Open Data Management in Agriculture and Nutrition

*This e-learning course is the result of a collaboration between **GODAN Action** partners, including **Wageningen Environmental Research** (WUR), **AgroKnow**, **AidData**, **the Food and Agriculture Organization of the United Nations** (FAO), **the Global Forum on Agricultural Research** (GFAR), and **the Institute of Development Studies** (IDS), **the Land Portal**, **the Open Data Institute** (ODI) and **the Technical Centre for Agriculture and Rural Cooperation** (CTA).*

GODAN ACTION
Global Open Data
for Agriculture & Nutrition

UKaid
from the British people

GODAN Action is a three-year project UK's Department for International Development to enable data users, producers and intermediaries to engage effectively with open data and maximise its potential for impact in the agriculture and nutrition sectors. In particular we work to strengthen capacity, to promote common standards and best practice and to improve how we measure impact. [www.godan.info]

UNIT 2: USING OPEN DATA

LESSON 2.2: QUALITY AND PROVENANCE

Photo by U.S. Department of Agriculture licensed under CC BY 2.0

Aims and learning outcomes

Open data becomes usable when a human can understand it and a machine can manipulate it. Users of open data need the permission of its publisher, granted by an open licence. But the open licence alone is not enough to guarantee the quality and usability of the data. It is unlikely that people will be able to engage with data to derive insights and demonstrate benefits unless it is usable.

After studying this lesson, you should be able to:

- *list, describe, and apply the factors that affect usability of open data*
- *use tools that help you evaluate usability of open data*
- *sross check your data against other data sources*
- *identify the provenance of an open dataset (for machine-readable/human-readable)*
- *describe what useable data is from different points of view*

Contents

List of figures

List of tables

1. Introduction

Quality and provenance are two important aspects that determine the usability of a dataset. This lesson takes a broad look at the different aspects that make a quality dataset and a number of the best practice guidelines that aid the publication of high-quality usable data.

Part of the quality of a dataset is dictated by the history, or provenance, of that dataset. Knowing that the data is from a reliable source and that it was collected using reliable methods (or via methods with known constraints) can often be more important than having a dataset with a well controlled vocabulary or schema.

While data is technical in nature, not all quality measures are technical; ensuring a dataset is machine readable doesn't always mean it is usable; not all quality requirements are technical. The evolution of best practice guidelines reflects this and in this lesson we look at the technical and non-technical aspects that make a high-quality usable dataset.

2. Quality marks and measures

Assessing the quality of open data cannot be done quickly. There are a number of community-based standards and quality marks that can help you assess the quality and usability of data.

One of the first quality marks to emerge for open data is the '5 stars of linked open data'. Awarding of each star is sequential and starts with the requirement to apply an open license to data. The remaining stars are split into two that focus on open data being available 'on the web' to download and two that focus on data being 'in the web' to use via an API which can instantly retrieve resources. Other than the aspect regarding licensing, the 5-star guideline is focused on the technical availability of data and lacks the non-technical aspects that make up quality usable data.

Similarly the FAIR principles[1] provide a similar guideline for publishers. According to these principles data should be: **Findable, Accessible, Interoperable and Reusable (FAIR)**. While they largely focus again on technical aspects, the principles do however set out some terms relating to provenance, stating that 'Published Data Objects should refer to their sources with rich enough metadata and provenance to **enable proper citation'**.

Both the 5-star schema and the FAIR principles are outlined in more detail in Unit 4: Exchanging Open Data. This lesson is going to take a look at the other

[1] Force11. Guiding Principles For Findable, Accessible, Interoperable And Reusable Data Publishing Version B1.0. https://www.force11.org/fairprinciples

aspects of usability and quality as defined by the Open Data Institutes Open Data Certificates.

3. Open data certificates

The Open Data Certificate is a free online tool developed and maintained by the Open Data Institute, to assess and recognise the sustainable publication of quality open data. It addresses the legal, practical, technical and social aspects of publishing open data using best practice guidance.

