Quick-start guide to the Stat-JR 1.0.6 TREE interface

This document provides a 'quick-start' guide to using Stat-JR, via its *TREE (Template Reading and Execution Environment)* interface. The development of Stat-JR was a collaborative project, funded by the UK's ESRC, between the Universities of Bristol and Southampton. More details about Stat-JR can be found on its webpage http://www.bristol.ac.uk/cmm/software/statjr/.

TREE is one of the interfaces available for Stat-JR, alongside an eBook-reading interface (*DEEP*), a workflow system (*LEAF*; first released as a beta version with Stat-JR 1.0.4), and also a command line interface. For more detailed instructions on how to use *TREE*, together with worked point-and-click examples, see the *Beginner's Guide to Stat-JR's TREE interface* and also the *Advanced User's Guide to Stat-JR*; the DEEP and LEAF interfaces also have their own user guides; all manuals are available via the Stat-JR website.

ISBN: To be confirmed

Printed in the United Kingdom

This documentation was written by Richard Parker* with additional updates from Christopher M.J. Charlton*

The screen shots for the 1.0.6 version were updated by Rhiannon Moore.

If you use Stat-JR in your research, then please cite it as:

Charlton, C.M.J., Michaelides, D.T., Parker, R.M.A., Cameron, B., Szmaragd, C., Yang, H., Zhang, Z., Frazer, A.J., Goldstein, H., Jones, K., Leckie, G., Moreau, L. and Browne, W.J. (2017) *Stat-JR version 1.0.6*. Centre for Multilevel Modelling, University of Bristol & Electronics and Computer Science, University of Southampton, UK.

The initials of Stat-JR are taken from those of the late Jon Rasbash, whose vision was instrumental to its conception. The Stat-JR software system has been primarily developed by Chris Charlton* and Danius Michaelides**, with algebra system development by Bruce Cameron*, and with additional input from William Browne* and Richard Parker*. Core template development by Chris Charlton*, William Browne*, Richard Parker*, Camille Szmaragd* and Zhengzheng Zhang*.

The Stat-JR:TREE software interface was primarily developed by Chris Charlton* and Danius Michaelides**, with additional input from Richard Parker* and William Browne*.

* Centre for Multilevel Modelling, University of Bristol, UK

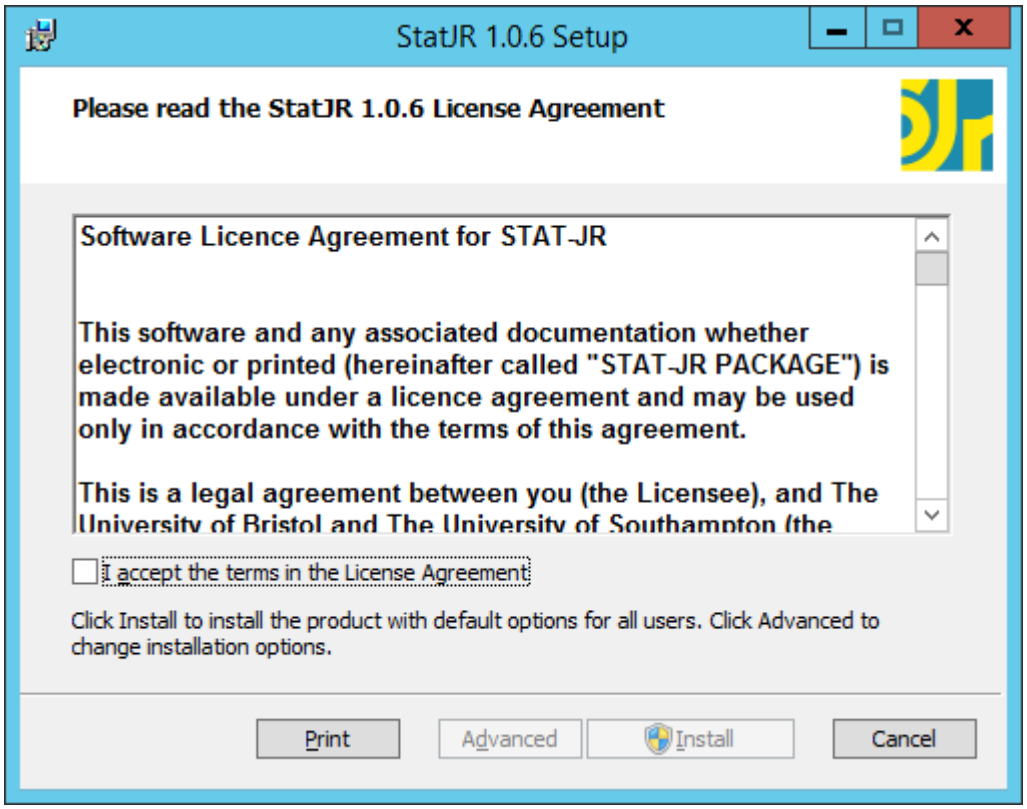
** Electronics and Computer Science, University of Southampton, UK.

Installing the software

Downloading & installing Stat-JR

Details of how to order a copy of Stat-JR can be found on its website: http://www.bristol.ac.uk/cmm/software/statjr/order-statjr/.

To install it, you need to start the *StatJR.msi* file that you will have been asked to download; on doing so, you will be greeted by the following screen:

