rsdmx - Tools for reading SDMX data and metadata documents in 

Emmanuel Blondel
emmanuel.blondel1@gmail.com

CEO - International Consultant

September 27, 2015

Outline

SDMX

Statistical Data and Metadata Exchange (SDMX)

Joint initiative created in 2001 by international & regional institutions [1]

- Promote and develop standards and guidelines for the exchange and sharing of statistical **data** and **metadata**
  - Definition of an abstract information model
  - Development of standard formats
  - Design of web-services architectures and tools
- Continuous process of improving the exchange of statistical data & metadata
  - an evolving set of specifications: SDMX 1.0, 2.0, 2.1
  - a variety of formats: SDMX-ML, SDMX-EDI, SDMX-JSON
  - a variety of service architectures: SOAP, REST
- Main SDMX format used across institutions: SDMX-ML

[1] Bank for International Settlements (BIS), European Central Bank (ECB), Statistical Office of the European Union (EUROSTAT), International Monetary Fund (IMF), Organization for Economic Co-operation and Development (OECD),United Nations (UN) and World Bank

Motivation
Conciliating SDMX and R

- Need of Interoperability between statistical systems, formats and tools
- Need to co-analyse and process statistical data
  - from a variety of domains (demography, socio-economics, health, environment, agriculture, fishery, etc.)
  - from scattered data providers (national, regional & international institutions)
  - by a growing range of actors (*e.g.* government institutions, statistical institutes, non-profit organizations, universities, research centers, companies)
- Fundamental and growing role of R as plateform for statistical computing
- ...and the need of tools to facilitate reading of SDMX data and metadata in R

rsdmx
Introduction

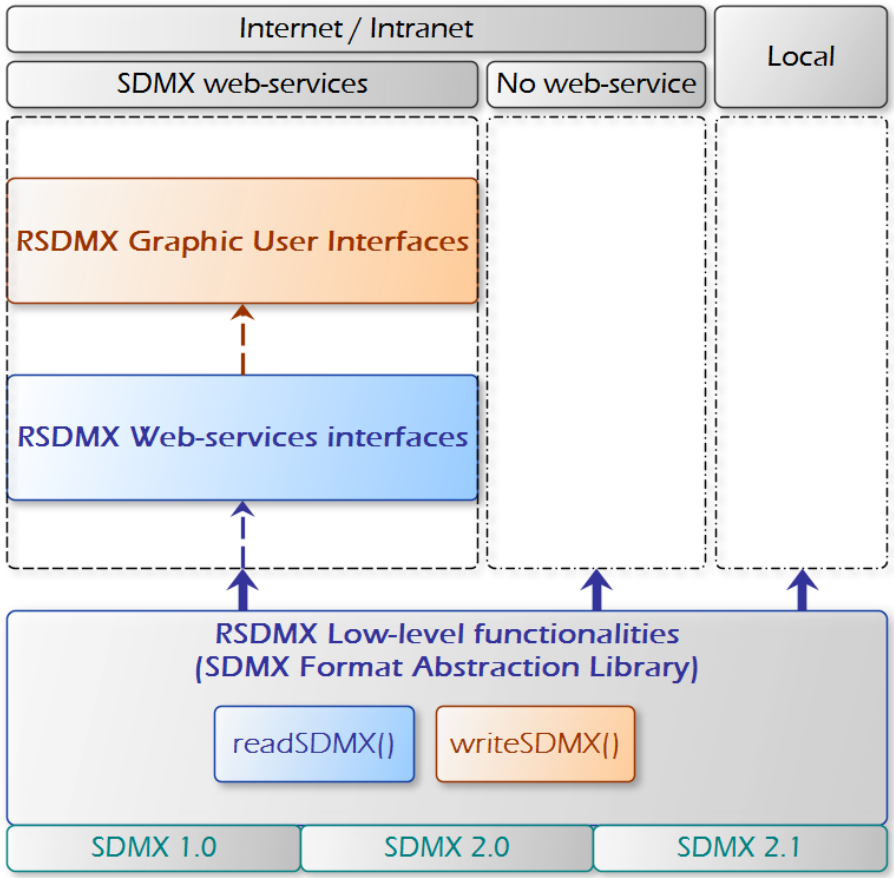
- First initiative to read SDMX in R made available to the R community
- Set of tools to read SDMX-ML data and metadata documents

