DEMOGRAPHIC RESEARCH
A peer-reviewed, open-access journal of population sciences

DEMOGRAPHIC RESEARCH

VOLUME 49, ARTICLE 40, PAGES 1117–1162
PUBLISHED 13 DECEMBER 2023
https://www.demographic-research.org/Volumes/Vol49/40/
DOI: 10.4054/DemRes.2023.49.40

Research Material

Programmatic access to open statistical data for population studies: The SDMX standard

Frans Willekens

Contents

Demographic Research: Volume 49, Article 40
Research Material

Programmatic access to open statistical data for population studies: The SDMX standard

Frans Willekens[1]

Abstract

BACKGROUND
The public sector publishes vast amounts of open data and metadata. APIs (application programming interfaces) are transforming the way data are collected, documented, and disseminated. The transformation is slow, however, due to differences in communication protocol, data definition, and data format. The development is of particular relevance to demography, being a data-intensive science. It paves the way to the automation of data acquisition and the integration of data acquisition and data analysis. Together with the parallel development of literate programming, which allows the integration of text and computer code in a single document, programmatic access to data makes workflows transparent, verifiable, and easy to replicate by others. The Statistical Data and Metadata Exchange (SDMX) standard, which has emerged as a popular option for data and metadata exchange, makes finding and retrieving data and metadata easy and swift. Query strings form URLs with a standardised syntax.

OBJECTIVE
The aim of this paper is to describe the SDMX standard and demonstrate its benefits to our profession by retrieving demographic data and the associated metadata from online databases disseminated by a variety of data providers. The software environment used is R.

CONTRIBUTION
This is the first review of the SDMX standard aimed at the study of population. The paper includes the R code to access databases and download data and metadata. The paper includes several hyperlinks to relevant documents issued by data providers, giving readers immediate access to the referenced material.

[1] Netherlands Interdisciplinary Demographic Institute (NIDI-KNAW)/University of Groningen, the Netherlands. Email: willekens@nidi.nl.

1. Introduction

Five developments are making programmatic access to data the new norm. The first is the rise of open data. For open data to be useful in the digital age, they should be FAIR (findable, accessible, interoperable, and reusable).[2] The second is the rise of data portals and in particular application programming interfaces, or APIs. The third is the rise of instruments in R, Python, and other programming languages to automate data access and to integrate data retrieval and data analysis. The fourth is the emergence of common standards for data exchange: the Statistical Data and Metadata Exchange (SDMX) standard for aggregate data, such as counts and rates, and the Data Documentation Initiative (DDI) standard for microdata. With a common standard, users can use a single procedure to access diverse databases. The fifth is the growing interest in reproducible research. Using computer code (script) to retrieve data makes data sources fully transparent and allows researchers to easily repeat data retrieval and verify sources of data. By accessing the metadata, researchers can verify the interpretation of the data. In addition to these five developments, other developments are in favour of programmatic data access in population studies. They include the increased computer skills and programming capabilities of demographers, the rise of computational demography, and the growing interest in comparative population studies, which requires harmonised data and extensive data documentation (metadata) in a uniform format.

The harmonisation of the format of statistical data across domains and a uniform data description have been goals for decades. The establishment of the European Monetary Union in 1992 triggered the development of a standard to facilitate the exchange of statistical data with Eurostat, the statistical agency of the European Union and a Directorate General of the European Commission. In 2001, seven international organisations,[3] including Eurostat, launched the initiative to develop a global standard for an API-based exchange of statistical data and metadata. A standard is a set of rules to define, describe, and transmit data, applied by all actors endorsing the standard. The

[2] Data are findable when they have a unique and persistent identifier, are described by sufficiently rich metadata, and are registered or indexed in a searchable resource. Data are accessible when they can be obtained by humans and machines through a well-defined and universal protocol. Concepts used to describe an object or artefact are interoperable if they can be used in different domains. Concepts need to be given a precise and correct meaning. If the semantics is right, data convey meaningful information (Gillman 2023). The precise meaning of concepts are provided in vocabularies. Data are interoperable if they are described by concepts that are interoperable. Data are reusable if they can be used by others, and the information the user needs to use the data correctly is included in the metadata. For an introduction to FAIR data, see Wilkinson et al. (2016) and European Commission Expert Group on FAIR Data (2018). Recently, the International Union for the Scientific Study of Population (IUSSP) initiated the development of FAIR vocabularies on demography (IUSSP – CODATA Working Group on FAIR Vocabularies 2023).

[3] The OECD, the World Bank, Eurostat, the International Monetary Fund (IMF), the United Nations Statistics Division (UNSD), and the International Labour Organization (ILO).

initiative resulted in the SDMX standard. The organisations participating in the initiative formed what became known as the SDMX community. The standard was released in 2004. In 2005, the International Organization for Standardization (ISO) approved SDMX as the international standard (ISO 17369) (updated in 2013) for data and metadata exchange (see here and here). The ISO certification was an important landmark in the propagation of the SDMX standard (Stahl and Staab 2018: 77). In January 2021, the SDMX community launched version 3.0 of the SDMX standard.

SDMX is not the only standard for the exchange of statistical data, although it is the only ISO standard. Statistics Sweden developed PxWeb (see Section 2.1), and Google developed the Dataset Publishing Language (DSPL) as part of the Google Public Data Explorer. The three standards use the same data model to structure statistical data – namely, the data cube or multidimensional table. The data cube is described in Section 2.

The initiative for a common standard acted as a catalyst for the harmonisation of statistical concepts and terminology and the harmonisation of the documentation of statistical data. These activities are essential to interpret data and to turn data into information. They also play a central role in data transparency (Gillman 2023). Demographic research, and in particular comparative research, is often handicapped by data inadequacies. Many attempts to harmonise data fail because data are inadequately documented.

The structure of the paper is as follows. Section 2 provides some background information that should help situate the transition to the SDMX standard in a wider context. It also describes the SDMX data model, which is the theoretical foundation of the SDMX standard. The data model is a standardised description of the structure of the data. Knowledge of the data structure is not needed when an entire database is downloaded (bulk download), but it is essential to retrieve subsets of data. Section 3 covers API-based data access and retrieval. It consists of three subsections. The first shows how to obtain information on data providers and their data catalogues. The structure of data queries is the subject of the second subsection. Data queries are URLs with a fixed format (syntax). The construction of data queries is the subject of the third subsection. The section includes the R code to access data and metadata of international organisations (OECD, Eurostat, International Labour Organization (ILO), United Nations Statistics Division (UNSD), World Bank, and Asia Development Bank) and national statistical offices (Statistics Canada, Statistics Lithuania, and Australian Bureau of Statistics). The section demonstrates the advantage of having a common standard for data and metadata exchange. Section 4 concludes the paper.

Throughout the paper, R (version 4.2.1, 2022) is used (R Core Team 2022). The R package rsdmx (version 0.6–3) by Blondel (2015, 2023a, 2023b) is used to retrieve data and metadata. Two other packages are used: jsonlite (Ooms 2014) to retrieve data in the JSON format and httr (Wickham 2023) to retrieve data in the HTML format. These packages are used mainly to retrieve data catalogues, which are not available in the SDMX format. The JSON data format is also used by the Population Division of the United Nations, which is a significant data provider for demographers. Two R packages are used to import images and produce publishable tables: knitr and kableExtra. Appendix A lists the packages used in this paper and shows how to install and load the packages. The paper was prepared using R Markdown in RStudio. For more details, see Appendix B. All packages used in this paper are freely available on the Comprehensive R Archive Network (CRAN). The R Markdown file, including the bibliography, and the R code have been uploaded on the Demographic Research website as replicable material and to the Zenodo repository (DOI: https://zenodo.org/records/10221972).

2. The SDMX standard

2.1 Background

The exchange of data between computers over the internet requires a reliable connection between the client computer and the server. Once the connection is established, the computers should be able to communicate, transmit data, and make sense of the data transmitted. To meet these requirements, developers introduced protocols, which are systems of rules. The first set of rules governs the communication between computers, the second specifies the IT architecture used to implement web services and enable data exchange, the third specifies the format in which the data are sent over the internet, and the fourth describes the data and metadata to be transmitted. A rule set is identified by a name. This section lists the rule sets by name and gives their main characteristics. More detail is given in Appendix C, which offers a brief technical introduction to APIs.

The most used communication protocol is hypertext transfer protocol (HTTP). It is used by web browsers for communicating with web servers. The two leading technologies being used to implement web services and exchange messages are the representational state transfer (REST) architecture and simple object access protocol (SOAP). REST is simpler than SOAP, which is used when a high level of security is required. HTTP sends data in Hypertext Markup Language (HTML) format. Web APIs usually use another format to transmit data: the JavaScript Object Notation (JSON) or Extensible Markup Language (XML) format. Both are text files that can be read by humans and machines. JSON's code looks like JavaScript. XML, similar to HTML, contains markup tags. But unlike HTML, where markup tags describe the structure of a page, in XML the markup tags describe the meaning of the data contained in a file. The XML format helps interpret the data sent over the internet.

The SDMX standard has these four components. It uses the HTTP communication protocol. The IT architecture is generally REST. Data are transmitted in XML. SDMX advocates a common syntax and semantics valid across domains. APIs that use the REST architecture and the XML format and comply with the SDMX guidelines to describe the data are said to be SDMX-based RESTful APIs or SDMX REST APIs.

Data queries include information on the data requested, their location (web address), the authorisation to access the data, and the format in which the data should be sent over the internet. Queries must also satisfy the protocol that governs the communication between computers. The uniform resource locator (URL) is the common format to access a web address. The specification of a URL that conforms to the required URL syntax is often a challenge. The CRAN repository of R packages includes packages with functions that convert a request for data into a URL with a correct syntax required by the targeted API.

The implementation of the SDMX standard is a work in progress. Providers of statistical data are in different stages of implementation. Some organisations (e.g., OECD and Eurostat) make most of their data available via an SDMX-based API. The Eurostat API is also used by national statistical offices to transmit data to Eurostat. In the European Statistical System, more than 40% of the data transmitted to Eurostat follow the SDMX standard (see here). The United Nations makes many data available online, but only a fraction can be accessed using SDMX-API queries. The UN promotes the SDMX format for the exchange of data on progress made towards the Sustainable Development Goals (SDGs). The International Monetary Fund (IMF) requests countries to use the SDMX standard or a simplified version to submit economic and financial statistics (ECOFIN) to the IMF.[4] Close to 100 countries use SDMX or ECOFIN to submit their data to the IMF.[5] The countries include the United States (Department of the Treasury) and Germany (Bundesbank), two countries in which statistical offices have not yet introduced the SDMX standard. The Asia Development Bank has adopted the SDMX standard to provide access to its Key Indicators Database. In cooperation with the IMF, the African Development Bank is developing a road map for the implementation of the SDMX standard as part of the African Information Highway initiative (see here and here).