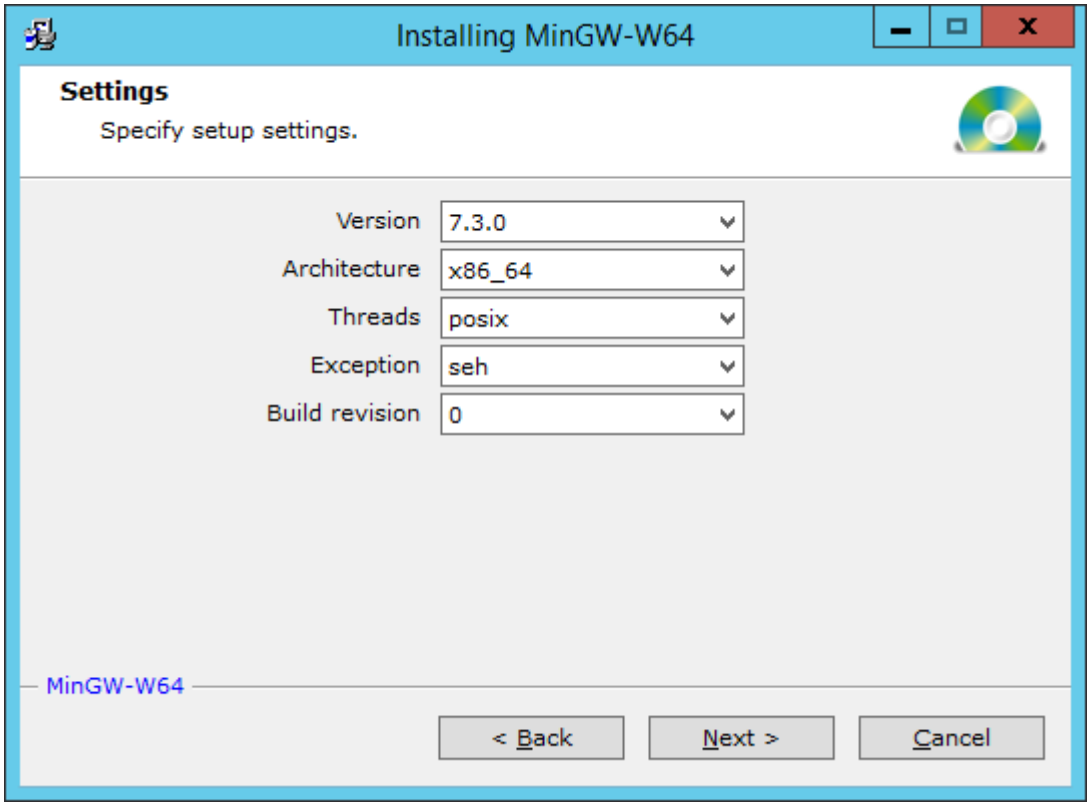


Assuming that you want to use the default settings, tick **Accept** and click **Install**; the software installation will then begin.

Installing a C++ compiler

Once the Stat-JR software is installed you will also need to install a C++ compiler. We recommend MinGW-w64: to download it, open the page https://sourceforge.net/projects/mingw-w64/ in a web browser and click the **Download** button for *mingw-w64-install.exe*.

Running this file will open another installation program. After clicking **Next** on the initial page, you will be presented with a screen asking which version of the compiler you want to install:



Note that we have set the details in the screenshot above to match those expected, by default, by Stat-JR, so if you use these settings then Stat-JR should find the C++ compiler automatically. If you wish, though, you can use a newer version, or a different build revision, than that displayed here, but doing so will mean that you will need to adjust Stat-JR's settings accordingly (we will describe how to do this shortly). Avoid installing version 8.1.0, revision 0 as this contains changes that prevent it from working with the supplied BLAS/LAPACK libraries.

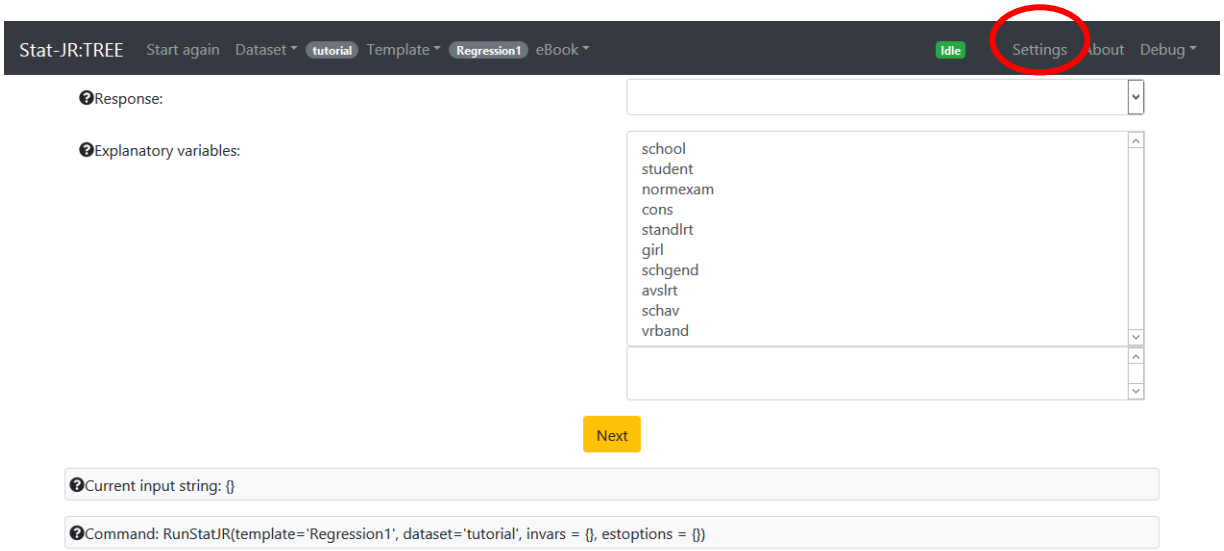
Having made your choices, click the **Next** button (repeatedly), and then **Finish**, to begin the installation.

Opening Stat-JR & modifying the settings

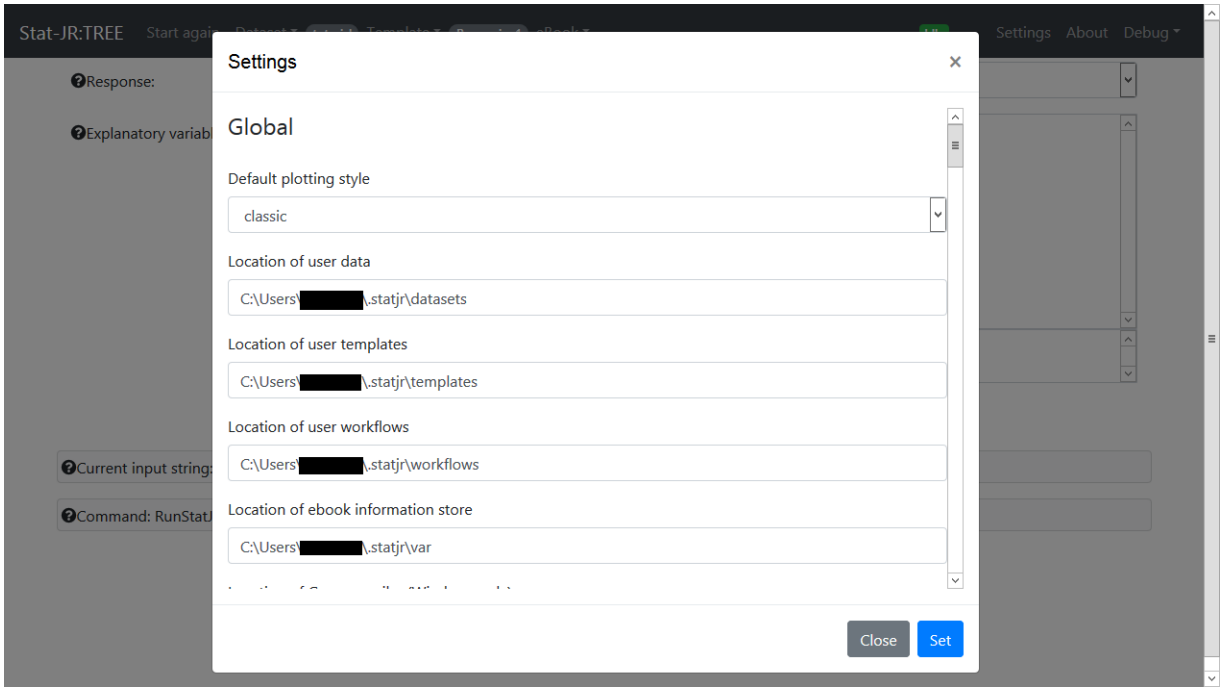
Now that you have installed the necessary software it is time to start Stat-JR:TREE up and ensure that the settings are correct for your machine. To do this, click on the Windows **Start** button and choose:

