SDMX Constructor: User Manual

International Labour Organization: Department of Statistics

2023-04-24

Contents

Preface

Welcome to the SDMX Constructor User Manual!

SDMX Constructor[1] is a powerful desktop software tool that helps users model aggregate data per the SDMX standards [2] . It eases generating and editing SDMX artefacts and ultimately supports data availability and access through SDMX-compliant data portals (such as ones built with .Stat Suite[3]).

SDMX Constructor is one of the tools that comprise the ILO SDMX toolkit [4], alongside SMART (Statistical Metadata-driven Analysis and Reporting Tool) and the SDMX Excel Add-in.

This user manual for the SDMX Constructor provides step-by-step instructions on using the tool in a user-friendly and accessible manner. It provides an in-depth understanding of SDMX Constructor's features and functionalities. It is an essential resource for anyone using the tool to manage and share data following the SDMX standards.

Below is a screenshot of the tool's landing page as an example of the user interface.

Click here to enlarge the image

---

[1]It was previously called DSD Constructor, it supported creating and editing Data Structure Definitions (DSDs) and their related artefacts (i.e., concept schemes and code lists). The significant improvements and additions to the software justify the name change of the tool from DSD Constructor to SDMX Constructor. With the latest update, the tool can now translate SDMX artefacts in multiple languages, integrate with the .Stat Suite, and manage reference metadata. Additionally, the tool now includes Table Modeller functionalities that make it easier for users who are new to SDMX to understand and use the software. The updated software is no longer just a DSD Constructor but a complete SDMX Constructor capable of handling various aspects of SDMX data and metadata. The new name better reflects the software's expanded capabilities, making it more relevant and recognizable to the target audience (https://ilostat.github.io/dsdc/).

[2]An ISO standard (since 2005, version 2.1 since 2013): https://www.iso.org/obp/ui/fr/#iso:std:iso:17369:ed-1:v1:en and https://sdmx.org/?page_id=5008

[3]To learn more about the .Stat Suite platform please refer to the documentation here: https://siscc.org/stat-suite/. The SDMX Constructor has been optimised to seamlessly integrate with the .Stat Suite platform, as its back-end client.

[4]https://ilostat.ilo.org/resources/sdmx-tools/

Figure 1: A screenshot of SDMX Constructor

Audience and use cases

The SDMX Constructor is designed to meet the needs of a wide range of users, from beginners who are new to SDMX to data managers who need advanced capabilities for managing large and complex datasets. In addition to these user types, there may be other groups of users with specific requirements or use cases. This user manual provides guidance and instructions for using the SDMX Constructor, focusing on the needs of these various user types.

SDMX beginners: These users are new to SDMX and want to use the SDMX Constructor to view, edit, and create new SDMX structural artefacts. They may need assistance understanding the SDMX concepts, terminology, and the software's user interface.

For SDMX beginners, this manual explains using the SDMX Constructor to access, view, edit, and create SDMX structural artefacts from SDMX registries. While one can find information on SDMX concepts and terminology from other sources, this manual focuses explicitly on the user interfaces of the SDMX Constructor. The SDMX beginners may be keen to know how to access SDMX artefacts from default SDMX registries and connect to a new SDMX registry from the SDMX Constructor.

Data managers: These users are likely familiar with the SDMX concepts and terminology and may require information about advanced functionalities offered by the SDMX Constructor. For example, they may want to use the SDMX Constructor as a backend client to manage SDMX artefacts for the .Stat Data Lifecycle Manager (DLM). For such cases, they may also use the SDMX Constructor to build the initial structural metadata when creating a new .Stat

Suite instance.

There are several topics that data managers may be interested in learning when it comes to creating SDMX artefacts from scratch. Firstly, they may want to start by setting up a registry as a local folder. Next, they can prepare inputs and create several artefacts, including the AgencyScheme, ConceptScheme, and Codelist and the DSD, Dataflow, ContentConstraint, and CategoryScheme. After creating these artefacts, they may want to learn how to upload the XML file to the Data Lifecycle Manager (DLM). Additionally, they may want to know how to access SDMX artefacts from default SDMX registries and connect to a new SDMX registry for editing SDMX artefacts directly in the DLM.

SDMX metadata managers: These users manage SDMX artefacts and ensure their accuracy and consistency. They may use the SDMX Constructor to model data and modify SDMX structural artefacts, including translating SDMX artefacts in various languages, managing annotations and creating Metadata Structure Definition (MSD).

Scope and assumptions

This comprehensive manual provides a step-by-step guide on getting started with the software, including installing it and creating and managing SDMX artefacts. The manual includes a detailed overview of the tool's user interfaces, including menu items, navigation, and other essential features. The manual does not, however, delve into the intricacies of the SDMX standard itself. Describing the SDMX standard is beyond the scope of this manual, as it is an extensive and complex topic requiring more in-depth discussion and is available elsewhere. Instead, the manual assumes that readers are familiar with the SDMX standard and focuses on explaining how to use the tool within that context.

