SRQM: Programs

The folder contains some teaching utilities:

- the `packages_required` command to check for the existence of commands installed through additional packages
- the `srqm` utilities to set up a computer for our course, as explained in the README file of the `SRQM` folder
- the `tsst` command to export summary statistics tables as tab-separated values

All commands were written to assist students in completing their research projects.

packages_required

Checks whether a given list of commands are currently installed in Stata, and if not, tries to install the corresponding package at the SSC archive.

Syntax

```
packages_required fre spineplot tabout etc
```

The command is used in course do-files to send a legible one-line warning about package dependencies without slowing down execution with multiple `ssc install` commands.

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]
```

The commands and subcommands form three 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 folder
srqm check packages
srqm check course
```

Finally, the mock symbolic link to the `SRQM` folder can be 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
```

Unless the additional `nolog` option is specified, all commands send moderately 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.

`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 occasionally fails on Windows Vista and Windows 7 due to an issue with admin privileges in which that case users have to set their working directory manually in each session.

If the `SRQM` folder has been renamed to `SRQM-USB`, the subcommand ignores the Stata application folder and sets instead packages to be installed to a temporary `Packages` folder in the `SRQM` folder.

`srqm setup packages`

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

The subcommand 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`.

`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 check course`

Runs the whole course (weekly sessions and draft assignments) as a single sequence to test the executability of the course code. Usually runs in less than ten minutes.

`srqm clean`

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

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.

tsst

Produces a simple table of summary statistics as required for the course research project. The code draws on a tutorial by Ben Jann.

The command is very barebones and means 'tabbed summary statistics table' because it produces tab-separated values in plain text format, for maximum compatibility with text and spreadsheet editors.

Syntax

```
tsst using stats.txt [weight], su(v1 v2) fr(v3 v4) [f(1) replace verbose]
```

The `su` option is meant produces five-number summaries out of continuous variables. The `fr` option produces frequencies out of categorical variables. Both results are combined in a single table.

Options

- The `tsst` command accepts analytical, frequency and probability weights `[aw,fw,pw]`. It does not, however, support the `svy` survey prefix command.
- The command uses a default precision of one decimal that can be changed by specifying a different number `n` with the `f(n)` option.
- The command will overwrite the file to which it exports its results if the `replace` option is specified. Note that there is no `append` option.
- For testing purposes, it is possible to view the plain text result of the command in the Stata Results window by specifying the `verbose` option.

Far more sophisticated output options appear in packages like `estout` or `tabout`. Tamás Bartus is also developing the `publish` command, which comes close to the spirit of the `tsst` command.

Help

- Type `tsst` to view a quick syntax sheet.
- Type `tsst using options` for a list of options.
- Type `tsst using example` for a working example.

Students are referred to the documentation of their text or spreadsheet editor to learn how to import a tab-separated values document, or how to convert tab-delimited text into tabular output.