All Programs > Centre for Multilevel Modelling > StatJR – TREE

Initially a command output window will open (you may want to refer back to this as the software runs, as messages relating to the current progress will appear here; if you encounter any problems with the software, the information here is also useful when requesting technical support). Then, after a short while, your default web browser will start-up (**Chrome or Firefox are recommended**), and the following page will be displayed:



We will outline the various options on this page in detail further below, but for present purposes note the **Settings** button towards the top-right of the screen (circled above). Clicking on this brings up the following:



Here you can change a number of details regarding how Stat-JR operates, as well as how it links to other third party software packages.

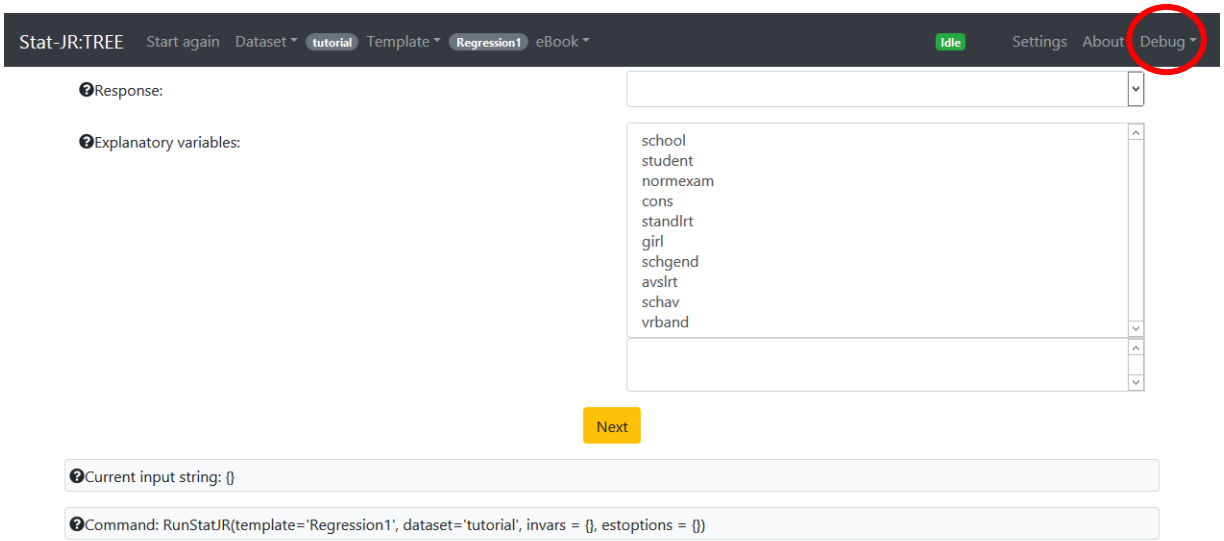
The first option, **Default plotting style**, allows you to set the style of plots created by the Stat-JR system. By default this is set to **classic** for backwards compatibility with previous Stat-JR versions, however if you can easily change to another provided style (see https://matplotlib.org/gallery/style_sheets/style_sheets_reference.html?highlight=style%20sheets%20reference for details of individual styles) or define your own (although this has to be done for each of the included Stat-JR modules – see the mpl-data\stylelib directory within each module).

The next three options – **Location of user data / templates / workflows** – tell Stat-JR where to search for these respective objects. You should not need to change these, however the locations are worth noting if you plan to add any of your own resources to the system.

The next option (**Location of eBook information store**) defines where temporary files used by the DEEP system are stored; again, you should not need to change this.

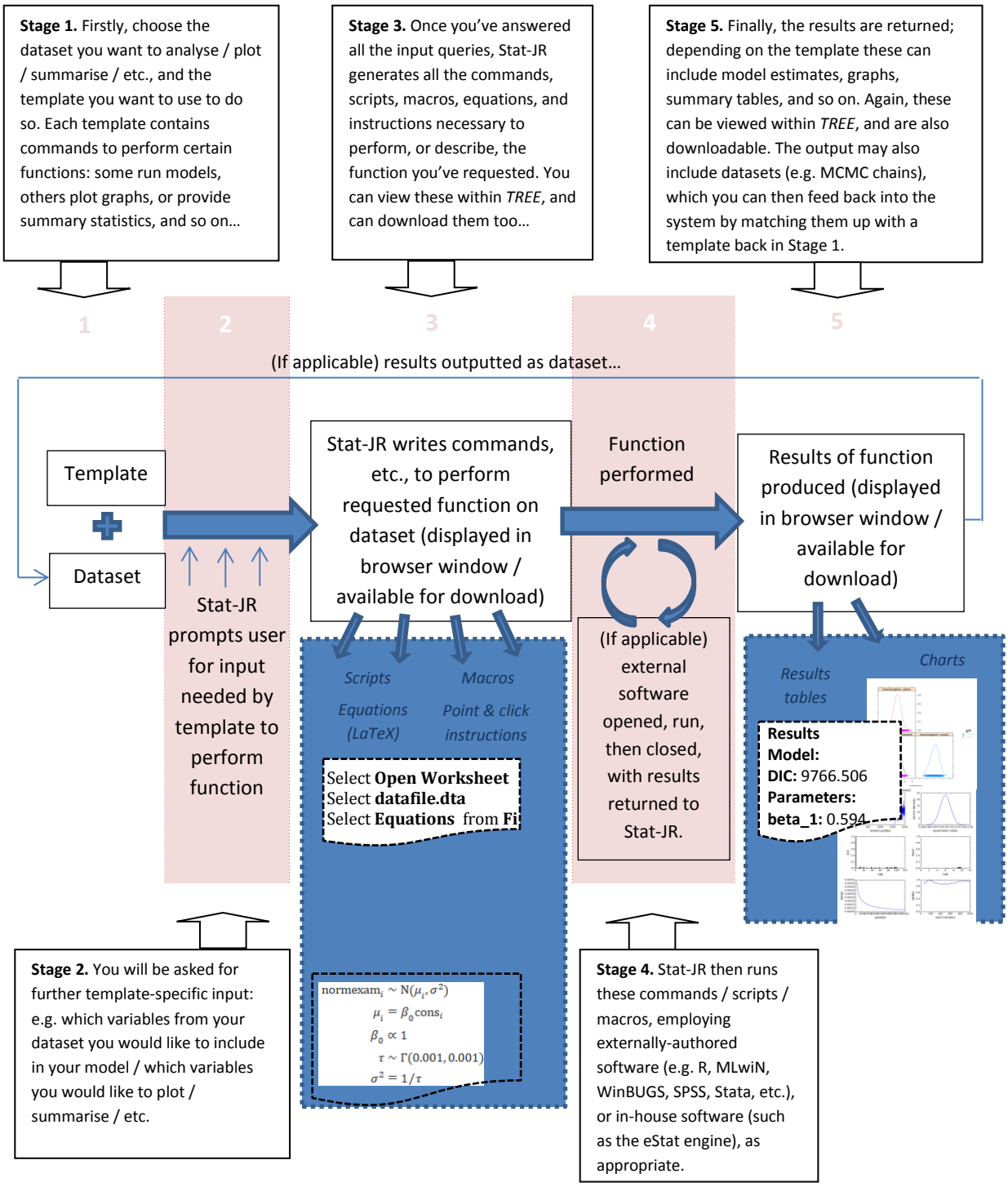
The final system-wide setting (**Location of G++ compiler**), allows you to specify were Stat-JR will locate the C++ compiler. If you have used the C++ compiler recommended earlier in this document, you will not need to change this, however if you have installed a different version, or installed it into a different location, then you will need to adjust this to match.