An increasing number of national statistical offices make data available via APIs and adopt the SDMX standard. In 2022, the Italian National Institute of Statistics (Istat) started sharing its data through an SDMX-based system: IstatData (see also IstatData). In the same year, the Australian Bureau of Statistics (ABS) introduced its data API (beta release) as part of the part of the ABS Data Explorer ABS.Stat. The ABS data API uses the REST protocol and is compliant with the SDMX standard. The US Census Bureau

[4] For an introduction to the ECOFIN standard see here.

[5] The data submitted are publicly available and can be downloaded using the procedures described in this paper.

makes data available via APIs but uses a different standard (see here).[6] The UK Office for National Statistics, which uses its own standard as well, is investigating the adoption of the SDMX standard (see here). The PxWeb standard, developed by Statistics Sweden, is used in the Nordic countries and statistical offices in more than 90 countries around the globe (see Appendix C). Version 2.0 of the PxWeb API, introduced in autumn 2023, conforms better to the RESTful design (see here).

The implementation of the SDMX standard by providers of open data is a work in progress. It is also a learning process because ensuring that data documentation follows the strict rules so the data and metadata are machine readable is a challenge. The transition from provider-specific syntaxes to a common global standard is bound to face obstacles. Contributing factors are (a) the common standard is evolving, and backward compatibility is not guaranteed (see here); and (b) all organisations that adopt the common standard do not implement the standard in exactly the same way.

2.2 The SDMX data model

In order to make sense of data, we need to know what they represent. For example, on its own, the number 35 is meaningless, but if we know it is age on 1 January 2023, it starts making sense. The value or metric 17.850 is obscure, but if we know it is the number of residents of the Netherlands on 1 May 2023, the number is clearer, but key information is still missing. Additional information needed to correctly interpret the figure is that (a) the figure is for males and females combined and all ages combined; (b) the population is expressed in millions; (c) the figure is an estimate; (c) the source is StatLine, the data portal of Statistics Netherlands, more specifically this table; and (e) the number is copied on 18 June 2023. The latter information is useful because data are occasionally updated.

The SDMX data model provides a framework to accommodate this description in a way both humans and computers can understand. It includes (a) a conceptual framework, consisting of a definition of the concepts used to describe the data; (b) a data structure, which enables the location of selected data in a database; and (c) additional information on the data deemed relevant to interpret the data and assess their validity. The central component of the data model is the data structure. In SDMX, data are structured as

[6] In 2022, the Consensus Study Panel on Transparency and Reproducibility of Federal Statistics (United States) stated that "adoption of standards such as DDI or SDMX is worthwhile, not only because transparency and reproducibility are inherently desirable as qualities that enhance the trustworthiness and reliability of statistical products and processes, but also because these standards were developed with the intention of helping national statistical agencies do more with less in a way that is interoperable and mutually intelligible" (National Academies of Sciences, Medicine et al. 2022: 129).

multidimensional tables. In SDMX, they are referred to as 'data cubes.'[7] The dimensions of the data cube describe the structure of the dataset. The structure of a dataset permits the location of each data point in the dataset. The value 17.850 is a marginal total in a country-sex-age table. One of two types of concepts the SDMX standard distinguishes is dimension. The other concept is attribute. Attributes give relevant information on a data point or set of data points, not provided by their location in the data cube. These concepts to describe the data (descriptor concepts) and the other standardised concepts used in SDMX ensure a uniform language to describe statistical data. For the standardised terminology used in the SDMX language, see this glossary.

In SDMX, a data point can be a numeric value or a character string. The content of a data point is called measure. The location of a data point in the multidimensional table is identified by dimensions. Place of residence, sex, and age are dimensions. A dimension has a limited number of possible categories (i.e., the cross-classified variables are discrete variables). Standardised concepts and terms are used to denote categories. They are listed in codelists, one for each dimension. For the official SDMX codelists, see here. Consider age: Age may be measured in completed years (completed age) or as age reached (determined by calendar date and year of birth). Eurostat uses these two concepts as categories of the dimension 'age definition' in the SDMX data cube. OECD does not make the distinction but uses current age. The age dimension has many possible categories because ages may be grouped in a variety of ways. Common age intervals are one or five years in length. Broad age classes have varying length. To accommodate all these possibilities, the SDMX standard allows many age categories, several of which are overlapping. Producers and users of statistical data select a subset of categories. 'Total' (i.e., all ages combined) and 65 and over are categories included in the codelist. The dimension 'place of residence' or 'geographical location' is another dimension with many possible categories. They include countries and geographical areas at different scale. Eurostat uses a hierarchical system, known as Nomenclature of Territorial Units for Statistics (NUTS) classification, for dividing up the European territory. The geographical units considered by data providers are generally restricted to a territory and do not cover the entire world. For instance, the OECD gives immigration statistics for OECD countries only, but the list of countries of origin includes all countries of the world. The dimensions determine the structure of the multidimensional table, while the codelists determine the size of the table. Relevant information on a data point beyond its value and location in a contingency table is included as attributes of the data point. Is the value observed or estimated? Other relevant information includes the unit of measure (year or month), the data source (census, population register, survey, digital footprints), the name

[7] The SDMX data cube is compatible with the data cube vocabulary used in the Resource Description Framework, a standard designed by the World Wide Web Consortium (W3C) for the exchange of linked (multidimensional) data on the web (World Wide Web Consortium 2014).

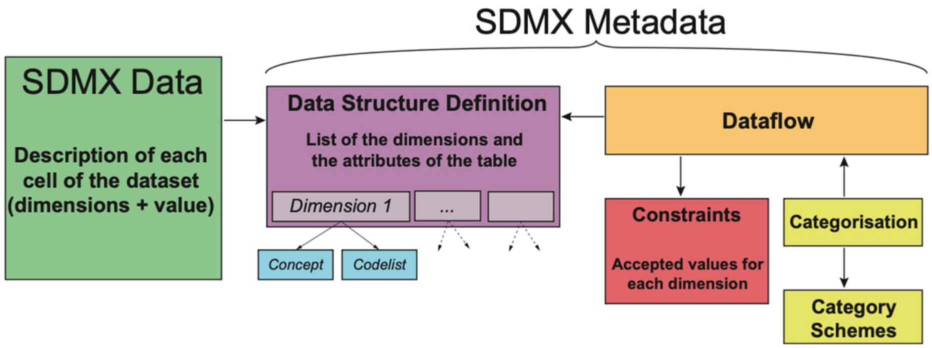
of the agency maintaining the data, the dates at which values of data points were observed or updated, the conditions under which the data are provided. That information is essential to interpret the value of a data point. If the value is estimated, the estimation method may be added. The metadata offer a description of a data point for interpretation and validation. In SDMX, a distinction is made between structural metadata and reference metadata. Structural metadata identify data and the dimensions of the cross-tabulation. Reference metadata give information on the content and the quality of the data. For an example of metadata, see here and here.

Datasets may pertain to different domains. Since most concepts are domain specific, the harmonisation of concepts across domains is a challenge. For instance, biologists and demographers have defined fertility and fecundity differently for a long time. The harmonisation of concepts is a necessary condition for the harmonisation of data. For many years, demographers tried to harmonise the concepts they use. It resulted in (multilingual) controlled vocabularies (IUSSP – CODATA Working Group on FAIR Vocabularies 2023).[8] In SDMX, the concepts used to describe measures, dimensions, and attributes are part of the data model and represented by a data structure definition (DSD). The SDMX community is developing a global repository for structural metadata and registered data sources that comply with the SDMX standard. It is the SDMX Global Registry. The registry supports the implementation of SDMX by making its metadata material, DSDs, and related artefacts (concept schemes, metadata structure definitions, code lists, etc.) publicly and centrally available. The ultimate aim is the harmonisation of concepts, methodology, and processes. The IUSSP – CODATA Working Group on FAIR Vocabularies recommends that cross-domain identifiers are FAIR and the SDMX Global Registry serves as a central resource for all vocabulary identifiers (IUSSP – CODATA Working Group on FAIR Vocabularies 2023: 42). The replacement of local concepts (and registries) by global concepts and a global registry is, however, a slow process with many hurdles.

Figure 1 summarises the SDMX standard. The central component is the data structure and its description, the DSD. Metadata include structural metadata on the location of data points and reference metadata with additional information on the data points or the entire dataset. To describe data, concepts included in a controlled vocabulary are used. Permitted categories of dimensions are stored in codelists. The structure of a particular dataset and the concepts used to describe the data are given in a dataflow. A dataflow is more specific than a DSD because it is for one particular subject-matter domain. Therefore, a dataflow includes a subset of the codes included in the DSD.

[8] A controlled vocabulary is a standardised set of words and phrases used to organise and describe knowledge (IUSSP – CODATA Working Group on FAIR Vocabularies 2023: 15).

Figure 1: SDMX data organisation


SDMX Metadata
SDMX Data
Description of each cell of the dataset (dimensions + value)
Data Structure Definition
List of the dimensions and the attributes of the table
Dimension 1
...
Concept
Codelist
Dataflow
Constraints
Accepted values for each dimension
Categorisation
Category Schemes

Source: Thiry, Manolescu, and Liberti 2020.

The SDMX data model is endorsed by a large number of data providers. That broad support base has several advantages. A first is an easier exchange of statistical data over the internet. A second is harmonisation of data, which has been a challenge for decades and continues to be a challenge. The concerted effort to produce a common conceptual framework for the description of statistical data contributes to the harmonisation of statistical data across domains. Demographers participate in the concerted effort by providing concepts for the description of population processes and controlled vocabularies that are valid across domains (IUSSP – CODATA Working Group on FAIR Vocabularies 2023). The SDMX data model is of particular interest to social scientists. The data structure (multidimensional table) is a popular data structure in the social sciences and the basis for discrete multivariate analysis or contingency table analysis (Agresti 2013; Bishop, Fienberg, and Holland 1975).

2.3 Structure of data queries: URL syntax

In SDMX, a query is a request for resources, which is a container concept. It refers to data, data structure, dataflow, and codelist. A request includes information on the data provider, the resource requested, the dataset of interest, and, for data, the location of the requested data points in the dataset. A data query is a character string – more particularly, a URL. The SDMX URL consists of several components in a fixed format. The components are the communication protocol, the web service entry point, the resource requested, the subset of data or metadata requested by the user, and the time series requested. The components are described in some detail below. Some examples are included.

a. *protocol*: http or https. The protocol https is an extension of http. It uses encryption for secure communication.

b. *wsEntryPoint*: The web service (ws) entry point or host name is the entry point to the web address where the requested data or metadata can be found. Table 1 gives the entry points of the SDMX-based APIs of a selection of data providers. Note that an entry point is part of a URL, not a complete URL.

