online resources and consists of originally 13 (later 15) metadata elements. First published in 1995 (Weibel, 1995) and formalized in 1998, the Dublin Core metadata standard can be applied to a wide range of data formats, e.g. HTML/XHTML, XML, or Resource Description Framework (RDF). The Dublin Core metadata elements are title, subject, description, type, source, relation, coverage, creator, publisher, contributor, rights, date, format, identifier, and language. (Weibel et al., 1998) The advantages of Dublin Core are its simple and generic metadata schema and the broad applicability of the metadata framework to various data types. The disadvantage is the missing metadata entries on the variable level.

Another important standard that is mainly used by statistical offices is the SDMX (Statistical Data and Metadata eXchange). It was first published in 2004 and recognized as an ISO-standard (version 2.1: ISO 17369:2013) in 2005. The SDMX initiative is sponsored by several

large international organizations that provide data themselves: BIS, Eurostat, ECB, IMF, World Bank, UN, and OECD. The emphasis of the SDMX standard is the standardization and exchange of data and metadata across these organizations and their statistical platforms (IMF, 2014).

To describe a dataset, a data provider generates a data structure definition (DSD) that documents the dimensions and concepts of a dataset. Together with the DSD, datasets can be processed by other statistical offices. SDMX further developed definitions of cross-domain concepts, code lists, and glossaries to harmonize datasets, variables, and vocabularies. (Stahl et al., 2018). Since the standard was established for data sharing across these organizations, the metadata is designed for international comparison and harmonization. The disadvantage of SDMX is that the documentary metadata comes up short. While this is not a problem for data with well-known measures of economics and other disciplines, the metadata concept is not suitable for other data types like survey data, where documentation is crucial for researchers to understand the concept and measurements of variables.

Another interesting open data format that combines both data and metadata in one xml file is StatDataML, developed by Meyer et al. (2004). The XML file contains a description element containing dataset metadata and the dataset element containing the data. The description contains five metadata elements: title, source, date, version, and comment. The dataset element can be either a list or an array. The array or list element contains a dimension element, (in case of an array) a type element, and the data element. StatDataML supports the data types logical, categorical, numeric (integer, real, complex), character, and datetime. For categorical variables, there is an element for the categorical mode (unordered, ordered, cyclical) and elements for labels and label codes. The authors provide software packages to support StatDataML in R. There were supposed to be software packages for Matlab, Octave, SPSS, and Gnumeric but they are presently neither available nor findable. StatDataML format has not succeeded yet in gaining widespread use. The data format has some drawbacks because of the limited metadata fields. Besides value labels, no metadata for variables is provided. Further, its suitability for large data sets is unclear.

As shown above, a variety of metadata standards exist, but most are not well suited for modeling variable metadata, as needed for the here proposed open data format. An exception is the DDI metadata standard, first published in 1996 and since evolved to a successful metadata standard in social sciences. Today DDI is a well-established set of metadata standards for describing data produced by surveys and other observational methods in the social, behavioral, economic, and health sciences (Vardigen et al., 2008). DDI has developed several versions of its metadata standard to meet the different requirements for describing metadata. The most important are DDI Codebook (version 2), and DDI Lifecycle (version 3). DDI Codebook (newest version 2.5) is a metadata standard for simple data files while DDI Lifecycle (newest version 3.3) is a more comprehensive metadata standard designed for documentation of

complex longitudinal studies including instruments and multiple datasets. For both standards, metadata is stored in an XML file.[4]

A significant effort to expand DDI's applicability beyond its traditional domains is the upcoming DDI-Cross-Domain Integration (CDI). Designed to be used with research data from any domain, DDI-CDI offers a domain-neutral specification that covers a wide range of commonly used data structures. Thus, this new standard provides a groundbreaking mechanism for interoperating diverse data from multiple disciplines and domains at the most granular level.

The DDI Codebook Standard would meet the needs for an open data format in terms of complexity and scope. DDI Codebook supports metadata on various levels: on the study level, the data file/dataset level, on the variable level, as well as on the level of values and value labels. Furthermore, the DDI approach is highly flexible and can be customized for specific use cases. A custom metadata scheme can be defined with a DDI Profile describing which elements are optional and mandatory in a metadata scheme (*DDI Profiles | Data Documentation Initiative*, 2015). These profiles can be used in the CESSDA validator to validate any metadata XML file. Therefore, we decided to build the metadata component upon the DDI Codebook standard.

2.3. Considerations for a New Data Format

For developing a new data format, two central aspects must be considered: the storage of raw data in an open format that can be processed efficiently alongside the features and storage of accompanying metadata in an open, human readable, and standardized format.

For storing raw tabular data, the typical CSV-format is well understood and highly suitable from a technical perspective, as it has a very simple structure that is lightweight and human readable. Due to its simple structure, it can be easily imported in almost every software package. Additionally, thanks to its openness and human-readability, it is suitable for long-term archiving. These features support compliance with the FAIR criteria, as the simple structure increases both searchability/findability and accessibility, as well as interoperability and reusability. While CSV has some advantages, there are also some shortcomings. For instance, one must know (or guess) the character encoding and the column separator and handle cell content that contains the column separator. And most importantly, CSV lacks the possibility to store metadata.

As stated above, for the accompanying metadata component we decided to use an XML file using the DDI Codebook standard. To ensure interoperability, we select a fixed set[5] of generic

4 XML stands for Extensible Markup Language (XML) and is a markup language and file format widely used for encoding structured data. It is an open and both human and machine-readable format where data is organized hierarchical in a tree-structure (similar to html).