The remaining options relate to specific estimation packages (e.g. R, MLwiN, Win/OpenBUGS, JAGS, Stata, etc.), usually defining where Stat-JR can find the necessary files to run the related software. If you want to use any of these you will need to adjust these locations to match the version on your machine (assuming you have these packages installed). Note that if you change these paths, and want to use the packages immediately in the current session, then you will need to select **Reload Packages** from the **Debug** drop-down menu (circled below). If you look in the command window after doing so you can determine whether or not the new software has been located.


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
eBook
Idle
Settings
About
Debug
Response:
Explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
Next
Current input string: {}
Command: RunStatJR(template='Regression1', dataset='tutorial', invars = {}, estoptions = {})

Everyday operation of Stat-JR:TREE – an overview

Now we've got the settings sorted, we next provide an overview of everyday operation of Stat-JR:TREE, in which you generally proceed through the following five stages:



We briefly highlight the main features, with screenshots, of each of these five stages below.

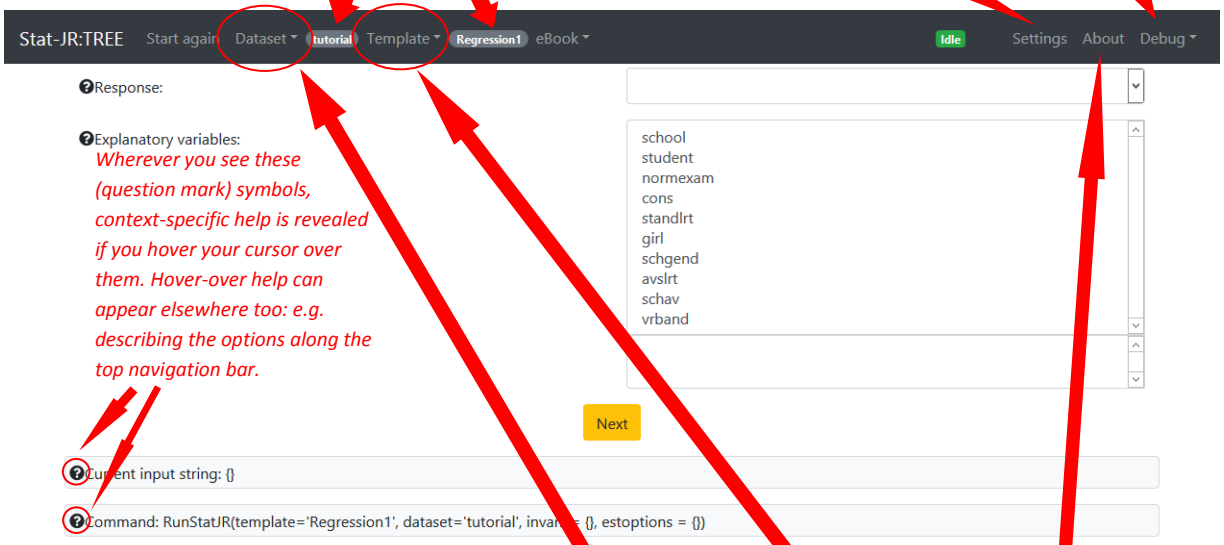
Stage 1: Selecting a template & dataset

- You have seen this opening screen in an earlier section, but here we provide further notes on the various options it presents:

*If you have modified any files in the **templates**, **datasets** or **packages** folders, then you can reload their contents into the current session via the Debug menu.*

This link allows you to change settings such as paths to user data and user template folders, paths to interoperating software, and optimisation settings for generated code.

Here you can see which dataset and template are currently selected. Hovering your cursor over these names will reveal a description of each (if available).

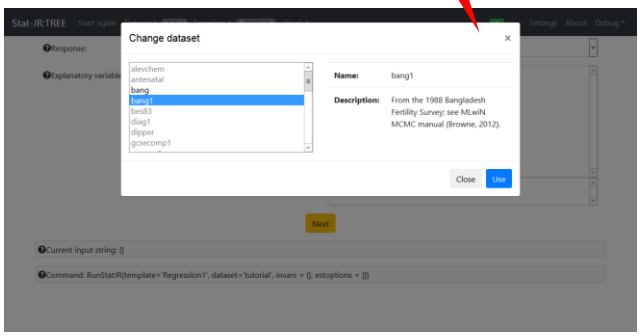


Wherever you see these (question mark) symbols, context-specific help is revealed if you hover your cursor over them. Hover-over help can appear elsewhere too: e.g. describing the options along the top navigation bar.