The entry point of a data provider can be obtained by calling the function findSDMXServiceProvider() of the rsdmx package. The following code snippet returns the entry points of the SDMX RESTful APIs of the OECD and Eurostat (ESTAT). Before you can call the function findSDMXServiceProvider(), you need to install the package rsdmx. Appendix A shows how to install and load packages used in this paper.

```
# wsEntryPoint of OECD
oecd <- rsdmx::findSDMXServiceProvider("OECD")
entrypoint_oecd <- oecd@builder@regUrl
# wsEntryPoint of Eurostat (ESTAT)
eu <- rsdmx::findSDMXServiceProvider("ESTAT")
entrypoint_eurostat <- eu@builder@regUrl
```

c. *resources*: Four resources are distinguished:

- Datastructure: The datastructure describes the structure of a dataset. A complete description of a dataset includes its structure, characteristics of the dataset (dimensions, attributes, measures), and codelists. That information is given in the DSD.
- Dataflow: A dataflow gives information on the data structure of a particular dataset. A dataflow is identified by DF_<DSD_ID> (i.e., the prefix DF, followed by the ID of the dataset). For instance, the data flow DF_MIGR_IMM8 gives information on Eurostat’s MIGR_IMM8 dataset. The information includes the agency maintaining the dataset; the code of the dataset (MIGR_IMM8); the version; the name of the dataset (‘Immigration by age and sex’), often in multiple languages; and the date at which the most recent version of the dataset was prepared.
- Codelist: The codelist gives, for each dimension, the possible categories. A codelist is identified by CL_<CLY_CODE> (i.e., the prefix CL, followed by the code of the dataset and the code of the dimension for which the categories are

requested). For instance, CL_MIG_CO2 refers to a codelist of the dimension CO2 (country of birth) of the OECD dataset MIG.

- Conceptscheme: A conceptscheme is a container of concepts.
- Data: This refers to data to be retrieved.

d. *flowRef*: This component of the URL refers to the resource to be returned (i.e., data structure, dataflow, codelist, or data). The syntax has three parts: (a) agency_id, which is the identifier of the agency maintaining the resource; (b) flow_id, which is the identifier of the resource; and (c) the version of the resource. The components are separated by a comma (,) or a slash (/). For example, ILO,MFL_FPOP_SEX_CBR_NB refers to the dataset MFL_FPOP_SEX_CBR_NB maintained by the ILO, and WDI/A.SP_POP_TOTL refers to dataset SP_POP_TOTL included in the World Bank's World Development Indicators. The agency identifier and the dataflow version are sometimes omitted.

e. *key*: A key is a data filter. It gives the location of a data point in a multidimensional table. A series key is a series of keys that gives the location of a series of data points. The location is given by the coordinates on the axes defined by the dimensions of the contingency table. Keys and series keys are character strings. In the character string, the dimensions of the table are given in the order defined by the DSD related to the dataflow, and dimensions are separated by a full stop (.). Wildcarding is supported by omitting the value for the dimension to be wildcarded. The logical 'or' operator is supported using the plus sign (+). Conventionally, the first dimension is the frequency at which data points are observed (e.g., annually, monthly). Let's extract data from dataset B11 of the International Migration Database (MIG) of the OECD, which contains data on the number of persons by nationality, sex, and country of current residence. The dimensions of the data cube are shown in Table 4. The string MEX.B11.TOT.USA requests the extraction of the number of migrants with a Mexican nationality (CO2) to the United States (COU), males and females combined (TOT) in a given year. The following key retrieves the number of migrants with a nationality of Mexico, India, or China to the United States, sexes combined: MEX+IND+CHN.B11.TOT.USA. The number of immigrants in the United States by nationality is retrieved if the series key .B11.TOT.USA is used, and the string .B11.TOT. retrieves the number of migrants by nationality (235 nationalities) and country of current residence (38 OECD countries).

f. *startPeriod* and *endPeriod*: These refer to the starting and ending dates of the period for which observations should be returned. The values should be given according to the syntax defined in ISO 8601 or as SDMX reporting periods. The supported formats are

- – YYYY for annual data (e.g., 2013)
- – YYYY-MM for monthly data (e.g., 2013-01)
- – YYYY-MM-DD for daily data (e.g., 2013-01-01)

For instance, the string ?startTime=2000&endTime=2021 requests the data for the years 2000 to 2021. The question mark says that a specific query is being performed.

Table 1: SDMX-based API entry points

Organisation	Entry point
OECD	https://stats.oecd.org/restsdmx/sdmx.ashx/
Eurostat	https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1
World Bank	http://api.worldbank.org/v2/sdmx/rest/
Asia Development Bank	https://kidb.adb.org/api/v2/sdmx/
United Nations	https://data.un.org/ws/rest/
International Labour Organization	https://www.ilo.org/sdmx/rest/
Statistics Canada	https://www150.statcan.gc.ca/t1/wds/sdmx/statcan/v1/rest/
Statistics Lithuania	https://osp-rs.stat.gov.lt/rest_xml/
Australian Bureau of Statistics	https://api.data.abs.gov.au/

3. API-based data access and retrieval

The section consists of two subsections. The first shows how to retrieve a list of data providers and, for each provider, the catalogue of datasets disseminated. The second shows how to retrieve resources (data structure, dataflow, codelist, and data). The function readSDMX() of the rsdmx package is used to retrieve the requested lists or resources. The rsdmx package is implemented in the object-oriented S4 system of the R language. S3 and S4 are two generations of functional programming in R. S3, the most widely used system, is less formal and simpler. S4 is a formal approach to implementing object-oriented programming. S4 objects have a formally defined class and structure. The structure of an object consists of named components called slots, which are accessed using the subsetting operator @ or the slot() function of the methods package of base R. The function getSlots() of the same methods package returns a description of all slots of a class. For more information, see Wickham (2019). The operator @ is analogous to the $ subsetting operator in S3. The subsetting operator @ should not be confused with the at sign of an e-mail address.

The CRAN repository features about 20,000 contributed packages. Different packages may include functions that have the same name.[9] To ensure that the correct function is used, the R Core Team recommends to explicitly refer to external functions using the syntax package::function(). That recommendation is followed in this paper.

3.1 Data providers and data catalogues

A list of data providers that have implemented the SDMX standard is given here.[10] The list can also be obtained by calling the function getSDMXServiceProviders() of the rsdmx package. The following code retrieves the list of data providers and creates a data frame with, for each provider, the identification code and the name of the provider.

```r
# Retrieve the list of data providers
dp0 <- rsdmx::getSDMXServiceProviders()
dp1 <- sapply(dp0@providers, slot, "agencyId")
dp2 <- sapply(dp0@providers, slot, "name")
# Create a dataframe wit ID and name of provider
dp <- data.frame(id = dp1, name = dp2)
```

A selection of data providers is covered in this paper. Table 2 shows the list and adds the name and the web address of their data portal. The entry points of their SDMX-based APIs are listed in Table 1.

In this section, a uniform method is used to download the data catalogues of the organisations listed in Table 2. After the data catalogues are downloaded, they are merged. The result is a data catalogue that includes datasets from all providers covered in this paper. A most simple method is presented to search the composite data catalogue and identify datasets that are of direct interest to demographers.

Before turning to a selection of data providers, note that the code presented in the paper should connect with the server of the data provider and import the selected data into your R workspace without problems. Occasionally you cannot connect to the server due to server maintenance or other reasons. It is recommended to try again later. You can also test the connection by using the GET() function of the httr package or the readLines function of base R. The two functions are described in Appendix D.

[9] That situation arose in the preparation of this paper: The packages restatapi and eurostat (not used in this paper) have functions with the name get_eurostat_toc().

[10] Note that the list is included in the documentation of sdmx, a Python package (see here and here). The list is not included in the documentation of the rsdmx package.

Table 2: Data portals of organisations covered in this paper

Organisation	Name	URL
OECD	OECD data	https://data.oecd.org/ (phased out)
OECD	OECD.stat	https://stats.oecd.org/ (phased out)
OECD	Explorer	https://data-explorer.oecd.org/ (new)
Eurostat	Data	https://ec.europa.eu/eurostat/web/main/data
ILO	ILOstat	https://ilostat.ilo.org
UNSD	UNdata	https://data.un.org
World Bank	WB Open Data	https://data.worldbank.org
Asia Development Bank	Data Library	https://data.adb.org
Statistics Canada	Data portal	https://www150.statcan.gc.ca/n1/en/type/data?MM=1
Statistics Lithuania	Open data sets	https://open-data-sets-ls-osp-sdg.hub.arcgis.com
Australian Bureau of Statistics	ABS Statistics	https://www.abs.gov.au/statistics

3.1.1 OECD

The OECD databases are listed on the OECD Data portal and the OECD Statistics website. At the beginning of 2024, the two portals will be discontinued and replaced by the OECD Data Explorer. The catalogue of OECD indicators is available here and the list of databases here. The OECD API provides programmatic access to datasets in the catalogue of OECD databases. The data are available in JSON and XML format. An introduction to the OECD SDMX-ML REST API is available here and here. Datasets made available through the API are imported into the R workspace by using the readSDMX() function of the rsdmx package. The function argument is a URL. The first two lines of the following code implement the procedure. The function call readSDMX(url) retrieves selected data specified by URL. The structure of the URL is described in Section 3.2 when the SDMX data model is discussed. The function returns an S4 object with four slots. The function as.data.frame() of base R converts the S4 object into a data frame, which is an S3 object familiar to most readers. To turn the S4 object into a tibble, which is a new version of a data frame and often used in teaching, use the function as_tibble() of the tibble package, which is part of the tidyverse collection of R packages for data science.

```
url <- "https://stats.oecd.org/RestSDMX/sdmx.ashx/GetKeyFamily/all"
d <- rsdmx::readSDMX(url)
# Create a data frame (tabular data)
toc_oecd <- base::as.data.frame(d)
```

The data frame toc_oecd contains identifiers (id) and names (in French and English) of 1,676 datasets (at time of writing 24 September 2023). Some of the entries in the list are databases (collections of datasets). The names of the datasets are given in the character vector toc_oecd$Name.en.

