The Stata Journal (2018)
18, Number 4, pp. 863–870

sdmxuse: Command to import data from statistical agencies using the SDMX standard

Sébastien Fontenay
Université catholique de Louvain
Louvain-la-Neuve, Belgium
sebastien.fontenay@uclouvain.be

Abstract. In this article, I present the command **sdmxuse**, which allows users to download and import statistical data from international organizations using the Statistical Data and Metadata eXchange standard (SDMX). The data structure is reviewed to show how users can send specific queries and import only the required time series.

Keywords: dm0097, sdmxuse, data import, data management, European Central Bank, Eurostat, International Monetary Fund, Organisation for Economic Co-operation and Development, United Nations, World Bank

1 Introduction

In 2001, statistical agencies around the world launched an initiative to facilitate the exchange of statistical data. The organizations involved were the Bank for International Settlements, the European Central Bank, Eurostat, the International Monetary Fund, the Organisation for Economic Co-operation and Development (OECD), the United Nations, and the World Bank. To this end, they developed an International Organization for Standardization (ISO) standard called the Statistical Data and Metadata eXchange (SDMX).[1]

The SDMX initiative revolved around three axes (SDMX 2017). First, it set technical standards for compiling statistical data. Second, it sought to harmonize terminology by developing statistical guidelines for most frequently used concepts (for example, seasonal or price adjustments). This is particularly important for metadata that provide information about other data.[2] Third, it promoted tools to deploy web services[3] that facilitate the access to data and metadata.

The primary goal of the participating organizations was to foster data sharing among themselves. The web services allowed agencies to "pull" data from another provider. Dissemination of data to final users was somehow secondary. But the web services are accessible to the public, and users can download a dataset by sending a request to the

[1] With certification number ISO 17369:2013.

[2] For example, the data point 9.9 for June 2016 is not useful unless it comes with the metadata explaining that it is a measure of the total unemployment rate (according to the International Labour Organization definition) for France, after seasonal adjustment but no calendar adjustment.

[3] Web services are systems designed to support machine-to-machine communication, specifically for transferring machine-readable file formats such as Extensible Markup Language (XML).

URL of the service. The result is a structured SDMX-ML file that should be converted into a human-readable format.

The sdmxuse command makes it easy to download and import the file directly from within Stata. It might prove useful for researchers who need frequently updated time series and wish to automate the downloading and formatting process. One can think of modern methods for forecasting economic series that exploit many predictors—often hundreds of time series that could be used as soon as they are released.

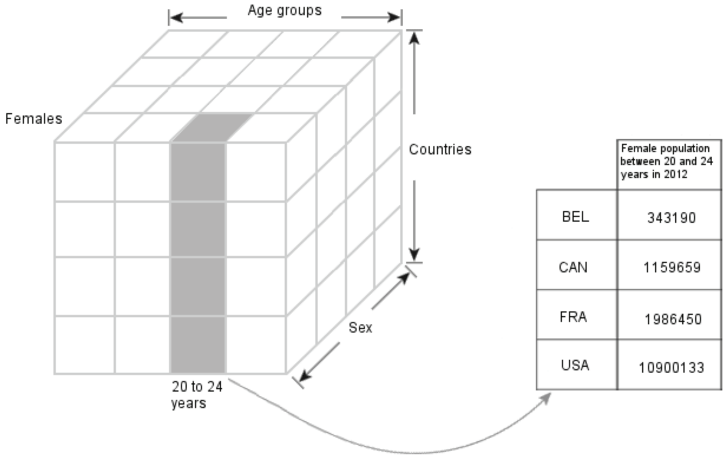
2 Data structure

Each statistical agency has its own repository with hundreds of publicly available datasets. Users can find a particular dataset by looking into the "data flow definition" that contains a description of the data together with the identifier of the dataset (for example, the dataset RPOP from the OECD repository contains demographic data). But the datasets are often quite large, whereas users may be seeking to download only a few series (for example, population data with specific characteristics, such as females aged 20−24 years). This is why the statistical agencies have implemented a genuine database service that is capable of processing specific queries. It is therefore important to understand the data structure to be able to download only the series of interest and save processing time.

2.1 Multidimensional cube

The datasets are organized along dimensions. It is useful to think about this organization as a cube, a structure commonly used for data warehousing. Each side of the cube corresponds to a dimension, and the combination of the values for all dimensions gives a unique identifier (called a key) for each cell of the cube. Figure 1 illustrates the cube structure of the dataset RPOP with three dimensions: age, gender, and country.

The main benefit of organizing a dataset in a cube structure is that it allows users to retrieve only a subset of the data contained in a few cells. In figure 1, we could "slice" the cube by processing a specific query to obtain only the total female population aged between 20 and 24 years in 4 OECD countries.