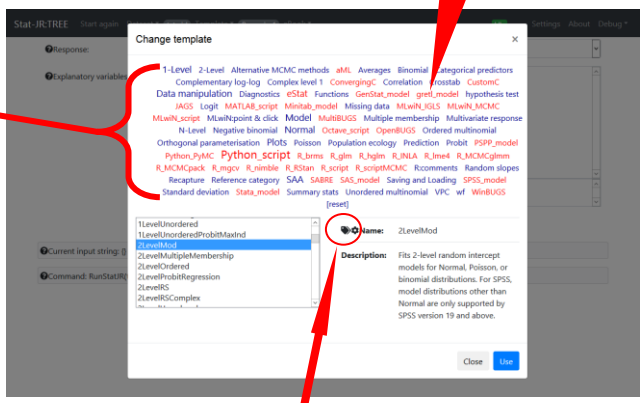
*Clicking on the down arrow symbol just to the right of the **Dataset** heading in the top bar will bring up a menu. Select **Choose** to bring up the window, below, allowing you to nominate a dataset other than that currently selected…*

*…and likewise for the **Template**…*

This button opens author information (including details of how to cite Stat-JR), with a link to Stat-JR webpages which contain further support, including frequently asked questions & a user forum



You can select one or more of these terms to help you find relevant templates; the blue tags describe the functional aspects of a template, whilst the red terms describe the engines / packages supported by a template. To unselect terms, press [reset]



Clicking the 'label' symbol brings up a list of tags, whilst clicking the 'cog' symbol brings up a list of supported engines / packages.

- Note that on this page, and on other screens, wherever you see the question mark symbols, **context-specific help** is revealed if you hover your cursor over them. Hover-over help can appear elsewhere too: e.g. describing the options along the top navigation bar.

- Selecting **Dataset > Choose** or **Template > Choose** from the top bar will reveal lists of available datasets and templates. For each, find the one you want from the list, and then press **Use**.

- Once you have specified the **template** and **dataset** you want to use, you can begin to specify your **inputs**.

- Note, when choosing a template, you can use the **cloud terms to help your search**: the blue tags describe functional aspects of the templates, whilst the red terms describe which engines / packages the templates support (you can combine search terms by clicking on more than one, and cancel your selections by pressing **[reset]**).

Stage 2: Providing template-specific input

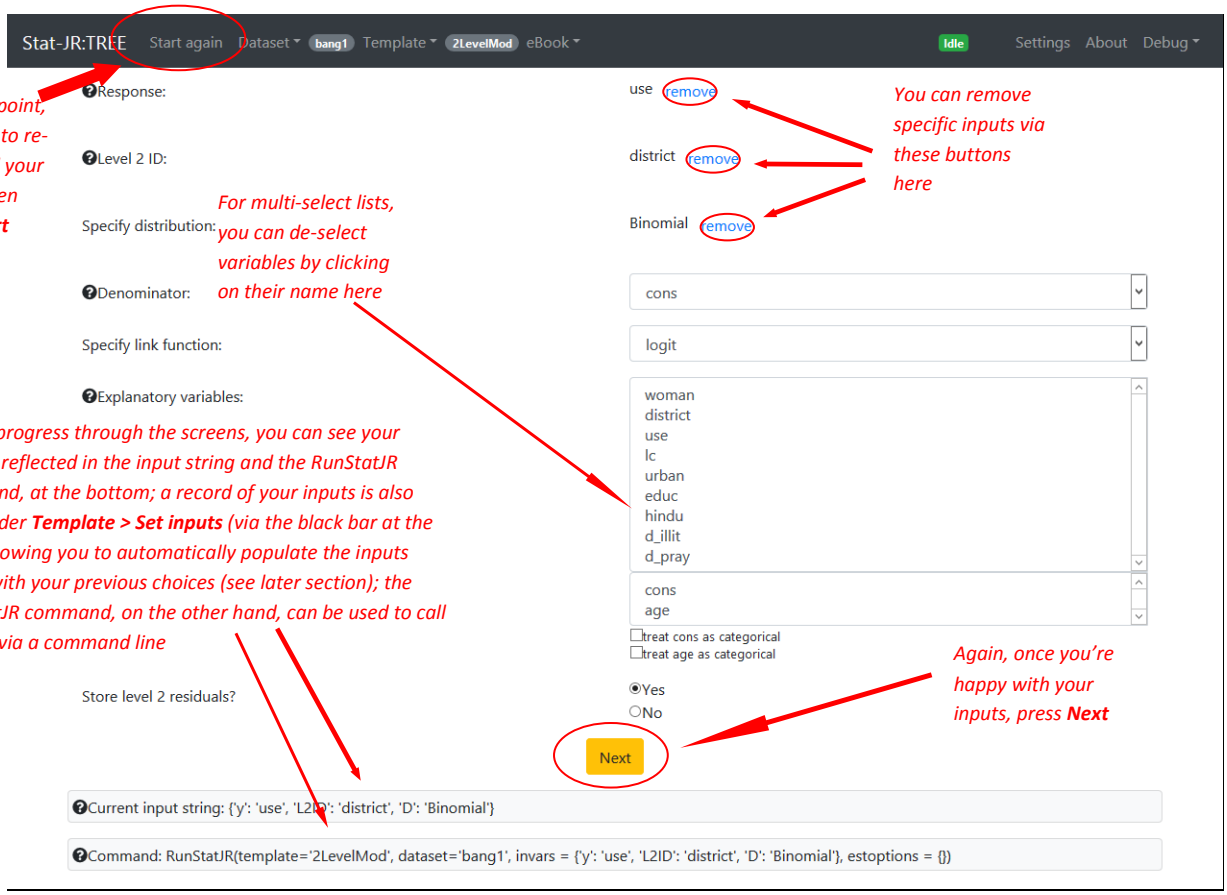
- Once your desired **Dataset** and **Template** are selected, you can start answering the input questions. These are required by Stat-JR to allow the template to perform the appropriate executions with your dataset; these inputs vary between templates, and also within templates too, depending on your earlier choices as you progress through the screens.