rsdmx
Introduction

- Generic SDMX abstraction library for R
- Read SDMX documents in a flexible way:
  - Support for SDMX-ML 1.0, 2.0 and 2.1 standard formats
  - Support for *remote* or *local* sources,
  - No restriction to the SDMX web-services standard specifications (for *remote* sources)
- Variety of SDMX documents:
  - **Data** (generic, compact, structure-specific, etc.)
  - **Metadata** (Data structure definition - DSD, Codelists, Concepts, etc.)

rsdmx
Introduction

- a single `readSDMX` function, with a **large bandwidth of use**:
  - "raw" approach (read from *url* or *file*)
  - "helping" approach (read from a list of well-known service providers, with no need to specify the entire request)
- a set of generic methods to convert SDMX data into tabular data (`data.frame`)



Outline

rsdmx

Architecture - Object-oriented model

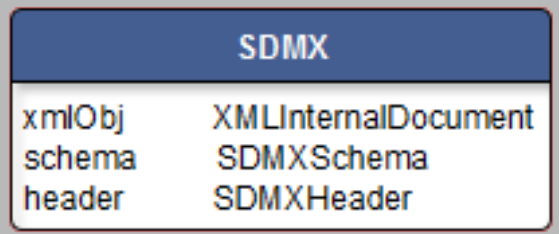
- SDMX model represented in R with S4 classes and methods
- In S4 modelling, a `class` is made of *slots* (properties)
- the general structure of SDMX-ML document is represented with an `SDMX` abstract class
- each SDMX-ML document has a corresponding R `SDMX*` class that extends the `SDMX` class

SDMX R representation

```
Class "SDMX" [package "rsdmx"]
Slots:

Name:                xmlObj        schema        header
Class: XMLInternalDocument    SDMXSchema    SDMXHeader

Known Subclasses:
"SDMXGenericData", "SDMXCompactData", "SDMXMessageGroup",
"SDMXConcepts", "SDMXCodelists", "SDMXDataStructures",
"SDMXDataStructureDefinition"
```



rsdmx

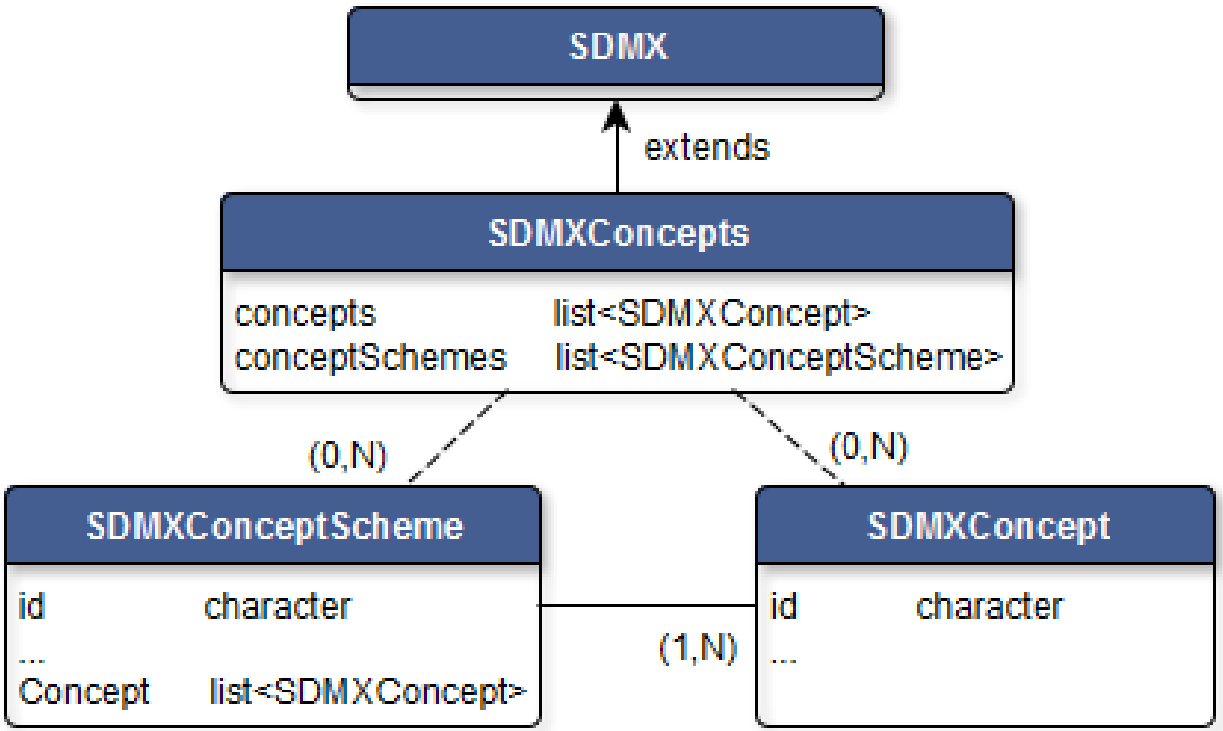
Architecture - Supported SDMX-ML documents

