SRQM: Programs

This document lists the programs included with the Statistical Reasoning and Quantitative Methods (SRQM) course run at Sciences Po by François Briatte and Ivaylo Petev.

All commands were written to assist students in completing their research projects. This file describes all commands in more detail than necessary to run them in class.

`betaplot`

Currently in the works. This command will produce a `plotbeta`-style plot of coefficients. See the `plotbeta` package.

`burd`

The series of `scheme-burd` files contain reversed versions of the `RdBu` ColorBrewer theme, as well as a replacement for the `s2color` scheme. The scheme is used for the course plots (and students get to use it if they want to). See example plots appear on the course wiki.

`repl`

Not in use due to limited testing. Creates a replication folder out of a do-file. The folder will contain the do-file, log and all plots that were assigned an optional `name()` in the code. It will also contain any file exported by the do-file and a short file manifest in a `README` file.

`srqm_data`

Prepares the course datasets, which are slight variations on the original versions. All datasets are saved in Stata 9/10 format for compatibility. The `ess2008` and `nhis2009` datasets need to be downloaded by hand from the sources and renamed to these handles.

`srqm`

The `srqm` utilities rely on the architecture of the `SRQM` folder, a.k.a the 'Teaching Pack', which contains the course material. All `srqm` commands should be run with this folder set as the working directory.

The `srqm` utilities require one command and optionally one subcommand to execute:

```
srqm command [subcommand] [, nolog forced]
```

The commands and subcommands form four blocks. The first of them, `setup`, is called from the `profile.do` file of the `SRQM` folder and is used to set up computers for the course:

```
srqm setup
srqm setup folder
srqm setup packages
```

To minimize trouble with working directory errors, the setup creates a mock symbolic link to the `SRQM` folder and performs a quick folder integrity check at startup. A few more similar checks are available:

```
srqm check
srqm check course
srqm check folder
srqm check packages
```

The mock symbolic link to the `SRQM` folder is generally erased at the end of the semester, and a few more cleanup utilities can be used for testing purposes:

```
srqm clean
srqm clean folder
srqm clean packages
```

The utilities also include a built-in course update system that recognizes where files go and keeps older files as backups. The command works with `week#.do` files (code), `week#.pdf` files (slides) and setup (`.ado`) files:

```
srqm fetch week7.pdf
srqm fetch week11.do
srqm fetch stab.ado // caution with that
```

Finally, the course datasets can be rebuilt by calling `srqm data` followed by `all` or by the name of a course dataset. The command calls the `srqm_data` ado-file.

When the additional `log` option is specified, all commands send verbose output to a log in order to help users report issues.

`srqm setup`

Tries to permanently set up a few options like screen breaks and scrollback buffer size in Stata, including `memory` on software versions older than Stata 12. Also sets the `burd` scheme (documented above).

`srqm setup folder`

Tries to tell Stata to automatically run from the `SRQM` folder by copying its path to the global macro `$srqm_wd`. The macro is saved into a `profile.do` file located in the Stata application folder.

The `profile.do` file saved to the Stata application folder acts as a symbolic link that also runs the `profile.do` file from the SRQM folder, to perform further integrity checks on the course setup.

This command requires to run Stata as administrator on Windows Vista and 7, or it will fail due to a system restriction in recent versions of Windows.

`srqm setup packages`

Installs the additional Stata packages used in the course do-files. Requires Internet access to execute properly. Usually runs in less than five minutes.

The command will try to run as quickly as possible by skipping packages that are already installed. This behaviour can be overriden by passing the `forced` option.

`srqm check`

Produces a report on the current setup, covering the basic system options obtained with `query` and the list of installed packages obtained with `ado dir`.

If the `demo' option is specified, as in`srqm check, demo(1/4 12)`, the command runs the course do-files to test their executability (the whole course usually runs in less than ten minutes).

`srqm check folder`

Checks the existence of the `Datasets` and `Replication` folders used in class. This subcommand is automatically run with the `nolog` option every time Stata loads with the `SRQM` working directory.

`srqm check packages`

Checks whether the additional packages installed from the SSC server by the `srqm setup packages` subcommand are up to date.

srqm clean

Cleans up the `SRQM` folder by erasing temporary workfiles produced by the course do-files. The workfiles consist of logs.

srqm clean folder

Unlinks Stata from the `SRQM` folder by erasing the `profile.do` file that is installed in the application folder as part of the setup for the course. Used at the end of the semester.

srqm clean packages

Uninstalls course packages. Used for testing purposes.

stab

Currently in the works. Produces summary statistics and correlation matrix tables in plain text format, for maximum compatibility with text and spreadsheet editors (the course recommendation is to use Google Documents).

The basic syntax for `stab` is:

```
stab using week8 [aw,fw] [if,in] [, su(v1 v2) corr fr(v3 v4) ttest by(varname) f(0) replace]
```

The `su` option produces five-number summaries for continuous variables. The `fr` option produces frequencies for categorical variables. Both results are combined into a single table.

The `by(varname)` option creates multiple tables based on the categories of `varname`. The `corr` option adds a correlation matrix of the continuous variables specified in `su`.

Less frequent options are `ttest` to add *t*-tests for all groups, and `float(n)` to modify the level of decimal precision. The command also supports analytical and frequency survey weights, `if` and `in`.