Like FAIR, the Open Data Certificates[2] process takes an alternative but complementary view to the 5-star scheme. A certificate measures how effectively someone is sharing a dataset for ease of reuse. The scope covers more than just technical issues[3] including rights and licensing, documentation, and guarantees about availability. A certificate therefore offers a more rounded assessment of the quality of publication of a dataset.

For data publishers the process of assessing a dataset provides insight into how they might improve their publishing process. The assessment process is therefore valuable in itself, but the certificate that is produced is also of value to reusers.

3.1. Being a data reuser

For data users, the technical quality at the point of use might be enough for their particular use. However, for reusers sustainability and support are likely to be more dominant in the decision-making process than file format.

Reusers need to be offered assurance that their access to the data will be consistent and reliable. An open data certificate challenges publishers to think beyond the data to address key policy considerations in the support of the data. These considerations are broken down into four categories crucial for reusers.

3.2. A reuser's checklist

Using the open data certificates as a guide, the following presents a reuser's checklist for open data. The checklist is split into four categories reflecting the sections of the Open Data Certificate.

Legal

- Is the data openly licensed and legally usable?
- Is the data model, format or structure also openly licensed and legally usable?

[2] https://certificates.theodi.org

[3] https://certificates.theodi.org/about

- Are copyright statements clear?
- Are any data or parts of the data that are not openly licensed described?
- Are any privacy and potential ethical constraints to the data use outlined?

Practical

- Is the data well described?
- Is the reason the data is collected clear?
- Is the publisher's use for the data clear?
- Are any other existing uses of the data outlined?
- Is the data accessible?
- Is the data timestamped or up to date?
- Will the data be available for at least a year?
- Will the data be updated regularly?
- Is there a quality control process?

Technical

- Is the data available in a format appropriate for the content?
- Is the data available from a consistent location?
- Is the data well structured and machine readable?
- Are complex terms and acronyms in the data defined?
- Does the data use a schema or data standard?
- Is there an API available for accessing the data?

Social

- Is there an existing community of users of the data?
- Is the data already relied upon by large numbers of people?
- Is the data officially supported?
- Are service level agreements available for the data?
- It is clear who maintains and can be contacted about the data?

If a published has completed an open data certificate and applied the quality mark to their data then a reuser can quickly find out the answers to all the questions on the reuser's checklist above. Alternatively a complete list of all certificates is available in the certified datasets registry[4].

3.3. A reuser's guide to provenance

All of the best practice guidelines introduced so far are somewhat focused on data producers, who already own and manage the source data. However, not all open data originates with the publisher; a vast amount of data is derived from other data. For example a weather forecast is derived by applying complex models to meteorological data. With there being many sources of meteorological data and many organisations that use different models to

[4] https://certificates.theodi.org/en/datasets

create a forecast, this can lead to situations where even forecasts based upon the same input data can be vastly different, with potentially devastating consequences[5].

3.3.1. Provenance checklist

The checklist below will help established the provenance of a dataset and help establish the level of trust in that dataset.

- Is the data wholly owned and produced by the data provider?
- Does anyone else produce comparable data for cross checking?
- Is it clear if the data has been derived from other sources of data?
- Are the other sources of data clear?
- Are the other sources of data trustworthy and comparable with other data providers?
- Is it clear if and how any data has changed (from any source) prior to being made available as open data at your point of access?

Following these points will help establish how trustworthy different open data sources are. It may even reveal potential to bypass the current data source and follow the trail of provenance back to an original source that may be more trustworthy or offer a more completed and/or supported data service.

4. Post-access quality checking

Establishing trust in a data provider, potentially obtaining a service agreement and then accessing the data is not the end of the quality checking process. Once access to the data has been obtained it is essential to verify the data, especially with your own use in mind.

This stage of checking and preparing data to be ready for use has many aspects, most of them based on technical processes. It is more than likely that problems or inconsistencies will be found in the data at this stage. This is when it is crucial to be connected to the community or provides and/or have access to quality control procedures to help with your understanding of the data or to fix these problems.