- *Structure* types, *i.e.* the elements that define the data structure, including:
  - `Concepts`: characteristics of a statistical dataset (dimensions, attributes, measures)
  - `Codelists`: description of a *dimension* with a list of codes and labels
  - `Datastructures`: description of the dataset structure
  - `Data Structure Definitions (DSD)`: complete description of a data structure including the 3 previous types
- *Dataset* types:
  - `GenericData`: generic SDMX data format
  - `CompactData`: compacted data format
  - `StructureSpecificData`: structure specific data format
  - `UtilityData`: utility data format
  - `MessageGroup`: specific message type developed to enable the exchange of several data or metadata messages of a single type in a unique SDMX-ML document. Currently enabled for

rsdmx

Architecture - Object-oriented model (SDMX Concepts)

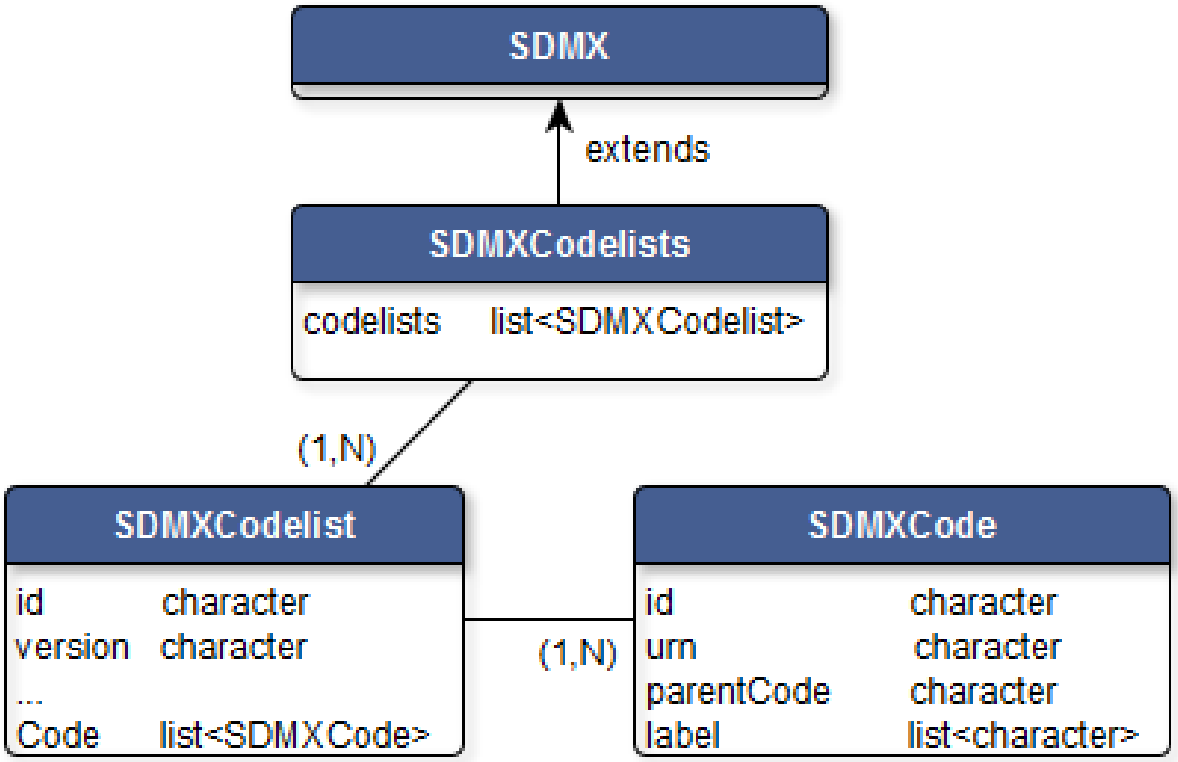
- an `SDMXConcepts` object handles concepts either through `concepts` or `conceptSchemes` (depending on the SDMX version)
- each concept is modeled with the `SDMXConcept` class