	Female population between 20 and 24 years in 2012
BEL	343190
CAN	1159659
FRA	1986450
USA	10900133

Figure 1. Slicing a data cube

The total number of cells of the cube in this example is actually 6,498, corresponding to all possible crossings of the variables: age groups (38) × countries (57) × sex (3). But new dimensions could be added (for example, educational attainment or employment status). In fact, even though it is called a cube, it is actually multidimensional (that is, it allows more than three dimensions).

2.2 The data structure definition file

Following the explanation above, we understand that users must identify the dimensions of a given dataset before being able to make a specific query. To this end, the SDMX standard provides structural metadata describing the organization of a dataset in the form of a data structure definition (DSD) file. The latter gives information about the number of dimensions (called "concepts") of the dataset, the order of the dimensions,[4] and the values (called "codes") for each dimension.

In the example of the `RPOP` dataset, the DSD file reveals that it is organized along four concepts, namely, `COUNTRY`, `DAGEGR`, `DSEX`, and `DSTATUS`. Each of these concepts includes several codes. For instance, the concept `COUNTRY` stores the country codes, while the concept `DAGEGR` identifies the age groups.

However, note that the DSD does not make any guarantees about the presence of data, and sometimes the dataset may be a "sparse cube" (that is, there may not be data for every possible key permutation). For instance, in the `RPOP` dataset, there are no data available for boys and girls between 10 and 14 years old in Sri Lanka.

[4] Dimensions must be specified in a certain order when sending the request to the provider's web service.

2.3 Processing raw data

The result of the query will be an SDMX-ML file, which is built around XML syntax. Box 1 below shows the raw data for a request of two series for total female population between 20 and 24 years old in Belgium and the United States.[5] The series is saved to a text file called `tmp_sdmxfile.txt` using the command `copy`. The series is a string of characters with text elements (data and metadata) and structural markers (called tags). The tags are encapsulated between lower-than and greater-than symbols to distinguish them from the content.[6] To process the file in Stata, you need to distinguish two types of tags:

- `<SeriesKey>`, which contains the identification key of a given series; and
- `<Obs>`, which contains the observations' values.

```
<Series> <SeriesKey><Value concept="COUNTRY" value="BEL"/><Value
concept="DAGEGR" value="2024"/><Value concept="DSEX" value="2"/><Value
concept="DSTATUS" value="90"/></SeriesKey> <Obs><Time>2010</Time>
<ObsValue value="329176"/></Obs> <Obs><Time>2011</Time> <ObsValue
value="337483"/></Obs> <Obs><Time>2012</Time> <ObsValue
value="343190"/></Obs> </Series>

<Series> <SeriesKey><Value concept="COUNTRY" value="USA"/><Value
concept="DAGEGR" value="2024"/><Value concept="DSEX" value="2"/><Value
concept="DSTATUS" value="90"/></SeriesKey> <Obs><Time>2010</Time>
<ObsValue value="10478470"/></Obs> <Obs><Time>2011</Time> <ObsValue
value="10680913"/></Obs> <Obs><Time>2012</Time> <ObsValue
value="10900133"/></Obs> </Series>
```

Box 1: Raw data

We observe from the SDMX-ML output in box 1 that under each `<SeriesKey>`, we have not one but several observations' values that are indexed by another dimension: time. In fact, following the analogy of the cube, we understand that each cell stores a time series and not a single number.

Before we import the text file in Stata, we add a carriage return to the `<SeriesKey>` and `<Obs>` tags (using the command `filefilter`). Then, we separate the data and metadata from the structural markers. This is facilitated by the use of the package `moss`, created by Robert Picard and Nicholas J. Cox (2011), which allows one to find substrings matching complex patterns of text using regular expressions.

Concretely, we first extract the metadata that allow one to identify a particular series and are located between the tags `<SeriesKey>` and `</SeriesKey>`. Then, we create two more variables for the time dimension and the observation's value, which are

[5] The request sent is the following:
https://stats.oecd.org/restsdmx/sdmx.ashx/GetData/RPOP/BEL+USA.2024.2./all?.

[6] There exist so-called start-tags (<*tag*>) and end-tags (</*tag*>).

located between each `<Obs>` and `</Obs>` tags. Here is an example of syntax[7] to extract the different concepts from the `<SeriesKey>` substring:

```
moss var1, match(‘"concept="([a-zA-Z0-9_]+)""’) regex
```

3 The sdmxuse command

The `sdmxuse` command allows for retrieving three types of resources:

- dataflow, which is a complete list of publicly available datasets with their identifiers and a description;
- DSD, which is metadata describing the structure of a dataset (that is, the order of dimensions and the distinct values for each dimension); and
- time-series data.

3.1 Syntax

The syntax varies according to the type of resource one wishes to download:

```
sdmxuse dataflow provider [, clear]
```

```
sdmxuse datastructure provider, dataset(identifier) [clear]
```

```
sdmxuse data provider, dataset(identifier) [clear dimensions(string)
   attributes start(string) end(string) timeseries panel(panelvar)
   mergedsd]
```

The provider acronym should be written in capital letters. Six *provider*s are currently available: European Central Bank (`ECB`), Eurostat (`ESTAT`), International Monetary Fund (`IMF`), Organisation for Economic Co-operation and Development (`OECD`), United Nations Statistics Division (`UNSD`), and World Bank (`WB`).

