# Stat-JR Workflow & eBook Workshop

3rd September 2015

Bristol

1. Introduce eBook project

1. Introduce eBook project
2. Overview of Stat-JR software package:
   – past & current developments

1. Introduce eBook project
2. Overview of Stat-JR software package:
   – past & current developments
3. Bringing the two together: developing Stat-JR tools & content via the eBook project

# eBook project funded by ESRC

E·S·R·C
ECONOMIC
& SOCIAL
RESEARCH
COUNCIL

eBook project funded by ESRC

Research objectives include:

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies

## eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes
  - statistical analysis assistant

eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
    - library of case studies
    - library of methodological advice / notes
    - statistical analysis assistant

Project uses Stat-JR package…

eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes
  - statistical analysis assistant

Project uses Stat-JR package…

Stat-JR

# Stat-JR

- “Stat-juh”

# Stat-JR

- “Stat-juh”
- Jon Rasbash

# Stat-JR

- “Stat-juh”
- Jon Rasbash

# Stat-JR

- “Stat-juh”
- Jon Rasbash
- ESRC-funded

# Stat-JR

- "Stat-juh"
- Jon Rasbash
- ESRC-funded
- Developed by CMM members & other colleagues at...

# Stat-JR

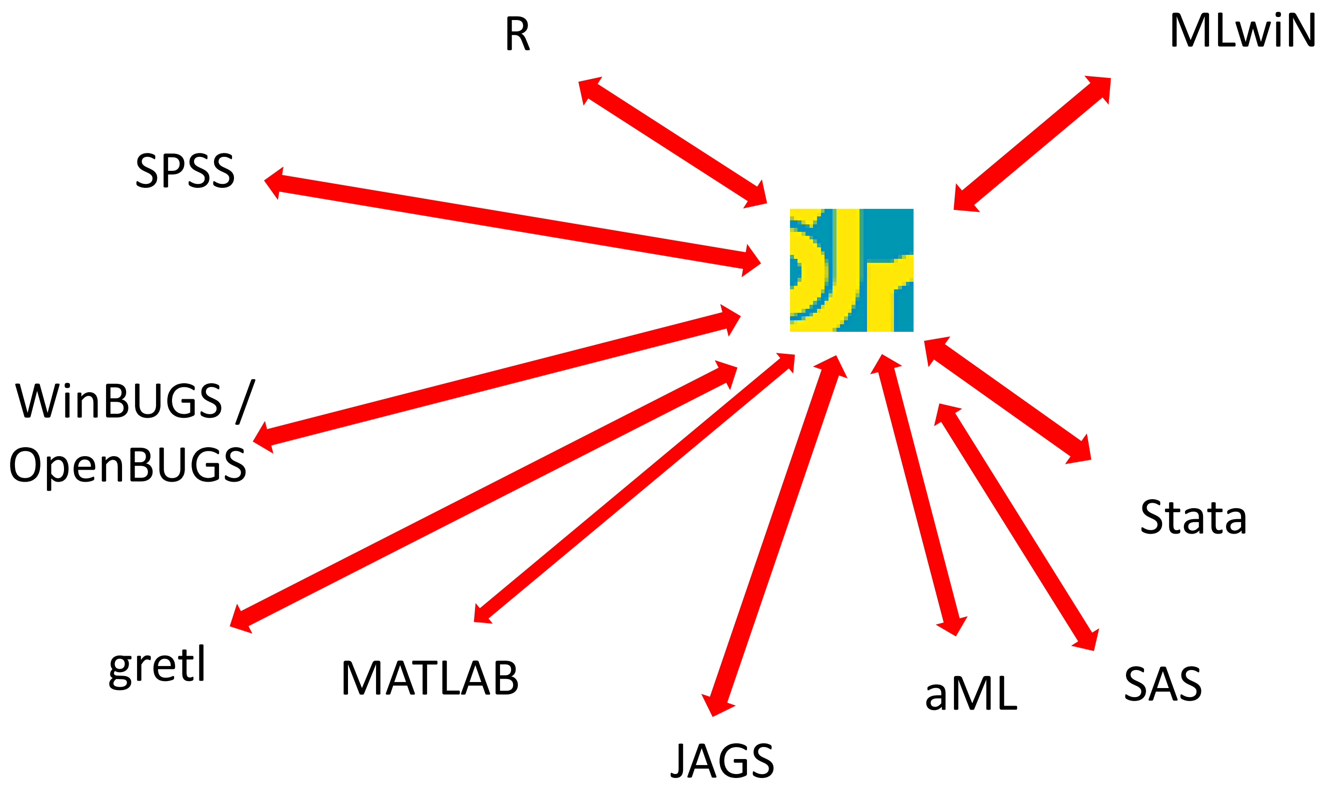
- "Stat-juh"
- Jon Rasbash
- ESRC-funded
- Developed by CMM members & other colleagues at…
- Current release version distributed with Bristol's MLwiN software package…

# Stat-JR

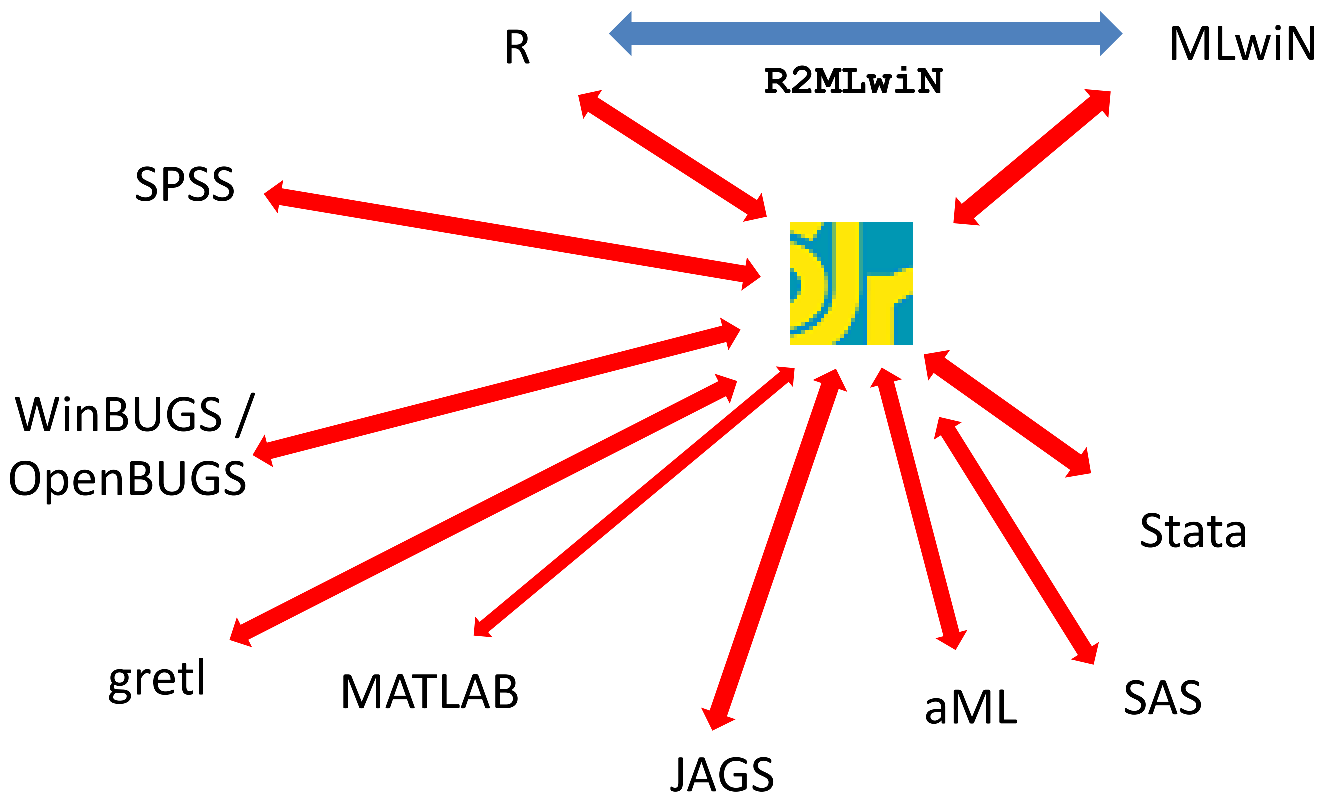
- "Stat-juh"
- Jon Rasbash
- ESRC-funded
- Developed by CMM members & other colleagues at…
- Current release version distributed with Bristol's MLwiN software package…
- …which is free to UK academics, but other users need to pay.

# Stat-JR can interoperate with other packages

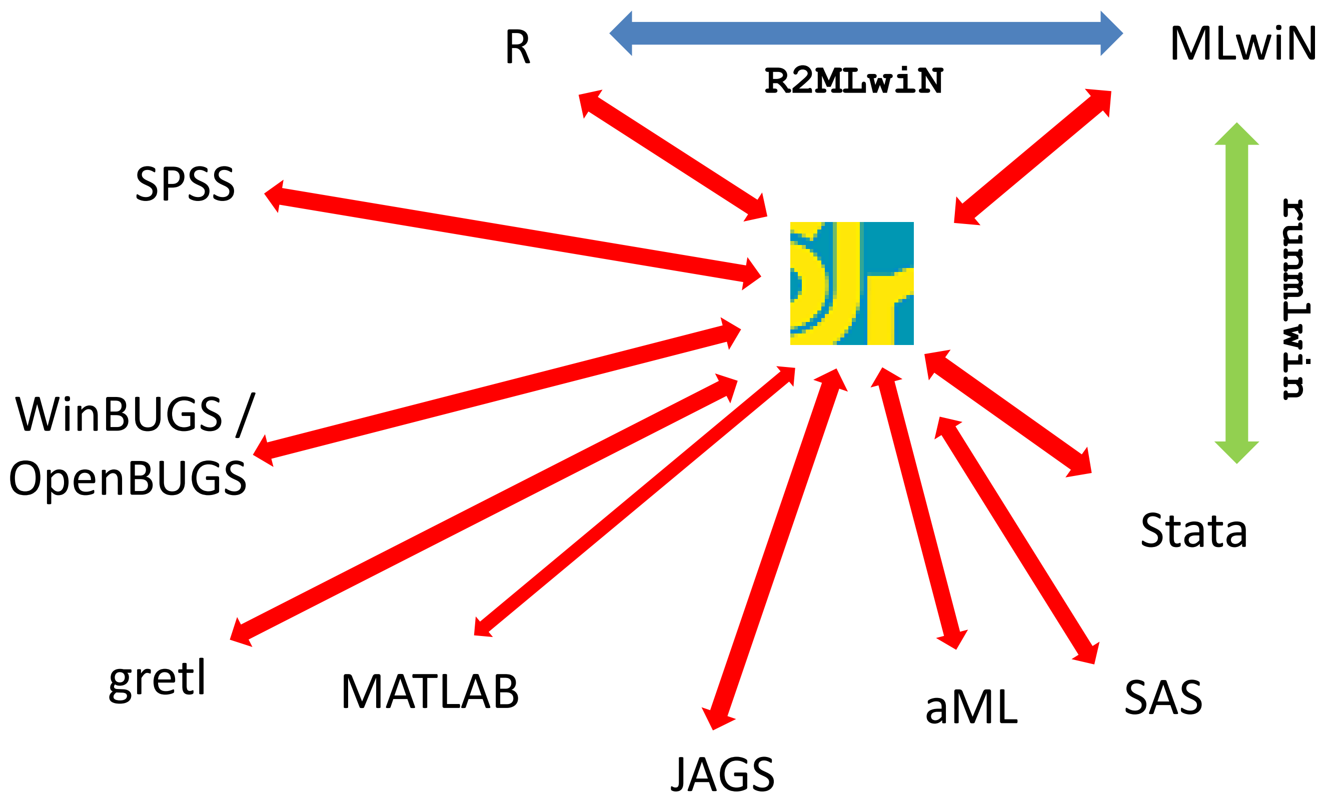
# Stat-JR can interoperate with other packages



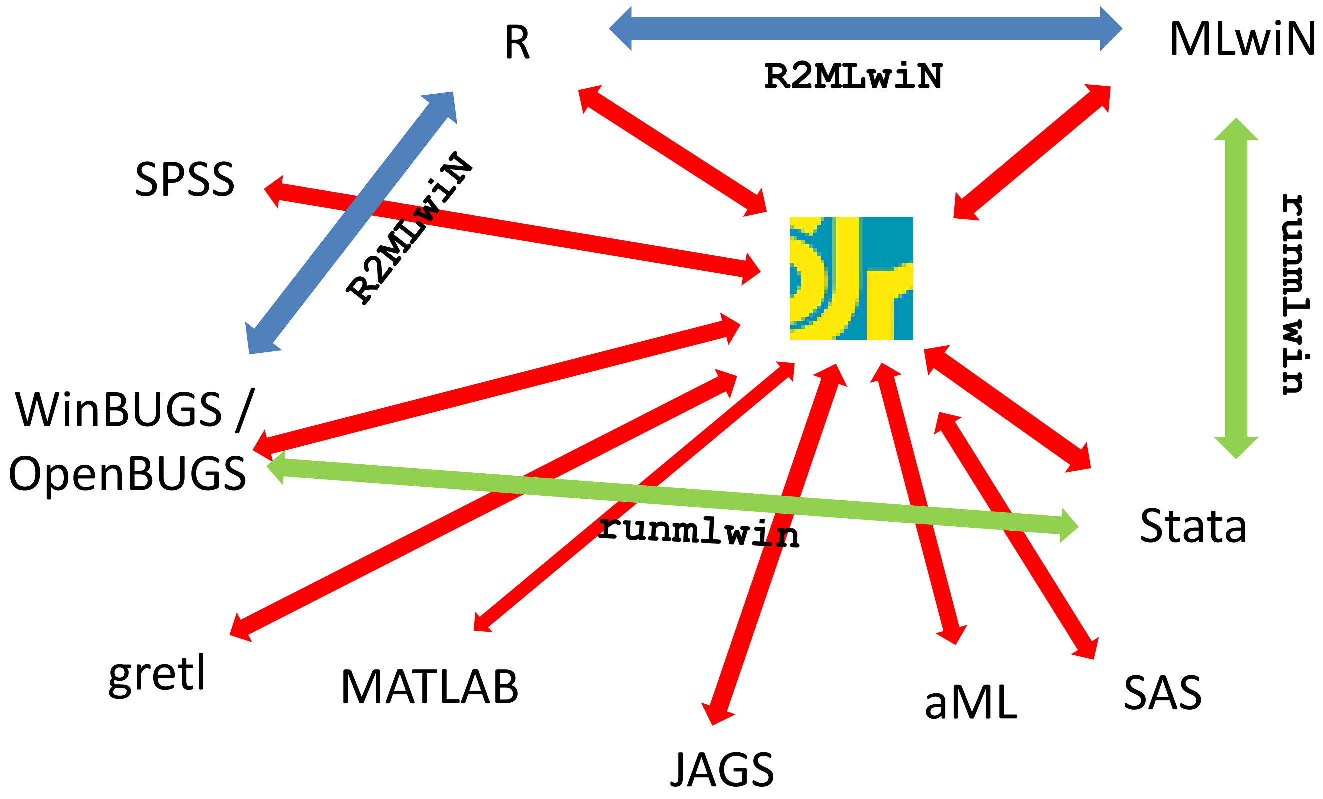
# Stat-JR can interoperate with other packages