Overview

The chapters in this manual build up toward providing comprehensive guidance that covers topics users need to know to get started with the application, understand its navigation, and use its features and functionalities to model data effectively. The manual includes the following chapters.

- Chapter 1: Benefits of SDMX Constructor
- Chapter 2: Getting started
- Chapter 3: User interface
- Chapter 4: Using SDMX Constructor
- Chapter 5: Special Topics
- Chapter 6: Glossary

Contact information

For more information and to seek basic technical assistance or support on the tool, please reach out International Labour Organization - Department of Statistics at sdmx.support@ilo.org.

Chapter 1

Benefits of SDMX Constructor

The first purpose of SDMX Constructor is to help users model their data in accordance with SDMX standards. This is essential for data providers who need to ensure that their data can be shared and used by others in a consistent and standardized way. With SDMX Constructor, users can easily define data structures, create data flows, and specify data concepts, all in accordance with SDMX standards.

The second purpose of SDMX Constructor is to enable users to generate and edit SDMX artefacts in a user-friendly environment. This includes creating and editing code lists, concept schemes, data structures, data flows and content constraints. SDMX Constructor provides a user-friendly interface that makes it easy for users to create and modify these artefacts without needing to be an expert in SDMX.

Finally, SDMX Constructor supports data availability and access through online data portals. By using SDMX Constructor, data providers can ensure that their data is available and accessible through online data portals that are built on SDMX standards. This allows data users to easily access and use the data they need for their research, analysis, and other activities.

1.1 A section

All chapter sections start with a second-level (##) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a `{.unnumbered}` or the shorter `{-}` at the end of the heading, like in this section.

Chapter 2

Getting Started

Cross-references make it easier for your readers to find and link to elements in your book.

2.1   System requirements

2.2   Installation

2.3   Software updates

2.4   Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
   - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world = # Hello world {#hello-world}`.
   - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter **??**.
   - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

2.5   Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 2.1.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```
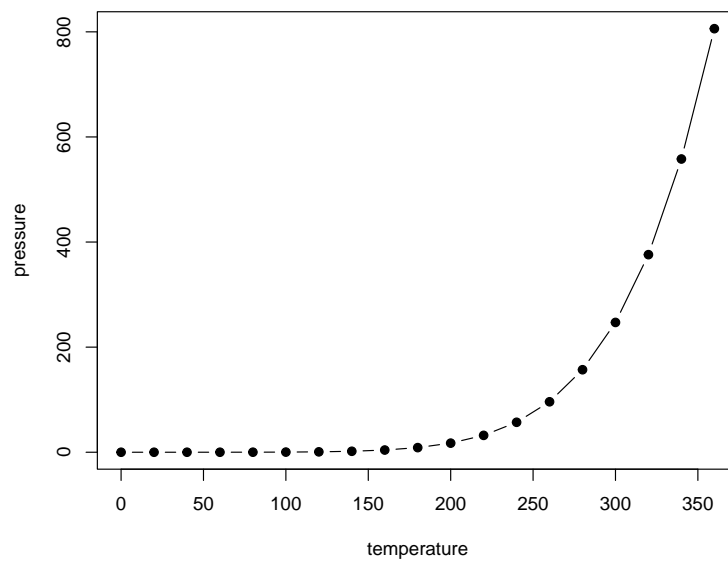

Figure 2.1: Here is a nice figure!

Don't miss Table 2.1.

```
knitr::kable(
  head(pressure, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

Table 2.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Chapter 3

User Interface

3.1 Interface overview and navigation

3.2 Input and output methods

3.3 Translation

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 4

Using SDMX Constructor

4.1   Accessing SDMX artefacts from SDMX registries

4.1.1   Default SDMX registries

4.2   Creating SDMX artefacts from scratch

4.2.1   Setting up a registry as a local folder

4.2.2   Preparing inputs

4.2.3   Creating AgencyScheme

4.2.4   Creating ConceptScheme & Codelist

4.2.5   Creating DSD, Dataflow, ContentConstraint and CategoryScheme

4.3   Working with .Stat Suite

4.3.1   Uploading XML file to the Data Lifecycle Manager (DLM)

4.3.2   Connect to a new SDMX registry

4.4   Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

---

[1]This is a footnote.

4.5 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

Chapter 5

Special Topics

5.1   Annotations

5.2   Table Modeller

5.3   Translations using Google API/DeepL

5.4   Equations

Here is an equation.

$$f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

5.5   Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 5.1.

Theorem 5.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

5.6   Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 6

Sharing your book

6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book-all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL https://CRAN.R-project.org/package=bookdown. R package version 0.32.