4.1. Data validation

One of the first things to do with the data is to verify it against any available data schema and description of the data structure. This will reveal potential omissions from the documentation for terms which have emerged in the data. Verifying the data against the schema will also help verify that the correct data exists within the dataset.

[5] http://www.bbc.co.uk/news/av/world-24713504/michael-fish-s-denial-of-a-hurricane-in-1987

If your data does not have a schema, it might be necessary to transform the data into a format where one can be easily applied or to make your own schema for the dataset.

4.1.1. Designing schemas

Schemas play a key role in enabling the wide, automated reuse of data. A schema is a blueprint for data that defines a set of integrity constraints and rules relating to the structure and contents of a data resource. A schema defines three key things as listed below.

Column/Key titles in the data: Defining a consistent set of column titles (or keys) for a dataset is essential to ensure that datasets of the same type can be merged and analysed easily. Often column titles will change or be abbreviated to save time however this causes a lot of problems when analysing data over long periods of time. Adding column titles is less of a problem but has to be taken into account when analysing data.

Value types: With the column title/keys defined it is important to define the valid data type for the values, e.g. number, text, date, coordinate etc.

Value constraints: With the value type defined, valid constraints such as being required, needing to be unique, being in a certain unit (e.g., gallons (UK)) or be within a certain range should be defined. For example a column might be entitled 'Cost (£m)'; thus any values should be numbers (without commas). Setting a valid range also help avoid and explain any errors in the data. For example setting a range of 0.001-100 on the 'Cost (£m)' (if it is known that cost cannot exceed £100m). Range validation stops people accidentally misreading the column title/units and entering 100000000 instead of 100 for £100m.

4.1.2. Using schemas

All spreadsheet packages allow you to create validation rules for data relating to each column title. However very few packages allow the exporting of schemas alongside the data for others to use without adding complex developer extensions.

One of the main reasons for this is the connection between schemas and hierarchical databases dating back to earlier database design by E F Codd (1970). Schemas are a key part of a database where multiple tables are linked with pre-defined relationships. Since this point implementations have been led by this theoretical model and applied technically in relational database packages like MySQL.

The development of the eXtensible Markup Language as a mechanism for sharing data emerged much later in 1996. Five years later in 2001, the now popular XML Schema specification was released to help formalise the sharing of consistent and verifiable data. During the mid-90s to late 2000s and still to some extent today XML was the standard of choice for the representation of exchangeable data; designed to be both machine and human readable.

The following example data (shown here in tabular form) is an example of some data taken from a dataset.

Table 1 Example data

FirstName	LastName	Instrument	DateOfBirth
John	Lennon	Vocal	1940-10-09
Paul	McCartney	Bass Guitar	1942-06-18
George	Harrison	Guitar	1943-02-25
Ringo	Starr	Drums	1940-07-07

The same data is shown below (Figure 1) in XML with associated extracts from the schema.

XML (Download)

```
<People>
  <Person>
    <FirstName>John</FirstName>
    <LastName>Lennon</LastName>
    <Instrument>Vocal</Instrument>
    <DateOfBirth>1940-10-09</DateOfBirth>
  </Person>
  <Person>
    <FirstName>Paul</FirstName>
    <LastName>McCartney</LastName>
    <Instrument>Bass Guitar</Instrument>
    <DateOfBirth>1942-06-18</DateOfBirth>
  </Person>
  <Person>
    <FirstName>George</FirstName>
    <LastName>Harrison</LastName>
    <Instrument>Guitar</Instrument>
    <DateOfBirth>1943-02-25</DateOfBirth>
  </Person>
  <Person>
    <FirstName>Ringo</FirstName>
    <LastName>Starr</LastName>
    <Instrument>Drums</Instrument>
    <DateOfBirth>1940-07-07</DateOfBirth>
  </Person>
</People>
```