rsdmx

Architecture - Object-oriented model (SDMX Codelists)

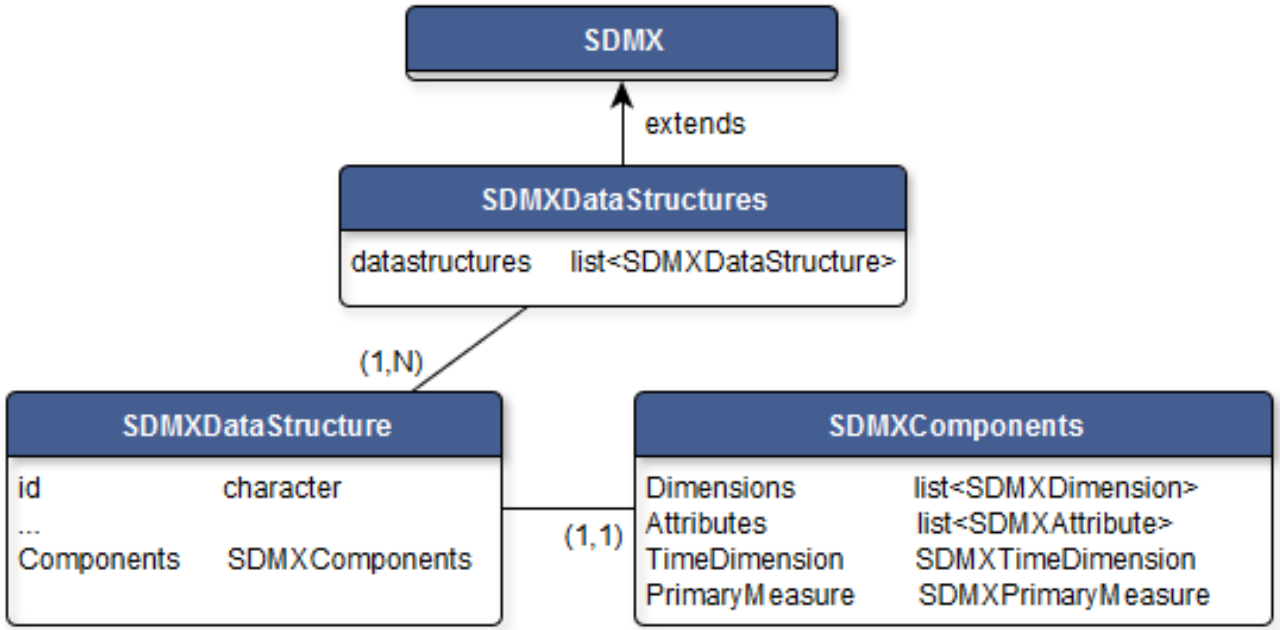
- an `SDMXCodelists` object handles one or more codelists
- each codelist is modeled with the `SDMXCodelist` class. It includes a list of `SDMXCode`



rsdmx

Architecture - Object-oriented model (SDMX Data structures / Key Families)