The following code searches for datasets of direct interest to demographers. Datasets with the keywords ‘population,’ ‘migration,’ or ‘immigrant’ in their name are selected. The datasets on migration and immigrants are listed in Table 3. To access the migration databases using the web browser, see here. The MIG database is used later in the paper and includes eight datasets, listed in Table 5.

```
j <- grep(pattern = "Population|population", x = toc_oecd$Name.en)
j <- grep(pattern = "migration|immigrant", x = toc_oecd$Name.en,
         ignore.case = TRUE)
a2 <- toc_oecd[j, c("id", "Name.en")]
z <- knitr::kable(a2,format = "latex",
 caption = "OECD datasets on migration and immigrant populations")
kableExtra::kable_styling(z, full_width = FALSE,
         latex_options = "HOLD_position")
```

Table 3: OECD datasets on migration and immigrant populations

id	Name.en
DIOC_CITIZEN_AGE	Immigrants by citizenship and age
DIOC_SEX_AGE	Immigrants by sex and age
DIOC_OCCUPATION_DET	Immigrants by detailed occupation
DIOC_LFS	Immigrants by labour force status
DIOC_DURATION_STAY	Immigrants by duration of stay
DIOC_OCCUPATION	Immigrants by occupation
DIOC_FIELD_STUDY	Immigrants by field of study
DIOC_SECTOR	Immigrants by sector
HEALTH_WFMI	Health Workforce Migration
MIG	International Migration Database

3.1.2 Eurostat

The Eurostat data portal is located here and the API here. For an introduction to the Eurostat API, see here. The following code retrieves the data catalogue of Eurostat:

```
url <- paste0("https://ec.europa.eu/eurostat/api/dissemination/",
              "sdmx/2.1/dataflow/ESTAT/",
              "all?detail=allstubs")
toc <- rsdmx::readSDMX(url)
toc_eurostat <- as.data.frame(toc)
```

The catalogue lists 7,481 datasets (at time of writing). The names of the datasets are stored in vector toc_eurostat$Name.en. The dataset with migration data, MIGR_IMM8, is used later in this paper.

3.1.3 ILO

ILOSTAT is the ILO's data portal for labour statistics. The portal has a bulk download facility and a SDMX RESTful API. The following code retrieves the catalogue of the 948 datasets ILO makes available through its API.

```
url <- "https://www.ilo.org/sdmx/rest/dataflow/"
d <- rsdmx::readSDMX(url)
toc_ilo <- as.data.frame(d)
toc_ilo$id <- substr(toc_ilo$id,start=4,stop=nchar(toc_ilo$id))
```

Note that the names of the datasets include the prefix DF_, which refers to the dataflow. The last line of the code removes the prefix.

3.1.4 UNSD

The UNdata portal (here) includes a RESTful SDMX API (here). The following code produces a list of 17 databases.

```
url <- "https://data.un.org/WS/rest/dataflow/"
d <- rsdmx::readSDMX(url)
toc_unsd <- as.data.frame(d)
```

A particularly relevant database is the Global Indicators Database, called SDG Harmonized Global Dataflow with identification code DF_SDG_GLH. The UNSD also refers to the database as the SDG API (see here). It includes data on the progress of countries towards the SDGs (see here and here). The code snippet below retrieves the list of 231 SDG indicators, also available here. The default response format of the API is not SDMX but JSON. The function fromJSON() of the jsonlite package (version 1.8.5) is used to import the list as a data frame.

```
url2 <- paste0("https://unstats.un.org/sdgs/UNSDGAPIV5/v1/sdg/",
              "SDMXMetadata/GetSeries")
toc_sdg <- jsonlite::fromJSON(url2)
```

In the object toc_sdg, some indicators are represented by different measures. For instance, the SDG 1.3.1 (proportion of population covered by a social protection programme) is represented by 12 measures. They are obtained as follows:

```
sdg131 <- toc_sdg[which (toc_sdg$indicator=="1.3.1"),]
```

The next code snippet extracts from the 658 indicators in the SDG database population-related datasets (a total of 87) and datasets on population in poverty (a total of 3). Later in the paper, the identification code of the dataset on the population living below the national poverty line is used to download the data.

```
# Extract datasets on population
ipop <- grep("population", toc_sdg$description, ignore.case = TRUE)
list_Pop <- toc_sdg$description[ipop]
# Extract datasets on population in poverty
jj <- grep("poverty", list_Pop, ignore.case = TRUE)
poverty <- toc_sdg[ipop[jj], c(5, 6)]
```

The Population Division of the United Nations maintains its own data portal, which includes a data portal API with a total of 60 indicators. The default response format of the API is JSON. The code to import the list into R is given below. The names of the indicators are stored in the vector toc_unpd$name.

```
url <- "https://population.un.org/dataportalapi/api/v1/indicators"
list <- jsonlite::fromJSON(url) # list object
# Description of the indicators
toc_unpd <- list$data
```

By way of example, we search the UN Population Division databases for expectation of life.

```
keyword <- "expectation of life"
j <- grep(keyword, toc_unpd$description, ignore.case = TRUE)
out_search <- toc_unpd[j, c("id", "name", "shortName", "displayName")]
```

The UN Population Division data portal API does not include the UN World Population Prospects. The 2019 Revision of World Population Prospects (wpp2019) is available on CRAN (here) and the 2022 Revision (wpp2022) is available here and as an R package on GitHub (Ševčíková 2023) (See here).

3.1.5 World Bank

The World Bank (WB) open data webservice is available here. It has a reference to the data catalogue and the data bank, an analysis and visualisation tool that contains collections of time series data on a variety of topics. For an introduction, see here. Recently, a data catalogue API has been launched. The catalogue enables users to retrieve the contents of the data catalogue along with data and metadata about individual datasets and their resources. The API is still in development (see here). The code below imports the catalogue into the R workspace. For technical reasons, the list of 6,712 datasets is downloaded in parts and merged afterwards. The WB catalogue is returned in HTML format, not in XML. The function GET() of the httr package (version 1.4.6) is used to retrieve the list of datasets. The function fromJSON() of the jsonlite package is used to convert the content of the response in HTML format into a data frame. The code is as follows:

```
url <- list()
url[[1]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=1000"
)
url[[2]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=2000&$skip=1000"
)
url[[3]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=3000&$skip=2000"
)
url[[4]] <- paste0(
```

```r
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=4000&$skip=3000"
)
url[[5]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=5000&$skip=4000"
)
url[[6]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=6000&$skip=5000"
)
url[[7]] <- paste0(
  "https://datacatalogapi.worldbank.org/ddhxext/",
  "DatasetList?$top=7000&$skip=6000"
)
toc_wb <- NULL
for (k in seq_along(url))
{
  z <- httr::GET(url[[k]])
  dd <- jsonlite::fromJSON(httr::content(z, as = "text"))
  toc_wb <- rbind(toc_wb, dd$data)
}
toc_wb <- toc_wb[, c("dataset_id", "name")]
colnames(toc_wb) <- c("id", "Name.en")
```

The code below searches for datasets on migration (j1) and datasets that include the keywords migration and bilateral (j2). j1 and j2 refer to the locations (lines) of the names of the datasets in the list of datasets. The latter returns the Global Bilateral Migration Database.

```r
j1 <- which(grepl("Migration", toc_wb$Name.en, ignore.case = FALSE))
j2 <- which(grepl("Migration", toc_wb$Name.en, ignore.case = FALSE) &
         grepl("Bilateral", toc_wb$Name.en, ignore.case = FALSE))
list <- data.frame(line = j2, name = toc_wb$Name.en[j2])
```

3.1.6 National statistical offices: Statistics Canada, Statistics Lithuania, and Australian Bureau of Statistics

The above procedure is used to obtain the data catalogues of the data portals of Statistics Canada (Web Data Service), Statistics Lithuania (here), and the Australian Bureau of Statistics (ABS). Statistics Canada lists 7,123 datasets, Statistics Lithuania 8,847, and ABS 1,188. Statistics Canada returns the data catalogue in JSON format, the other providers in XML.

```
# Statistics Canada
url <- paste0("https://www150.statcan.gc.ca/t1/wds/rest/",
          "getAllCubesList")
toc_statcan0 <- jsonlite::fromJSON(url)
toc_statcan <- data.frame(id = toc_statcan0$productId,
                  Name.en = toc_statcan0$cubeTitleEn)
# Statistics Lithuania
url <- "https://osp-rs.stat.gov.lt/rest_xml/dataflow/"
d <- rsdmx::readSDMX(url)
toc_LTU <- as.data.frame(d)

# ABS
url <- "https://api.data.abs.gov.au/dataflow"
d <- rsdmx::readSDMX(url)
toc_ABS <- as.data.frame(d)
```

3.1.7 A most simple global dataset search engine

The catalogues of the data providers covered in this section may be combined into a single data frame.

```
toc <- cbind(toc_oecd[,c("agencyID","id","Name.en")])
toc <- rbind(toc, cbind(toc_eurostat[,c("agencyID","id","Name.en")]))
toc <- rbind(toc, cbind(toc_ilo[,c("agencyID","id","Name.en")]))
toc <- rbind(toc, cbind(toc_unsd[,c("agencyID","id","Name.en")]))
toc <- rbind(toc, cbind(agencyID = rep("WorldBank", nrow(toc_wb)),
toc_wb))
toc <- rbind(toc, cbind(agencyID = rep("StatisticsCanada",
                   nrow(toc_statcan)),toc_statcan))
```

The combined catalogues list 24,038 datasets at time of writing. Let's search for datasets on migration.

```
keyword <- "migration"
j <- grep(keyword, toc$Name.en, ignore.case = TRUE)
# ID and title of the datasets on migration: toc[j,]
```

The search produces a list of 189 datasets by OECD, Eurostat, the World Bank, and Statistics Canada.[11]

3.2 URL query builder

Several tools exist to compose URL queries from components supplied by the user, but not all comply with the SDMX standard. Some are offered by data providers (e.g., OECD at here and Eurostat at here), while other were developed by unrelated individuals. The latter category includes wbstats (Piburn 2020), which helps build queries for the World Bank RESTfull API (not SDMX based). The ILO offers a menu-driven SDMX query builder (see here). CRAN includes packages with functions that help build URL queries. The package rsdmx used in this paper builds URLs for a range of SDMX-based APIs, while restatapi (Mészáros 2023) builds queries for the Eurostat's SDMX-based REST API. In this section, the focus is on generic SDMX query building. To illustrate the generic query builder, we retrieve data from the OECD, Eurostat, ILO, UNSD, World Bank, and Asia Development Bank (ADB). The queries built are used to retrieve data and metadata (data structures, dataflows, and codelists).