5 To have a metadata scheme with a fixed set of entries instead of a flexible number of metadata fields improves interoperability because it minimizes the risk of information loss when implementing import and export filters in different statistical software. All possible metadata fields are known to any developer of software implementations and therefore a lossless conversion can be achieved more easily. A flexible set of metadata fields makes it more difficult for developer, because all possible metadata fields in the XML file have to be anticipated by the developer when writing import and export filters for any statistical software.

and non-domain-specific metadata fields: Each variable can be described by a label, a description, and a URL (the same applies to the dataset itself). If needed, pairs of values and labels can describe the categories of the variables. This relatively minimalistic approach, where basic information is accompanied by a URL for external details, was chosen because the data format needs to be implemented within existing, diverse statistical software packages.

With implementation necessities in mind, the metadata schema does not include some information, such as missing values. Including features that are not supported by some statistical software would lead to compatibility problems. Another relevant aspect that influenced the decision to include specific metadata fields is their contribution to understanding the data itself. While Stata and SPSS recognize the concept of value labels, this differs fundamentally from similar structures in R. Although one could argue that R's approach (using so-called factors, where different categories are associated with integer levels starting from 1) is theoretically well-founded, we believe that the implementation in Stata and SPSS is more flexible (as all kinds of numeric values can be labeled) and also allows for the enrichment of legacy data.

To further enhance the accessibility of the open data format, the format enables the provision of translated versions for all metadata fields containing natural language (e.g., labels, value labels, and descriptions). This is particularly valuable when datasets are made available to an international scientific community, as multilingual metadata offers significant added value.

This leads to the specification of the ODF, which consists of a combination of a CSV file for the data and an XML file in DDI-Codebook format—a combination also suited for long-term dataset archiving. The implementation in statistical software packages must be able to import datasets in the new data format and process the metadata using the features provided by the package. Detailed features and their implementation are described in the next section.

3 Specification

3.1. Format

To comply with FAIR principles, the ODF brings together data and metadata using only open and non-proprietary formats that are both human and machine readable. Data and metadata are organized in two separate files (see Figure 3). The data is stored in CSV format and the metadata is in XML. Both documents are encoded in UTF-8.

The upcoming DDI-Codebook 2.6 metadata schema is the basis for the specification of the XML metadata file in the ODF. Further elaborations on the XML file schema are in the successive sections. The CSV file with the raw dataset is a simple comma-separated file containing numeric and character values separated by commas (,). The first row contains variable names. To classify character variables, quotation marks ("") are used as text classifier. For data exchange, both files are packed into a ZIP file. Therefore, the Open Data Format has the extension .zip.

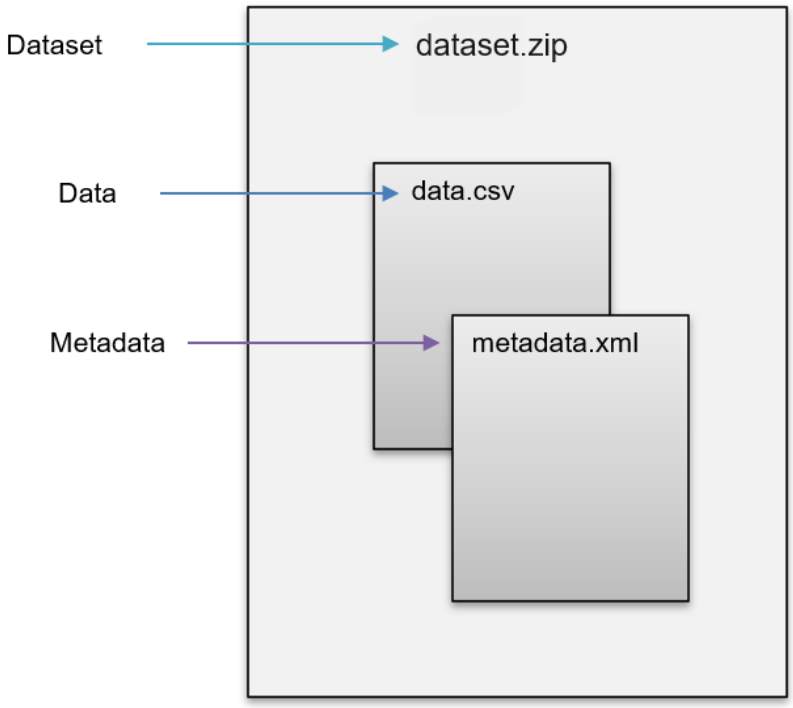


Figure 3: ODF File Specification

3.2. Metadata in the ODF

The Open Data Format provides a generic set of metadata entries that are applicable to a broad set of data types. The available metadata fields or entries are displayed in Table 1. The metadata for the dataset comprise of the *study* the dataset is derived from, the *name* of the dataset, *labels* and *descriptions* for the dataset in multiple languages, and a *URL*. For each variable the ODF provides metadata entries for *labels* and *descriptions* in different languages, the *variable type* (character or numeric), a *URL* and *Value Labels* for different values in different languages.

The multilingual concept of the data format offers the opportunity to include metadata in several languages for labels, value labels, and descriptions. Each label or description entry should be in a specific language, but a label or description without a language tag is possible as well. Similarly, for each labelled value there can be several labels in different languages. There can be one label and one description per language and variable. The ODF does not require a complete translation of all labels, descriptions, and value labels to all languages.

The language of a metadata entry is coded in ISO 639-1, an international standard that defines the codes that uniquely represent the names of languages (ISO 639-1, 2002).