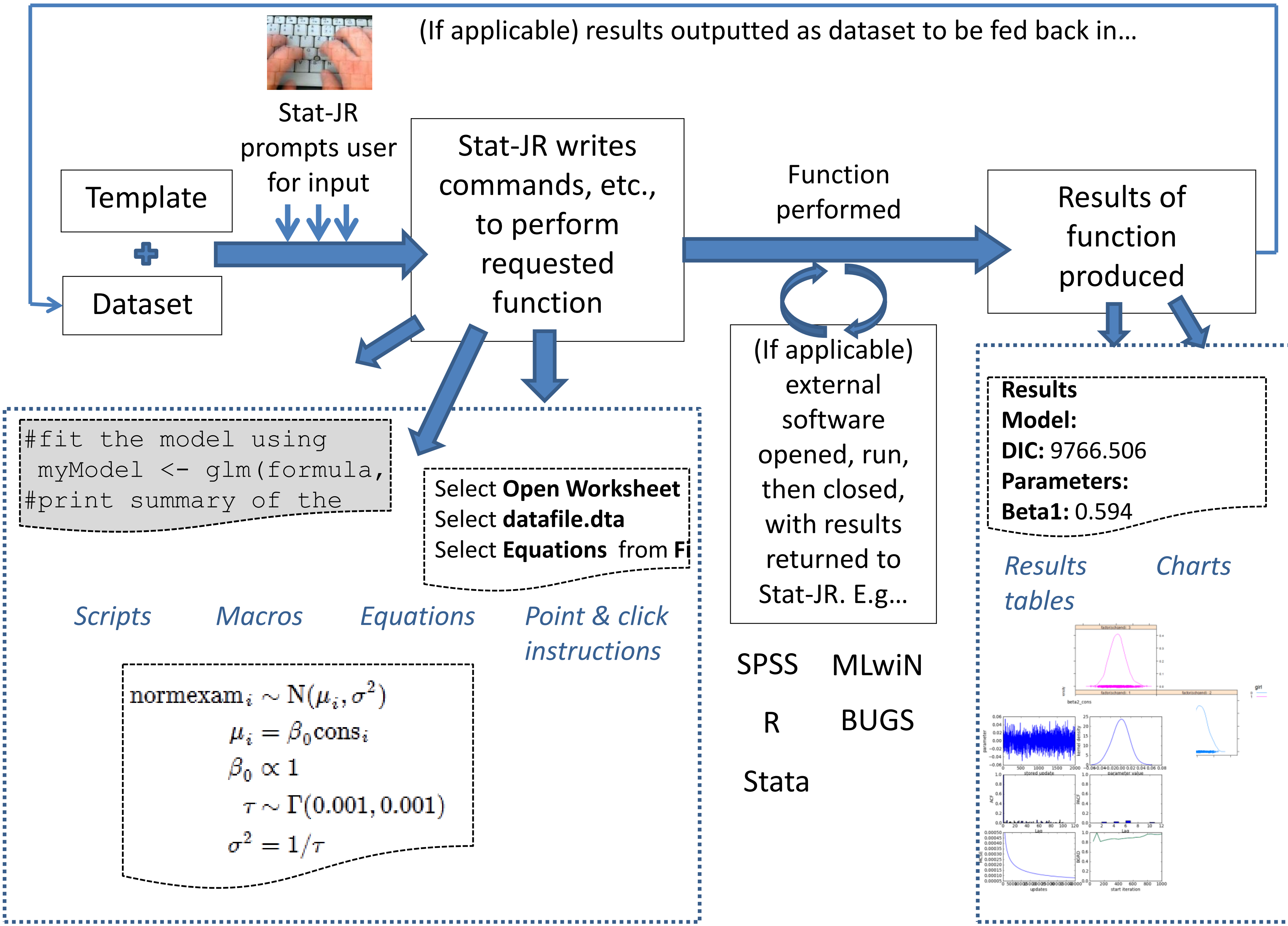


# Stat-JR can interoperate with other packages



# Stat-JR can interoperate with other packages





(If applicable) results outputted as dataset to be fed back in…
Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g…
SPSS
MLwiN
R
BUGS
Stata
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
$\mathrm{normexam}_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
Results
Model:
DIC: 9766.506
Parameters:
Beta1: 0.594
Results tables
Charts

# Choice of interface

Three different ways to interact with Stat-JR:

1. **Point-and-click menu-driven** interface (TREE)

2. **eBook** interface (DEEP)

3. **Command line** interface (runStatJR)

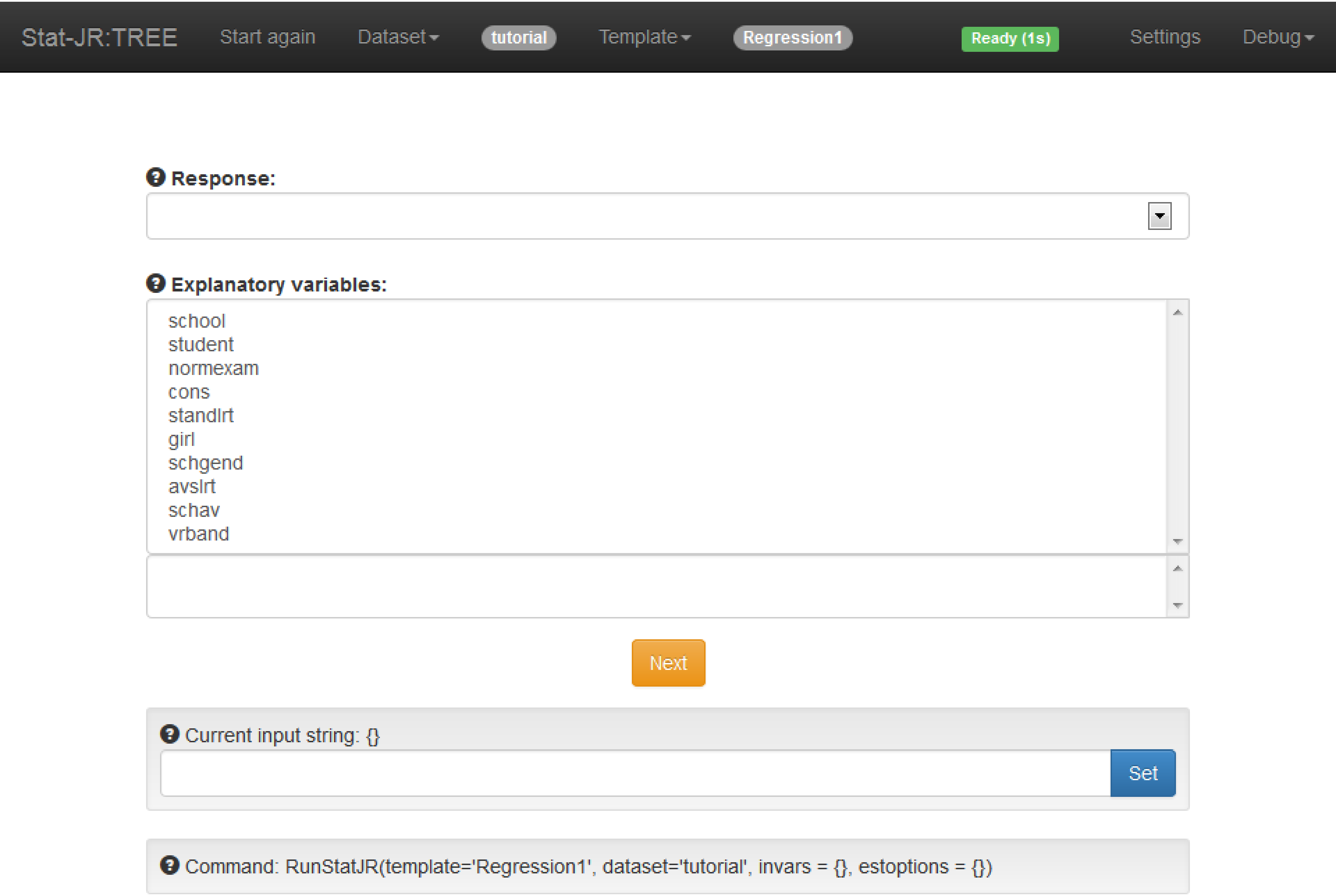
# Choice of interface

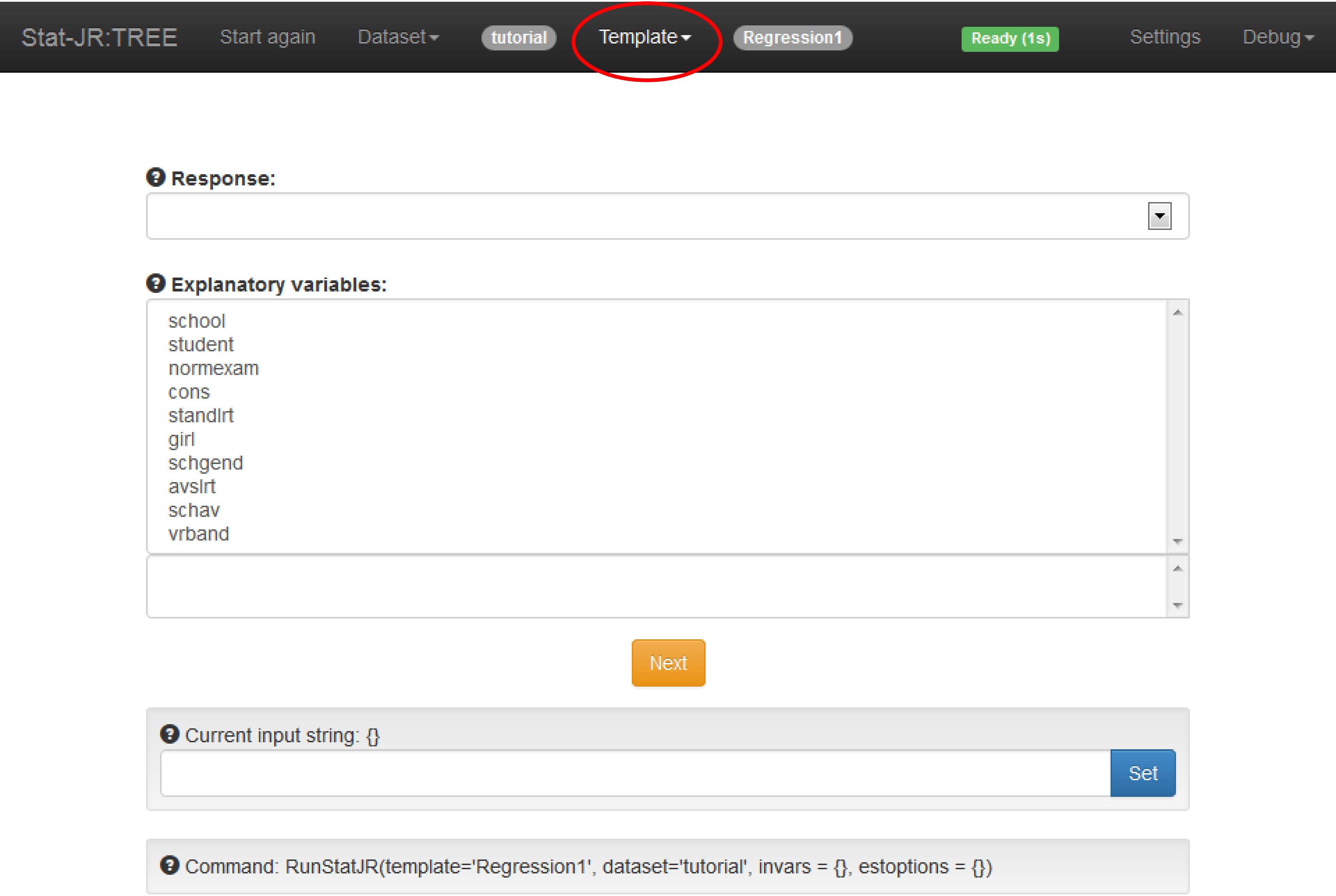
Three different ways to interact with Stat-JR:

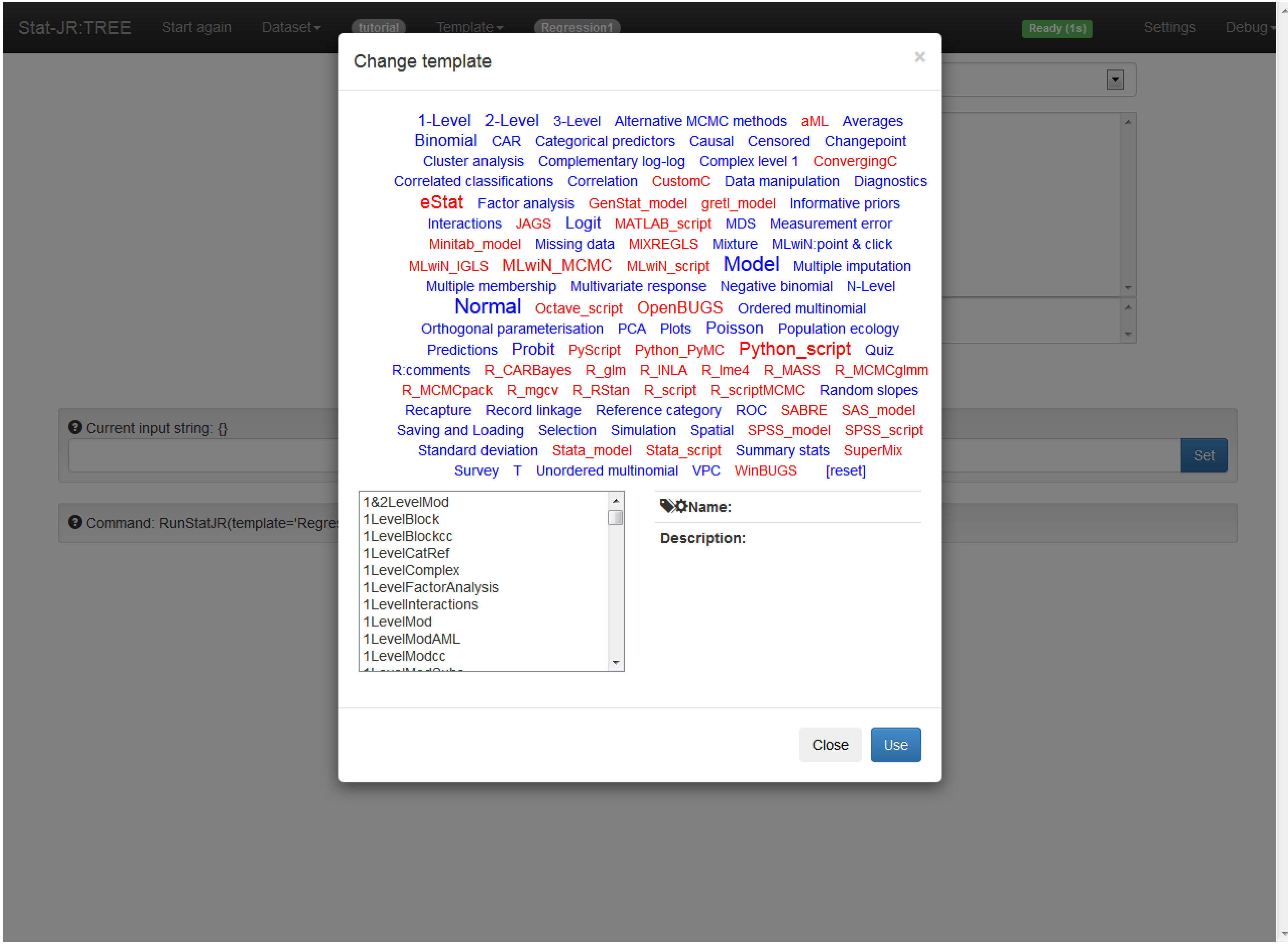
1. **Point-and-click menu-driven** interface (TREE)

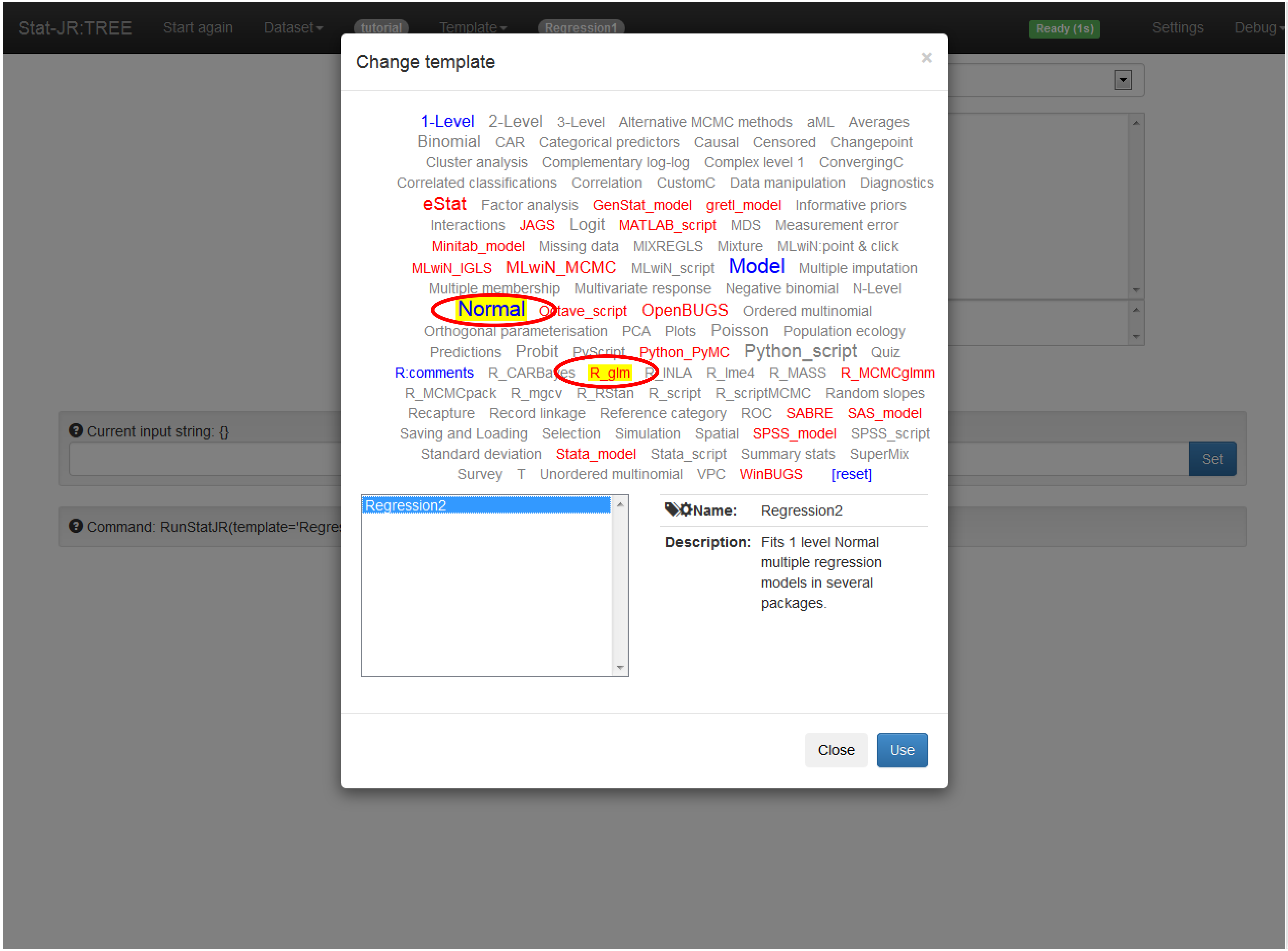
2. **eBook** interface (DEEP)

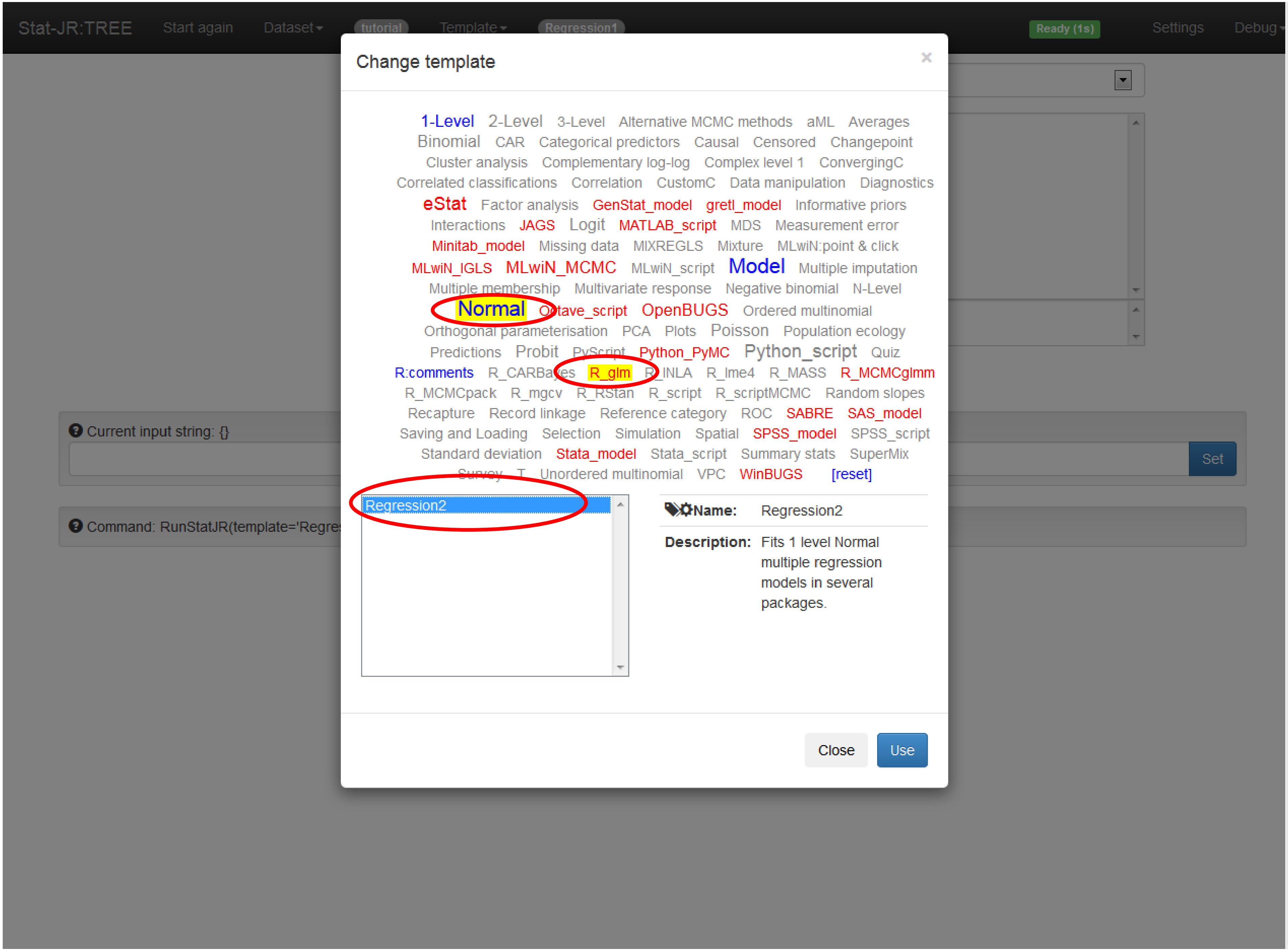
3. **Command line** interface (runStatJR)

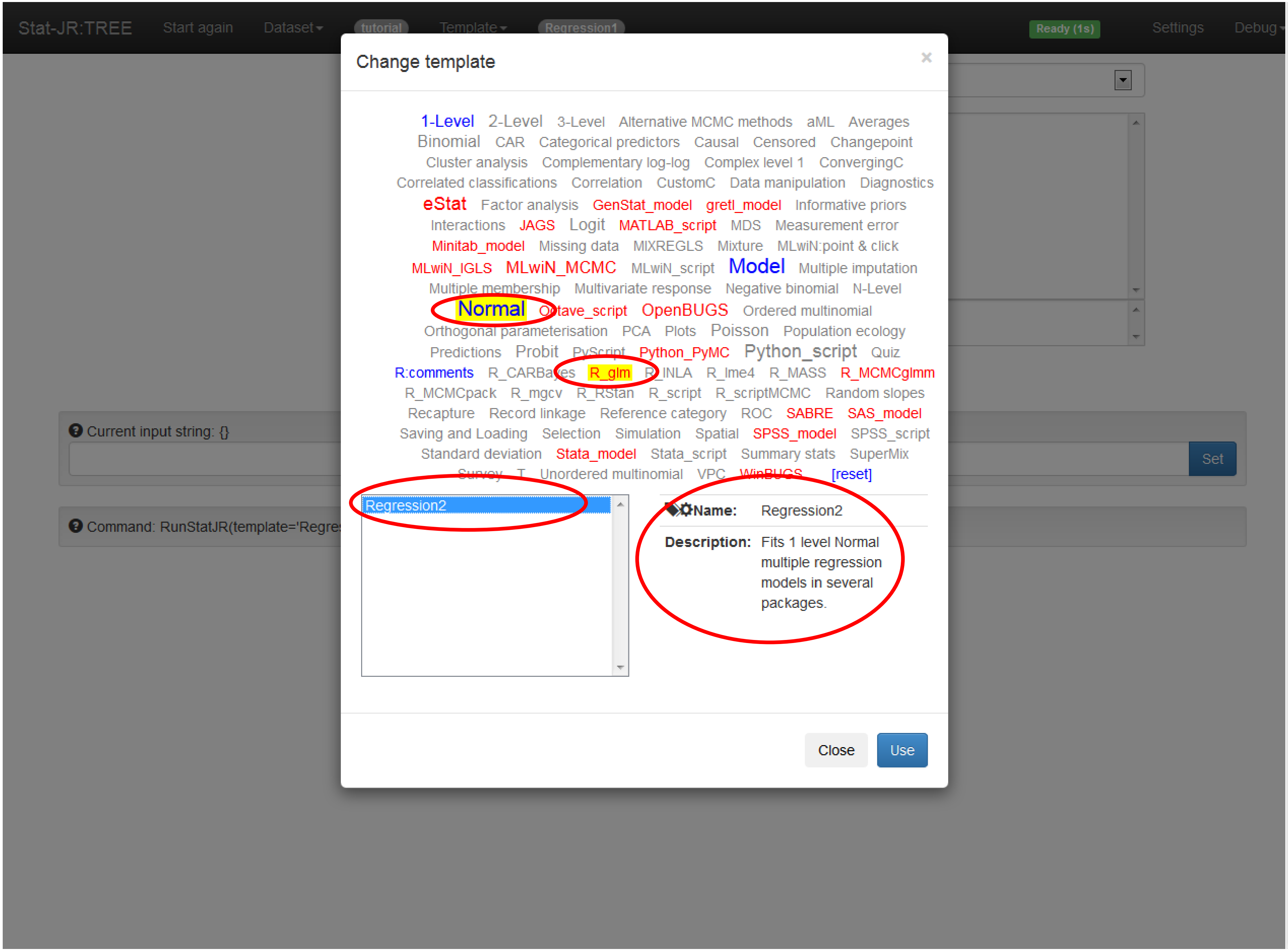

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Response:
Explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
Next
Current input string: {}
Set
Command: RunStatJR(template='Regression1', dataset='tutorial', invars = {}, estoptions = {})