3.2.1 OECD

To illustrate the procedure, some data are retrieved from the OECD's International Migration Database MIG, which gives the annual number of immigrants in OECD countries by sex and nationality in the period 1995–2021. Only individuals with a foreign nationality are included in the inflow. The database can be accessed here. The metadata associated with MIG indicate that the OECD derives the flow data from population registers, work permits, or specific surveys. The exact meaning of 'immigrant' varies between OECD countries. For details on the B11 dataset, see here and here.

[11] Dataset search is an emerging subject. In 2018 Google launched a beta version of a Dataset Search Engine, followed by a full version in 2020 (see here). The outcome is still rudimentary and does not include a facility to connect users to the websites of data providers.

The code below imports the entire database into the R workspace. The database is large (64 MB), and downloading it, and particularly the conversion into a data frame object takes time (minutes). The database is saved as an R data file in a folder specified by path to be provided by the user.

```
url <- "https://stats.oecd.org/restsdmx/sdmx.ashx/GetData/MIG"
d <- rsdmx::readSDMX(url)
data <- as.data.frame(d)
# Save the database
save(data,file=paste0(path,"MIG.RData"))
```

To illustrate the SDMX syntax, we retrieve the number of immigrants in the United States by sex and one of three nationalities (Mexico, India, and China) for each year from 2000 to 2021.

To build the URL, we need to know the structure of the multidimensional table in the MIG database. That information is found in the DSD of the MIG database. The URL is constructed from its components (see Section A in following code snippet). The URL is https://stats.oecd.org/restsdmx/sdmx.ashx/GetDataStructure/MIG/.

The readSDMX() function of the rsdmx package retrieves the DSD. The DSD object dsd is an S4 object. The code snippet C extracts from the object the dimensions. Table 4 shows the dimensions. Note also that slot(dsd, “concepts”) is equivalent to the subsetting dsd@concepts.[12] The sequence of the dimensions is important to construct the key defining the requested subset of data.

```
# A. query builder: Building blocks for the URL
protocol <- "https"
entrypoint <- "stats.oecd.org/restsdmx/sdmx.ashx/"
resource <- "GetDataStructure"
flowRef <- "MIG"
# Construct URL
url0 <- paste0(protocol, "://", entrypoint, resource, "/", flowRef)
url <- paste0(url0, "?references=children&detail=referencepartial")

# B. Import DSD into R workspace
dsd <- rsdmx::readSDMX(url)

# C. Extract from the DSD the id and name of each dimension of MIG
dimensions <- t(sapply(slot(slot(dsd, "concepts"),"concepts"), function
```

[12] To get the slots of the dsd object, use getSlots(class(dsd)).

```
(x)
      {y1 <- slot(x, "id")
       y2 <- slot(x, "Name")$en
       c(y1, y2)
      }))
colnames(dimensions) <- c("id","Name.en")
dimensions <- data.frame(dimensions)
```

Table 4: Dimensions of the OECD MIG database

id	Name.en	ncategories
CO2	Country of birth/nationality	500
VAR	Variable	16
GEN	Gender	6
COU	Country	76
YEA	Year	56
OBS_VALUE	Observation value	42
TIME_FORMAT	Time format	10
OBS_STATUS	Observation status	N/A

The following code gets, for each dimension of MIG, the names of the categories. Dimension COU includes OECD countries, while dimension CO2 includes all countries of the world and selected groups of countries. Dimension GEN has three categories: women, total, and percent women. The MIG database does not store data on males. They are obtained by subtracting the number of females (WMN) from the total (TOT). Dimension VAR includes the names of the datasets stored in the MIG database. The categories of the VAR are shown in Table 5. The table shows each dimension's identification code and name and the maximum number of categories allowed by the DSD.

```
# Get codelists associated with MIG: ID and description
codelists <- lapply(dsd@codelists@codelists, function(x)
      {codes0 <- sapply(x@Code, function(z) {
                             z1 <- slot(z, "id")
                             z2 <- slot(z, "description")$en
                             z <- c(z1,z2)
                             })
     codes00 <- t(codes0)
      })
names(codelists) <- dimensions$id[seq_along(codelists)]
```

```
# Show dimensions
yy <- sapply(codelists,function(x) length(x))
dimensions$ncategories <- c(yy,NA)
z <- knitr::kable(dimensions,format = "latex",
 caption = "Dimensions of the OECD MIG database")
kableExtra::kable_styling(z, full_width = FALSE,
                          latex_options = "HOLD_position")
```

The code to display Table 5 is

```
z <- knitr::kable(codelists[["VAR"]],format = "latex",
  caption = "Structure of the OECD MIG database", "simple")
kableExtra::kable_styling(z, full_width = FALSE,
                            latex_options = "HOLD_position")
```

Table 5: Structure of the OECD MIG database

VAR	Description
B11	Inflows of foreign population by nationality
B12	Outflows of foreign population by nationality
B13	Inflows of asylum seekers by nationality
B14	Stock of foreign-born population by country of birth
B15	Stock of foreign population by nationality
B16	Acquisition of nationality by country of former nationality
B21	Inflows of foreign workers by nationality
B22	Inflows of seasonal foreign workers by nationality

Once the structure of the multidimensional table is known, the data query can be constructed. In the R code below, the URL is assembled from its components, and the function readSDMX() imports the requested data into R. The as.data.frame() function converts an S4 object into a data frame. The first few lines of the data frame d_a are shown in Table 6.

```
protocol <- "https"
entrypoint <- "stats.oecd.org/restsdmx/sdmx.ashx/"
resource <- "GetData"
flowRef <- "MIG"
key <- "CHN+IND+MEX.B11.WMN+TOT.USA"
# Define the parameters: start and end of time series
parameters <- paste0("startPeriod=2000&endPeriod=2022")
# Construct URL
```

```
url <- paste0(protocol, "://", entrypoint, resource, "/", flowRef, "/",
              key, "/?", parameters)

# Import data into R workspace
d <- rsdmx::readSDMX(url)
d_a <- as.data.frame(d)
    z <- knitr::kable(head(d_a),
  format = "latex",
  caption = "First lines of data selected from MIG database")
kableExtra::kable_styling(z, full_width = FALSE,
  latex_options = "HOLD_position")
```

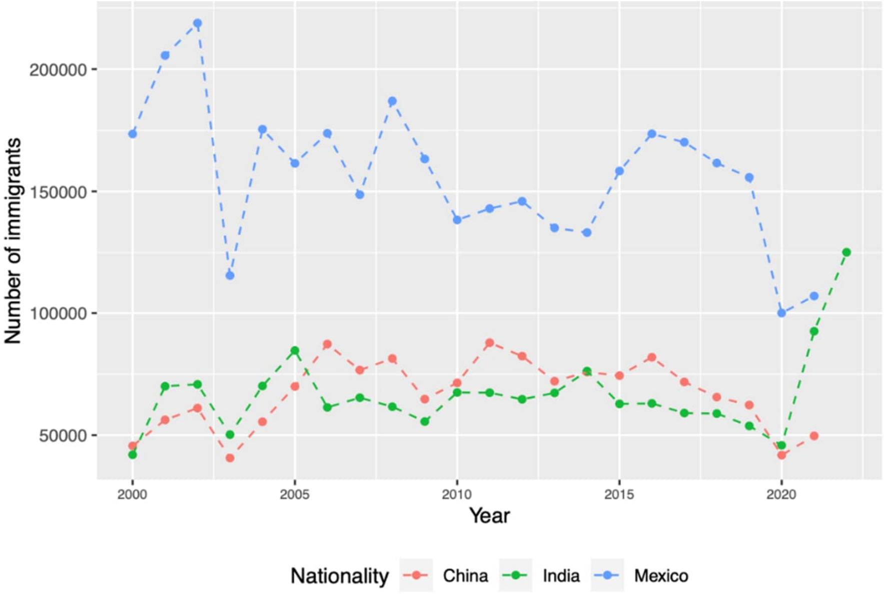
Table 6: First lines of data selected from MIG database

CO2	VAR	GEN	COU	TIME_FORMAT	obsTime	obsValue
CHN	B11	WMN	USA	P1Y	2002	36537
CHN	B11	WMN	USA	P1Y	2003	25396
CHN	B11	WMN	USA	P1Y	2004	33724
CHN	B11	WMN	USA	P1Y	2005	41448
CHN	B11	WMN	USA	P1Y	2006	50243
CHN	B11	WMN	USA	P1Y	2007	43149

Figure 2 shows the inflows of foreign population in the United States by nationality and year. The metadata (here) state that inflow is the number of lawful permanent residents (LPRs) (‘green card’ recipients). Inflows include persons already present in the United States who changed status (i.e., were admitted conditionally and are required to remove their conditional status after two years). They are counted as LPRs when they first enter. Data cover the fiscal year (October of the year indicated to September of the following year). The data producer is the Department of Homeland Security. This description illustrates the type of metadata needed for a correct interpretation of the data. The code below produces the figure.

```
library(ggplot2)
dd <- reshape2::melt(d_a[d_a$GEN=="TOT",],
          id.vars=c("COU","VAR","TIME_FORMAT","GEN","CO2", "obsTime"))
ggplot(dd, aes(as.numeric(obsTime), value, colour = CO2)) +
  geom_point() + geom_line(linetype = "dashed") +
  xlab("Year") +
  ylab("Number of immigrants") +
  theme(axis.text.x = element_text(size = 7)) +
  scale_colour_discrete(name="Nationality",
    breaks=c("CHN", "IND","MEX"),
    labels=c("China", "India","Mexico")) +
  theme(legend.position="bottom")
```

Figure 2: **Inflows of foreign population in the United States, by nationality**



Source: OECD’s International Migration Database (MIG).

3.2.2 Eurostat

Let’s retrieve data from the Eurostat Migration and Asylum Database. To access the database in the web browser, see here. The website lists several migration and asylum datasets. Select the immigration dataset (migr_immi), which is also a collection of datasets. Select migr_imm8. The dataset gives the number of immigrants in countries of the European Union and the European Free Trade Organization and in the United Kingdom, Montenegro, North Macedonia, and Ukraine by age and sex. Data for Ukraine are missing. To access the data in the web browser, see here, and for the metadata, see here.