XML Schema extract (Download)

```
…
<xs:simpleType name="birthsDate">
  <xs:restriction base="xs:date">
    <xs:minInclusive value="1800-01-01"/>
    <xs:maxInclusive value="2017-07-31"/>
    <xs:pattern value=".{10}"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="instrument">
  <xs:restriction base="xs:token">
    <xs:enumeration value="Vocal"/>
    <xs:enumeration value="Guitar"/>
    <xs:enumeration value="Bass Guitar"/>
    <xs:enumeration value="Drums"/>
  </xs:restriction>
</xs:simpleType>
…
<xs:complexType>
  <xs:sequence>
    <xs:element name="FirstName" type="xs:string"/>
    <xs:element name="LastName" type="xs:string"/>
    <xs:element name="Instrument" type="instrument"/>
    <xs:element name="DateOfBirth" type="birthsDate"/>
  </xs:sequence>
</xs:complexType>
…
```

Figure 1 Example data in XML format with associated extracts from schema

Such schema can be used by machines to automatically validate the structure and contents of a datasets; there is also an online validation tool[6] which can be used with the two examples from above.

While XML has seen wide adoption, 2005 was a significant year for data on the web with the emergence of Ajax (Garrett 2005). Standing for Asyncronous JavaScript and XML, Ajax is a set of web development techniques that uses JavaScript to dynamically load data into web applications. The initial goal was to allow the dynamic use of XML data in web applications, further encouraging the release of data. In practice, modern applications commonly substitute JSON (JavaScript Object Notation) in place of XML due to the advantages of JSON being native to JavaScript, 21% faster to work with as a result and substantially less verbose.

Similarly to XML, JSON provides a JSON Schema[7] specification; however this is currently not as fully developed at the XML Schema specification and lacks the ability to validate a range of inputs (e.g. minimum and maximum values permitted). Figure 2.2.2 below shows the same data as the XML example, this time in JSON with the equivalent JSON schema.

<table>
<tr><th>JSON</th><th>JSON Schema (Download)</th></tr>
<tr><td>

```
[
  {
    "FirstName": "John",
    "LastName": "Lennon",
    "Instrument": "Vocal",
    "DateOfBirth": "1940-10-09"
  },
  {
    "FirstName": "Paul",
    "LastName": "McCartney",
    "Instrument": "Bass Guitar",
    "DateOfBirth": "1942-06-18"
  },
  {
    "FirstName": "George",
    "LastName": "Harrison",
    "Instrument": "Guitar",
    "DateOfBirth": "1943-02-25"
  },
  {
    "FirstName": "Ringo",
    "LastName": "Starr",
    "Instrument": "Drums",
    "DateOfBirth": "1940-07-07"
  }
]
```

</td><td>

```
{
  "fields": [
    {
        "name": "FirstName",
        "type": "string",
        "constraints": {
            "required": true
        }
    },
    {
        "name": "LastName",
        "type": "string",
        "constraints": {
             "required": true
        }
    },
    {
        "name": "Instrument",
        "enum": ["Vocal", "Guitar", "Bass Guitar", "Drums"],
        "constraints": {
            "required": true
        }
    },
    {
        "name": "DateOfBirth",
        "type": "string",
        "format": "date-time"
    }
  ]
}
```

</td></tr>
</table>

[6] http://www.utilities-online.info/xsdvalidation/#.WhK94rZ0cUs

[7] http://json-schema.org

Figure 2 Example data in JSON format with associated extracts from schema

Shown here is the JSON schema for our tabular data file. Try it for yourself by downloading the tabular data[8] (in CSV) and schema[9] (in JSON) and then using csvlint[10] (a tabular data validator) to validate that the data conforms to the schema.

One of the challenges with both these formats is the requirement to have two files: the data and the schema. This leads to the predictable situation where the data is maintained and shared, but the schema gets forgotten and lost. The use of namespaces and linked data as a technical solution goes some way to providing a solution however the main problem still lies with the lack of integration of such standards in off-the-shelf data-management packages.