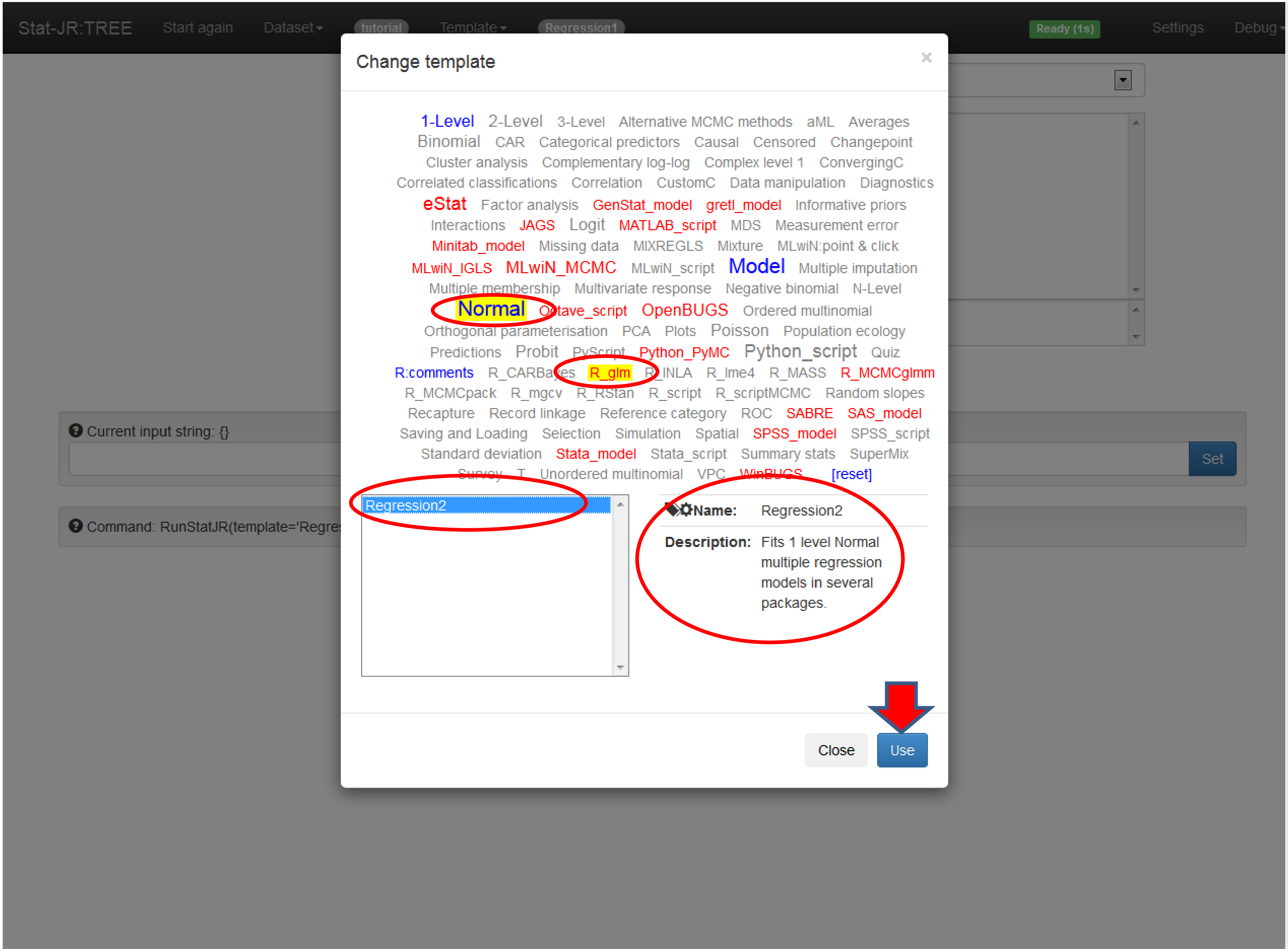

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Response:
Explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
Next
Current input string: {}
Set
Command: RunStatJR(template='Regression1', dataset='tutorial', invars = {}, estoptions = {})


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Change template
1-Level 2-Level 3-Level Alternative MCMC methods aML Averages
Binomial CAR Categorical predictors Causal Censored Changepoint
Cluster analysis Complementary log-log Complex level 1 ConvergingC
Correlated classifications Correlation CustomC Data manipulation Diagnostics
eStat Factor analysis GenStat_model gretl_model Informative priors
Interactions JAGS Logit MATLAB_script MDS Measurement error
Minitab_model Missing data MIXREGLS Mixture MLwiN:point & click
MLwiN_IGLS MLwiN_MCMC MLwiN_script Model Multiple imputation
Multiple membership Multivariate response Negative binomial N-Level
Normal Octave_script OpenBUGS Ordered multinomial
Orthogonal parameterisation PCA Plots Poisson Population ecology
Predictions Probit PyScript Python_PyMC Python_script Quiz
R:comments R_CARBayes R_glm R_INLA R_lme4 R_MASS R_MCMCglmm
R_MCMCpack R_mgcv R_RStan R_script R_scriptMCMC Random slopes
Recapture Record linkage Reference category ROC SABRE SAS_model
Saving and Loading Selection Simulation Spatial SPSS_model SPSS_script
Standard deviation Stata_model Stata_script Summary stats SuperMix
Survey T Unordered multinomial VPC WinBUGS [reset]
1&2LevelMod
1LevelBlock
1LevelBlockcc
1LevelCatRef
1LevelComplex
1LevelFactorAnalysis
1LevelInteractions
1LevelMod
1LevelModAML
1LevelModcc
Name:
Description:
Close
Use
Current input string: {}
Set
Command: RunStatJR(template='Regre

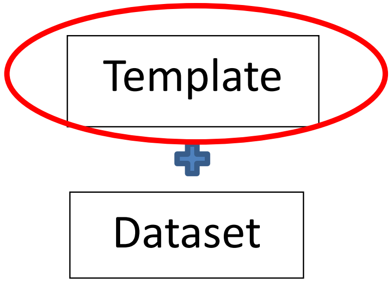

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Change template
1-Level 2-Level 3-Level Alternative MCMC methods aML Averages
Binomial CAR Categorical predictors Causal Censored Changepoint
Cluster analysis Complementary log-log Complex level 1 ConvergingC
Correlated classifications Correlation CustomC Data manipulation Diagnostics
eStat Factor analysis GenStat_model gretl_model Informative priors
Interactions JAGS Logit MATLAB_script MDS Measurement error
Minitab_model Missing data MIXREGLS Mixture MLwiN:point & click
MLwiN_IGLS MLwiN_MCMC MLwiN_script Model Multiple imputation
Multiple membership Multivariate response Negative binomial N-Level
Normal Octave_script OpenBUGS Ordered multinomial
Orthogonal parameterisation PCA Plots Poisson Population ecology
Predictions Probit PyScript Python_PyMC Python_script Quiz
R:comments R_CARBayes R_glm R_INLA R_lme4 R_MASS R_MCMCglmm
R_MCMCpack R_mgcv R_RStan R_script R_scriptMCMC Random slopes
Recapture Record linkage Reference category ROC SABRE SAS_model
Saving and Loading Selection Simulation Spatial SPSS_model SPSS_script
Standard deviation Stata_model Stata_script Summary stats SuperMix
Survey T Unordered multinomial VPC WinBUGS [reset]
Regression2
Name: Regression2
Description: Fits 1 level Normal multiple regression models in several packages.
Close
Use
Current input string: {}
Set
Command: RunStatJR(template='Regre

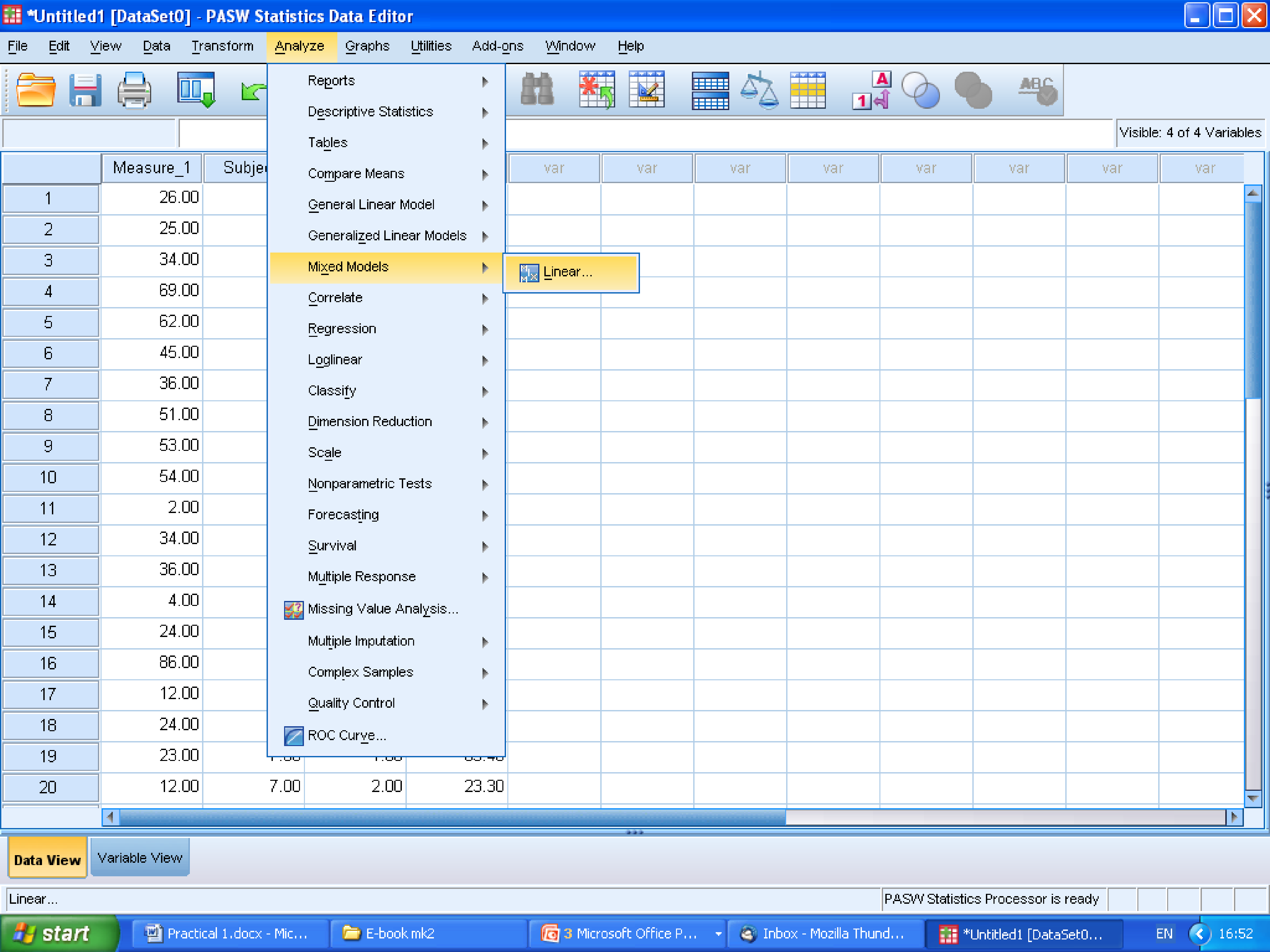

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Change template
1-Level 2-Level 3-Level Alternative MCMC methods aML Averages Binomial CAR Categorical predictors Causal Censored Changepoint Cluster analysis Complementary log-log Complex level 1 ConvergingC Correlated classifications Correlation CustomC Data manipulation Diagnostics eStat Factor analysis GenStat_model gretl_model Informative priors Interactions JAGS Logit MATLAB_script MDS Measurement error Minitab_model Missing data MIXREGLS Mixture MLwiN:point & click MLwiN_IGLS MLwiN_MCMC MLwiN_script Model Multiple imputation Multiple membership Multivariate response Negative binomial N-Level Normal Octave_script OpenBUGS Ordered multinomial Orthogonal parameterisation PCA Plots Poisson Population ecology Predictions Probit PyScript Python_PyMC Python_script Quiz R:comments R_CARBayes R_glm R_INLA R_lme4 R_MASS R_MCMCglmm R_MCMCpack R_mgcv R_RStan R_script R_scriptMCMC Random slopes Recapture Record linkage Reference category ROC SABRE SAS_model Saving and Loading Selection Simulation Spatial SPSS_model SPSS_script Standard deviation Stata_model Stata_script Summary stats SuperMix Survey T Unordered multinomial VPC WinBUGS [reset]
Regression2
Name: Regression2
Description: Fits 1 level Normal multiple regression models in several packages.
Close
Use
Current input string: {}
Set
Command: RunStatJR(template='Regre