The following code imports the complete dataset (20 MB) into R and saves the dataset as an R data file in a folder specified by path. The bulk download, and more particularly the conversion into a data frame, takes a few minutes. The code is given but is not executed.

```
url <- paste0("https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1
/",
          "data/migr_imm8")
d <- rsdmx::readSDMX(url)
data <- as.data.frame(d)
# Save the dataset in a folder with path provided by the user
save(data,file=paste0(path,"migr_imm8.RData"))
```

To illustrate SDMX, we extract the annual number of immigrants in Germany and France by age (age reached) and sex during the period 2000–2022. First we retrieve the data structure and extract the dimensions and codelist.

```
# Dimensions
protocol <- "https"
entrypoint <- "ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/"
resource <- "datastructure"
flowRef <-"ESTAT/migr_imm8"
# Construct URL
url <- paste0(protocol, "://", entrypoint, resource, "/", flowRef,
              "?references=children")
# Import DSD into R workspace
dsd <- rsdmx::readSDMX(url)
# Dimensions
dimensions <- t(sapply(slot(slot(dsd, "codelists"),"codelists"),
                       function(x)
                         { y1 <- slot(x, "id")
                           y2 <- slot(x,"Name")$en
                           yy <- c(y1,y2)
                       }))
colnames(dimensions) <- c("id","Name.en")
dimensions <- data.frame(dimensions)
```

Second we retrieve the codelists of migr_imm8 and determine the number of categories:

```
# Codelists
codelists <- lapply(dsd@codelists@codelists,
                function(x)
       { codes0 <- sapply(x@Code,
                          function(z)
                            { id <- slot(z, "id")

                            })
       })
# For each dimension: number of categories
dimensions$ncategories <- sapply(codelists,function(x) length(x))
```

Table 7 shows the dimensions and, for each dimension, the number of categories in the codelists.

```
z <- knitr::kable(dimensions,format = "latex",
 caption = "Dimensions of the Eurostat migr\\_imm8 dataset")
kableExtra::kable_styling(z, full_width = FALSE,
                          latex_options = "HOLD_position")
```

Table 7: Dimensions of the Eurostat migr_imm8 dataset

id	Name.en	ncategories
FREQ	Time frequency	11
AGEDEF	Age definition	2
AGE	Age class	655
UNIT	Unit of measure	707
SEX	Sex	7
GEO	Geopolitical entity (reporting)	4,041
OBS_FLAG	Observation status (Flag)	58

Third, we retrieve the data. The data filter (series key) retains the immigration in Germany and France of individuals of age 25. The dimension ‘status of the observation (OBS_FLAG)’ is disregarded.

```
protocol <- "https"
entrypoint <- "ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/"
resource <- "data"
flowRef <- "migr_imm8" # Note: no agency identifier
key <- "A.REACH.Y25.NR.T.DE+FR"
# Define the parameters: start and end of time series
```

```
parameters <- paste0("startPeriod=2000&endPeriod=2022")
# Construct URL
url <- paste0(protocol, "://", entrypoint, resource, "/", flowRef,
              "/", key, "/?", parameters)
# Import data into R
d <- rsdmx::readSDMX(url)
data <- base::as.data.frame(d)
```

3.2.3 ILO

From the ILO databank, we retrieve data on the working-age population of selected countries. The database is POP_XWAP_SEX_AGE_NB. To access the data in the ILO web browser, see here.

The following code snippet retrieves the location of the database in ILO's data catalogue:

```
line <- which(toc_ilo$id=="POP_XWAP_SEX_AGE_NB")
```

To retrieve the information in the data catalogue related to this dataset, use toc_ilo[line,]. The information includes the identification code, the name, and the description of the dataset in English, French, and Spanish.

Let's retrieve the data structure and the title of the database.

```
# Retrieve structure of dataset POP_XWAP_SEX_AGE_NB
protocol <- "https"
entrypoint <- "/www.ilo.org/sdmx/rest/"
resource <- "datastructure"
flowRef <- "ILO/POP_XWAP_SEX_AGE_NB"
url <- paste0(protocol, "://", entrypoint, resource, "/",
              flowRef, "?references=children&detail=referencepartial")
dsd <- rsdmx::readSDMX(url)

# Get dimensions and codelist
codelist <- lapply(dsd@codelists@codelists, function(x)
      {y <- slot(x, "id")
       codes0 <- sapply(x@Code, function(z) zz <- slot(z, "id"))
      })
kk <- sapply(dsd@codelists@codelists,
              function(x) slot(x, "id"))[seq_along(codelist)]
names(codelist) <- substr(kk, start = 4, stop = nchar(kk))
dimensions <- names(codelist)

# Retrieve the title of the table
protocol <- "https"
entrypoint <- "/www.ilo.org/sdmx/rest/"
resource <- "dataflow"
url <- paste0(protocol, "://", entrypoint, resource, "/", flowRef)
ds <- rsdmx::readSDMX(url)
title_table <- slot(ds@dataflows[[1]], "Name")$en
```

We now download data on the size of the working-age population (in thousands) of Brazil, China, India, and South Africa, males and females combined.

```
protocol <- "https"
entrypoint <- "www.ilo.org/sdmx/rest/"
resource <- "data"
flowRef <- "ILO,DF_POP_XWAP_SEX_AGE_NB"
key <-  "BRA+IND+CHN+ZAF.A.POP_XWAP_NB.SEX_T.AGE_5YRBANDS_TOTAL"
parameters <- paste0("startPeriod=2010&endPeriod=2021")
url <- paste0(protocol, "://", entrypoint, resource, "/", flowRef, "/",
              key, "/?",  parameters)
d <- rsdmx::readSDMX(url)
data <- as.data.frame(d)
# Select relevant columns and a few selected years
data2 <- data[data$obsTime%in%c("2010","2020"),c("REF_AREA","obsTime",
"obsValue")]
```

The results are shown in Table 8.

```
z <- knitr::kable(head(data2),
  format = "latex",
  caption = paste0("Working-age population, selected countries and",
                   " selected years (thousands)"))
kableExtra::kable_styling(z, full_width = FALSE,
  latex_options = "HOLD_position")
```

To retrieve the entire database, the URL is https://www.ilo.org/sdmx/rest/data/ILO,DF_POP_XWAP_SEX_AGE_NB. Retrieval takes time.

The ILO implemented an extension of their API to make it easier to query data for country groups, such as BRICS for Brazil, Russia, India, China, and South Africa (see here, page 13).

Table 8: Working-age population (thousands), selected countries, and selected years

REF_AREA	obsTime	obsValue
BRA	2020	166968.22
CHN	2010	1038171.24
CHN	2020	1141071.56
IND	2010	707840.04
IND	2020	832070.34
ZAF	2010	35418.83

3.2.4 UNSD

To obtain the structure of the SDG Indicators dataset, the agency ID must be provided. The ID is IAEG-SDGs, which stands for Inter-agency and Expert Group on SDG Indicators. The R code returning the dimensions and the codelists is

```r
url0 <- paste0(
  "https://registry.sdmx.org/ws/public/sdmxapi/rest/",
  "datastructure/IAEG-SDGs/SDG"
)
url <- paste0(url0, "?references=children")
dsd0 <- rsdmx::readSDMX(url)

# Dimensions
dimensions <- t(sapply(slot(slot(dsd0, "codelists"), "codelists"),
  function(x) {
  y1 <- slot(x, "id")
  y2 <- slot(x, "Name")$en
  c(y1, y2)
  }))
colnames(dimensions) <- c("id", "Name.en")
dimensions <- data.frame(dimensions)
```

```
# Codelists
codelist <- sapply(dsd0@codelists@codelists, function(x) {
  y <- slot(x, "id")
  yy <- sapply(x@Code, function(z) {
    k <- slot(z, "id")
  })
})
kk <- sapply(dsd0@codelists@codelists, function(x) y <- slot(x, "id"))
names(codelist) <- substr(kk, start = 4, stop = nchar(kk))
```

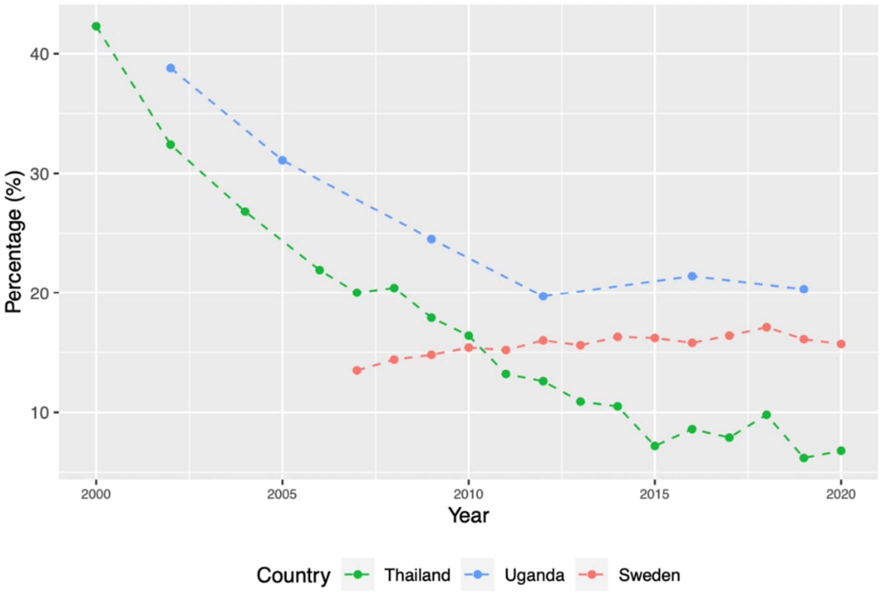
The following code retrieves a selection of data from the SDG database. Retrieved is the percentage of population living below the national poverty line (“SI_POV_NAHC”) in Uganda (REF_AREA 800), Thailand (764), and Sweden (752) for the years from 2000 to 2022 or for the years in this period for which the data exist. If the data exist for none of the years selected, the following error message is returned: HTTP request failed with status: 404. The data filter (key) needs ISO numeric codes of the selected countries. To find the ISO codes of a given country, use the ISO search platform (see here) and select country codes.

```
url <- paste0(
  "http://data.un.org/WS/rest/data/DF_SDG_GLH/",
  "..SI_POV_NAHC.800+764+752..........",
  "?startPeriod=2000&endPeriod=2022"
)
d <- rsdmx::readSDMX(url)
dd <- as.data.frame(d)
```

Figure 3 shows the percentage of the population living below the national poverty line. The percentage in poverty depends on the income level used in the definition of poverty.

```
require (ggplot2)
ggplot(dd, aes(as.numeric(obsTime), obsValue, colour = REF_AREA)) +
  geom_point() +
  geom_line(linetype = "dashed") +
  xlab("Year") +
  ylab("Percentage (%)") +
  theme(axis.text.x = element_text(size = 7)) +
  scale_colour_discrete(
    name = "Country",
    breaks = c("764", "800", "752"),
    labels = c("Thailand", "Uganda", "Sweden")
  ) +
  theme(legend.position="bottom")
```

Figure 3: Percentage of population living below the national poverty line (SDG 1.2.1)



Source: UNSD Sustainable Development Goals Indicators database.