Metadata Fields	Description	Occurrence
DatasetStudy	Name of the study or data collection where the data of the dataset was generated	Mandatory, one entry
DatasetName*	Name of the dataset	Mandatory, one entry
DatasetLabel*	Label of the dataset in different languages	Optional, several entries in different languages
DatasetDescription	Description of the dataset in different languages	Optional, several entries in different languages
DatasetURL	URL or DOI for dataset	Optional, one entry
VariableLabel*	Label of each variable in different languages	Optional, several entries in different languages
VariableDescription*	Label of the variable in different languages	Optional, several entries in different languages
VariableURL	URL to e.g. documentation website for variable	Optional, one entry per column
VariableType	Variable type ("numeric" or "character")	Optional, one entry per column
VariableValue	Value that should be labeled, can be a string	
VariableValueLabel*	Labels for values for different variables in different languages	Optional, several entries in different languages

Table 1: Metadata in the Open Data Format

** Labels, Descriptions and Value Labels can be defined in several languages.*

3.3. DDI Codebook Specification for Metadata XML

The metadata is stored in an XML-file structured in the upcoming DDI Codebook 2.6 XML standard. As mentioned on the DDI Alliance website,[6] "DDI is a very flexible and complex standard that may be used by various projects or organizations in 'customized' ways that best answer specific needs" (see DDI Alliance: DDI Profiles). To date, DDI-Codebook specifies 252 different elements: 243 global and 9 local ones. To define which metadata elements are mandatory and which are optional for the Open Data Format, we developed a DDI profile (*DDI Profiles | Data Documentation Initiative*, 2015) for the specification of the metadata xml-file. The profile defines mandatory and optional elements in the metadata XML and can be used to validate any ODF XML-file using the CESSDA Metadata Validator[7] (Mühlbauer and Morris, 2023).

A DDI-Codebook compliant XML file has the element <codeBook> on the first level. On the second level within the <codeBook>-element, five different elements are available: the mandatory <stdyDscr>, and the optional elements <docDscr>, <fileDscr>, <dataDscr>, and <otherMat>. In the ODF XML (profile) three of the five elements are used to store metadata: <stdyDscr>, <fileDscr>, and <dataDscr>.

6 https://ddialliance.org/

7 https://www.cessda.eu/Tools/Metadata-Validator

In the <stdyDscr> the study is documented, from which the dataset is derived. It is stored in the <titl> element (see figure 4).

```
<stdyDscr>
   <citation>
      <titlStmt>
         <titl>DatasetStudy</titl>
      </titlStmt>
   </citation>
</stdyDscr>
```

Figure 4: stdyDscr element and all sub elements in a ODF XML file

All other metadata regarding the dataset level are stored in the <fileDscr> element (see figure 5). Within the <fileTxt> element, the dataset name is stored in the element <fileName>, the labels for different languages are stored in the <titl xml:lang="language"> and <parTitl xml:lang="language"> elements (which are on sublevels of <fileCitation> and <titlStmt> – elements) with the attribute of the respective language,[8] and the descriptions are stored in <fileCont xml:lang="en"> elements with the respective language attributes.[9] Further a URL for the dataset is stored in the <notes> element in the URI attribute of the sub element <ExtLink>.

```
<fileDscr>
   <fileTxt>
      <fileName>DatasetName</fileName>
      <fileCitation>
         <titlStmt>
            <titl xml:lang="en">DatasetLabel (English)</titl>
            <parTitl xml:lang="de">DatasetLabel (German)</parTitl>
         </titlStmt>
      </fileCitation>
      <fileCont xml:lang="en">DatasetDescription (English)</fileCont>
      <fileCont xml:lang="de">DatasetDescription (German)</fileCont>
   </fileTxt>
   <notes>
      <ExtLink URI="DatasetURL"/>
   </notes>
</fileDscr>
```

Figure 5: fileDscr element and all sub elements in a ODF XML file

It is worth noting that the profile for the xml-file fulfills the DDI Codebook 2.5 scheme with one exception: the element <fileCont> (a sub element of in <fileText>, which is a sub element of <fileDscr>) is allowed multiple times for different languages (defined through the "xml:lang"-attribute) in the DDI profile for ODF, while according to DDI Codebook 2.5 XML schema it can

8 Note that there is only one <titl>-element allowed. All dataset labels in other languages must be stored in <parTitl> elements.

9 The language attribute is optional; however you can only have one label and description without language tag. For both the dataset and each variable.

occur only one time. However, in the upcoming DDI Codebook 2.6 XML schema the <fileCont> will be allowed to occur several times in different languages.[10]

Variable Metadata are stored in the element <dataDscr> (see figure 6). Within the <var> element, the variable labels are stored in the <labl xml:lang="en"> elements with the xml:lang attribute indicating the respective language of the label. The variable description is stored in the <txt xml:lang="en"> element with the xml:lang attribute indicating the respective language. For each pair of value and value label there is a <catgry> element containing the labelled value in <catValu> (character or numeric) and the value labels in the <labl xml:lang="en"> elements with the respective language attributes. The variable type is stored in the <varFormat type="variable type"/> element within the type attribute (which can take the values numeric and character) and a variable URL or PID can be stored in the URI attribute of the <ExtLink URI="Variable URL or PID"/> within the <notes> element.

```
<dataDscr>
   <var name="VariableName">
      <labl xml:lang="en">VariableLabel (English)</labl>
      <labl xml:lang="de">VariableLabel (German)</labl>
      <txt xml:lang="en">VariableDescription (English)</txt>
      <txt xml:lang="de">VariableDescription (German)</txt>
      <catgry>
         <catValu>VariableValue (e.g. -1)</catValu>
         <labl xml:lang="en">VariableValueLabel (English)</labl>
         <labl xml:lang="de">VariableValueLabel (German)</labl>
      </catgry>
      <catgry>
         <catValu>VariableValue (e.g. -2)</catValu>
         <labl xml:lang="en">VariableValueLabel (English)</labl>
         <labl xml:lang="de">VariableValueLabel (German)</labl>
      </catgry>
      <varFormat type="VariableType ("numeric" or "character")"/>
      <notes>
         <ExtLink URI="VariableURL"/>
      </notes>
   </var>
   <var name="Variable Name">
      [...]
   </var>
   [...]
</dataDscr>
```