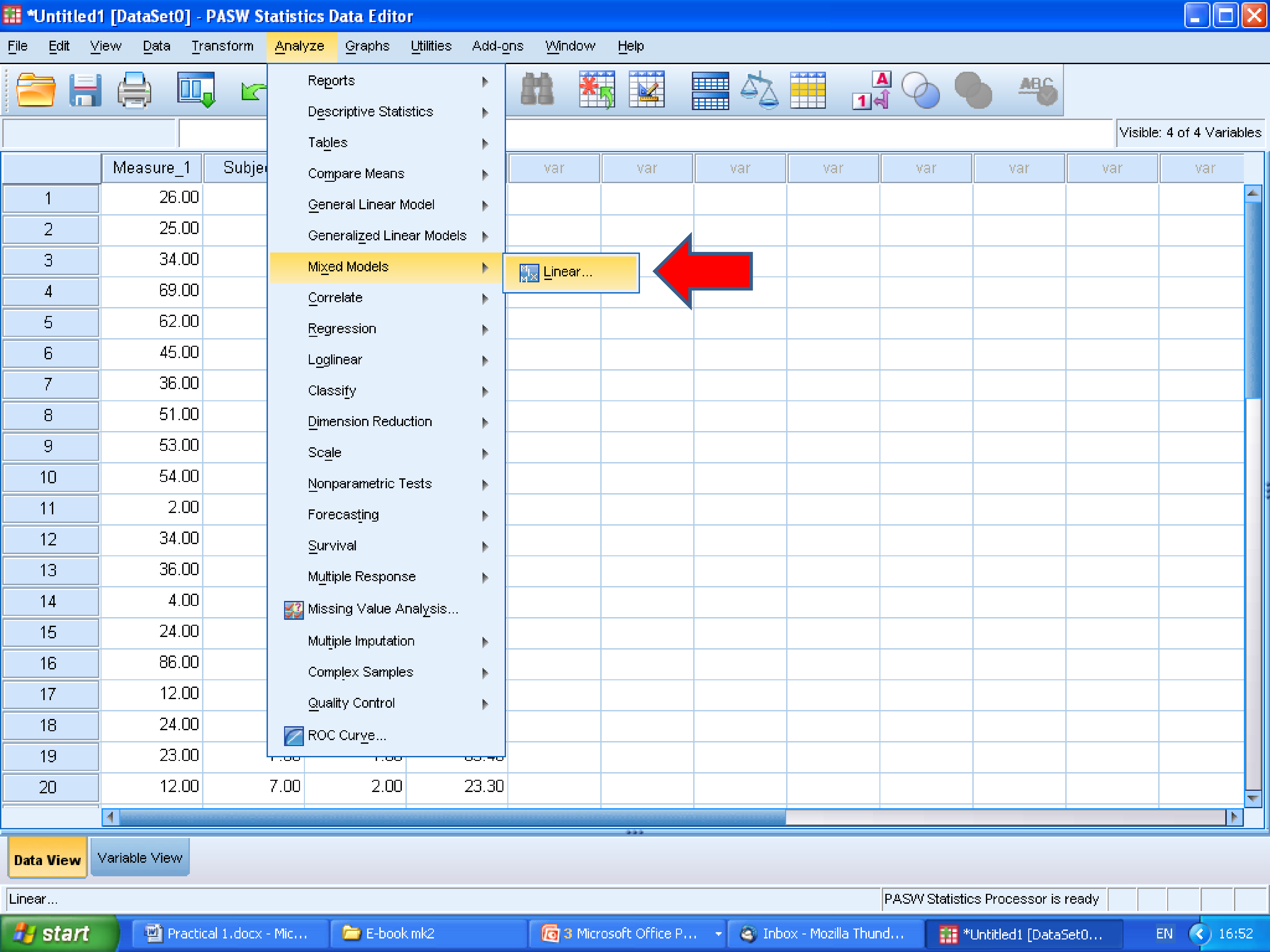

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Change template
1-Level 2-Level 3-Level Alternative MCMC methods aML Averages
Binomial CAR Categorical predictors Causal Censored Changepoint
Cluster analysis Complementary log-log Complex level 1 ConvergingC
Correlated classifications Correlation CustomC Data manipulation Diagnostics
eStat Factor analysis GenStat_model gretl_model Informative priors
Interactions JAGS Logit MATLAB_script MDS Measurement error
Minitab_model Missing data MIXREGLS Mixture MLwiN:point & click
MLwiN_IGLS MLwiN_MCMC MLwiN_script Model Multiple imputation
Multiple membership Multivariate response Negative binomial N-Level
Normal Octave_script OpenBUGS Ordered multinomial
Orthogonal parameterisation PCA Plots Poisson Population ecology
Predictions Probit PyScript Python_PyMC Python_script Quiz
R:comments R_CARBayes R_glm R_INLA R_lme4 R_MASS R_MCMCglmm
R_MCMCpack R_mgcv R_RStan R_script R_scriptMCMC Random slopes
Recapture Record linkage Reference category ROC SABRE SAS_model
Saving and Loading Selection Simulation Spatial SPSS_model SPSS_script
Standard deviation Stata_model Stata_script Summary stats SuperMix
Survey T Unordered multinomial VPC WinBUGS [reset]
Regression2
Name: Regression2
Description: Fits 1 level Normal multiple regression models in several packages.
Current input string: {}
Set
Command: RunStatJR(template='Regre
Close
Use

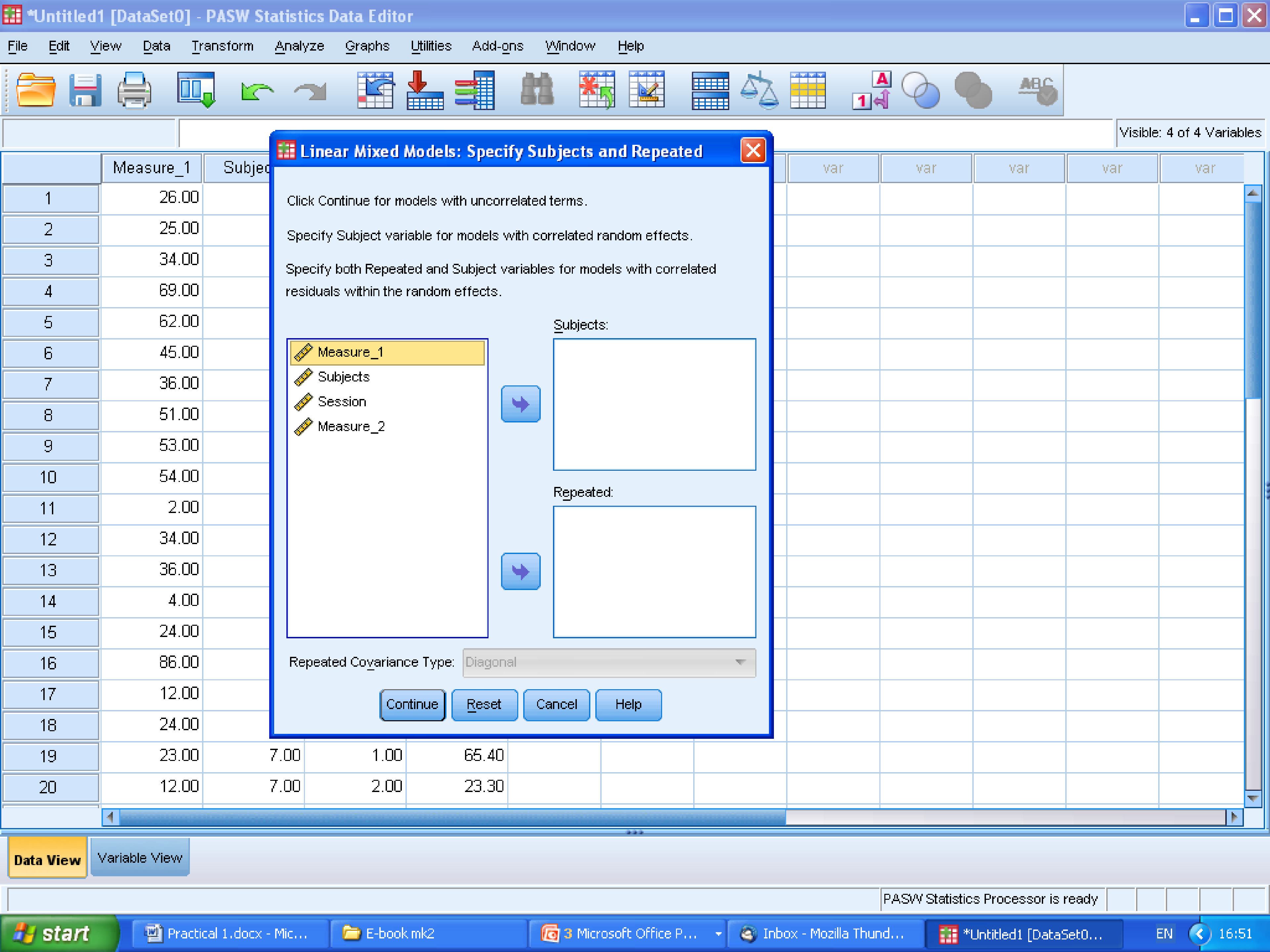


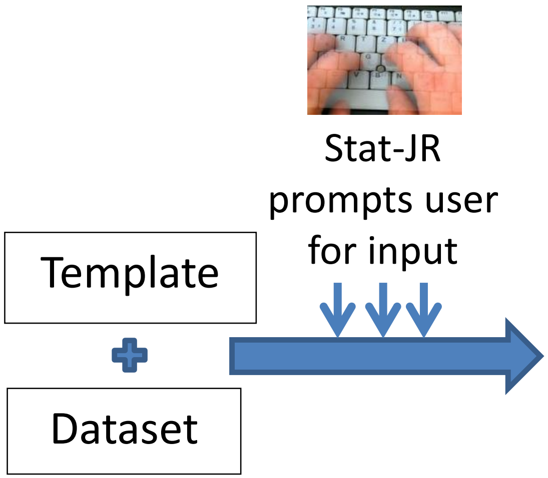
Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression1
Ready (1s)
Settings
Debug
Change template
1-Level 2-Level 3-Level Alternative MCMC methods aML Averages
Binomial CAR Categorical predictors Causal Censored Changepoint
Cluster analysis Complementary log-log Complex level 1 ConvergingC
Correlated classifications Correlation CustomC Data manipulation Diagnostics
eStat Factor analysis GenStat_model gretl_model Informative priors
Interactions JAGS Logit MATLAB_script MDS Measurement error
Minitab_model Missing data MIXREGLS Mixture MLwiN:point & click
MLwiN_IGLS MLwiN_MCMC MLwiN_script Model Multiple imputation
Multiple membership Multivariate response Negative binomial N-Level
Normal Octave_script OpenBUGS Ordered multinomial
Orthogonal parameterisation PCA Plots Poisson Population ecology
Predictions Probit PyScript Python_PyMC Python_script Quiz
R:comments R_CARBayes R_glm R_INLA R_lme4 R_MASS R_MCMCglmm
R_MCMCpack R_mgcv R_RStan R_script R_scriptMCMC Random slopes
Recapture Record linkage Reference category ROC SABRE SAS_model
Saving and Loading Selection Simulation Spatial SPSS_model SPSS_script
Standard deviation Stata_model Stata_script Summary stats SuperMix
Survey T Unordered multinomial VPC WinBUGS [reset]
Regression2
Name: Regression2
Description: Fits 1 level Normal multiple regression models in several packages.
Close
Use
Current input string: {}
Set
Command: RunStatJR(template='Regre


Template
Dataset


*Untitled1 [DataSet0] - PASW Statistics Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Reports
Descriptive Statistics
Tables
Compare Means
General Linear Model
Generalized Linear Models
Mixed Models
Linear...
Correlate
Regression
Loglinear
Classify
Dimension Reduction
Scale
Nonparametric Tests
Forecasting
Survival
Multiple Response
Missing Value Analysis...
Multiple Imputation
Complex Samples
Quality Control
ROC Curve...
Visible: 4 of 4 Variables
Measure_1
Subje
var
1 26.00
2 25.00
3 34.00
4 69.00
5 62.00
6 45.00
7 36.00
8 51.00
9 53.00
10 54.00
11 2.00
12 34.00
13 36.00
14 4.00
15 24.00
16 86.00
17 12.00
18 24.00
19 23.00
20 12.00 7.00 2.00 23.30
Data View
Variable View
Linear...
PASW Statistics Processor is ready
start
Practical 1.docx - Mic...
E-book mk2
3 Microsoft Office P...
Inbox - Mozilla Thund...
*Untitled1 [DataSet0...
EN
16:52


