**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 two big types of complementary types of analysis:

- The first analysis is static.

*The static analysis allows checking for within the survey consistency. For example:*

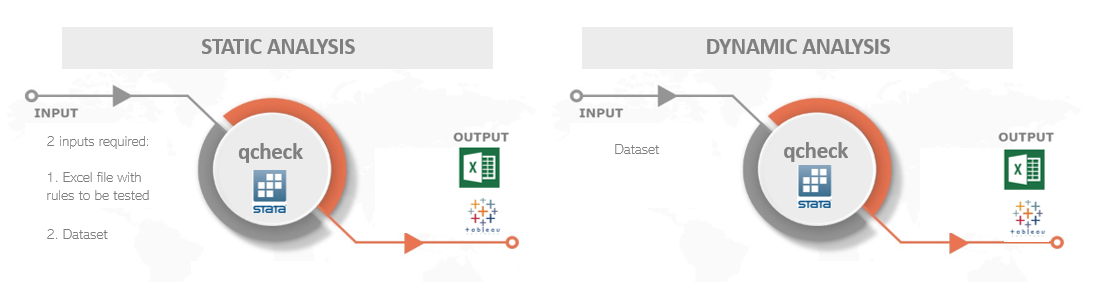
The static analysis of qcheck verifies the internal consistency of each variable and its relationship with other variables in the same dataset. It verifies that a variable is consistent with its definition (e.g., age is always a positive number) and checks the consistency with the other variables (e.g., 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.

- The second analysis is dynamic.

*Dynamic analysis performs two types of analysis: basic and categorical.*

1. *Basic: create basic statistics from all, one, or multiple variables. The basic statistics store all the estimations 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. These analyses are shown in a tabulated format.*
2. *Categoric: run tabulates with the mean values for each category of categoric variable(s) specified by the user.*

## package components



For the **static** analysis, the qcheck package requires as input an Excel file. Here, 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.

Unlike static analysis, the **dynamic** analysis does not require additional input other than the dataset to be analyzed.

Once the Stata command qcheck has performed the assessment, results are 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.

## Setup and Installation

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

For the qcheck setup, the user should save all the 3 ado files in the “C:\ado\plus\q” folder.

* qcheck.ado
* qcheck\_static.ado
* qcheck\_dynamic.ado
* qcheck.sthlp

*Note: it is recommended to save the “qcheck\_NNN.xlsx” file into the same directory as the ado files “c:\ado\plus\q”. However, the user may place the Excel file in another directory and use the option ‘path()’ when running the qcheck.ado.*

## USE – STATIC ANALYSIS

### 1 step: Modify Excel file as needed (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-1)

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)**

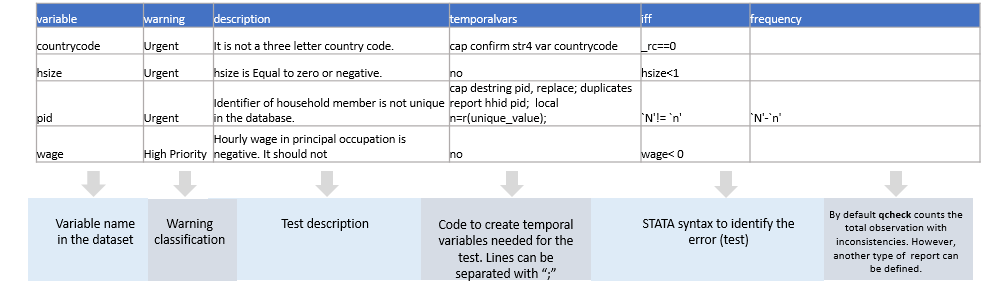
**count**

**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:

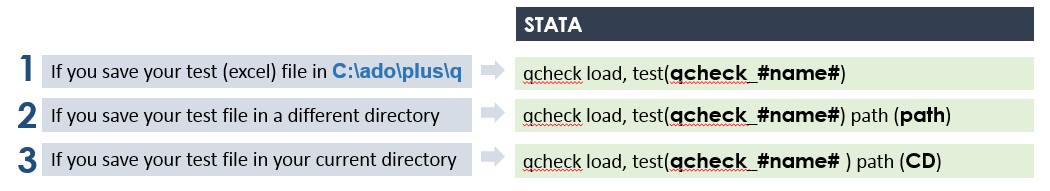


### 2 STEP: Modify Excel file as needed (spreadsheet “Variables”)

The dynamic assessment of qcheck performs different analyses depending on the variable type: welfare, categorical, and basic. Variables classified as ‘welfare’ are assumed to be continuous, and estimations of poverty and inequality are only performed with these variables. Categorical variables are numeric, but their values refer to a classification or characteristic of the observation rather than an ordinal correlation between its members. For instance, the variable of Labor Force Participation contains three numeric values: 1, 2, and 3. However, 1 means ‘employed,’ 2 means ‘unemployed,’ and 3 means ‘out of labor force.’ Finally, the basic classification of variables refers to variables that are either non-categorical or welfare aggregate.

### 3 step: Load Checks into qcheck

Before using qcheck in Stata, you must ‘load’ the checks into the system. To do so, specify the function ‘create’ in the qcheck command in Stata. Depending on where you saved the Excel file “qcheck\_NNN.xlsx,” you need to specify the directory path as indicated in the image below. You need to do this procedure for each “qcheck\_NNN.xlsx” input file.



## USE – DYNAMIC ANALYSIS

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-1)