Fusion Metadata Registry – validating SDMX data
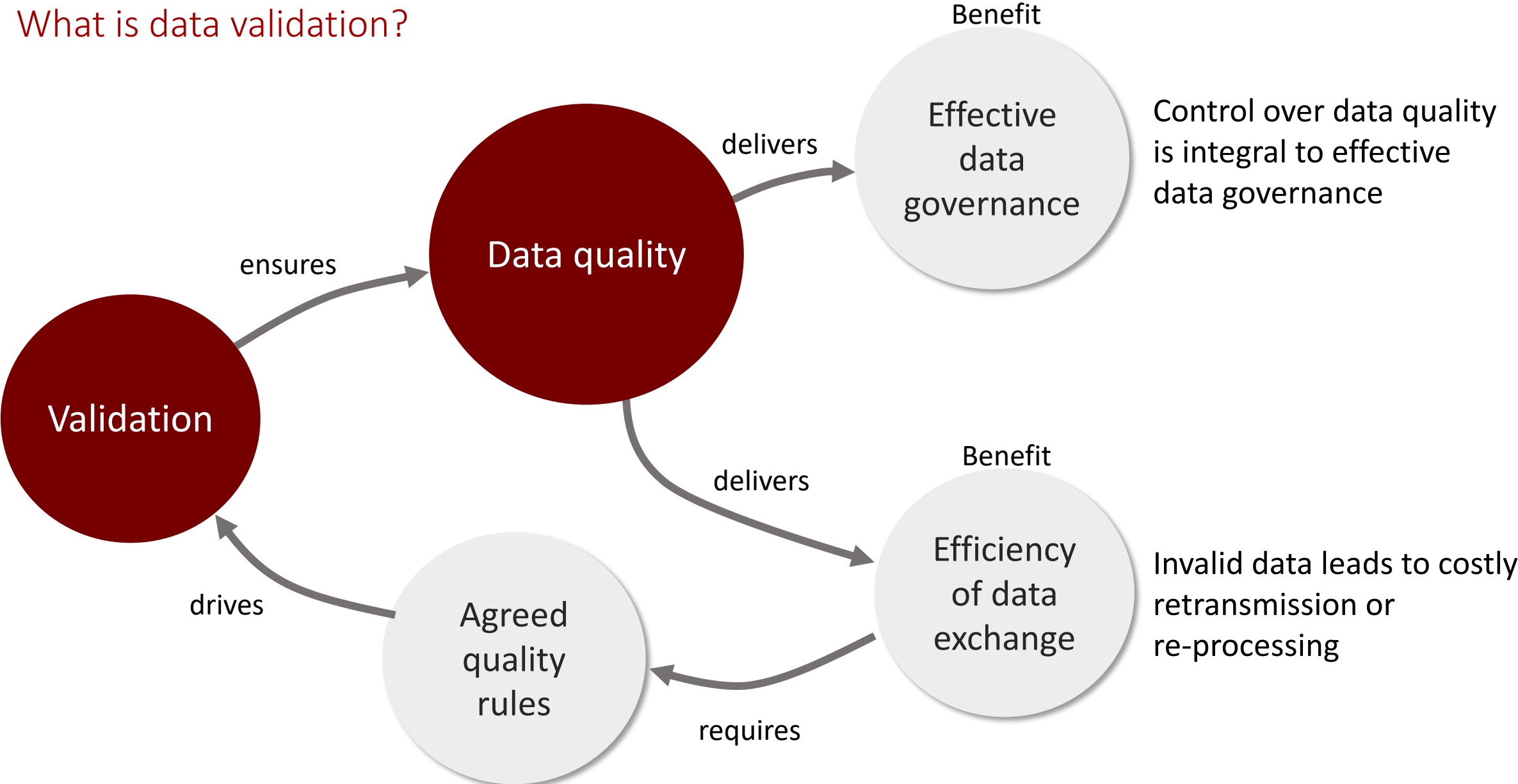
An introduction to SDMX data validation using Fusion Metadata Registry
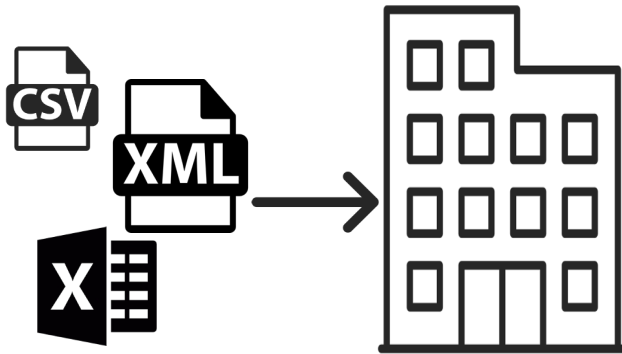
Glenn Tice

Topics

- FMR data validation use cases

- Validating data interactively using FMR's web user interface

- The validation rules available in FMR

- Using Concept representations and Constraints to define the universe of valid data

- Checking balance equalities using Validation Schemes

What is data validation?