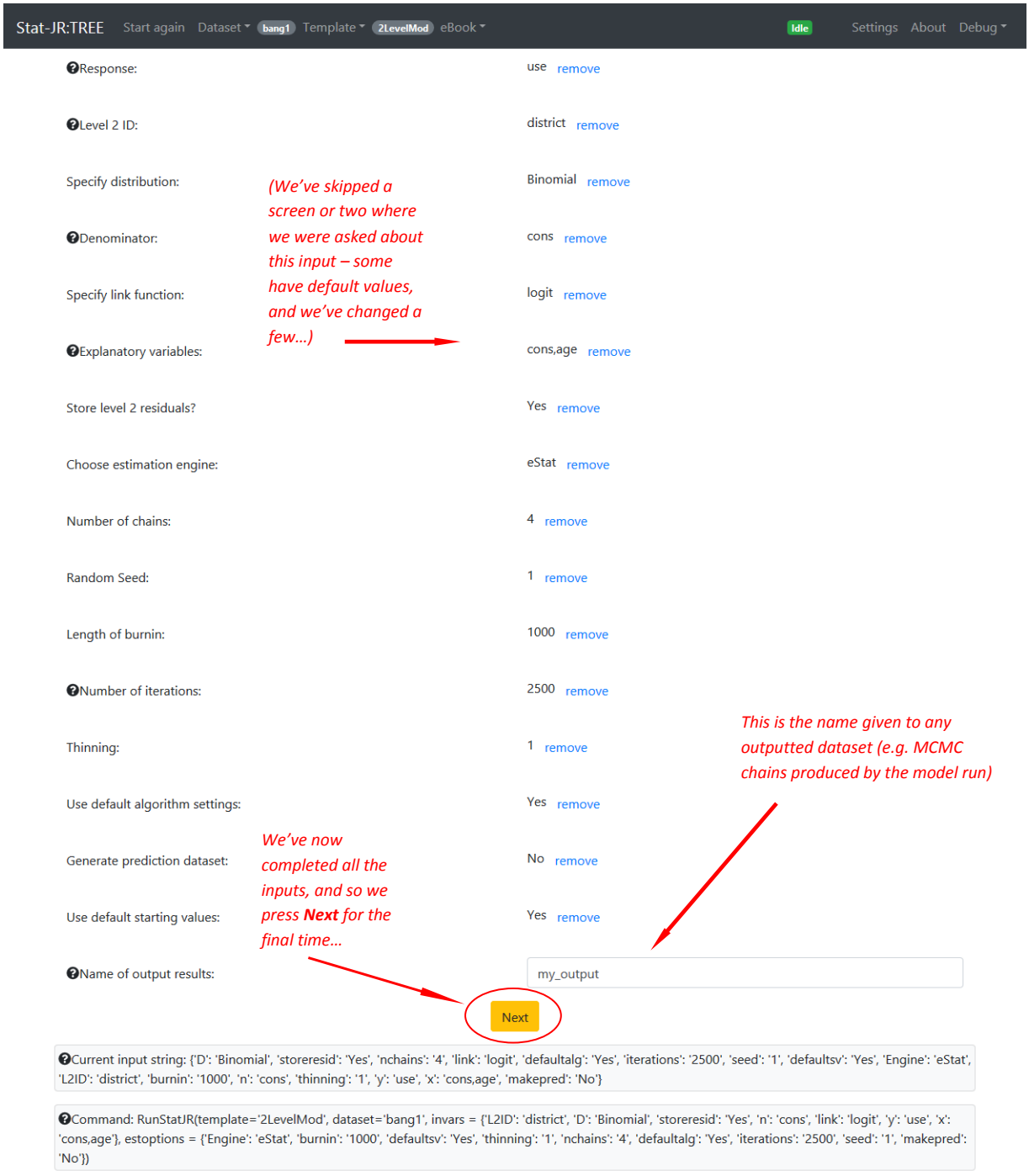
- For multi-choice lists you can de-select variables by simply clicking on their name in the list of selected items.

- Press **Next** each time you've completed the input questions on the current page.

- Then, if applicable, more inputs will be revealed for you to specify, although you can still change your previous input choices via the **remove** button which you'll see next to each one. Alternatively, to re-specify *all* your inputs, press **Start again** (in the top bar).

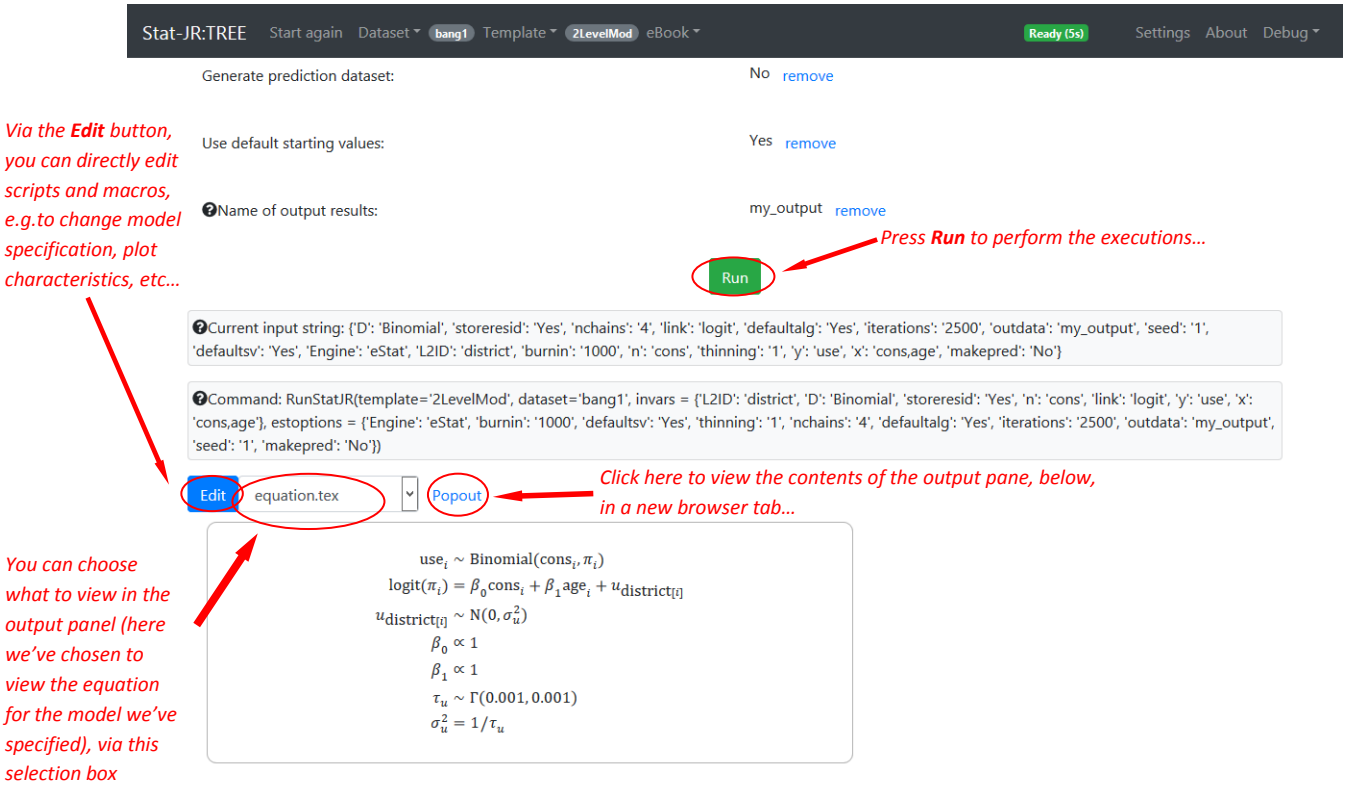
- When asked for the **Name of the output results**, this will be the name given to any outputted dataset which results from running the template (see Stage 5).