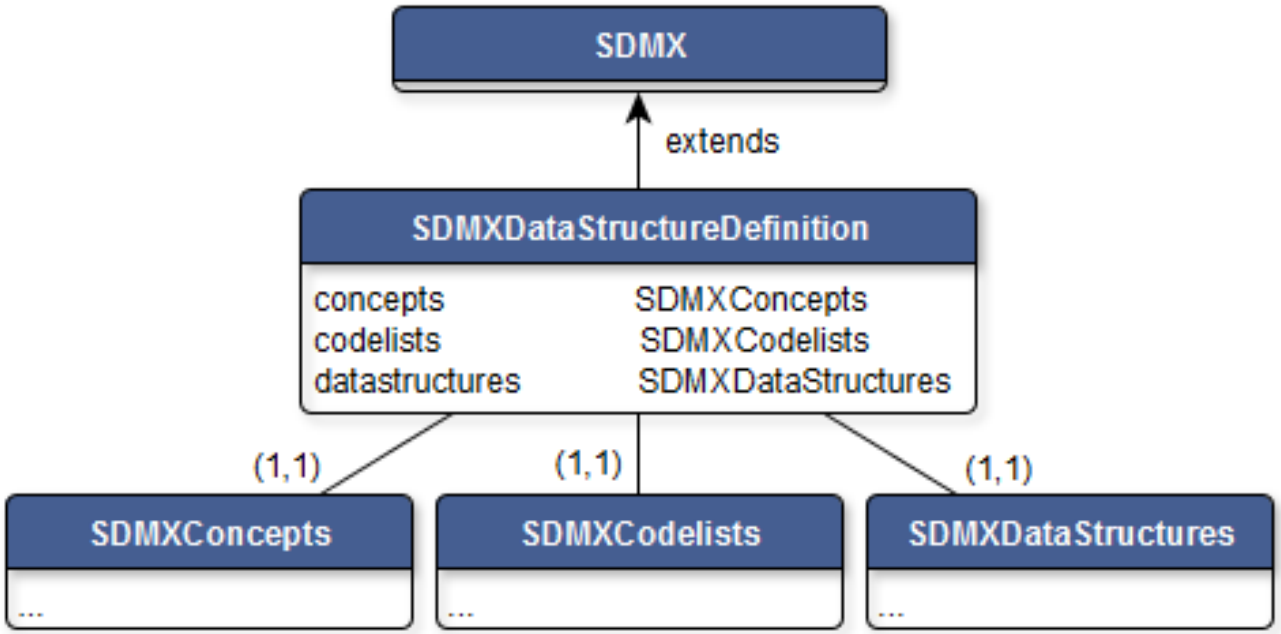
- an `SDMXDataStructures` object handles one or more data structures (or key families)
- each data structure is modeled with the `SDMXDataStructure` class. It includes a `SDMXComponents` object handling the dimensions, attributes, time dimension and measure



rsdmx

Architecture - Object-oriented model (SDMX Data structure Definition - DSD)

an `SDMXDataStructures` object handles concepts, codelists and data structures.



rsdmx

Architecture - readSDMX end-user function

`readSDMX` is the main function of `rsdmx` package. The function will do the following:

- download the SDMX-ML document
- determine the SDMX-ML message type and instantiate the corresponding `SDMX*` object
- in case of *Structure* message types, parse completely the document into a `S4` sub-model specific to the message type

rsdmx

Architecture - XML Parsing technics & strategies

2 different parsing technics:

- Initial and current technic: using **XPath**
  - requires loading the XML document tree in R
  - can cause R memory issues with large SDMX-ML documents
- Alternative approach (in factory): using the **Simple API for XML (SAX)**
  - does not require loading the XML document tree in R
  - avoids R memory issues with large SDMX-ML documents
- capacity to parse remote or local SDMX-ML files

2 different parsing strategies:

- for *Structure* types: when instantiating the `SDMX*` object (done by `readSDMX`)
- for *Dataset* types: when coercing the `SDMX*` object to a `data.frame` (done by `as.data.frame`)

Outline

rsdmx

Usage - Installing rsdmx

rsdmx can be installed:

- from CRAN

```
R> install.packages("rsdmx")
```

- from Github (requires `devtools` package)

```
R> require(devtools)
R> install_github("opensdmx/rsdmx")
```