Spreadsheet packages like Excel tend to make formatting data and setting up validation rules overly complex, as proven with the ease of setting them up in tools like airtable[11]. Again, however, exporting of the schema is not currently possible, neither is the use of namespaced and linked schemas.

The same is true for database software and the move towards noSQL database structures that are not controlled by tightly defined relational schema. While the move to new flat database structures is good for speed when managing big datasets, the use of namespaces to define schemas is still lacking.

Schemas clearly have their advantages and should be adopted where organisations rely heavily on high quality data.

4.2. Cleaning data

One of the biggest challenges when working with any data is dealing with errors. Often errors are not even noticed by data publishers because the data can change over many years. In other cases, errors can be the result of human mistakes in data entry, such as mistyping or incorrect abbreviations.

Even when a schema is available, errors and inconsistencies may exist in the data. When working with any data, it is important to know how to find errors and correct them to make the data more useful.

[8] http://odedu.learndata.info/D5/course/en/assets/d466f522c9b280bdf8e0050e0899efe7cb184c9d.csv

[9] http://odedu.learndata.info/D5/course/en/assets/e367089a2d91deea7449638787755946d8b5a91e.json

[10] http://csvlint.io

[11] https://airtable.com

This section introduces a number of different examples of errors and inconsistencies in data, outlines which can be fixed with schema validation and which need a more advanced tool like Open Refine[12].

4.2.1. Wrong date formats

Dates can be written in inconsistent ways and according to different standards. One of the biggest confusions exists between the US and UK ways of writing dates. In the US the month comes first, then the day (e.g. 12/30/2017), whereas it is the other way around in the UK (30/12/2017). This is easy to spot when the day is greater than the 12th, however can cause confusion otherwise (e.g. 6/7/2017?).

The ISO 8601 standard specifies a series of rules for writing dates and times to solve this and other problems. ISO 8601 specifies that dates must be written year first (e.g. YYYY-MM-DD HH:ii:ss). Not only is this still easy to read, it also works as a way to sort in date order with the most significant sort factor going first.

As dates are complicated, efforts have been made over the years to hide the management of dates from users. For example if you type '8-7' in any general cell in Excel it will automatically translate this into a date and save '08-Jul' in a CSV, which is not ISO standard. Formatting the cells as a date allows the specific formatting, however mixing US and UK dates is still possible and only visible with the content left (incorrect) or right (correct) aligned in the cell. Even more concerning is what happens on import of CSV data into Excel, for example an ISO standard date (YYYY-MM-DD HH:ii:ss) will be translated to a custom date format (DD/MM/YYYY HH:ii:ss) and this format will be saved back to your file upon completion of editing in Excel, unless the format is changed before it is saved. Most users will be completely unaware that Excel has done this translation prior to displaying the data.

Dates are difficult to manage, especially when software make assumptions on behalf of the user. Schemas can help providing the data does match the required format and translations do not happen somewhere in the middle.

4.2.2. Multiple representations

People often try to save time when entering data by abbreviating terms. If these abbreviations are not consistent, it can cause errors in the dataset. Schemas that use enumerated predefined lists of acceptable terms can help here, providing users are not able to easily add to the list and thus recreate the same error.

Other errors that exist in this category include differences in capitalisation, spacing, gender and pluralisation of adjectives (e.g., councilman vs councilmen), which can all cause interesting problems.

[12] http://openrefine.org

4.2.3. Duplicate records

A duplicate record is where the same piece of data has been entered more than once. Duplicate records often occur when datasets have been combined or because it was not known there was already an entry. Additionally record duplication can occur when one person might be referred to by two names (e.g. Dave and David). This might lead to instances where records need disambiguating to discover if the authors of publications are the same or different. This type of error cannot be caught with a schema validation.

4.2.4. Redundant or combined data

Redundant data is anything that is not relevant to your work with the dataset. Often a dataset has been created for a specific purpose which requires details you may not need. Common occurrences of redundant data include rows that represent total amounts. These often appear when a dataset in Excel has been exported into other formats without the 'Total' row first being removed. At other times columns of data have been combined or replicated in order to assist human readability.