Stat-JR:TREE
Start again
Dataset
bang1
Template
2LevelMod
eBook
Idle
Settings
About
Debug
Response:
Level 2 ID:
Specify distribution:
Denominator:
Specify link function:
Explanatory variables:
Store level 2 residuals?
use remove
district remove
Binomial remove
cons
logit
woman
district
use
lc
urban
educ
hindu
d_illit
d_pray
cons
age
treat cons as categorical
treat age as categorical
Yes
No
Next
Current input string: {'y': 'use', 'L2ID': 'district', 'D': 'Binomial'}
Command: RunStatJR(template='2LevelMod', dataset='bang1', invars = {'y': 'use', 'L2ID': 'district', 'D': 'Binomial'}, estoptions = {})
If, at any point, you want to re-specify all your inputs, then press Start again
For multi-select lists, you can de-select variables by clicking on their name here
You can remove specific inputs via these buttons here
As you progress through the screens, you can see your choices reflected in the input string and the RunStatJR command, at the bottom; a record of your inputs is also kept under Template > Set inputs (via the black bar at the top), allowing you to automatically populate the inputs boxes with your previous choices (see later section); the RunStatJR command, on the other hand, can be used to call Stat-JR via a command line
Again, once you're happy with your inputs, press Next


Stat-JR:TREE
Start again
Dataset
bang1
Template
2LevelMod
eBook
Idle
Settings
About
Debug
Response:
use remove
Level 2 ID:
district remove
Specify distribution:
Binomial remove
Denominator:
cons remove
Specify link function:
logit remove
Explanatory variables:
cons,age remove
Store level 2 residuals?
Yes remove
Choose estimation engine:
eStat remove
Number of chains:
4 remove
Random Seed:
1 remove
Length of burnin:
1000 remove
Number of iterations:
2500 remove
Thinning:
1 remove
Use default algorithm settings:
Yes remove
Generate prediction dataset:
No remove
Use default starting values:
Yes remove
Name of output results:
my_output
Next
(We've skipped a screen or two where we were asked about this input – some have default values, and we've changed a few...)
This is the name given to any outputted dataset (e.g. MCMC chains produced by the model run)
We've now completed all the inputs, and so we press Next for the final time...
Current input string: {'D': 'Binomial', 'storeresid': 'Yes', 'nchains': '4', 'link': 'logit', 'defaultalg': 'Yes', 'iterations': '2500', 'seed': '1', 'defaultsv': 'Yes', 'Engine': 'eStat', 'L2ID': 'district', 'burnin': '1000', 'n': 'cons', 'thinning': '1', 'y': 'use', 'x': 'cons,age', 'makepred': 'No'}
Command: RunStatJR(template='2LevelMod', dataset='bang1', invars = {'L2ID': 'district', 'D': 'Binomial', 'storeresid': 'Yes', 'n': 'cons', 'link': 'logit', 'y': 'use', 'x': 'cons,age'}, estoptions = {'Engine': 'eStat', 'burnin': '1000', 'defaultsv': 'Yes', 'thinning': '1', 'nchains': '4', 'defaultalg': 'Yes', 'iterations': '2500', 'seed': '1', 'makepred': 'No'})

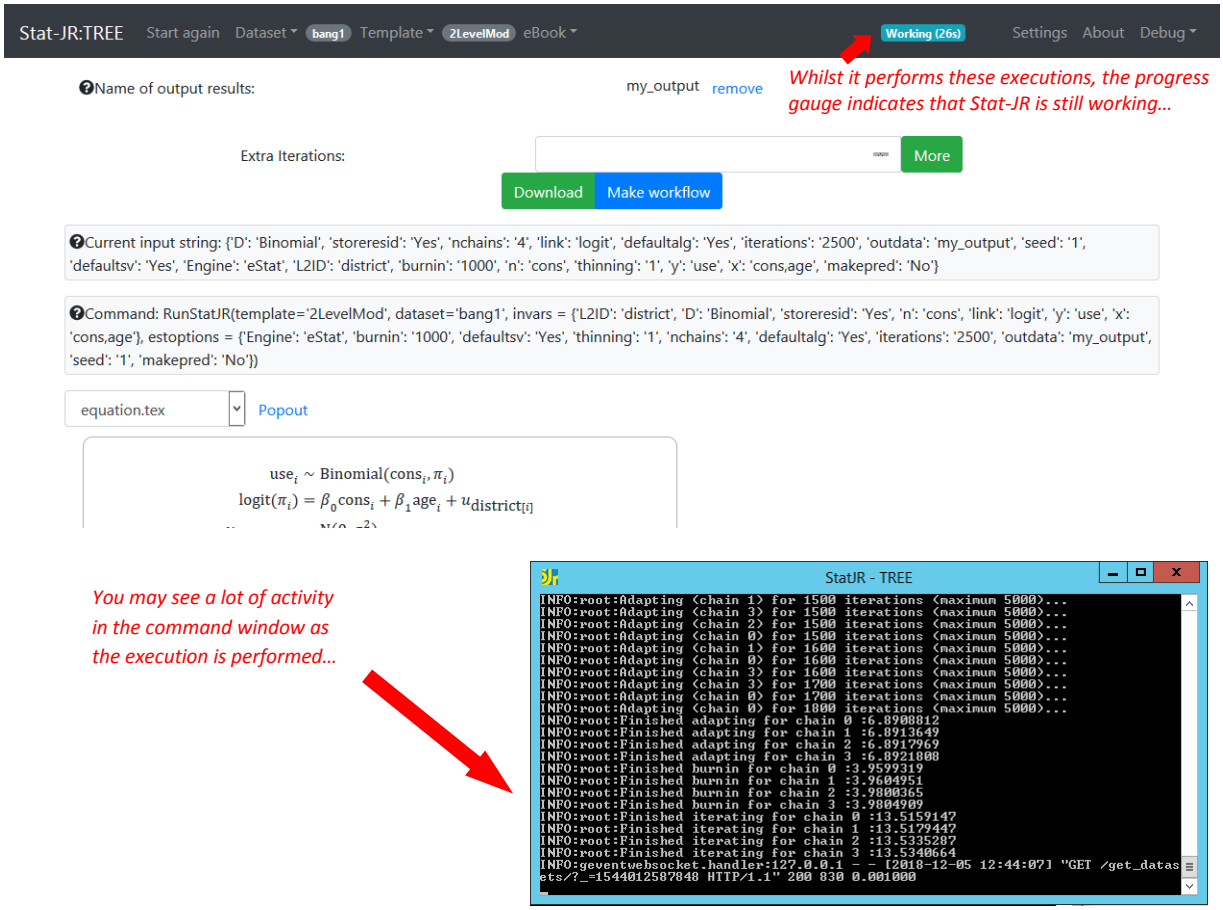
Stage 3: Outputting the files to run the desired execution

