**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 allows checking for within the survey consistency. The **static** analysis of qcheck 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). The user can create new tests, validations, and crosstabs to automate the assessment of variables across years, countries, and regions, among others.

For the **static** analysis, the qcheck package requires as input an Excel file. 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 is free to modify such a 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 revise the existence of the variables, and creates flags if the variable is not in the data, or if the variables was created all empty, either all zeros or missing. The diagram below indicates how the inspection is conducted.

#### Output

As a result, you obtain a dta per analyzed file with the number of observations and percentage in which the inconsistency is flagged. Not all the inconsistencies are errors, some simply may flag unexpected values or unexpected relation across variables. Other 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 equalt to employed without employment type defined in empstat | 5 | 0.01 |
| survey1 | relationharm | Urgent | Households without household head | 108 | 0.18 |

From the outcome above, the results can be filer to obtain the variables that are all missing in the data or were not created in the data.

|  |  |  |  |
| --- | --- | --- | --- |
| **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, look at 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 code, but each has a particular function. The fourth column (iff) contains the 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. As you see, 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 some inconsistencies. For instance, you may need to test whether the combination of household and person id is unique along the dataset. In order to do so, you can do the following:

**cap destring pid, replace**

**duplicates report hid pid**

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

**count**

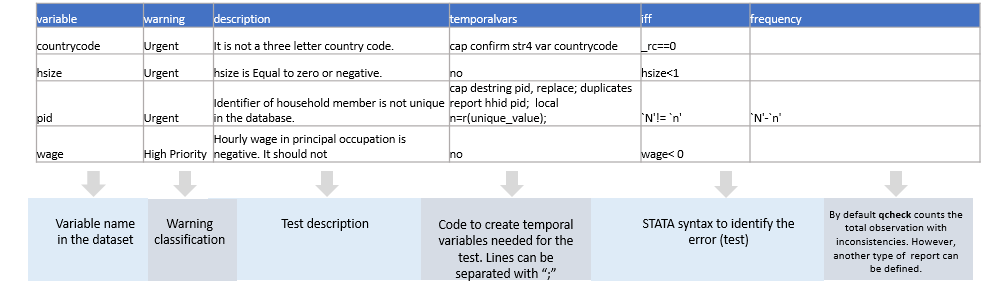
**count if r(N) != `n' // logical statement**

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 correctly, 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 store in a tabulated format.

Basic test workflow

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

Categorical test workflow

Unlike static analysis, the basic and categorical analysis does 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

<<Divya, add here the output for the categoric and dynamic, based on the reports you will create>>

## Setup and Installation

<<Divya, try if the following works>>

Manual Installation:

Download the .ado file from <https://github.com/worldbank/qcheck>.

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

Install the .ado file in Stata

* Open Stata.
* Use the cd command to change the working directory to the location where you downloaded the .ado file. For example, if you downloaded the file to your Downloads folder, you would type cd "~/Downloads" (replace with your actual file path).
* Use the ado install command followed by the name of the .ado file to install it. For example, if your file is named my\_file.ado, you would type ado install my\_file.

Here’s what the Stata commands might look like:

cd "~/Downloads"

ado install my\_file

Please replace "~/Downloads" and "my\_file" with your actual file path and file name. After these steps, the .ado file should be installed and ready to use in Stata.

From Github repository:

<<Divya, try if this is possible to do>>

First, you need to install the github package in Stata. [This package allows you to search and install Stata packages from GitHub](https://github.com/haghish/github). [Once the github package is installed, you can use the gitget command to install or update Stata packages from GitHub](https://github.com/haghish/github). [The gitget command is a wrapper for github install](https://github.com/haghish/github). [You only need to provide the package name](https://github.com/haghish/github). Here is the syntax:

ssc install github

gitget packagename

github install haghish/markdoc

[Please note that the gitget command relies on a complete list of Stata packages on GitHub to identify the URL of a project](https://github.com/haghish/github). [This list is created programmatically using a search program that detects Stata packages1](https://github.com/haghish/github). Remember to replace packagename with the name of the package you want to install. If the package is not found, make sure you have the correct name of the package and it is available on GitHub.

## example do-file, once qcheck installed

<<Divya, Add here some screenshots giving an example for one GMD survey, all the test.>>

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