**QCHECK**

# STATA PACKAGE FOR QUALITY CONTROL OF DATASETS

## Introduction

**qcheck** (shorthand for ‘quality check’) is a technical package for quality control of different data types. -qcheck- performs the following types of complementary types of analysis:

## package components

### - Static Analysis

The **static** analysis of qcheck allows checking for within survey consistency. This analysis verifies the internal consistency of each variable and its relationship with other variables in the same dataset. It verifies that a variable is consistent with its definition (e.g., age is always a positive number) and checks the consistency with the other variables (e.g., 5 years old with graduate-level education). A user can create new tests, validations, and crosstabs to automate the assessment of variables across years, countries, and regions, among others.

To perform the **static** analysis, the qcheck package requires an Excel file as input. From this input file, the Stata command *qcheck* retrieves all the information needed to perform the assessment. The user must create and complete the input Excel with logic statements about the variables in the data. The ado-file is complemented with an example of an Excel file with a basic set of tests to check the quality of an example database. The user could modify this file either by editing the tests or adding tests to it and running the qcheck analysis again to observe how the results change in response to the changes in the input file.

Static test workflow

Before performing the **static** analysis, qcheck revises the existence of the variables, and creates flags if the variable is not available in the data, or if the variables were created all empty, either all zeros or missing. The diagram below indicates how the inspection is conducted.

#### Output

As a result, the user obtains an analyzed file in the *.dta* format comprising the frequency and percentage of observations in which the inconsistency is flagged. Note that not all the inconsistencies are errors, some flags may simply be for unexpected values or unexpected relation across variables. Other flags indicate errors in the harmonized data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **file** | **variable** | **warning** | **flag** | **freq.** | **perc.** |
| survey1 | age | Urgent | Variable has extreme values (>120) | 5 | 0.01 |
| survey1 | computer | Flag | Household owns a computer but doesn't have access to electricity | 26 | 0.04 |
| survey1 | lstatus | Caution | lstatus equal to employed without employment type defined in empstat | 5 | 0.01 |
| survey1 | relationharm | Urgent | Households without household head | 108 | 0.18 |

Above is an example of the outcome *.dta* file. Using the results of the qcheck static analysis, information can be obtained about the variables that are all missing in the data or were not created in the data.

Below is an example outcome *.dta* file comprising flags for different types of missing variables:

|  |  |  |  |
| --- | --- | --- | --- |
| **file** | **variable** | **importance** | **flag** |
| survey1 | welfaredef | Regular | All missing .a, variable had not been harmonized |
| survey1 | welfareother | Optional | All missing .c, variable not harmonized, data not available |
| survey1 | industrycat10 | Regular | All missing, unknown reason |
| survey1 | industrycat4 | Regular | All missing, unknown reason |

#### INPUT: set up the inconsistency test in the auxiliar excel file, in the spreadsheet “Test”

The first step is to create the input Excel with the internal consistency logic statements. Before completing your input Excel, refer the example file “qcheck\_NNN.xlsx.” First, in the spreadsheet “TEST,” you can add, modify, or edit the set of quality checks of your database. Each row corresponds to a different check or logical statement, and each column corresponds to a particular check feature.

The first column contains the name of the variable to be checked. It may be the case that one variable has to be checked in relation to another variable so that both variables are checked jointly. It does not matter which variable name goes in the name as long as only one name is specified.

The second column, “Warning,” allows the user to specify the level of urgency. The purpose of this column is merely cosmetic. It allows the user to organize or filter the results easier in the Tableau dashboard or their own analyses.

The third and fourth columns are the checking codes, and each column has a particular function. The fourth column (*iff*) contains logical statements that check the consistency of the variable. For instance, if you wanted to test that the variable corresponding to the person’s age does not have negative values, positive values above 100, or missing values, you may type something like this: age < 0 | age > 100. Notice that the logical test flags those observations that meet the criterion as inconsistent.[[1]](#footnote-2)

The third column (*temporalvars*), is for code lines that must be executed before the logical statement in column “iff.” Sometimes, it is needed to create a temporal variable with certain characteristics in order to check for some inconsistencies. For instance, to test whether the combination of household and person id is unique along the dataset, sample code could be:

**cap destring pid, replace**

**duplicates report hid pid**

**local n = r(unique\_value)**

**count**

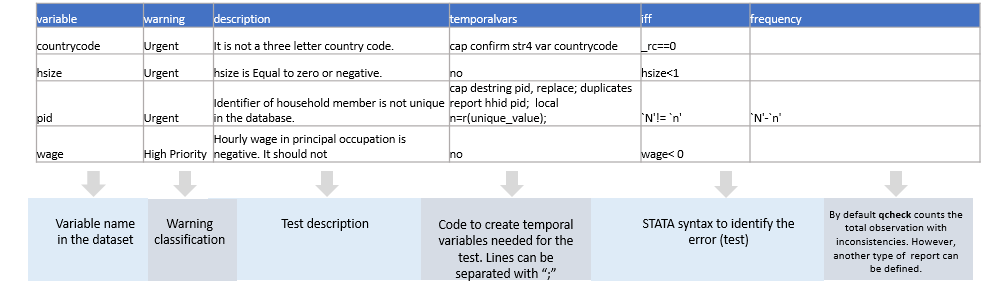
**count if r(N)! = `n' // logical statement**

The first four lines of the code above create a temporal macro that counts the number of observations in the dataset that have a unique value for the combination *hid* and *pid*. If the dataset was constructed accurately, the number in local *n* should be the same as the number of observations in the dataset. Therefore, the last line of code is the logical test that verifies the aforementioned statement. Several things should be kept in mind.

Given that there is only one cell for each check in column “*temporalvars*”, each line of code must be separated from the subsequent line with a semicolon (;) instead of a break of line.