Load rsdmx in R using:

```
R> require(rsdmx)
```

rsdmx

Usage - datasets

Read a SDMX generic dataset in R using the *raw* approach.

```
R> url <- paste("http://data.fao.org/sdmx/repository/data/CROP_PRODUCTION/",
                ".156.5312../FAO?startPeriod=2008&endPeriod=2008", sep="")
R> sdmxObj <- readSDMX(url)
R> class(sdmxObj)
```

```
[1] "SDMXGenericData"
attr(,"package")
[1] "rsdmx"
```

Convert the `SDMXGenericData` into tabular data (`data.frame`)

```
R> myData <- as.data.frame(sdmxObj)
R> head(myData)
```

```
  FREQ REF_AREA INDICATOR COMMODITY DOMAIN UNITS UNIT_MULTIPLIER obsTime obsValue OBS_STATUS
1 YEAR      156      5312       515      Q    No             1000    2008     8832       <NA>
2 YEAR      156      5312       526      Q    No             1000    2008      450          E
3 YEAR      156      5312       367      Q    No             1000    2008      700          E
4 YEAR      156      5312       572      Q    No             1000    2008     4000          E
5 YEAR      156      5312        44      Q    No             1000    2008    67435       <NA>
6 YEAR      156      5312       414      Q    No             1000    2008      730          E
```

rsdmx

Usage - concepts

Read a SDMX concepts document in 

```
R> url <- paste("http://data.fao.org/sdmx/registry/conceptscheme/FAO/",
                "ALL/LATEST/?detail=full&references=none&version=2.1", sep="")
R> sdmxObj <- readSDMX(url)
R> class(sdmxObj)
```

```
[1] "SDMXConcepts"
attr(,"package")
[1] "rsdmx"
```

Convert the SDMXConcepts into tabular data (data.frame)

```
R> concepts <- as.data.frame(sdmxObj)
R> head(concepts[,c("id","en")])
```

```
               id                       en
1       COMMODITY                COMMODITY
2       INDICATOR                INDICATOR
3        REF_AREA                 REF_AREA
4          DOMAIN                   DOMAIN
5     UNIT_MEASURE              UNIT_MEASURE
6            FREQ                     FREQ
7   FAO_MAJOR_AREA            FAO Major Area
8       UN_COUNTRY                UN Country
9      ENVIRONMENT               Environment
10         SPECIES ASFIS Species Alpha 3 Code
11       OBS_VALUE                OBS_VALUE
12      OBS_STATUS               OBS_STATUS
```

rsdmx

Usage - codelists

Read a SDMX codelists document in 

```
R> clUrl <- paste("http://data.fao.org/sdmx/registry/codelist/FAO/",
                  "CL_FAO_MAJOR_AREA/0.1", sep="")
R> clObj <- readSDMX(clUrl)
R> class(clObj)
```

```
[1] "SDMXCodelists"
attr(,"package")
[1] "rsdmx"
```

Convert the SDMXCodelists into tabular data (data.frame)

```
R> codelist <- as.data.frame(clObj)
R> head(codelist[,c("id", "label.fr", "label.es")])
```

```
  id                              label.fr                                  label.es
1 01          Afrique - Eaux continentales             África - Aguas continentales
2 02 Amérique du Nord - Eaux continentales América del Norte - Aguas continentales
3 03  Amérique du Sud - Eaux continentales   América del Sur - Aguas continentales
4 04             Asie - Eaux continentales              Asia - Aguas continentales
5 05          Europe - Eaux continentales           Europa - Aguas continentales
6 06          Océanie - Eaux continentales          Oceanía - Aguas continentales
```

rsdmx

Usage - Data structures

Read a SDMX Data Structure Definitions (DSD) document in R

```
R> dsdUrl <- "http://stats.oecd.org/restsdmx/sdmx.ashx/GetDataStructure/TABLE1"
R> dsd <- readSDMX(dsdUrl)
R> class(dsd)
```

```
[1] "SDMXDataStructureDefinition"
attr(,"package")
[1] "rsdmx"
```