4.2.5. Mixed use of numerical scales

Numerical values in datasets often use different scales to make it easier for a human to read. In budget datasets, for example, the units are often in the millions. $1,200,000 often becomes $1.2m. However, smaller amounts like $800,000 are still written in full. For a machine, this means they read the larger figure as $1.2, which causes errors. Alternatively if the column is meant to be in millions, then the second figure becomes $800,000,000,000.

Unfortunately schemas are not great at catching this type of error. This is because all the values could be value and errors could be caused at any level. Setting boundaries on can help but might not solve the problem. Making the units of measurement clear at the point of collection or use is essential here to guarantee data consistency and ensure disasters do not happen[13].

4.2.6. Numerical ranges

Data is sometimes measured in ranges, such as age or salary range. In order for a machine to understand these ranges it is important to separate the high and low values for easier analysis. It might even be necessary to create new brackets if they have changed over the years (e.g. people's ages or retirement age has risen).

4.2.7. Spelling errors

If you have lots of free text in the data, it is important to check for consistent spelling to ensure analysis can be performed with other datasets. The spelling

[13] http://www.nytimes.com/1983/07/30/us/jet-s-fuel-ran-out-after-metric-conversion-errors.html?mcubz=1

might not be 'correct', e.g. colour vs color, however it should be consistent to allow datasets to be comparable and interoperable.

4.3. Data-cleaning toolkit

When you are looking for errors in data, it may be necessary to download and upload datasets in many tools for cleaning and processing. It is also important to keep a note of what changes have been made and share these openly with others so that everyone can benefit from your work, particularly if the data you are cleaning is existing open data that has been published.

We have already looked at a number of schema validation tools, however there are a number of other tools that can help clear messy data.

OpenRefine

OpenRefine is a software tool designed to deal with uncleaned data. The tool is a column-based browser that allows you to fix errors across an entire open dataset in a single action. The errors that can be fixed include:

- date formats
- multiple representations
- duplicate records
- redundant data
- mixed numerical scales
- mixed ranges.

Spreadsheet programmes

OpenRefine is a key tool for cleaning data. However, it is sometimes easier to fix some errors in a spreadsheet programme:

- spelling errors
- redundant data
- numerical verification
- fixing shifted data.

Other tools

- Drake[14]
- Data Wrangler[15] (jointly developed tool from Stanford and Berkeley Universities in the United States)
- Data Cleaner[16]
- WinPure[17]

[14] https://github.com/Factual/drake

[15] http://vis.stanford.edu/wrangler/

[16] http://datacleaner.org

[17] http://www.winpure.com/article-datacleaningtool.html

Summary

For data to be usable there is a lot more than it simply being technically great data. Many guidelines try to simplify the requirements for data to be truly usable and each has its merits. The checklist created from the Open Data Certificates work goes to show the extent of the challenge of producing high-quality usable data. The Open Data Certificate is designed to assess the legal, practical, technical and social aspects of publishing open data using best practice guidance. However, even this does not consider the cross checking of data sources to ensure that the right data is used. Provenance of data is also key but again hard to follow fully on a dataset.

Once you have established trust in a dataset, verifying its contents is the next challenge awaiting. The creation and adoption of schemas and data validators helps to some extent but the need to clean and validate the data (potentially by hand) is not going away any time soon. In fact as we will explore in the next lesson, the process of preparing data ready for analysis could take up to 80% of your time!

References

- Dodds, L. 2015. Comparing the 5-star scheme with Open Data Certificates. Available at: https://theodi.org/blog/5-star-open-data-certificates-tim-berners-lee
- Codd, E. F. 1970. A relational model of data for large shared data banks. *Communications of the ACM* **13 (**6), 377–387.
- Garrett, J. J. 2005. Ajax: A new approach to web applications. Available at: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/