[7] The exact syntax varies from one provider to another because they are using different versions of the SDMX standard, version 2.0 or the latest version 2.1.

3.2 Options

The following options are available when downloading data:

`dataset(`*identifier*`)` can be found by first downloading `dataflow`. You can also find the dataset identifier on some providers' websites.[8] Users should realize that datasets' identifiers are case sensitive and may contain underscores or other symbols. The option `dataset()` is required for `sdmxuse datastructure` and `sdmxuse data`.

`clear` specifies that data in memory be replaced with the imported data.

`dimensions(`*string*`)` allows one to customize requests for data. Time series can be retrieved based on the value they take for each dimension. Dimensions should be separated by a dot (`.`) character and must respect the order specified in the DSD. A dimension can be left empty if all values are requested. Multiple values for a dimension are separated by a plus (`+`) sign.

`attributes` downloads attributes that give additional information about the series or the observations but do not affect the dataset structure itself (for example, observations' flags).

`start(`*string*`)` defines the start period. You can specify the exact value (for example, `2010-01`) or just the year (for example, `2010`).

`end(`*string*`)` defines the end period.

`timeseries` reshapes the dataset so that each series is stored in a single variable. Variables' names are made of the values of the series for each dimension.

`panel(`*panelvar*`)` reshapes the dataset into a panel. *panelvar* will often be the geographical dimension.

`mergedsd` merges the data (time series) and the DSD—particularly useful when dimensions' codes are not transparent (for example, `HRV` is the ISO alpha-3 code for Croatia).

3.3 Walk-through

The example below uses `sdmxuse` to download and import population data in OECD countries.

Step 1: Find all publicly available datasets from *provider* `OECD`, and search for those whose description contains the word `population`.

```
. sdmxuse dataflow OECD, clear
. list if regexm(lower(dataflow_description), "population"), noobs
  (output omitted)
```

[8] Eurostat, for instance, refers to the dataset identifier as "product code", indicated between brackets after the titles in the navigation tree: http://ec.europa.eu/eurostat/data/database.

Step 2: Find the DSD of the RPOP dataset. The command also returns a message to indicate the names and order of the dimensions.

```
. sdmxuse datastructure OECD, clear dataset(RPOP)
Order of dimensions: (COUNTRY.DAGEGR.DSEX.DSTATUS)
```

Step 3: Customize the request to obtain total population aged between 20 and 24 years. We leave the first and last dimensions empty, meaning that we want all values of those dimensions.

```
. sdmxuse data OECD, clear dataset(RPOP) dimensions(.2024.90.)
56 serie(s) imported
```

We can ask to merge the data with the DSD to get the country label stored in another variable.

```
. sdmxuse data OECD, clear dataset(RPOP) dimensions(.2024.90.) mergedsd
56 serie(s) imported
```

We can reshape the dataset to get all time series in separate variables or build a panel dataset. Here we ask series for men and women by specifying the values 1+2 in the dimension DSEX.

```
. sdmxuse data OECD, clear dataset(RPOP) dimensions(.2024.1+2.) timeseries
112 serie(s) imported
. sdmxuse data OECD, clear dataset(RPOP) dimensions(.2024.1+2.) panel(COUNTRY)
112 serie(s) imported
```

4 Conclusion

In this article, I presented the SDMX initiative and showed how the command sdmxuse can be used to download and import data directly from within Stata. I also reviewed the structure of the datasets in SDMX format to illustrate how users can build specific queries.

Some initiatives have already been implemented to facilitate the use of SDMX data for external users, but they rely on the Java programming language. Formatting the data directly within Stata has proved to be quicker for large datasets, but it also offers a simpler way for users to address potential bugs by amending the ado-file. The last argument is particularly important for a standard that is evolving.

5 Acknowledgments

The author thanks the participants of the Stata Users Group meetings in Brussels in 2016, London in 2016, and Paris in 2017 for their comments. He is also thankful to Evaristo Vidal Otero, Jan Helmdag, and Samuel Pinto Ribeiro for their suggestions for improvement. Finally, he is grateful to Robert Picard and Nicholas J. Cox (2011) for allowing him to reproduce their code of the package moss to deal with multiple occurrences of substrings.

6 References

Picard, R., and N. J. Cox. 2011. moss: Stata module to find multiple occurrences of substrings. Statistical Software Components S457261, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457261.html.

SDMX. 2017. Learning about SDMX basics. https://sdmx.org/?page_id=2555.

About the author

Sébastien Fontenay is a statistical consultant at the Université catholique de Louvain (Institute for Multidisciplinary Research in Quantitative Modelling and Analysis) and PhD candidate in economics at the Université Libre de Bruxelles (Département d'Économie Appliquée).