Get the codelists contained in this DSD...

```
R> cls <- slot(dsd, "codelists")
R> codelists <- sapply(slot(cls, "codelists"), function(x) slot(x, "id"))
R> codelists
```

```
 [1] "CL_TABLE1_OBS_STATUS"  "CL_TABLE1_DAC_DONOR"   "CL_TABLE1_DAC_PART"
 [4] "CL_TABLE1_TRANSACTYPE" "CL_TABLE1_FLOWS"       "CL_TABLE1_DATATYPE"
 [7] "CL_TABLE1_TIME"        "CL_TABLE1_UNIT"        "CL_TABLE1_POWERCODE"  "CL_TABLE1_TIME_FORMAT"
```

...and convert one codelist into tabular data (data.frame)

```
R> cl <- as.data.frame(slot(dsd, "codelists"),codelistId = "CL_TABLE1_FLOWS")
R> cl
```

```
    id          label.fr            label.en
1 1121   Versements Dons Grant Disbursements
2 1122  Versements PrOts  Loan Disbursements
3 1120  Versements Bruts Gross Disbursements
```

rsdmx

Usage - readSDMX with the helping approach

`rsdmx` now brings the capacity to query data from a set of well-known data providers, still using the single `readSDMX` function. `rsdmx` embedds a list of SDMX service providers by default.

The list of data providers "known" by `readSDMX` can be queried as follows:

```
R> providers <- getSDMXServiceProviders()
R> sapply(providers, slot, "agencyId")
```

```
[1] "ECB"   "ESTAT" "OECD"  "FAO"   "ILO"
```

The following example shows how to use `readSDMX` based on one of known data provider, OECD:

```
R> sdmx <- readSDMX(agencyId = "OECD", operation = "GetData", key = "MIG",
                                        filter = list("TOT", NULL, NULL),
                                        start = 2011, end = 2011)
```

rsdmx

Usage - readSDMX with the helping approach

It is also possible to **add your own SDMX service provider**, and make it "known" by `readSDMX`!

If you are interested, you can checkout the `rsdmx` documentation available online!

If you want to register your SDMX service endpoint in the default list of providers, please contact me.

Outline

rsdmx

Success stories - Variety of datasources

Used on a variety of datasources:

- with **international** & **regional** data sources: UN data portal, UN Food & Agriculture Organization (FAO), UN International Labour Organization (ILO) Organisation for Economic Co-operation and Development (OECD), EUROSTAT, European Central Bank (ECB), International Monetary Fund (IMF), World Bank
- with **national** data sources: Australian Bureau of Statistics (ABS), UK's Office of National Statistics (ONS), Deutsche Bundesbank, INSEE (France), and more!!

rsdmx

Success stories - Use in Projects

Used in different web projects, such as:

- the iMarine data e-infrastructure: within R statistical analysis processings made available through Web Processing Services (WPS).
- the Live Labour Force project: to allow reading SDMX datasets from the Australian Bureau of Statistics (ABS) portal (ABS.Stat).
- the SYRTO project

rsdmx
Perspectives

- enable the Simple API for XML (SAX) parsing technic for large datasets
- improve the existing functionalities, *e.g.* dataset time dimension format
- support for additional SDMX document types and formats
- extend the embedded list of SDMX service providers (`rsdmx` as web-service interface)
- develop a generic SDMX web-client with the `shiny` R web-framework (http://shiny.rstudio.com)

rsdmx
Looking for Sponsors

`rsdmx` can play a **fundamental role** for **exploiting** and **co-analyzing statistics** from scattered data sources in R.
Until now, `rsdmx` was born from a **voluntary initiative**, and is now a **published library** with a **growing number of users**. To guarantee the sustainability of `rsdmx`, we are seeking for **fundings**, throught *sponsoring* or *donations* to:

- **implement, test, validate and release improvements**
- **guarantee a quality maintenance** of the package
- **provide support** to users & institutions that take advantage of `rsdmx`

If you are interesting in supporting `rsdmx`, please do not hesitate to contact me!

rsdmx
on the web

- on Github:
  - source code: https://github.com/opensdmx/rsdmx
  - online documentation: https://github.com/opensdmx/rsdmx/wiki
- mailing list: rsdmx
- on the Comprehensive R Archive Network (CRAN): http://cran.r-project.org/package=rsdmx