Processing JASMIN1 Data from a LOTUS Results File

*Thomas Pronk*

[*tpronk@uva.nl*](mailto:tpronk@uva.nl)

*Version 1.0, 2016-04-12*

This is a step-by-step guide on:

1. **Decoding JASMIN1 data encoded in a LOTUS results file into a trial data CSV file.**
2. **Calculating scores from trial data**
3. **Merging scores together into a single file**

Note: This tutorial assumes you have already installed the SANDRA library, which you’ve already done if you’ve followed the steps of Tutorial *1. Installing SANDRA.docx*.

## ****Step 1. Setup the analysis framework****

The right processing scripts can be obtained from the SANDRA Github repository.

1. Download the whole repo in a ZIP via the link below. See “Download ZIP” button in the top right corner of the screen. <https://github.com/tpronk/SANDRA>
2. Unzip the file. The subdirectory *framework\_demos*  contains a SANDRA Analysis Framework, together with the processing scripts, and some example data to test it on.
3. Move the contents of *framework\_demos* to a comfortable location, then run “Install SANDRA (from GitHub).R” to set up the framework for that location.

If everything went well, you’ll see some messages saying that various directories already exist, and finally the message:

sandra::FrameworkFileIO. Succesfully constructed FileIO

## ****Step 2. Test the processing script with the example data****

1. The *scripts* sub-directory contains a collection of scripts for processing your data. See the page below for a brief description of what each script does: <https://github.com/tpronk/sandra/tree/master/framework_demos>
2. In general, you’ll execute a t.1, t.2, and t.3 script in order. By default, the scripts are set up to process a demo dataset named “jasmin1\_data.csv” located in the *original* sub-directory. Run a t.1, t.2, and t.3 script, and the files listed below should be produced in the *interim* folder. Check “date modified” to find out if they were actually just created.
   * t.1 decodes JASMIN data into trial data and metadata, the latter of which contains participant parameters and some reporting variables. See glossary below.
     1. It produces a *trialdata* file (one file per task; one row per trial) and *metadata* file (one file per dataset; one row per participation). The *trialdata* and *metadata* files can be joined together on the *set\_id­* variable.
     2. You can configure *fileSource* (which file to decode) and *participationID* (which columns identify a participation).
   * t.2 calculates scores (and/or split-halve reliabilities) from trial data with one row per participation.
     1. It produces a *scores* file (one per task; one row per participation)
     2. You can configure *fileSource* and *scorings* (how to score a task). Type **?calculateScores**into the R console to get information about how to setup the task scoring.
   * t.3 merges scores across tasks and sessions together into a single ‘wide’, with one row per participant and columns postfixed by task and session.
     1. It produces a *joined* file (one file per dataset, one row per participant)
     2. You can configure *fileSource*, participantID (which column identifies one participant), *sessionID* (which column identifies a session), and *tasks* (which task data to combine). Finally, you can drop artefacts before joining the files together via the *dropArtefacts* function.

## Step 3. Download a LOTUS Results File

Time to download your task data into a results file. To get it in the right format, take heed of the settings below:

1. Login to LOTUS.
2. Go to *Your project 🡪 View participants 🡪 Participants 🡪 Results*.
3. Click *Export*. At the export screen:
   1. Make sure to select any participant parameters you are interested in (such as *username* or *age)*.
   2. At Separator, remove the ‘;’ and enter ‘\t’ instead.
   3. At Escape, remove the ‘\’ and just leave that field empty.
   4. Click OK.
4. Unzip the downloaded file and put it in the *interim* sub-directory.
5. Be sure to give your unzipped results file an informative name, such as “*pretest\_alcohol.csv*”.

## ****Step 4. Process your own data!****

Now set up the right t.1, t.2, and t.3 scripts and process your data.

* + Note that with big datasets, this may take a while.
  + Don’t forget to set up scorings for each of the tasks encoded in your data
  + The manual *TP - CBM Tasks JASMIN Configuration.docx* provides more information about the meaning of all the variables in the trial data. You can find this manual in ADAPT Shared/Documentation/CBM Tasks

## ****Step 5. Inspect the output****

SANDRA will add a couple of variables to the metadata that tell you a bit about what happened during data processing. These variables are also added to the *scores* and *joined* files. The table below explains the meaning of these variables.

|  |  |
| --- | --- |
| **Folder** | **Explanation** |
| lotus\_says | Tells you how the task ended. There is only trialdata produced if the task completed successfully. Possible values:   * **task\_done**. Task completed successfully * **task\_start**. Task was restarted before it completed * **task\_error**. Task reported that an error occurred |
| sequence\_report | Reports suspicious data. If this variable is not empty, then the trial data of this set are not to be trusted. Possible values:   * **inconsistent.** Events with the same sequence number but different data * **missing.** Sequence numbers missing (1,2,3,5,6,7) * **negtime.** Client time decreased with successive sequence numbers |
| taskName | The type of task belonging to this set. |

*Table 2. Explanation about metadata variables*

<https://github.com/tpronk/sandra/tree/master/framework_demos/scripts>

1. R, which is a free and open source statistical programming language, with features that are similar to MATLAB. You can download it at <https://www.r-project.org/>)
2. A program for editing R-scripts. A plain text editor (such as Notepad/Kladblok) is a simple but sufficient solution. For a more powerful editor, consider installing **RStudio**: <https://www.rstudio.com>.

## Step 2. Run the SANDRA installation script

Create an *analysis directory* (or folder) on your hard-drive, which will contain your analysis scripts and data. Next, **start R** and copy-paste the installation script shown below into the **R console**. The most recent version of this script can be found here: <https://github.com/tpronk/sandra/tree/master/framework_demos>

* Windows users can run the script as-is. A folder picker will appear that allows you to select the analysis directory you just created. Note that you might need to minimize R/RStudio to make this folder picker visible.
* Mac users need alter the installation script to specify the full path to their analysis folder, for example as follows:

installAnalysisFramework( “/studies/analysis” );

# Install & load package devtools

install.packages( "devtools" );

library( "devtools" )

# Install & load package SANDRA

install\_github( "tpronk/SANDRA/src" );

library( "sandra" );

# Install SANDRA Analysis Framework

installAnalysisFramework(

# Enter path to analysis folder here (without trailing slash)

# Only for Windows: leave empty to get a folder picker

);

The installation script first installs the SANDRA R-package and next installs a SANDRA Analysis Framework in the analysis directory that you specified. A SANDRA Analysis Framework is a standard way of organizing your data. The installation scripts creates all scripts and directories required for the framework if they do not exist yet.

**Table 1.** Folder structure of the ADPT Framework

|  |  |
| --- | --- |
| **Directory or File** | **Explanation** |
| analysis/Load SANDRA.R | Loads SANDRA package and sets up your SANDRA Analysis Framework for this analysis directory |
| analysis/interim | Your interim data (datasets produced by processing other datasets) |
| analysis/original | Your original unprocessed data |
| analysis/scripts | Your analysis scripts |

If everything went well, then the files and directories listed in Table 1 have been created. Finally, SANDRA confirms successfully loading the framework by showing the text seen below in the console.

sandra::FrameworkFileIO. Succesfully constructed FileIO

## Step 3. Reload your SANDRA Analysis Framework

After installing, your analysis framework is installed and setup. You only need to install the SANDRA Analysis Framework once, but you need to set it up each time you restart R to run an analysis. To setup your framework, run the script in ‘Load SANDRA.R’, which has just been created in your analysis directory.