Figure 6: dataDscr and subelements in the ODF XML file

10 see DDI Alliance Jira Issue: fileCont in fileTxt is 0..1 so does not support multiple languages without repeating fileTxt. https://ddi-alliance.atlassian.net/browse/DDICODE-546?page=com.atlassian.jira.plugin.system.issuetabpanels%3Aall-tabpanel [accessed: 08/14/2024]

Further information about requirements of the XML file is published in the DDI Profile of the Open Data Format on Zenodo (Wenzig et al., 2024).

4 Software Packages with Import and Export Filters

To load data files in the Open Data Format to R or Stata, software packages were developed that provide import and export filters to read the datasets to R and Stata and to enrich the native dataset objects with the metadata from the ODF file. Additionally, the software packages provide functions to retrieve and display metadata and some helper functions.

4.1. R-Package opendataformat

The R Package is published on CRAN[11] and can be downloaded with the 'install.packages' command. The development version is available on Github.[12] The R-Package provides three main functions to read (*read_odf()*) and write (*write_odf()*) ODF files and to display metadata (*docu_odf()*). Further, it provides a function to retrieve specific metadata (*getmetadata_odf()*) and a function to set the active metadata language (*setLanguage_odf()*). Finally the package provides an S3-method for the merge function in R (*merge.odf()*).

4.1.1. The ODF Data Frame in R

Before loading an ODF data file in R, it is important to determine the appropriate type of object for storing the data and how to incorporate metadata into that object. Commonly used structures for handling tabular data in R include data.frames and tibbles. The latter is an enhanced version of the data.frame class, introduced in the tibble package (Müller and Wickham, 2023). Unlike statistical software such as Stata, not only does R lack a native framework for handling variable and value labels, it also does not natively support multilingual metadata.

One approach for adding metadata to an R tibble or data.frame object is to use attributes. However, a challenge arises with data.frames, as attributes are often lost during operations such as indexing or merging. In contrast, tibble offers greater persistence of attributes, making them a more robust choice for maintaining metadata.

The *opendataformat* package adds a set of standardized attributes to an R tibble to enrich it with metadata from the ODF file. Both the tibble object and the columns/variables of the data frame have attributes containing metadata. Since R has no native concept for multilingual metadata, the *opendataformat* package also uses attributes to store available metadata languages and to define an active language. The

11 https://cran.r-project.org/web/packages/opendataformat/index.html
12 https://github.com/opendataformat/r-package-opendataformat

resulting tibble object is enriched with metadata in the attributes and is assigned the class “odf” in addition to "tbl_df", "tbl", and "data.frame" in R.

Metadata Fields	Metadata in R tibble	Attribute name
DatasetStudy	Attribute study of the tibble	study
DatasetName*	Attribute name of the tibble	name
DatasetLabel*	Attribut label of the dataset in different languages	label_[language tag]
DatasetDescription	Attribut description of the dataset in different languages	description_[language tag]
DatasetURL	Attribute url of the tibble	url
VariableLabel*	Attribute label of the column in different languages	label_[language tag]
VariableDescription*	Label of the variable in different languages	description_[language tag]
VariabelURL	Attribute url of the column	url
VariableType	Attribute type of the column	type
VariableValue / VariableValueLabel*	Attribute with values that should be labeled (can be a string) and value labels in the namespace of the labelled values	labels_[language tag]
	Additional metadata attributes in R:	
	Attribute lang indicates the active/current language, by default, English is chosen as active language, if existing	lang
	Attribute languages for available metadata languages	languages

Table 2: Storing ODF Metadata in an R tibble

* DatasetDescription, DatasetLabel, VariableDescripton, VariableLabel and the VariableValue/ValueLabel in different languages are assigned to the respective attribute with the language tag as suffix of the attribute name.

The data frame and its columns/variables both have two language attributes: The *languages* attribute indicates the available languages for the metadata (labels and descriptions) in the data frame. The *lang* attribute contains the active metadata language of an ODF data frame object. It indicates the language of the metadata displayed by default by the *docu_odf()* and *getmetadata_odf()* functions. The active language can be changed using the *setLanguage_odf()* function.

The *study* attribute contains the name of the study or data collection where data of the dataset was generated. The dataset and its columns/variables also have a name attribute indicating the dataset's and columns’/variables’ name. The *url* attribute for the dataset and the variables/columns can contain a link to a webpage which can be displayed as interactive link.

Both the dataset and the variables also have *label* and *description* attributes in one or several languages. The English label for a dataset or for a variable/column for example is stored in the *label_en* attribute of the dataset or variable/column. Additionally, the label in the active language will be stored in the *label* attribute. By default, English is the active language if available. If the haven package is loaded, the variable labels in the active language are displayed in the data viewer in RStudio. Similarly to the labels, the English descriptions will be stored in the attribute *description_en* of the data frame and the columns/variables.

The variables/columns further have the *type* attribute indicating the variable type (e.g. numeric, character) and the *labels* attribute containing value labels in one or several languages. For example, the value labels in English are stored in the *labels_en* attribute, which is a numeric vector containing the values to label and the actual labels for each value in English in the namespace of each element of the numeric vector.

4.1.2. Functions in the opendataformat Package

The *opendataformat* package provides three main functions: an import filter to load ODF files as tibble in R (read_odf), a function to display metadata for a variable (docu_odf), and an export filter to save an R tibble (or data.frame) as ODF-file (write_odf).