Validation —ensures→ Data quality

Data quality —delivers→ Effective data governance (Benefit)

Control over data quality is integral to effective data governance

Data quality —delivers→ Efficiency of data exchange (Benefit)

Invalid data leads to costly retransmission or re-processing

Efficiency of data exchange —requires→ Agreed quality rules

Agreed quality rules —drives→ Validation

FMR data validation use cases

Data Reporting

Data producers use FMR to validate reporting data using the collectors' rules prior to transmission
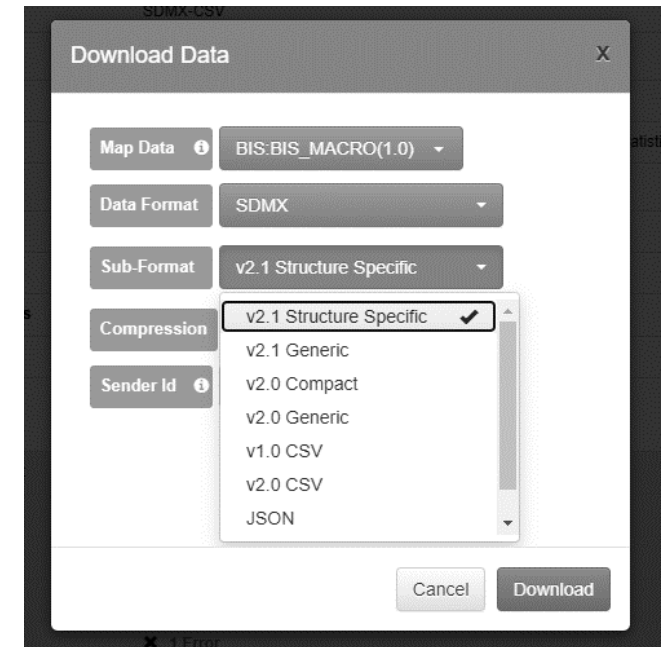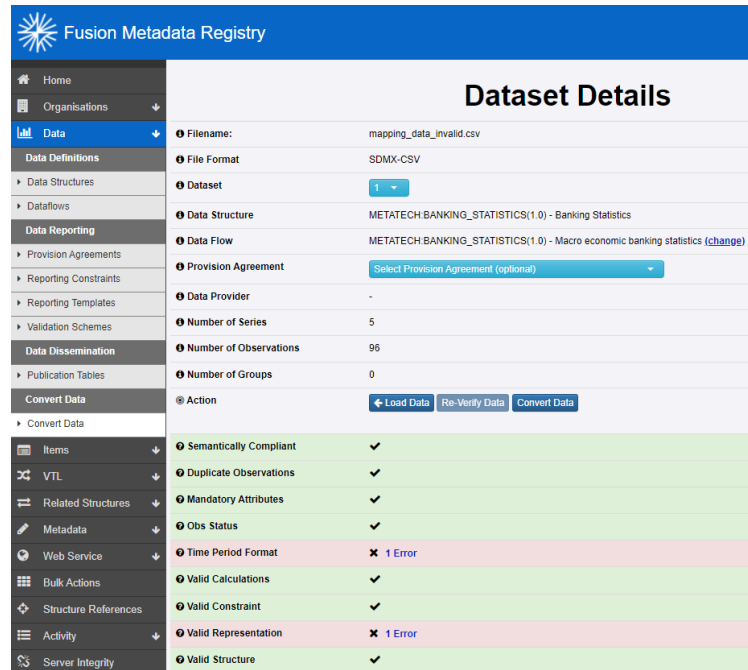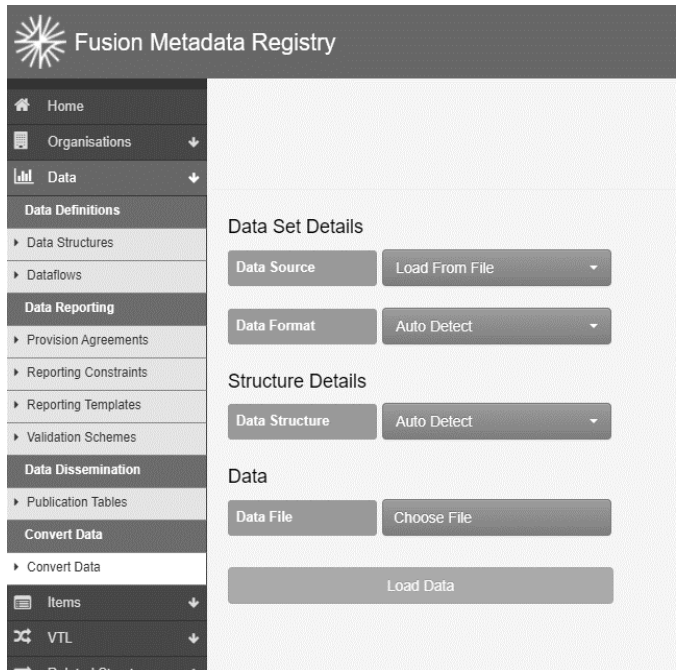
Data Collection