3.2.5 World Bank

At present, only the World Development Indicators (WDI) are available through the SDMX-based API (see here). Developer information about the World Bank API is available here.

Let’s retrieve data on the population size of Afghanistan, South Africa, and Panama.

```r
protocol <- "http"
entrypoint <- "api.worldbank.org/v2/sdmx/rest/"
resource <- "data"
flowRef <- "WDI"
key <- "A.SP_POP_TOTL.AFG+ZAF+PAN"
parameters <- "startPeriod=2015&endPeriod=2022"
url <- paste0(protocol, "://", entrypoint, resource, "/",
              flowRef, "/", key, "/?", parameters)
d <- rsdmx::readSDMX(url)
m <- as.data.frame(d)
```

3.2.6 Asia Development Bank

From the ADB Key Indicators Database, we retrieve data on the population of China (PRC), India (IND), Japan (JPN), and Australia (AUS) for selected years from 2005 until the last year available (in millions). To access the data on the ADB website, see here. In the code below, the data are retrieved programmatically.

```r
# Retrieve the data
protocol <- "https"
entrypoint <- "kidb.adb.org/api/v2/sdmx/"
resource <- "data"
flowRef <- "IMF" # / instead of ,
key <- "A.LP_PE_NUM_MOP.PRC+IND+JPN+AUS"
# Define the parameters: start and end of time series
parameters <- paste0("startPeriod=2005&endPeriod=2023")
# Construct URL
url <- paste0(
  protocol, "://", entrypoint, resource, "/",
  flowRef, "/", key, "/?", parameters
)
d <- rsdmx::readSDMX(url)
data <- as.data.frame(d)
```

4. Conclusion

The open data movement leads to vast amounts of statistical data that can be accessed relatively easily and at no cost. The common approach is to visit websites of data providers and download data as Excel files or comma-separated text files. With the introduction of APIs, access to data portals can be automated, meaning that a computer program communicates with the data portal of the provider and retrieves the requested data. The R community responded swiftly to the introduction of open data APIs by producing packages to import data directly into the R workspace and by making these packages and their source code publicly available on CRAN or GitHub. The package rsdmx used in this paper concentrates on SDMX-based REST APIs.

The implementation of the SDMX standard is a work in progress. When this standard is fully implemented, users need to learn a single standard to get access to open statistical data worldwide and to retrieve data and metadata. The automation of data retrieval using a uniform method for a variety of data providers is likely to change in a fundamental way how demographic research is done. An integration of data retrieval and data analysis will be the first step. Other steps will follow, such as dataset search engines, APIs with fully harmonised data, and natural language interfaces. Thiry, Manolescu, and Liberti (2020) describe a prototype chatbot based on the SDMX standard. For instance, the question “What has been the trend in the size of the female population over 85 in Japan?” triggers a series of operations that result in a correct answer presented in a format selected by the user. Metadata will be added to enable the user to interpret the response and assess its validity. That future may be nearer than we imagine. One of the reviewers of this paper informed the author that state-of-the-art models like ChatGPT with plug-ins like WolframAlpha can do this already (see here). In its Roadmap 2021–2025, the SDMX community endorses the development of chatbot conversations as a way to make statistical data more accessible. These developments and the 9th SDMX Global Conference, held in Bahrain from 29–31 October 2023, reflect the interest in artificial intelligence in the production and dissemination of data.

The implementation of the SDMX standard faces major challenges. A first one is the harmonisation of concepts and data filters (keys). Controlled vocabularies are fundamental for the harmonisation and sharing of data. As long as data providers use different concepts of age, household, migrant, address, or employment status and tools to harmonise data identifiers are lacking, automatic data retrieval does not resolve the issue of incomparable data. A second challenge is the development of common URL syntax. Data providers endorsing the SDMX standard continue to use slightly different SDMX syntax, meaning that the harmonisation of concepts and data filters is very much a work in progress. Implementation of the SDMX standard is a demanding process. Eurostat published a useful road map for the implementation of SDMX (see here). A third challenge is the production of the metadata needed to correctly interpret data. Some organisations publish data structure definitions (DSDs) with much detail, while others provide little detail. Harmonisation of metadata remains a major challenge. Ideally, metadata include descriptions of the methods used to produce statistics, observed or estimated. Guidelines for data harmonisation and controlled vocabularies are necessary but not sufficient. Data providers need incentives to comply with the guidelines.

Demographers participate in the development of the SDMX standard by contributing a FAIR vocabulary of demographic concepts (IUSSP – CODATA Working Group on FAIR Vocabularies 2023). A larger involvement would benefit the SDMX community but would also help demographers to achieve a long-standing ambition – namely, to make data more accessible and comparable. Fulfilling that ambition brings demography nearer to the ultimate goal: high-quality data and information on population.

5. Acknowledgements

I like to thank George Alter, University of Michigan and co-chair of the IUSSP – CODATA Working Group on FAIR Vocabularies, and two anonymous referees for their very helpful comments.

References

Agresti, A. (2013). *Categorical data analysis*. 3rd Edition. Wiley.

Bauer, P.C. and Landesvatter, C. (2023). Writing a reproducible paper with R Markdown and Quarto. doi:10.31219/osf.io/ur4xn.

Bishop, Y.M., Fienberg, S.E., and Holland, P.W. (1975). *Discrete multivariate analysis: Theory and practice*. Cambridge, Mass.: MIT Press. Reprinted by Springer in 2007. doi:10.1007/978-0-387-72806-3.

Blondel, E. (2015). rsdmx – tools for reading SDMX data and metadata documents in R(slides). https://www.slideshare.net/EmmanuelBlondel/rsdmx-tools-for-reading-sdmx-data-and-metadata-in-r.

Blondel, E. (2023a). *rsdmx – Tools for reading SDMX data and metadata in R (R package Version 0.6-3)*. https://github.com/opensdmx/rsdmx/wiki#package_overview.

Blondel, E. (2023b). *rsdmx Quickstart guide*. https://cran.r-project.org/web/packages/rsdmx/vignettes/quickstart.html.

European Commission Expert Group on FAIR Data (2018). *Turning FAIR data into reality*. Brussels: European Commission. doi:10.2777/1524.

Frank, M. and Hartgerink, C. (2017). RMarkdown for writing reproducible scientific papers. https://libscie.github.io/rmarkdown-workshop/handout.html.

Gillman, D. (2023). Achieving transparency: A metadata perspective. *Data Intelligence* 5(1): 261–274. doi:10.1162/dint_a_00188.

Gylling, K.C. (2019). *Pyscbwrapper 0.1.1*. https://github.com/kirajcg/pyscbwrapper.

IUSSP – CODATA Working Group on FAIR Vocabularies (2023). FAIR Vocabularies in population Research. Report of the IUSSP – CODATA Working Group on FAIR Vocabularies. Paris: IUSSP; CODATA. doi:10.5281/zenodo.7818157.

Macoveiciuc, A. (2020). Beginner’s guide to APIs, protocols and formats. *Frontend Digest* 29(April). https://medium.com/frontend-digest/beginners-guide-to-apis-protocols-and-data-formats-f80cf7f30425.

Magnusson, M., Kainu, M., Huovari, J., and Lahti, L. (2022). *pxweb: R Interface to PXWEB APIs. Version 0.16.2*. https://cran.r-project.org/web/packages/pxweb/index.html.

Mészáros, M. (2023). *Restatapi: Search and retrieve data from Eurostat Database (r Package Version 0.20.6).* https://cran.r-project.org/web/packages/restatapi/index.html.

National Academies of Sciences, Engineering, Medicine, and others (2022). *Transparency in statistical information for the National Center for Science and Engineering Statistics and All Federal Statistical Agencies*. Washington, DC: The National Academies Press. doi:10.17226/26360.

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between JSON data and R objects. *arXiv:1403.2805 [stat.CO]*. https://arxiv.org/abs/1403.2805.

Piburn, J. (2020). *wbstats: Programmatic Access to the World Bank API*. Oak Ridge, Tennessee: Oak Ridge National Laboratory. doi:10.11578/dc.20171025.1827.

Queljoe, M. de (2023). *readsdmx: Read SDMX-XML Data*. https://github.com/mdequeljoe/readsdmx.

R Core Team (2022). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ševčíková, H. (2023). *wpp2022 United Nations World Population Prospects 2022*. https://github.com/PPgp/wpp2022.

Stahl, R. and Staab, P. (2018). *Measuring the data universe. Data integration using statistical data and metadata exchange*. Springer. doi:10.1007/978-3-319-76989-9.

Thiry, G., Manolescu, I., and Liberti, L. (2020). A question answering system for interacting with SDMX databases. *NLIWOD 2020-6th natural language interfaces for the web of data/workshop (in conjunction with ISWC).* https://inria.hal.science/hal-03021075/ and https://ceur-ws.org/Vol-2722/nliwod2020-paper-1.pdf.

Wickham, H. (2019). *Advanced R*. 2nd edition. Boca Raton, Florida: CRC press. doi:10.1201/9781351201315.

Wickham, H. (2023). *httr: Tools for Working with URLs and HTTP (R Package Version 1.4.6)*. https://CRAN.R-project.org/package=httr.

Wickham, H., Hester, J., and Ooms, J. (2021). *Xml2: Parse XML (R Package Version 1.3.3)*. https://CRAN.R-project.org/package=xml2.

Wilkinson, M.D., Dumontier, M., Aalbersberg, Ij.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L.B. da, Bourne, P.E., and others

(2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data* 3(1): 1–9. doi:10.1038/sdata.2016.18.

World Wide Web Consortium (2014). The RDF data cube vocabulary. http://www.w3.org/TR/vocab-data-cube/.

Xie, Y. (2023). *Knitr: A General-Purpose Package for Dynamic Report Generation in R. Version 1.44*. https://cran.r-project.org/web/packages/knitr/index.html.

Appendix A: How to install the packages used in this paper

The following packages are used in this paper: rsdmx, jsonlite, httr, ggplot2, knitr, and kableExtra. To install the packages on your computer, run the following code in the console pane in R or RStudio:

```
install.packages(c("rsdmx","readsdmx","jsonlite","httr",
                   "ggplot2","knitr","kableExtra"))
```

The packages are stored in a folder. The user may specify the folder or leave the selection to the operating system. The system keeps track of the location and knows where to find a package when needed. To find out where the packages are located, type .libPaths() in the console.