The input to the import filter *read_odf()* is a path to an ODF zip-file or the name of an ODF zip-file in the working directory. The output is a tibble with ODF metadata stored in the attributes. Further, the function takes several optional arguments: *languages*, *nrows*, *skip,* and *select*.[13]

The *docu_odf()* function displays metadata for a data set or a variable. The input to the R-function is an ODF tibble with metadata in the attributes or one of its columns. Metadata for a data set or a variable is displayed in the Rstudio viewer (or alternatively in the R console). In Figure 7, we show the viewer output of the docu_odf-function with the metadata for the variable school leaving degree (pgpsbil) from the pgen dataset from the German Socioeconomic Panel (SOEP). Further the function takes several optional arguments: *style*, *variables*, *languages,* and *replace_missing_language*.[14]

13 The R-command with all default function arguments: **read_odf**(file, languages = "all", nrows = Inf, skip = 0, select = NULL). Only the file input is mandatory. By default, metadata in all available languages is imported. languages can be set to "all" or to a vector with language codes, e.g. languages=c("en", "de"). select can take a vector with column names or indices. By default, metadata in all languages is imported (languages="all"), and all rows (nrows = Inf, skip = 0) and columns (select = NULL) are read. The arguments nrows and skip take integers as inputs indicating the maximum number of rows to read and the number of rows to skip.

14 The R-command with all default function arguments: **docu_odf**(input, style = "print", variables = "yes", languages = " current", replace_missing_language = F). By default, the metadata is displayed in the viewer (style = "viewer") and in the current/active language (languages = " current"), which is English by default. Alternatively, it can be displayed in the console (style = "print" or style = "console") or in both (style = "both"). When metadata for labels and descriptions is missing for a language, labels and descriptions in other languages can be displayed instead (replace_missing_language = T). Together with the dataset metadata a list of the variables and variable labels in the data frame is displayed by default (variables = "yes").

To save a dataset from R as an ODF file, the *opendataformat* package provides the *write_odf()* function. All metadata saved in the attributes compatible with the ODF-specification are written to the metadata XML in the ODF-file. The *write_odf()* function requires the data frame object and the file name (or file path) to the new ODF zip-file as inputs. Further it has the optional arguments *languages* and *export_data*.[15]