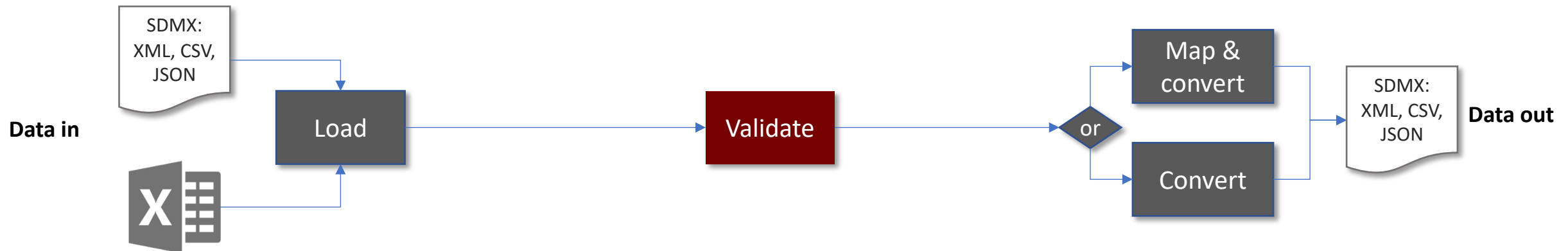Data collectors validate data received from data reporters

Statistics Production

Set and enforce dataset quality rules

Using FMR's web UI to validate a data set

FMR's nine validation rules – a quick overview

Rules		Test applied
❓ Semantically Compliant	✔	The XML, JSON, CSV or Excel is well formed
❓ Duplicate Observations	✔	Uniqueness - there is only one observation value reported for each time period
❓ Mandatory Attributes	✔	All mandatory attributes are reported
❓ Obs Status	✔	OBS_STATUS is consistent with the observation value
❓ Time Period Format	✔	E.g. FREQ=M means the TIME_PERIOD format must be YYYY-MM
❓ Valid Calculations	observation consistency	Balance equalities defined using Validation Schemes
❓ Valid Constraint	valid universe of data ($\mathbb{D}$)	The data is within the universe defined by **Data Constraints**
❓ Valid Representation		Each component complies with the **representation** defined in the DSD
❓ Valid Structure	✔	The dimensions and attributes are consistent with the DSD

Defining the valid universe of data ($\mathbb{D}$) – DSD component representations

An example DSD – BIS Macro-economic series

Dimensions
Frequency
BIS economic phenomenon
Reference area
BIS suffix
Measures
Observation value
Attributes
Observation comment

Infinite universe of data
Everything is valid!

Representation ✗
String
String
String
String
String
String

The dataset's universe is more specific

Representation ✓
Codelist: CL_FREQ
Codelist: CL_BIS_TOPIC
Codelist: CL_REF_AREA
Codelist: CL_BIS_SUFFIX
Number: $\mathbb{R} > 0$
String: $1 \leq \text{length} \leq 1050$

Setting DSD component representations in practice using FMR

Choose between coded (enumerated) or un-coded (described)

For coded components, pick the Codelist to use from those available in the registry

For coded representation, SDMX also allows further restriction rules to be applied, e.g. only codes that start with a specific string

Further refining the valid universe of data ($\mathbb{D}$) – Data Constraints

- A dataset's universe of valid data can still be large even with carefully designed representations

- **SDMX Data Constraints** allow further restrictions on the valid universe

- Constraints come in two flavours:
  - Cube Region
  - Series

Constraints can be attached at four different levels:

DSD

Dataflow

Data provider

Provision Agreement

More specific

Further refining the valid universe of data (𝔻) – Data Constraints

Two common use cases for Data Constraints

1. **Restrict the domain for a specific Dataflow**

- Problem: generic DSD that can be used for different datasets
  e.g. World Bank World Development Indicators DSD [WB:WDI(1.0)](WB:WDI(1.0))