Once installed, the packages must be loaded (i.e., imported) into the workspace. The traditional approach is to use the library() function. The function attaches a package to the search path and, when a function is called by name, R searches for packages in the order they are listed on the search path (e.g., the most recently loaded package is the first on the list). If two packages on the search path have a function with the same name, R takes the package first on the search path. It may not be the package the user wants. To prevent R from extracting the function from the wrong package, the R Core Team recommends specifying both the package and the function. The two names are separated by the double colon (::) operator. That practice is adopted in this paper. To install the RStudio Integrated Development Environment (IDE), go to the Posit (formerly RStudio) website here. If you are new to RStudio, you find an introduction for beginners here and a tutorial here.

Note that CRAN includes another package to read SDMX data: readsdmx (Queljoe 2023). The package has a single function, read_sdmx(), which imports SDMX data into R as a data frame. The package works well to download data but does not download data structures and metadata.

Appendix B: A note on reproducibility

Reproducibility refers to the ability to reproduce computations and obtain the same results. In computational science, reproducibility requires that a description of the results is accompanied by an adequate description of the method, the data, and the code used to obtain the results. Ideally, the code is interactive to allow the reader to run the code without the need to copy the code and/or leave the document. If the documentation is adequate, the reader can reproduce the entire workflow from data acquisition to presentation of the results. Publishing systems that integrate text, data, code, and results exist. They include R Markdown, Jupyter Notebook, and Quarto. The latter two support multiple programming languages, including R and Python. The publishing systems export documents to PDF, HTML, Microsoft Word, and a few other formats. The technology is a milestone on the road to reproducible research.

The paper was written with R Markdown using the rmarkdown package in RStudio. R Markdown is a variant of Markdown, a lightweight markup language. A markup language is a text-encoding system consisting of a set of symbols inserted in a text document to control its format. LaTeX, HTML, and XML are markup languages. Text in a markup language is difficult to read. The lightweight version is easier to read. R Markdown has the additional feature that the language may include R code. Snippets of R code may be executed by a simple click, without leaving the R Markdown document (extension Rmd). For an introduction to R Markdown, see Bauer and Landesvatter (2023) and for a short tutorial, see Frank and Hartgerink (2017).

BibTeX is used to manage and format the bibliography and the Citation Style Language (CSL) is used to prepare the citations in the format required by *Demographic Research*. The CSL is a collection of descriptions (XML format) for the formatting of citations and bibliographies. The Zotero Style Repository has the citation styles of major scholarly journals, including *Demographic Research* and *Demography*. Ilya Kashnitsky contributed the *Demographic Research* citation style (demographic-research.csl) to the Zotero Style Repository. The R packages knitr (Xie 2023) is used to convert the Rmd format to PDF and MS Word files.

The paper includes hyperlinks. A hyperlink is a link to another publication, data, website, or any other object available online. Hyperlinks point to the location (URL) of an object. It enables the reader to access an object without leaving a document. A simple click on a hyperlink takes you to the object. The link can also be copied (right click or two-finger tap). For instance, clicking on the linked text 'OECD' takes the reader to the OECD website.

For a paper to be reproducible, all objects cited in the paper or pointed at by hyperlinks should have unique and persistent identifiers. Many digital objects, including academic and professional publications and datasets, have a Digital Object Identifier (DOI). A DOI points to the location (URL) of the object and may point to other metadata of the object. DOIs are standardised by the International Organization for Standardization (ISO). For instance, the DOI of the OECD International Migration Database is 10.1787/data-00342-en. Clicking on it takes the reader to the database.

Appendix C: A brief technical introduction to APIs

An API (application programming interface) is a set of rules that allow programs to talk to each other and access data (resources). Four sets of rules may be distinguished. Each set is related to part of the technology. The parts are as follows:

a. The communication over the internet.
b. The architecture of an API.
c. The syntax of the data request.
d. The format of the data sent by the server to the client or user.

Several introductions to APIs are available on the internet. I found this and Macoveiciuc (2020) quite useful. The four parts of an API are described in this appendix. SDMX-based REST APIs are not the only standard used for the exchange of statistical data and metadata. Section e, ‘The PxWeb API and other APIs,’ offers a brief description of the PxWeb API developed by Statistics Sweden.

a. Communication over the internet

Computers connect and communicate with each other over the internet using the HTTP (hypertext transfer protocol). HTTP is a set of rules that allow web browsers and web servers to talk to each other. They include a request for action: GET to get data, POST to create a new entry in a database on a server, PUT to replace an existing entry in a database, and DELETE to delete data. The URL identifies the host computer to be contacted. The R package httr (Wickham 2023) is a collection of tools to perform HTTP requests and process the responses.

b. API architecture

Developers may use different architectural styles in API development. The architectural style determines how the API looks and how it deals with security issues. One style is REST (representational state transfer). REST uses HTTP for communication. REST is a relatively simple architecture. A REST-based API server does not retain session information on a request and its sender. The communication protocol is said to be stateless. It has no memory and cannot use information on previous requests. For instance, if authentication is required, a user (identified by the IP address) must authenticate every time a request is submitted. User authentication verifies the identity of a user attempting to gain access to a web service. Session information may, however, be stored on the user's computer. Cookies are small pieces of data collected during a session and stored by a website on a user's computer. A removal of cookies deletes all historical information on communication with a web service.

c. Data request syntax

A data request is a character string. The request consists of several elements in a fixed format determining the structure of the request. Think of a request as a sentence and syntax as the sequence of words creating a sentence others can understand. A well-structured sentence with a correct syntax does not imply that the sentence conveys the meaning intended by the sender. Semantics is the study of meaning in natural and computer languages.

A data request communicates to the server the precise location (on the server) of the data (resource) requested. The uniform resource locator (URL) is a request that points to the location of the resource requested. A pointer is an address. Many programming languages use pointers to locate objects (variables, files, etc.) on a computer's storage devices. Some programming languages, such as C and C++, allow the manipulation of the content of memory addresses. Pointers must conform to the syntax rules specified by a data provider. Otherwise, a server cannot locate the requested data. Since providers may organise their data in many different ways and dump the data on a server, syntax rules vary greatly between data providers. Users interested in combining data from different providers must learn different syntaxes. To enhance the dissemination and reuse of data, data providers are motivated to agree on a common syntax. A common syntax also helps producers of data to transmit their data to multiple data providers and to update data previously transmitted. It suffices to point to the precise location to store the new or updated data on a server. That motivation resulted in the initiative of data providers to develop a common standard for data and metadata exchange, which resulted in the SDMX standard released in 2004.

d. Format of data transmitted

The RESTful API returns the requested data in a particular format, usually the JSON (JavaScript Object Notation) or XML (Extensible Markup Language) format.[13] The

[13] A markup language is a language that allows users to insert symbols or annotations in a text document to control its structure and look. Examples include TeX and Markdown.

format is independent of the programming language used. Most programming languages include functions to generate and parse JSON- and XML-format data. R has the package jsonlite (Ooms 2014) to produce and process JSON-format data. The R package XML (Wickham, Hester, and Ooms 2021) has tools for parsing and generating data in XML format.

The most common version of the SDMX standard (version 2.1) relies on the XML data transmission format, but version 3.0 of the standard, released in September 2021, also allows for the JSON and CSV formats. Most organisations endorsing the SDMX standard continue to use the XML format. The R package rsdmx, which implement version 2.1 of the SDMX standard, reads data in XML format.

e. The PxWeb API and other APIs

The SDMX standard is not the only standard for the exchange of statistical data. In the 1980s, Statistics Sweden developed a system to disseminate statistical data in machine-readable form and implemented the system on the mainframe system Axis. Statistical data are organised as multidimensional tables. In preparation of the 1990 census, Statistics Sweden developed PC-Axis to disseminate the census results. Initially, the file format was a plain text file (ASCII) but was changed in the 1990s to Structured Query Language (SQL) consistent with relational databases. The PC-Axis files have the extension px. To disseminate data over the internet, PxWeb was developed. It generates tables automatically from a PC-Axis database. PxWeb is proprietary. It is free only for governments and international organisations. According the Statistics Sweden’s website, the organisation does not guarantee that PxWeb works and does not provide support. The open-source code of PxWeb is available here. For the px format, see here. Open data are available free of charge through the PxWeb API. Magnusson et al. (2022) publish R tools to access the PxWeb API, and Gylling (2019) publishes a Python wrapper for the PxWeb API. Statistical offices in Nordic countries and over 90 national statistical offices around the world use PxWeb. The format of the data transmitted over the internet is called px. Version 2.0 of the PxWeb API, introduced in autumn 2023, conforms better to a RESTful design than the previous versions (see here).

Many organisations use their own standard and URL syntax. The CRAN repository of R packages includes several packages designed to access specific data portals and retrieve data. These packages use provider-specific URL syntaxes. For instance, CRAN includes a total of 22 R packages to retrieve data from the US Census Bureau, a few retrieving data from the bureau’s APIs. For a list, see the ‘Guide to Working with US Census Data in R,’ available here.

Appendix D: How to test the connection between your computer and a server

Occasionally, trying to access data programmatically results in an error message. The message indicates that the server cannot be found or that the data cannot be read from the connection. The reason may be a wrong URL or a server being down. In the latter case, it is recommended to try again later or to test the connection using the method described in this appendix, which describes two methods to test an internet connection. The second method, which is very easy to use, relies on the first method. That explains why it gives the same error messages as the first method.

The HTTP has two important parts: the request (the data sent to the server) and the response (the data sent back from the server). The GET() function of the httr package makes contact with a server, and the server sends a response. The response is a list variable with 10 components. One component is the HTTP status code, which gives information regarding the outcome of the execution of the request on the server. Codes in the 100s and 200s mean the request was successfully executed; codes in the 300s mean the page was redirected; codes in the 400s mean there was a mistake in the way the client sent the request; codes in the 500s mean the server failed to fulfil an apparently valid request. A frequent error is 404, which means that the server is not found. Note that common reasons why the server IP address cannot be found is that the URL is wrong or the website is down. If the website is down, there is nothing you can do except wait for the website to come back online.

The code to test the connection with the server of the Asian Development Bank and obtain the status code is

```r
url <- paste0(
  "https://kidb.adb.org/api/v2/sdmx/data/IMF/",
  "A.LP_PE_NUM_MOP.PRC+IND+JPN+AUS/",
  "?startPeriod=2005&endPeriod=2023"
)
r <- httr::GET(url)
httr::status_code(r)
#> [1] 200
```

An alternative method to check an internet connection is to try to read the first line from the connection.

```r
 test <- try(base::readLines(url, n = 1))
```

If the URL is wrong or the server is down, the function call returns a warning stating that the connection cannot open and produces the error code generated by the GET() function.