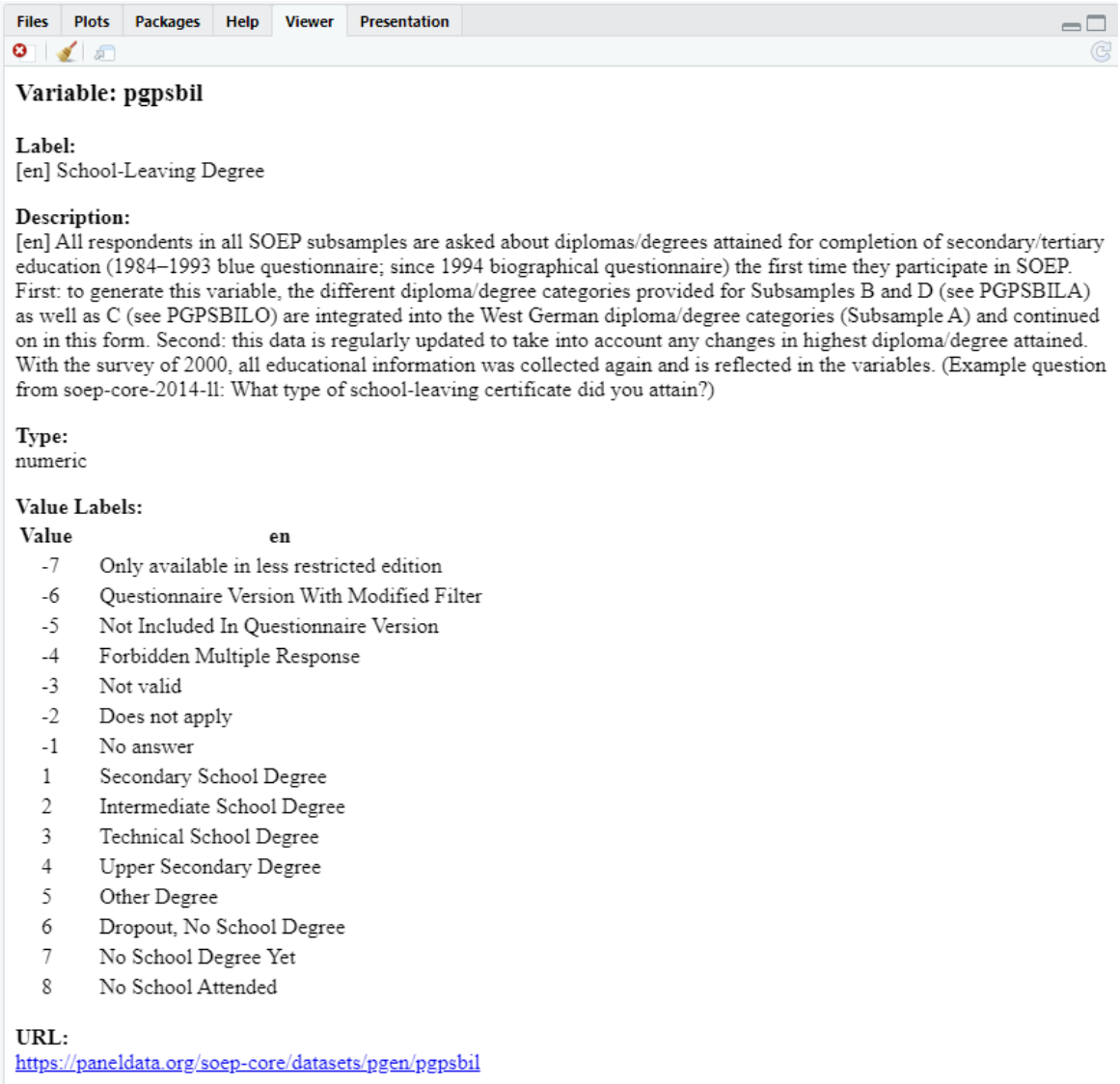


Figure 7: Function output docu_odf for a variable

The R package further has a function to change the active language of an ODF tibble in R (*setlanguage_odf*) and a function to retrieve labels and other metadata (*getmetadata_odf*). It also provides a S3-method for the generic merge function in R to keep attributes with variable

15 The R-command with all default function arguments: **write_odf**(x, file, languages = "all", export_data = TRUE). The first input x is a tibble object in R with metadata compatible to the ODF and stored in correctly named attributes. The second input is the name of the output file or file path ending with .zip. It is possible to export only the metadata XML (export_data = FALSE) and only metadata in specific languages (languages = c("en", "de")). By default, metadata in all available languages is exported (languages = "all") and the data CSV is exported together with the metadata (export_data = TRUE).

metadata when merging two odf tibbles. Merging ODF tibbles with the base merge command would lead to a loss of the variable metadata stored in the attributes of the columns.

4.2. Stata-Package opendf

The Stata package opendf is currently available at the Statistical Software Components (SSC) Archive (SSC) using the command *ssc install opendf* in Stata. The development version of the package is available on GitHub.[16] Like the R package, it provides three main functions to read ODF files, to write ODF files, and to display metadata.

4.2.1. Storing Metadata in Stata

To load an ODF data file in Stata, the same question arises: where to store the metadata in Stata. Stata has a more comprehensive native structures for metadata than R, it not only provides labels for the variables and the dataset but also value labels for each variable. Therefore, it is straightforward to store these metadata in the existing metadata fields. The characteristics in Stata can be used enrich the dataset with additional information. The *opendf* package uses characteristics to store the metadata of the ODF file in Stata.

The metadata is stored in following characteristics of the data frame: The *study* characteristic (*_dta[study]*) contains the study name where the dataset is derived from. The *dataset* characteristic (*_dta[dataset]*) contains the dataset name. The *url* characteristic for the dataset (*_dta[url]*) and the variables/columns (*varname[url]*) can contain a URL, a digital object identifier (DOI) or a persistent identifier (PID) for the dataset or a variable.

The description of the dataset and of the variables is stored in the *description_[lang_tag]* characteristic, with the respective language tag at the end of the characteristic name. E.g. the description of the dataset or variable in English is stored in the *description_en* characteristic (e.g. *_dta[description_en]* ; *varname[description_en]*). The dataset label, variable labels, and value labels in different languages are stored in the labels and value labels of Stata. The variables/columns further have the *type* characteristic indicating the variable type (e.g. numeric, character) (see table 3).

4.2.2. Functions in the opendf package

The Stata-Package provides three main functions: *opendf read, opendf docu*, and *opendf write.*

The *opendf read* function is an import filter to load opendf-files into Stata. The input to the Stata-function is a path to an ODF zip-file or the name of an ODF zip-file in the working directory. The dataset is loaded to Stata with ODF metadata stored in the labels and characteristics. Further the function has several optional parameters: *rowrange*, *colrange*, *clear*, *save*, *replace,* and *verbose*.[17]

16 https://github.com/opendataformat/stata-package-opendf

17 The Stata-command with all function arguments **opendf read** *input* [,**rowrange**([start][:end]) **colrange**([start][:end]) **clear save**() **replace verbose**]. **Rowrange,** and **colrange** indicate the ranges of rows

Metadata Fields	Metadata in Stata	Label / Characteristic name
DatasetStudy	Characteristic study of the dataset	study
DatasetName	Characteristic name of the dataset	name
DatasetLabel**	Label of the dataset in different label languages	stored as dataset label (up to 80 characters)
DatasetDescription*	Characteristic description of the dataset in different languages	description_[language tag]
DatasetURL	Characteristic url of the dataset	url
VariableLabel**	Labels of the variables in different label languages	stored as variable label
VariableDescription*	Characteristic description of the variables in different languages	description_[language tag]
VariableURL	Characteristic url of the variables	url
VariableType	Characteristic type of the variables	type
VariableLabel/ VariableValueLabels**	Values and Value labels of the variables in different label languages	stored as value label

Table 3: Storing ODF metadata in Stata

* *Description for dataset and variable in different languages are assigned to the description characteristic with the respective language tag as suffix of the characteristics name.*

** *Dataset, Variable and Value Labels in different languages are assigned to the respective label in the respective label language*

The *opendf docu* function displays metadata for the dataset or a variable. If the function is called without any input, the dataset metadata is displayed. Otherwise, the function displays the metadata for the indicated variable (see Figure 8). In Figure 8 we can see the output of the *opendf docu* function with metadata for the school leaving degree variable from the pgen dataset from the German Socioeconomic Panel (SOEP). The function further has the *languages* argument.[18]

To export a dataset from Stata as an ODF file, the *opendf* package provides the *opendf write* function. All metadata saved in the labels and in the characteristics compatible with the ODF-specification are written to the metadata XML in the ODF-file. The *opendf write* function

and columns to read. **clear** indicates to clear any existing loaded dataset. **save** indicates to directly save the dataset as dta-file with the indicated name. **Replace** ensures that any existing file is overwritten with the save command. **verbose** enables more warnings and messages.