- Solution: Add constraint to the <u>Dataflow</u> to make the domain specific

2. **Restrict what specific data providers can report**

- Problem: Each data provider must only report certain values

- Solution: Add a constraint to each <u>Provision Agreement</u>

Defining Data Constraints in practice using FMR

Set the valid values for each component:
- Included values
- Excluded values

Reporting Constraint Wizard

OECD:CC_SMEE_OUTLOOK(1.0)

| 1. Details | 2. Constrained Structure | 3. Included Values | 4. Excluded Values |

Select one of the four attachment levels

Structure Type — Dataflow

Constrained Structure(s)

OECD:DF_SMEE_OUTLOOK(1.0)

Add New — Remove Selected

Cube Region constraint example

Choose which components to constrain

Select Components to Include in Constraint

Selected	Component Id	Component Name	Codelist	Type
✓	COU	Country	CL_SMEE_OUTLOOK__COU[1.0]	Dimension
✓	INDICATOR	Indicator	CL_SMEE_OUTLOOK__INDICATOR[1.0]	Dimension
✓	YEAR	Year	CL_SMEE_OUTLOOK__YEAR[1.0]	Dimension
☐	OBS_STATUS	Observation Status	CL_SMEE_OUTLOOK__OBS_STATUS[1.0]	Attribute
☐	UNIT_MEASURE	Unit of Measures	CL_SMEE_OUTLOOK__UNIT_MEASURE[1.0]	Attribute
☐	UNIT_MULT	Multiplier	CL_SMEE_OUTLOOK__UNIT_MULT[1.0]	Attribute

Showing 1 to 6 of 6 entries   3 rows selected

Search:

Checking data set 'balance equalities' using FMR Validation Schemes

In some datasets, reported observations must be in balance

Dimension   REF_AREA
Balance rule   EUR = DE + FR + ES + IT

REF_AREA	2019	2020	2021
DE	5	6	4
FR	3	4	5
ES	7	5	5
IT	2	7	2
EUR	17	20	16

Reported values

Balance equality   EUR = DE + FR + ES + IT   ✅   ❌   ✅

20≠22

Defining balance equalities in practice using FMR Validation Schemes

Validation Scheme Wizard

BIS:VS1(1.0)

1. Details	2. Attachment	3. CSV Import	4. Expression builder

This step is optional. Use this step to import custom validation expressions. The CSV may include quotes for each value, example "AN_ID","A Name","A description". Any CSV rules will be added in addition to any rules which may exist for this Validation Scheme.

Each validation expression can contain 'meta' information such as an **Id**, **Name**, and **Dimension Id**.

The rules' expression is then broken down into: **Dimension Id** on which the rule is operating, e.g REF_AREA; **Output**, either numerical, or Code Id in square brackets e.g [EUROPE]; **Equality Operator**, valid values are =,<>,<,<=,>,>=; and **Expression**, where each Code Id in the expression must be placed in square brackets, for example [UK]+[FR]+[DE]. Valid operators are +,-,/,* and brackets '()' are supported.

Rule Details	Column Index	Rule Expression	Column Index
Rule Id	1	Dimension Id	3
Rule Name	2	Output	4
Rule Description		Equality Operator	5
		Expression	6

REF_AREA_BALANCE,Europe balance check,REF_AREA,[EUR],=,"[DE]+[FR]+[ES]+[IT]"

Equalities are defined for specific dimensions in a DSD - REF_AREA in this example

An expression defines the balance calculation – standard arithmetic operators are allowed (+ - / *)

Balances of the following form are also allowed:
0 = [EUR] − ([DE]+[FR]+[ES]+[IT])

We saw:

- FMR data validation use cases

- Validating data interactively using FMR's web user interface

- The validation rules available in FMR

- Using Concept representations and Constraints to define the universe of valid data

- Checking balance equalities using Validation Schemes

Questions

BIS MED IT
Glenn Tice
glennphilip.tice@bis.org

FMR data validation follow-on topics

- Series Constraints

- + and % operators for efficiently defining Series Constraints

- Code validity periods – Constraints, and at the code level

- Balance equalities using Hierarchical Codelists (Hierarchies in SDMX 3.0)

- Automating data validation using FMR's REST API

- Using FMR SDMX Codelists for validation in R

References

FMR Docker image   https://www.sdmx.io/resources/containers/fmr-docker-mysql/

Download FMR   https://www.sdmx.io/resources/download/fmr/

FMR product page   https://www.sdmx.io/tools/fmr/

FMR Wiki – general reference   https://fmrwiki.sdmxcloud.org/Main_Page

Data Validation Cookbook   https://data-cleaning.github.io/validate/