In the example above, the logical statement that goes in the corresponding cell of column “*iif*” is r(N) != `n', rather than count if r(N) != `n'. Given that by design, all the consistency checks count the number of observations with problems, it is inefficient to ask the user to type “count if” for each cell. Instead, it is only necessary to type the logical statement of the code line.

See a small example below:



The convention of the name of the Excel file is “qcheck\_NNN.xlsx” where NNN refers to a set of checks to be applied to a particular collection.

\*\*A word of caution here\*\*: it is expected that the suffix NNN of the “qcheck\_NNN.xlsx” file refers to the name of the collection to be tested. For example, the user may have the file “qcheck\_ABC.xlsx” to contain the check of the collection ABC. 1 step: Files location

### - BASIC and CATEGORIC Analysis

The **basic** analysis is useful to compare within and across surveys, for example, evolution of a categorical variable over time or across countries or across regions within a country. The basic analysis of qcheck store all the descriptive statistics provided by -sum var name, d- and the number of missing observations, number of non-missing observations, number of zeros, mean, standard deviation, maximum, minimum, skewness, kurtosis, and 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th ‘s percentiles. This output is stored in a tabulated format.

Basic test workflow

The **categoric** analysis runs tabulates for each category variable(s) specified by the user. The output is stored in a tabulated long format. This visualization allows the user to compare the evolution of the categories over time, or within regions, or by population groups. Such comparisons allow to identify anomalies or mistakes in the harmonization of the data.

Categorical test workflow

Unlike **static** analysis, the **basic** and **categorical** analysis do not require additional input other than the dataset to be analyzed. Once the Stata command *qcheck* has performed the assessment, results can be exported into long-formatted Excel files that can be read by Tableau/Power BI/R/Pivot tables Excel. We provide some examples, but the user may create their reports in the program of their preference or adapt the provided examples.

#### Output

As a result, the user obtains an excel file in the log format. Below is an example of an output in the form of Pivot tables in excel for **categoric** analysis:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **countrycode** | **year** | **varname** | **valuelab** | **label\_var** | **freq** |
| survey1 | 2012 | cellphone | 0 | Own\_mobile\_phone\_\_at\_least\_one\_ | 36.03 |
| survey1 | 2012 | cellphone | 1 | Own\_mobile\_phone\_\_at\_least\_one\_ | 63.96 |
| survey1 | 2012 | computer | No | Own\_Computer | 82.58 |
| survey1 | 2012 | computer | Yes | Own\_Computer | 17.41 |
| survey1 | 2012 | educat4 | No education | Level\_of\_education\_4\_categories | 47.80 |
| survey1 | 2012 | educat4 | Primary (complete or incomplete) | Level\_of\_education\_4\_categories | 22.46 |
| survey1 | 2012 | educat4 | Secondary (complete or incomplete) | Level\_of\_education\_4\_categories | 24.89 |
| survey1 | 2012 | educat4 | Tertiary (complete or incomplete) | Level\_of\_education\_4\_categories | 4.82 |

Similarly, the user obtains an excel file in the log format. Below is an example of an output in the form of Pivot tables in excel for **basic** analysis:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **countrycode** | **year** | **var** | **description** | **analysis** | **case** | **value** |
| survey1 | 2014 | educy | Years of education | Basic | Missing | 0.13 |
| survey1 | 2014 | educy | Years of education | Basic | Zero | 0.13 |
| survey1 | 2014 | educy | Years of education | Basic | Mean | 4.63 |
| survey1 | 2014 | educy | Years of education | Basic | SD | 4.93 |
| survey1 | 2014 | educy | Years of education | Basic | Max | 29 |
| survey1 | 2014 | educy | Years of education | Basic | Min | 0 |
| survey1 | 2014 | educy | Years of education | Basic | Num | 58423 |
| survey1 | 2014 | educy | Years of education | Basic | Skewess | 0.73 |
| survey1 | 2014 | educy | Years of education | Basic | Kurtosis | 2.38 |
| survey1 | 2014 | educy | Years of education | Basic | P1 | 0 |
| survey1 | 2014 | educy | Years of education | Basic | P5 | 0 |
| survey1 | 2014 | educy | Years of education | Basic | P10 | 0 |
| survey1 | 2014 | educy | Years of education | Basic | P25 | 0 |
| survey1 | 2014 | educy | Years of education | Basic | P50 | 3 |
| survey1 | 2014 | educy | Years of education | Basic | P75 | 8 |
| survey1 | 2014 | educy | Years of education | Basic | P90 | 12 |
| survey1 | 2014 | educy | Years of education | Basic | P95 | 14 |
| survey1 | 2014 | educy | Years of education | Basic | P99 | 16 |

## Setup and Installation

Download the following four .ado files from <https://github.com/worldbank/qcheck>":

1. qcheck.ado
2. [qcheckcat.ado](https://github.com/worldbank/qcheck/blob/master/qcheckcat.ado)
3. [qcheckstatic.ado](https://github.com/worldbank/qcheck/blob/master/qcheckstatic.ado)
4. qchecksum.ado

* Click the download icon on the right side of the screen to download a particular file.
* Alternatively, Click the Raw button on the right side of the screen to download a file. Right-click anywhere on the page and select Save As to download the file.

After downloading all the four ado files, place the files in the following folder: “C:\ado\plus\q”

## example do-file, once qcheck installed

After installing the qcheck, the user needs to execute the test in Stata. Below is an example of a .do file comprising all types of qcheck tests for a GMD survey:



1. Notice that the test identifies those observations with errors, and not those that are accurate. That is, the test should not be **inrange(age, 0, 100)**. [↑](#footnote-ref-2)