18 The Stata-command with all function arguments: **opendf docu** [*varname*, **languages**()]. The input is a variable name and is optional. The languages argument indicates in which language the metadata shall be displayed. If not indicated, metadata is displayed in the active label language. The argument can take a language tag (ISO639-1) or "all" as input to display metadata in the respective language or all available metadata in all languages.

requires the file name (or file path) to the new ODF zip-file as inputs. Further it has the arguments *input*, *languages, variables, replace* and *verbose*.[19]

Further the Stata package provides some additional functions: *opendf installpython* and *opendf removepython* install and remove a portable python version for the opendf functions since the functions require a working Python integration in Stata. Other functions convert ODF files or datasets in Stata into four CSV files containing data and metadata and vice versa (*opendf csv2dta*, *opendf dta2csv*, *opendf csv2zip* and *opendf zip2csv)*. The functions are provided to help data providers and others to convert data into the Open Data Format.

```
. opendf read "pgen.zip", clear

. opendf docu pgpsbil
Label: School-Leaving Degree
Description: All respondents in all SOEP subsamples are asked aboutdiplomas/degrees
attained for completion of secondary/tertiary education (1984-1993 blue
questionnaire;since 1994 biographical questionnaire) the first time theyparticipate in
SOEP. First: to generate this variable, thedifferent diploma/degree categories provided
for Subsamples Band D (see PGPSBILA) as well as C (see PGPSBILO) are integratedinto the
West German diploma/degree categories (Subsample A)and continued on in this form. Second:
this data is regularlyupdated to take into account any changes in highestdiploma/degree
attained.With the survey of 2000, all educational information wascollected again and is
reflected in the variables. (Example question from soep-core-2014-ll: What type of
school-leaving certificate did you attain?)
URL: https://paneldata.org/soep-core/datasets/pgen/pgpsbil
Variable Type: numeric
Value Labels en:
-7 :  Only available in less restricted edition
-6 :  Questionnaire Version With Modified Filter
-5 :  Not Included In Questionnaire Version
-4 :  Forbidden Multiple Response
-3 :  Not valid
-2 :  Does not apply
-1 :  No answer
1 :  Secondary School Degree
2 :  Intermediate School Degree
3 :  Technical School Degree
4 :  Upper Secondary Degree
5 :  Other Degree
6 :  Dropout, No School Degree
7 :  No School Degree Yet
8 :  No School Attended

. opendf write "output_dataset.zip", replace
 Dataset successfully saved in opendf-format to \\hume\soep-data\MA\kwenzig\opendataformat\
> soep-core/output_dataset.zip.
```

Figure 8: Function output opendf docu for a variable

19 The Stata-command with all function arguments: **opendf write** output [,**input**() **languages**() **variables**() **replace verbose**]. The output is the name (and path) of the ODF output file. It is possible to export only metadata in specific languages (e.g. **languages**("en")). Instead of writing the current data frame from the Stata environment, an existing DTA file can be converted to ODF using the **input**() option. To export only specific variables/columns the **variables**() argument indicates the variables for export. To display more warnings and messages the **verbose** option can be enabled.

5 Summary

Current data management practices in social, behavioral, and economic sciences show substantial deficits in meeting the FAIR principles for scientific data management and stewardship. Tabular data is mostly provided in CSV without embedded metadata, or in proprietary data formats that have serious interoperability problems across different software platforms. Both practices violate FAIR principles.

To address these challenges and promote the FAIR principles, this paper introduced the "Open Data Format" (ODF) — a new, non-proprietary, multilingual, and metadata-enriched format for tabular data. Additionally, the paper presented software packages for data users to work with ODF files: the R Package *opendataformat* and the Stata package *opendf*.

The Open Data Format is platform-independent, open-source, and designed to meet the diverse needs of data producers and users. It allows for the integration of multilingual metadata structured in the DDI Codebook standard and offers compression to optimize storage and processing efficiency. The format also facilitates the inclusion of supplementary information, enhancing data documentation and long-term archiving by removing reliance on proprietary software. By adopting the Open Data Format, data producers can reduce the need for multiple data formats, streamline data processing, and improve the accessibility and quality of metadata for users. This new format also supports direct links to data portals from within statistical software, saving time and improving research quality.

The Open Data Format adheres to the FAIR principles. To enhance **Findability**, metadata should be assigned a globally unique and persistent identifier and registered in a searchable resource with comprehensive metadata information. ODF features a URL area that allows metadata to be linked to online platforms. Additionally, a persistent identifier is planned for integration into the schema in the next development phase. **Accessibility** is prioritized in ODF by providing a widely recognized and free data format that is accessible to all users and system environments. In ODF data and metadata are both included in a machine and human readable format. The multilingual feature further enhances accessibility. **Interoperability** is achieved through the standardized metadata schema DDI Codebook that underpins ODF. This schema enables a formal, accessible, shared, and broadly applicable data description. The clear definition of existing metadata fields alongside import and export filters developed for different statistical software ensure lossless conversion and interoperability across platforms. Lastly, ODF provides a rich description of data with a variety of accurate and relevant attributes, promoting **Reusability**. Moreover, it recognizes the importance of machines in data-rich environments and advocates for machine-readable formats for data and metadata.

The provided software packages introduce functions to read (*opendf read* in Stata, *read_odf()* in R) and write (*opendf write* in Stata and *write_odf()* in R) ODF data files into the respective environment and to display the metadata (*opendf docu* in Stata and *docu_odf()* in R). We are currently working on a software package for Python with similar functionalities.