*Untitled1 [DataSet0] - PASW Statistics Data Editor
File
Edit
View
Data
Transform
Analyze
Graphs
Utilities
Add-ons
Window
Help
Reports
Descriptive Statistics
Tables
Compare Means
General Linear Model
Generalized Linear Models
Mixed Models
Linear...
Correlate
Regression
Loglinear
Classify
Dimension Reduction
Scale
Nonparametric Tests
Forecasting
Survival
Multiple Response
Missing Value Analysis...
Multiple Imputation
Complex Samples
Quality Control
ROC Curve...
Visible: 4 of 4 Variables
Measure_1
Subje
var
Data View
Variable View
Linear...
PASW Statistics Processor is ready
start
Practical 1.docx - Mic...
E-book mk2
3 Microsoft Office P...
Inbox - Mozilla Thund...
*Untitled1 [DataSet0...
EN
16:52


*Untitled1 [DataSet0] - PASW Statistics Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Visible: 4 of 4 Variables
Measure_1
Subjec
var var var var var
1 26.00
2 25.00
3 34.00
4 69.00
5 62.00
6 45.00
7 36.00
8 51.00
9 53.00
10 54.00
11 2.00
12 34.00
13 36.00
14 4.00
15 24.00
16 86.00
17 12.00
18 24.00
19 23.00 7.00 1.00 65.40
20 12.00 7.00 2.00 23.30
Linear Mixed Models: Specify Subjects and Repeated
Click Continue for models with uncorrelated terms.
Specify Subject variable for models with correlated random effects.
Specify both Repeated and Subject variables for models with correlated residuals within the random effects.
Measure_1
Subjects
Session
Measure_2
Subjects:
Repeated:
Repeated Covariance Type: Diagonal
Continue Reset Cancel Help
Data View
Variable View
PASW Statistics Processor is ready
start
Practical 1.docx - Mic...
E-book mk2
3 Microsoft Office P...
Inbox - Mozilla Thund...
*Untitled1 [DataSet0...
EN
16:51


Stat-JR
prompts user
for input
Template
Dataset

Stat-JR:TREE  Start again  Dataset  tutorial  Template  Regression2  Ready (1s)  Settings  Debug

**Response:**

**Explanatory variables:**

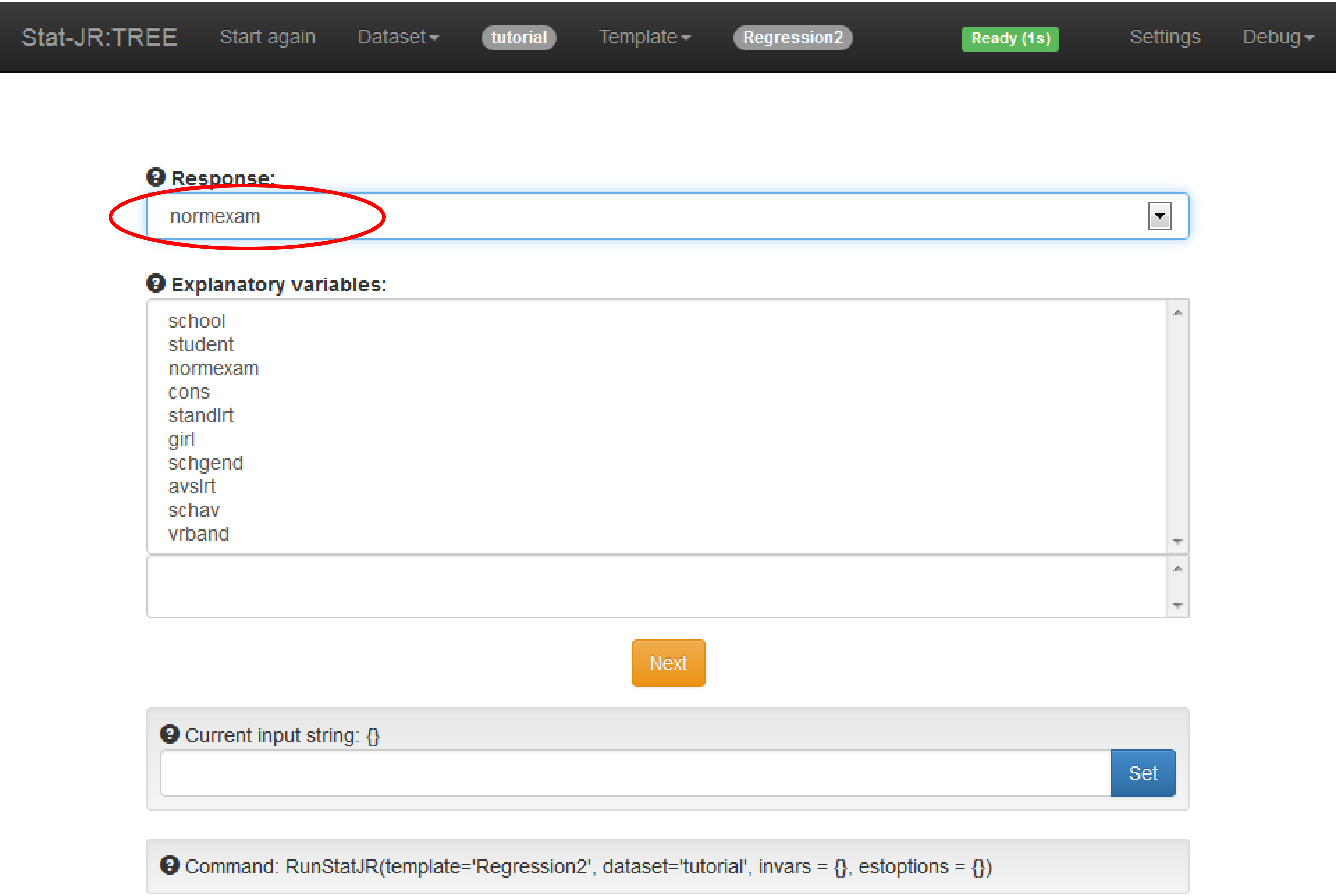
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband

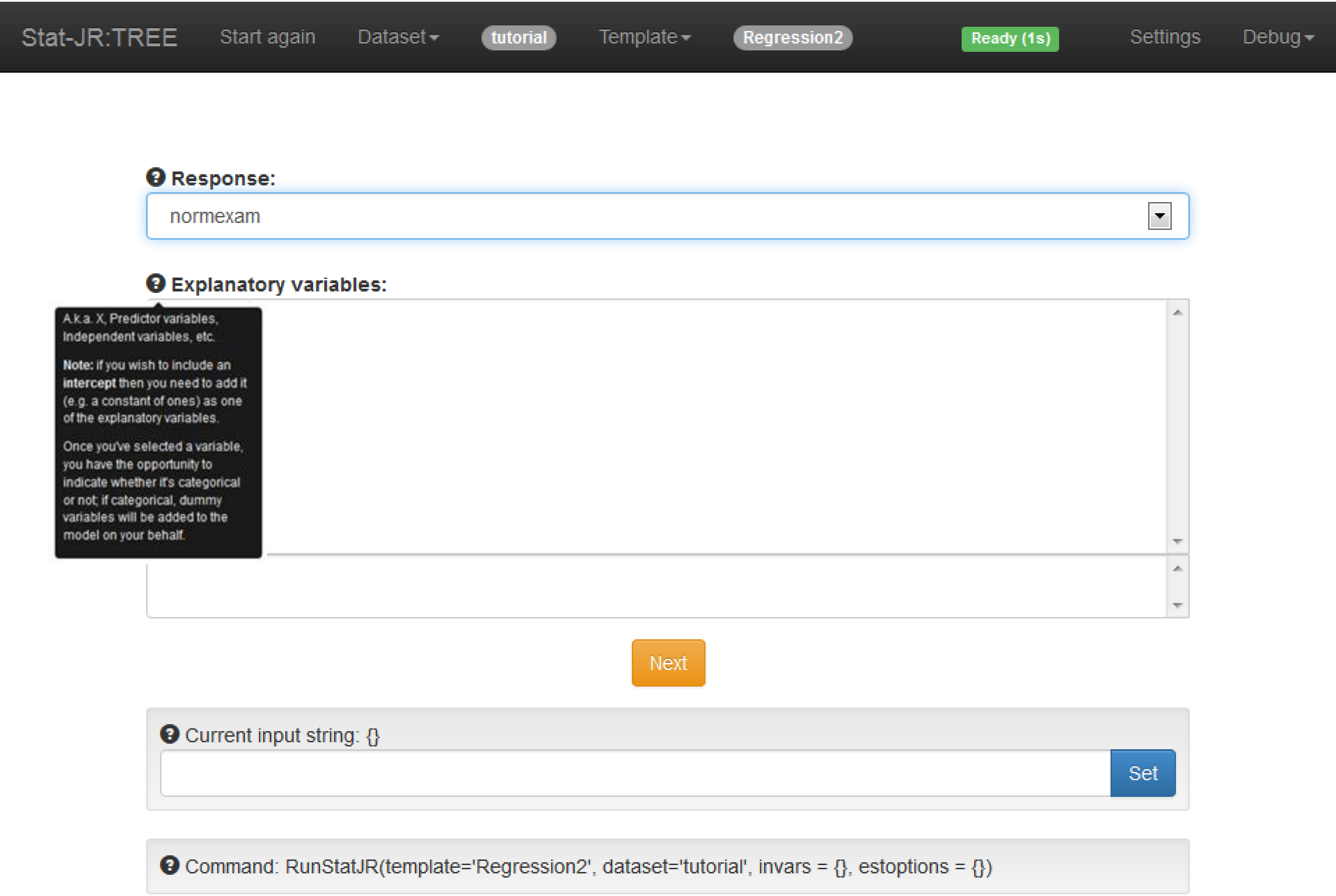
Next

Current input string: {}

Set

Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {}, estoptions = {})


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam
Explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
Next
Current input string: {}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {}, estoptions = {})



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam
Explanatory variables:
A.k.a. X, Predictor variables, Independent variables, etc.
Note: if you wish to include an intercept then you need to add it (e.g. a constant of ones) as one of the explanatory variables.
Once you've selected a variable, you have the opportunity to indicate whether it's categorical or not; if categorical, dummy variables will be added to the model on your behalf.
Next
Current input string: {}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {}, estoptions = {})


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
eStat
WinBUGS
OpenBUGS
MLwiN_MCMC
MLwiN_IGLS
R_MCMCglmm
R_glm
Stata_model
Python_PyMC
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {})

Stat-JR:TREE   Start again   Dataset   tutorial   Template   Regression2   Ready (1s)   Settings   Debug

