Quality assessment of numeric data: QAMyData

Louise Corti, Myles Offord & Christina Magder
UK Data Service

September 2019

UK Data Service

When does data need quality assessing?

For data publishing

- A researcher submitting data to a repository
- A repository for checking quality
- Peer review of published analysis for a journal
- Supporting FAIR Principles

For using data

- A researcher using a new data source
- Preparation of data for students learning about quality

When you want your data to be safe and healthy and beautiful

UK Data Service

QAMyData Tool

- UK Data Service project to develop a light weight, open-source tool for quality assessment of research data
- A 'data health check' tool that identifies the most common problems in data submitted in disciplines that utilise quantitative methods
- Helps set Data Quality Profiles for data publishers using own default settings

UK Data Service

Cleaning data manually vs a tool

- Getting to know your data:
  - check structure, find issues
  - incorrect, missing, inconsistent values
  - check for unanticipated/accidental disclosure risk
- Tool can:
  - flag issues: enable a machine or human to resolve the problems
  - Be deployed as a service for self deposit repository, eg DataVerse for a submission health check
  - Be deployed into data publishing pipelines

Scoping the tests

- In-house procedures for data checking
- Prepared ‘dirty’ test datasets for evaluation
- Reached out to other archives/data publishers to gather information on their own QA checks
- Feedback from tool use in early training

UK Data Service

Data: what to look for

- Basic file checks
- Metadata issues
- Data integrity issues
- Disclosure control

UK Data Service

Basic file checks

File opens	Checks whether acceptable format
Bad filename check, regular expression via RegEx pattern	Regex requires quotes "[a-z]". To use a special characters, e.g. a backslash (\) a backslash before is required e.g. \\

Metadata checks

Report on number of cases and variables	Always run
Count of grouping variables	
Missing variable labels	Must be set to true . If set to false the test will not run
No label for user defined missing values e.g. - 9 not labelled	SPSS only
'Odd' characters in variable names and labels	User specifies the characters
'Odd' characters in value labels	User specifies the characters
Maximum length of variable labels, e.g. >79 characters	User specifies the length
Maximum length of value labels, e.g. >39 characters	User specifies the length
Spelling mistakes (non-dictionary words) in variable labels using a dictionary file	User specifies a dictionary file.
Spelling mistakes (non-dictionary words) in value labels using a dictionary file	User specifies a dictionary file

Data integrity checks

Report number of numeric and string variables	
Check for duplicate IDs	User specifies the variables. Multiple variables can be added on new lines e.g. - Caseno - AnotherVariableHere
'Odd' characters in string data	User specifies the characters
Spelling mistakes (non-dictionary words) in string data using a dictionary file (can check if date format set correctly!)	User specifies a dictionary file
Percentage of values missing ('Sys miss' and undefined missing)	User sets the threshold, e.g. more than 25%

Disclosure control checks

Identifying disclosure risk from unique values or low thresholds (frequencies of categorical vars or minimum values)	User sets the threshold value, e.g. 5
Direct identifiers using a RegEx pattern search	User runs separately for postcodes, telephone numbers etc. Advise tests are separately as may be resource intensive
Direct identifiers/named entities in string data using a dictionary file (to be added)	Specify a dictionary file containing lists of stop words or named entities e.g. for places, names etc. Advise tests are separately as may be resource intensive

Formats and test selection

- Accepts: SPSS, Stata, SAS, CSV
- Configure data type, threshold, modular design
- `# Checks can be commented or omitted to exclude them from the checks to be run`

```
qamd run "I:\dcmagd\QAMD\data\teaching    data%set.sav" --include-locators --output-format html --output results_teaching_data.html --config "I:\dcmagd\QAMD\config\default.toml"
```

```
# Variable Configuration

[variable_config.odd_characters]
setting = ["!", "#", "  ", "@", "ë", "ç", "ô", "ü"]
desc = "Variable names and labels cannot contain certain 'odd' characters."

[variable_config.missing_variable_labels]
setting = true
desc = "Variables should have a label."

[variable_config.label_max_length]
setting = 79
desc = "Variable labels cannot exceed a max length”
```