There are still some challenges for the further development of the data format. Some compatibility challenges for different platforms are still present and need to be considered either in the data format or in the implementation with the software packages, such as NaN (not a number) and Inf (infinity) values or value labels for continuous values. The generic metadata schema may also be unsuitable for other disciplines. Thus, it is planned to integrate the emerging metadata standard DDI CDI (Cross Domain Integration). Further optimizations in the performance of the software packages are also planned.

6 References

Akdeniz, E., & Moilanen, K. (2023). CMM CESSDA Metadata Model. In Zenodo (CERN European Organization for Nuclear Research). CMM CESSDA Metadata Model (zenodo.org)

Betancort Cabrera, N., Bongartz, E. C., Dörrenbächer, N., Goebel, J., Kaluza, H., & Siegers, P. (2020). White paper on implementing the FAIR principles for data in the social, behavioural, and economic sciences. https://doi.org/10.17620/02671.60.

Boeckhout, M., Zielhuis, G. A., & Bredenoord, A. L. (2018). The FAIR guiding principles for data stewardship: fair enough?. *European journal of human genetics*, *26*(7), 931-936. https://doi.org/10.1038/s41431-018-0160-0.

DDI Profiles | Data Documentation Initiative. (2015). *Ddialliance.org*. https://ddialliance.org/learn/resources/ddi-profiles

European Commission (2015). Directorate-General for Research and Innovation, Open innovation, open science, open to the world – A vision for Europe. Publications Office. https://data.europa.eu/doi/10.2777/061652

IMF (2014). A Data Collection Strategy: Leveraging SDMX Standards. BOPCOM - 14/08, Washington, D.C

Ionescu, S. (2009). Creating a DDI Profile, DDI Working Paper Series -- Best Practices, No. 6., http://dx.doi.org/10.3886/DDIBestPractices06

ISO 639-1 (2002). "ISO 639 Language code". https://www.iso.org/iso-639-language-code

Kent, J. J., & Schuerhoff, M. (1997, August). Some thoughts about a metadata management system. In Proceedings. Ninth International Conference on Scientific and Statistical Database Management (Cat. No. 97TB100150) (pp. 174-185). IEEE.

Meyer, D., Leisch, F., Hothorn, T., & Hornik, K. (2004). StatDataML: An XML format for statistical data. Computational Statistics, 19, 493-509.

Mühlbauer, A., & Morris, M. (2023). CESSDA Metadata Validator - Core component (1.1.0). Zenodo. https://doi.org/10.5281/zenodo.7961900

Müller K, Wickham H (2023). *tibble: Simple Data Frames*. https://tibble.tidyverse.org/, https://github.com/tidyverse/tibble.

da Silva Santos, L. O. B., Burger, K., Kaliyaperumal, R., & Wilkinson, M. D. (2023). FAIR data point: a FAIR-oriented approach for metadata publication. *Data Intelligence*, *5*(1), 163-183. https://doi.org/10.1162/dint_a_00160.

SOEP User Survey (2023). Unpublished internal document. https://www.diw.de/en/diw_01.c.603784.en/soep_user_survey.html

Stahl, R., Staab, P., Stahl, R., & Staab, P. (2018). The Main Elements of SDMX. Measuring the Data Universe: Data Integration Using Statistical Data and Metadata Exchange, 85-101.

StataCorp. 2023. Stata 18 Base Reference Manual. College Station, TX: Stata Press.

Twenty-Seventh Meeting of the IMF Committee on Balance of Payments Statistics

Vardigan, M., Heus, P., & Thomas, W. (2008). Data documentation initiative: Toward a standard for the social sciences. International Journal of Digital Curation, 3(1), 107-113.

Weibel, S. L. (1995). Metadata the foundation of resource description. Annual review of OCLC research, 52-56.

Weibel, S., Kunze, J., Lagoze, C., & Wolf, M. (1998). Dublin core metadata for resource discovery (No. rfc2413).

Wenzig, K., Han, X., Hartl, T., & Saalbach, C. (2024). Open Data Format DDI XML profile (1.0). Zenodo. https://doi.org/10.5281/zenodo.13329882

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific data, 3(1), 1-9.

Imprint:

Contact:
Sozio-oekonomisches Panel | Socio-Economic Panel (SOEP)
Deutsches Institut für Wirtschaftsforschung e.V. (DIW Berlin) | German Institute for Economic Research (DIW Berlin)
Mohrenstraße 58
10117 Berlin
https://www.diw.de/de/diw_01.c.412809.de/sozio-oekonomisches_panel__soep.html
kwenzig@diw.de, xhan@diw.de, thart@diw.de

November 2024

KonsortSWD Working Paper:
KonsortSWD, as part of the National Research Data Infrastructure, is expanding offerings to support research with data in the social, behavioural, educational, and economic sciences. Our mission is to strengthen, expand and deepen the research data infrastructure for the study of society. It should be user-oriented and consider the needs of the research communities. An important cornerstone is the network of research data centres that has been built up for more than two decades by the German Data Forum (RatSWD).
This series contains articles on research data management that are produced in the context of KonsortSWD. Articles that have been externally and double-blind peer-reviewed are marked accordingly.

KonsortSWD is funded by the German Research Foundation (DFG) within the framework of the NFDI – project number: 442494171.

DOI: https://doi.org/10.5281/zenodo.14215268

Citation suggestion:
Han, X., Hartl, T. & Wenzig, K. (2024). *Introducing Open Data Format: A Platform-Independent, Non-Proprietary, Metadata-Enriched, Multilingual Data Format and its Implementation in R and Stata.* KonsortSWD Working Paper 10/2024. Konsortium für die Sozial-, Verhaltens-, Bildungs- und Wirtschaftswissenschaften (KonsortSWD).
https://doi.org/10.5281/zenodo.14215268