**Response:**

normexam remove

**Explanatory variables:**

cons,standlrt remove

**Choose estimation engine:**

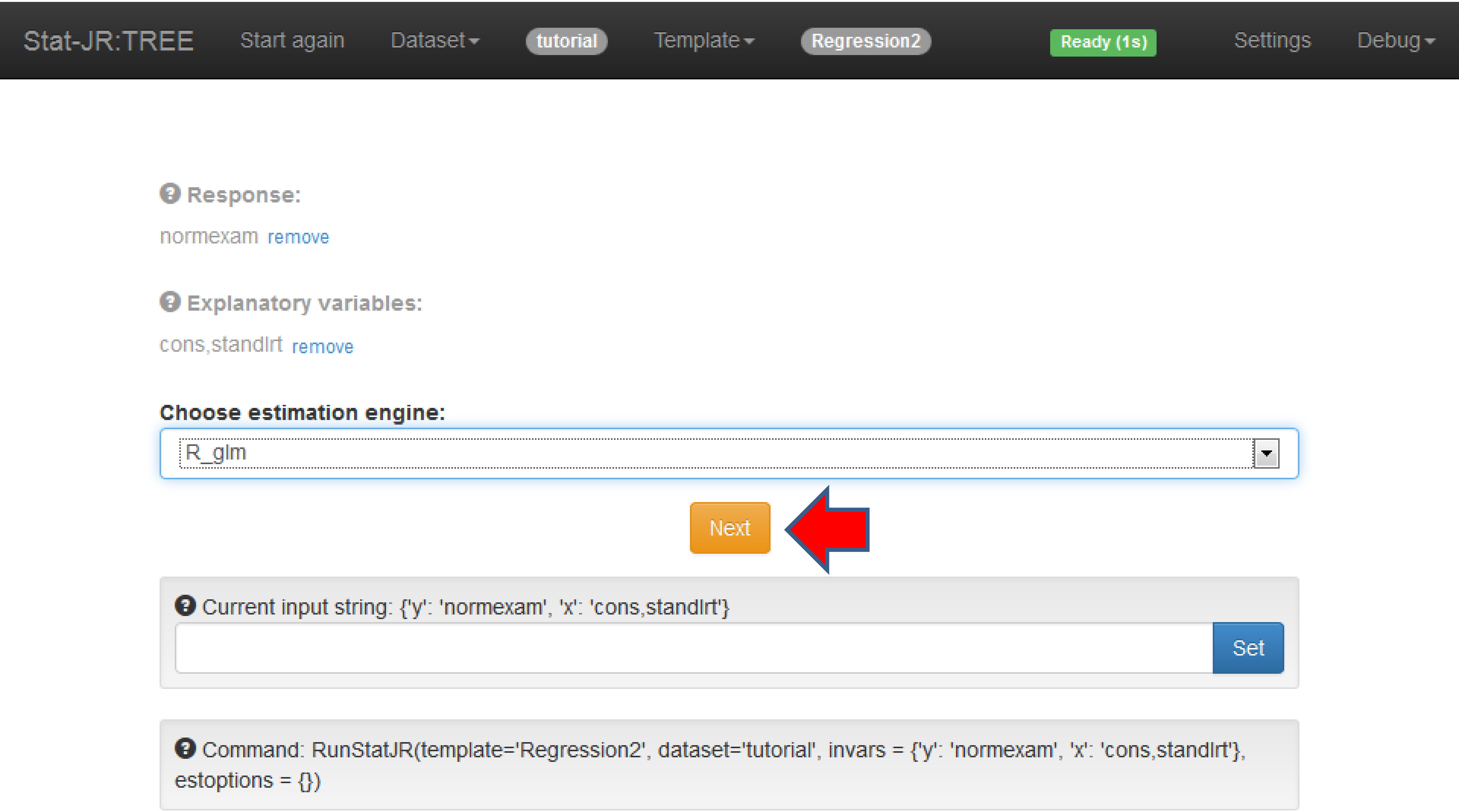
R_glm

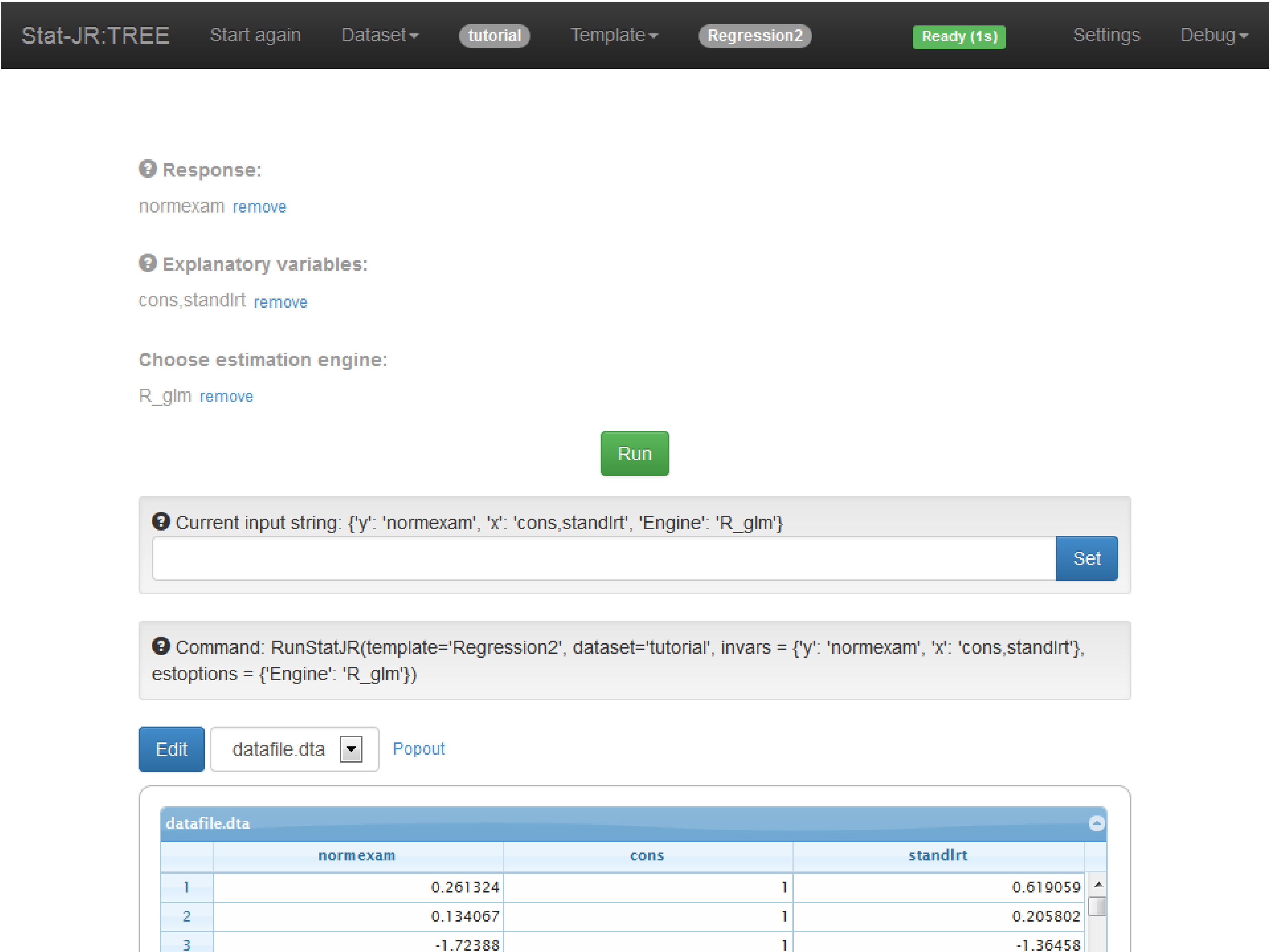
Next

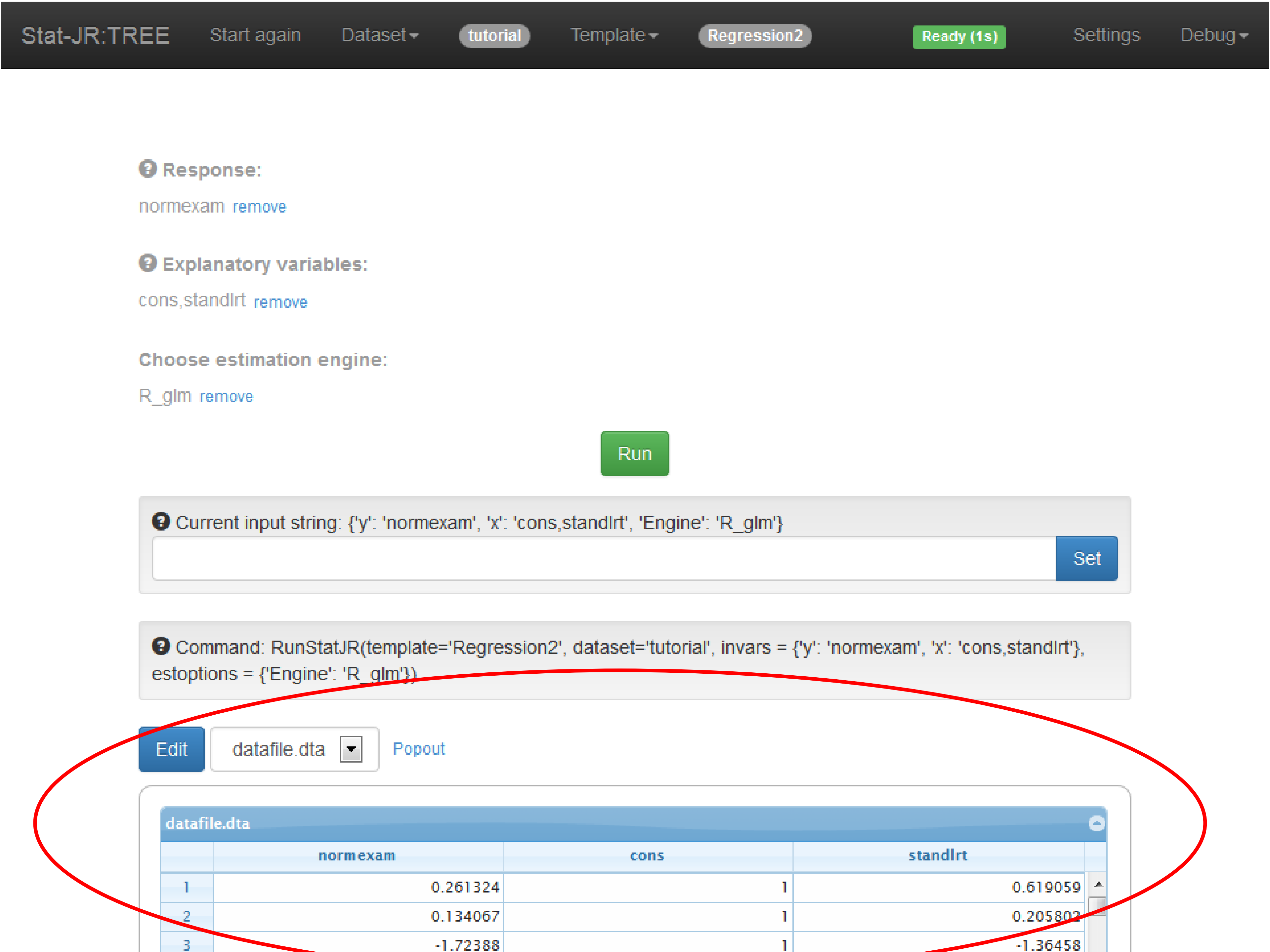
Current input string: {'y': 'normexam', 'x': 'cons,standlrt'}

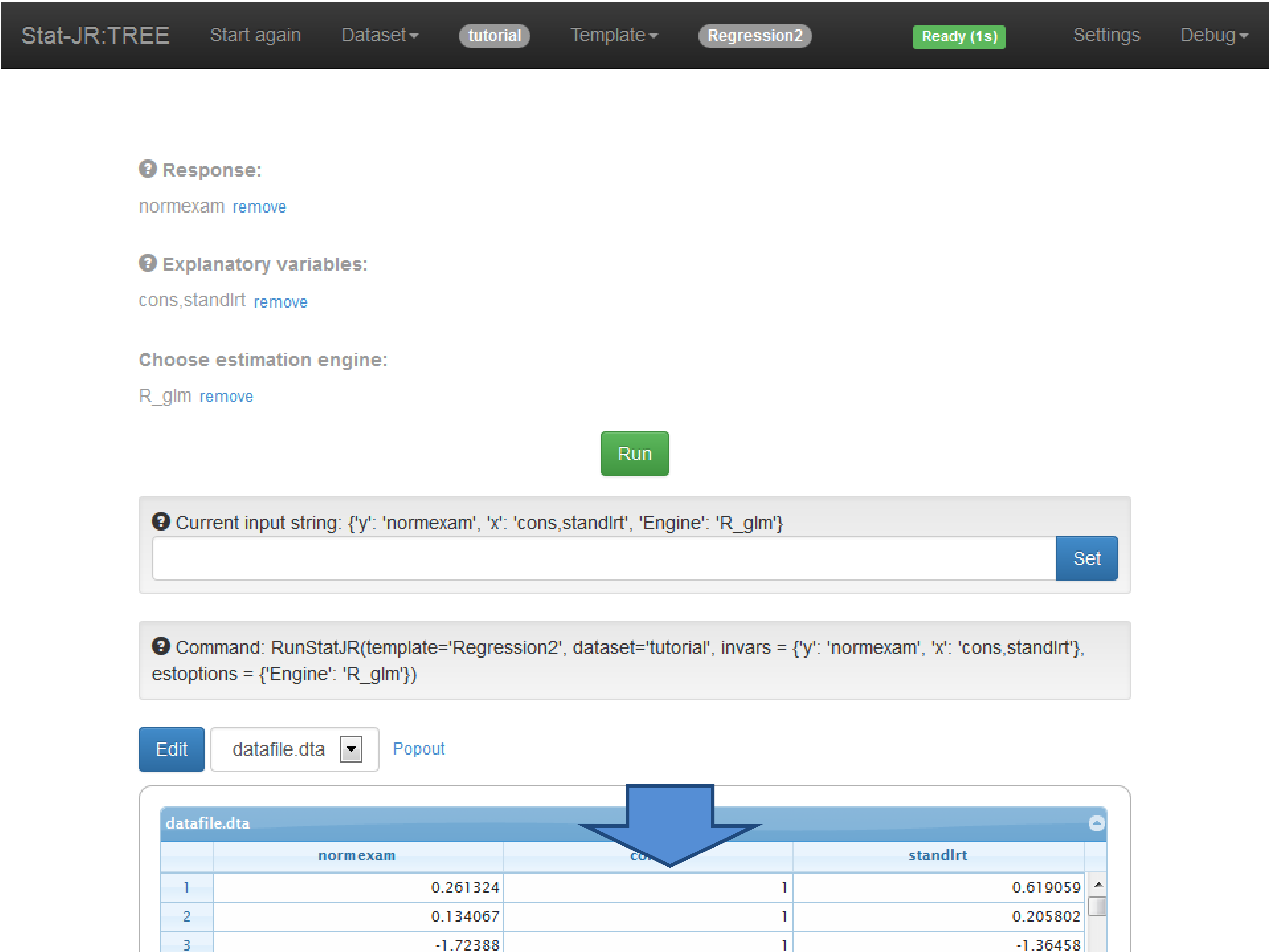
Set

Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {})


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm
Next
Current input string: {'y': 'normexam', 'x': 'cons,standlrt'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {})


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
datafile.dta
Popout
datafile.dta
normexam
cons
standlrt
1
0.261324
1
0.619059
2
0.134067
1
0.205802
3
-1.72388
1
-1.36458


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
datafile.dta
Popout
datafile.dta
normexam
cons
standlrt
1
0.261324
1
0.619059
2
0.134067
1
0.205802
3
-1.72388
1
-1.36458


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
datafile.dta
Popout
datafile.dta
normexam
standlrt
1 0.261324 1 0.619059
2 0.134067 1 0.205802
3 -1.72388 1 -1.36458

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (1s) Settings Debug

Edit datafile.dta Popout

datafile.dta

| | normexam | cons | standlrt |
|---|---|---|---|
| 1 | 0.261324 | 1 | 0.619059 |
| 2 | 0.134067 | 1 | 0.205802 |
| 3 | -1.72388 | 1 | -1.36458 |
| 4 | 0.967586 | 1 | 0.205802 |
| 5 | 0.544341 | 1 | 0.371105 |
| 6 | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