UK Data Service

RegEx

```
# Checking for specific patterns by using RegEx (as this step is resource intensive it has
been commented out from the initial config file, please see the User Guide for more
information about RegEx and how to run this step)
#[value_config.regex_patterns]
#setting = [
  # Simple Email address Regex
  # "^([\\w\\.\\-]+)@([\\w\\-]+)((\\.(\\w){2,4})+)$"
  # UK post code regex
  # "([Gg][Ii][Rr] 0[Aa]{2})|((([A-Za-z][0-9]{1,2})|(([A-Za-z][A-Ha-hJ-Yj-y][0-9]{1,2})|(([A-
Za-z][0-9][A-Za-z])|([A-Za-z][A-Ha-hJ-Yj-y][0-9]?[A-Za-z])))))\\s?[0-9][A-Za-z]{2})",
  # Email addresses as per RFC 2822
  # "((([a-zA-Z0-9!#$%&'*+/=?^_`{|}~-]+(\\.[a-zA-Z0-9!#$%&'*+/=?^_`{|}~-]+)*)|(\"(([\\x01-\
\x08\\x0B\\x0C\\x0E-\\x1F\\x7F]|[\\x21\\x23-\\x5B\\x5D-\\x7E])|(\\[\\x01-\\x09\\x0B\\x0C\
\x0E-\\x7F]))*\"))@(([a-zA-Z0-9!#$%&'*+/=?^_`{|}~-]+(\\.[a-zA-Z0-9!#$%&'*+/=?^_`{|}~-]+)*)|(\
\[(([\\x01-\\x08\\x0B\\x0C\\x0E-\\x1F\\x7F]|[\\x21-\\x5A\\x5E-\\x7E])|(\\[\\x01-\\x09\\x0B\
\x0C\\x0E-\\x7F]))*\\])))"
#]
#desc = "Values matching a regex pattern fail the check."
```

UK Data Service

Reporting

- JSON: used to build a detailed report
- HTML output: human readable data at a glance
- Summary and detailed report for each test that fails
- Locates issues - variable (and row)

Health check report

QAMyData

teaching-data%set.sav

Raw Case Count: 10210
Aggregated Case Count: 0
Total Variables: 188
Data Type Occurrences: Numeric: 186, String: 2
Created At: 2019-02-18 13:37:39
Last modified at: 2019-02-18 13:37:39
File Label:
File Format Version: 2
File Encoding: WINDOWS-1252
Compression type: Rows

Basic File Checks

Name	Status (N)	Description
Bad file name	failed (1)	File name should match the user specified pattern

Metadata Checks

Name	Status (N)	Description
Missing variable labels	failed (8)	Variables should have a label
Variable odd characters	failed (2)	Variable names and labels should not contain the specified characters ["&", "#", " ", "@", "*", "ç", "ô", "ü"]
Variable label max length	failed (6)	Variable labels should not exceed the defined number of characters (79 characters)

Variable odd characters

# (limited to 1000)	Variable	Row number
1	OwnTV	-
2	V137	-

Technologies

Using the ReadStat library

- A command-line tool and C library for reading files from popular stats packages
- Originally developed for Wizard for free stat data analysis on a Mac
- Supports SPSS, Stata and SAS files – new & old formats
- In active development since 2012, continually receiving security patches and bug fixes from the open source community
- Currently used by the R library Haven, part of the Tidyverse collection of R packages for data science

RUST programming language

- Tried the following wrappers: Java, Clojure, R, Python
- RUST from the Mozilla Foundation for improving the Firefox browser: an environment that demands things just 'work'
- Performance: Rust generates executables that run very fast, without needing to write low level C code; easily integrates with other languages
- Reliability: enables the developer to eliminate many classes of bugs at compile-time
- Productivity: has great documentation, a friendly compiler with useful error messages, and excellent tooling

Deployment

- Downloadable to run on Linux, Windows, Mac
- Simple to install and deploy from our Github
- Lightweight to run and set tests - edit config file
- Wiki with documentation
- Space for suggesting new tests
- Released under MIT Licence

UK Data Service

Tool evaluation

First tier of evaluation

- UKDS data curation staff
- Peer data repositories in our international network

Second tier:

- University repositories - introductory webinar and visits
- Data owners,, researchers, data managers, quant. methods lecturers, journal publishers who run data peer review
- Voluntary testing – YOU!

Advocate data publishers to develop a Data Quality Profile with stated thresholds

UK Data Service

Resources

- Table of Available Tests
- Download and Run Guide – includes step by step for editing config. file to set your own thresholds and searches
- Teaching resources, slides, exercises and test data
- A blog

https://www.ukdataservice.ac.uk/about-us/our-rd/qamydata.aspx

UK Data Service

Acknowledgements

Thank you to our QAMyData team:

- Myles Offord: open source developer
- Jon Johnson: lead specs and development
- Cristina Magder: teaching materials and user guides
- Anca Vlad: input into tests and testing
- Louise Corti: PI

UK Data Service

Contact

corti@essex.ac.uk
moffor@essex.ac.uk
dcmagd@essex.ac.uk

@UKDataService
#QAMyData

UK Data Service

UK • DATA ARCHIVE

University of Essex