- Once you're pressed **Next** after the final input, Stat-JR returns a number of initial outputs which you can view in the output pane at the bottom of the window.
- Note that Stat-JR hasn't done everything you want it to do yet: it's just producing preliminary files telling you what it's going to do, and how it's going to do it.
- To select particular content to view in the output pane, use the drop-down menu just above it.
- The **Popout** button, just above the output pane, allows you to view its contents in a new browser tab.
- Pressing **Run** performs the executions described by the scripts, etc., returned in the output pane.



Stage 4: Running the execution

- Once you've pressed **Run**, the executions specified by you are peformed.
- Depending on your choices, this may take anything from a second or two (e.g. to produce a simple plot, fit a model using certain methods of estimation, produce summary data, etc.), to many minutes (e.g. to run MCMC chains for a large number of iterations).
- If appropriate (e.g. if the template supports inter-operability, and if you have chosen to employ it when prompted), externally-authored software packages (e.g. R, MLwiN, WinBUGS, SPSS, etc.) are opened, run, then closed, and the results are returned to Stat-JR (note that you will need these third-party software packages installed to exploit this feature; once you have installed the third-party packages of interest to you, you can tell Stat-JR where to find them by specifying the path via the **Settings** button in the black bar at the top: once you have done so press **Set**, and then **Debug > Reload packages**).
- Whilst the execution runs, you may see a lot of activity in the black command window, which may help you keep a track of progress.

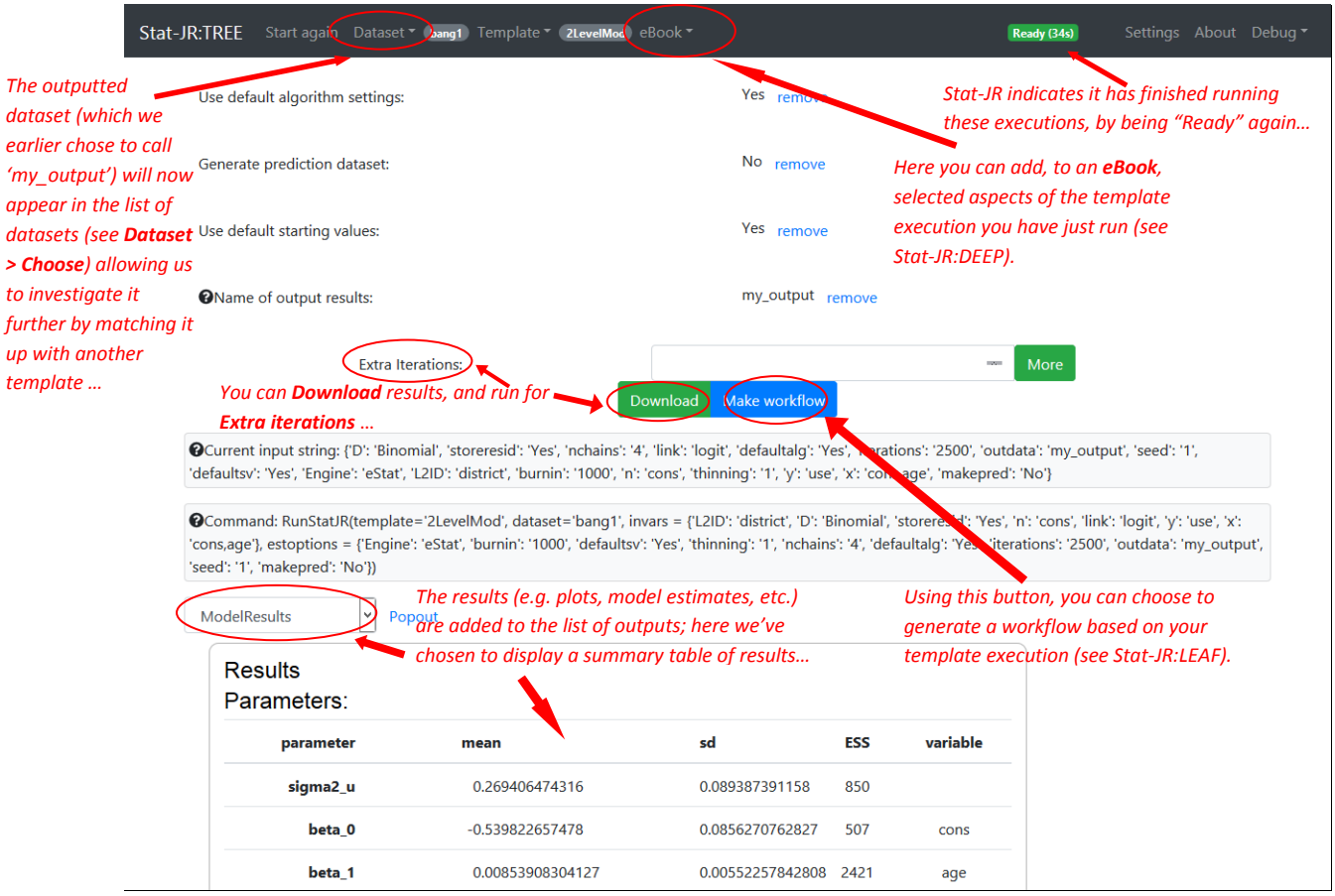


Stage 5: The results

- Once the executions have run, the progress guage, in the top-right corner, will change from “Working” to “Ready”, and the drop-down list, just above the output pane, will now be populated with more results.

- Depending on the template, a range of buttons / boxes appear above the output pane allowing you to e.g. **Download** the results, **Make workflow**, and run chains for **Extra iterations**.

- If applicable, an outputted dataset now appears in the list of datasets (see **Dataset > Choose**, via the top bar).



This ends the quick-start guide to *TREE*; for a more detailed overview, see the *Beginner’s Guide to Stat-JR’s TREE interface* and also the *Advanced User’s Guide to Stat-JR*, plus further information and guidance on the Stat-JR website, http://www.bristol.ac.uk/cmm/software/statjr/.