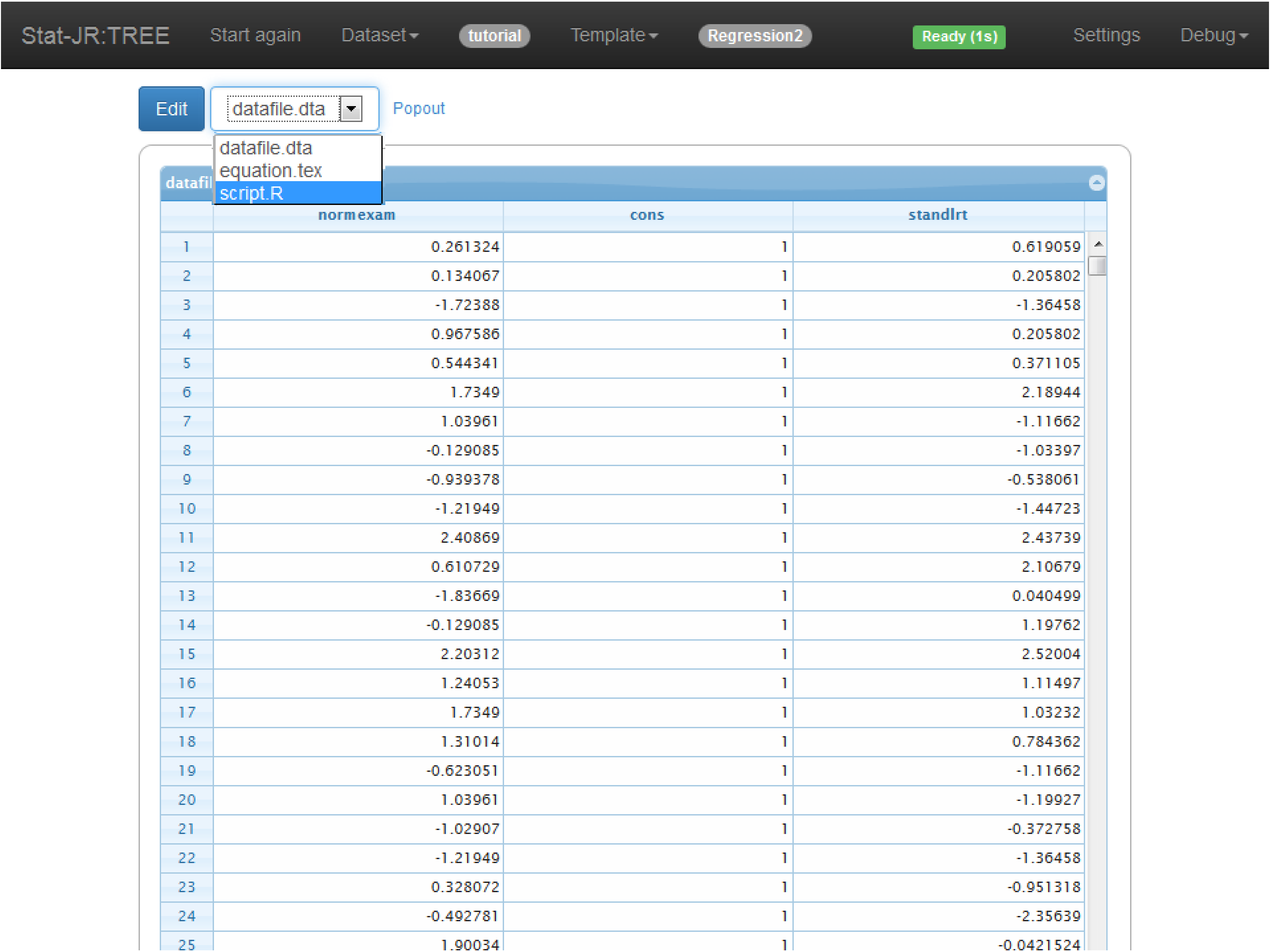
Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (1s) Settings Debug

Edit datafile.dta Popout

datafile.dta
equation.tex
script.R

datafil

| | normexam | cons | standlrt |
|---|---|---|---|
| 1 | 0.261324 | 1 | 0.619059 |
| 2 | 0.134067 | 1 | 0.205802 |
| 3 | -1.72388 | 1 | -1.36458 |
| 4 | 0.967586 | 1 | 0.205802 |
| 5 | 0.544341 | 1 | 0.371105 |
| 6 | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

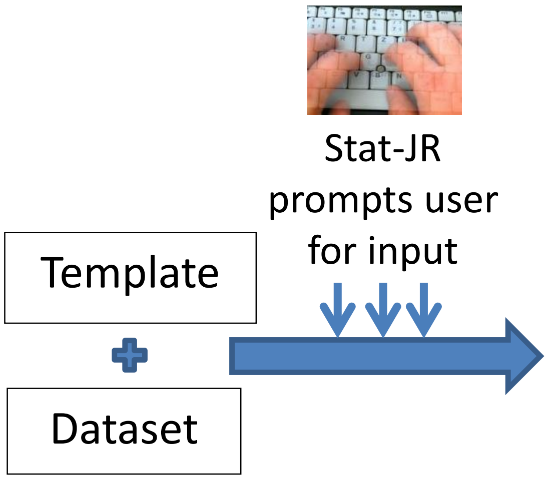


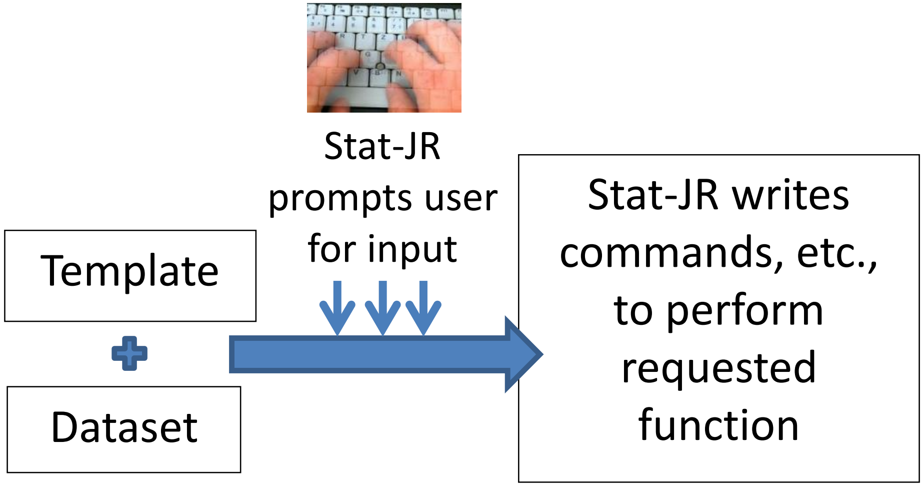

| | normexam | cons | standlrt |
|---|---|---|---|
| 1 | 0.261324 | 1 | 0.619059 |
| 2 | 0.134067 | 1 | 0.205802 |
| 3 | -1.72388 | 1 | -1.36458 |
| 4 | 0.967586 | 1 | 0.205802 |
| 5 | 0.544341 | 1 | 0.371105 |
| 6 | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

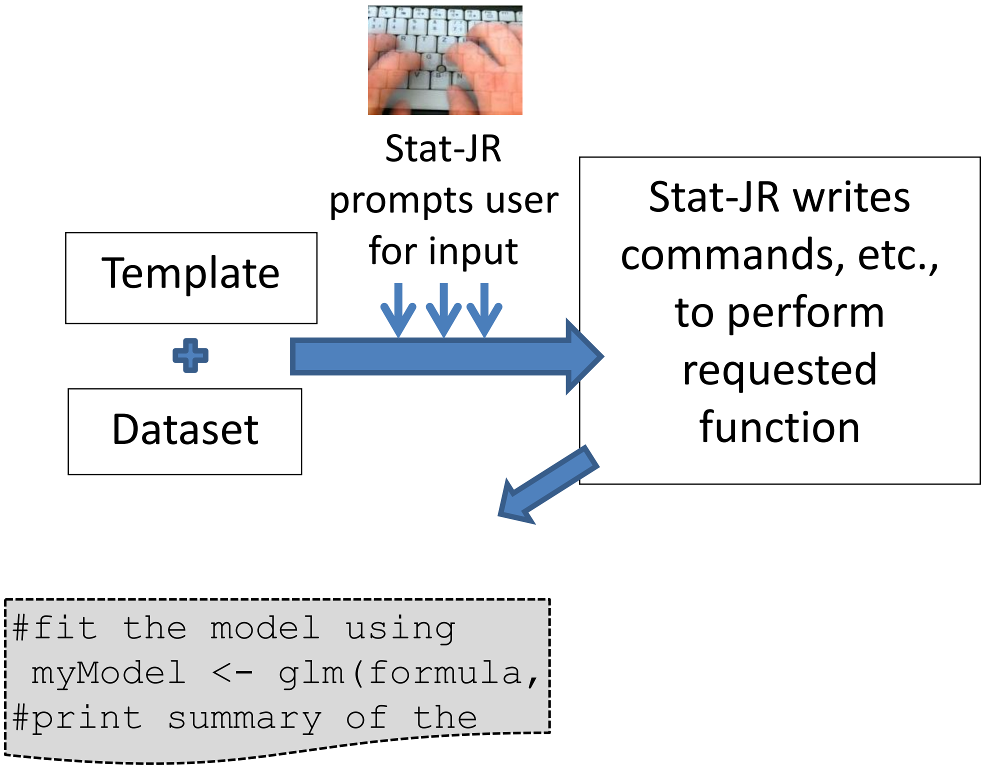
Stat-JR:TREE   Start again   Dataset   tutorial   Template   Regression2   Ready (1s)   Settings   Debug

Edit   script.R   Popout

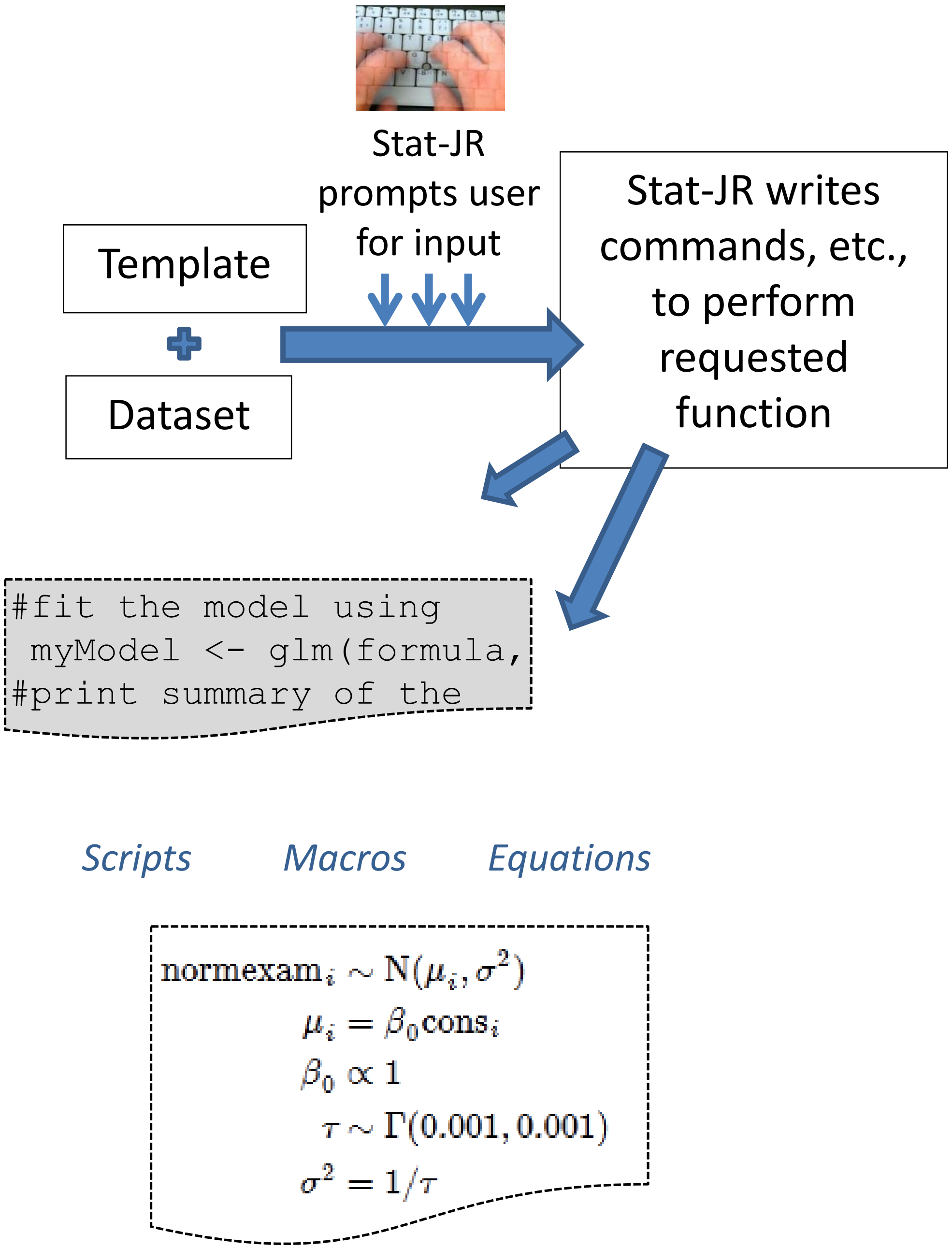
```
local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Note that when Stat-JR interoperates with R, it sets the working
# directory to wherever the user's temporary files are stored, i.e.
# workdir = tempdir(). The data to be modelled, this script, and the
# files exported from R, are all saved there.
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# test to see if foreign package is already installed, if not, then install it
if (!require(foreign)) {
    install.packages("foreign")
    library(foreign)
}

# read *.dta file (Stata format) into R data frame (requires foreign):
mydata<-read.dta("datafile.dta")
# print summary of the data
summary(mydata)

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Below we specify the model formula, formatted as y ~ x1 + x2 + ...
# Since Stat-JR assumes users have included the intercept in their list
# of explanatory variables, -1 removes the intercept which the glm
# function otherwise adds by default.
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

formula <- normexam ~ cons + standlrt - 1
# fit the model using the glm function, specifying the formula, data, and distribution (with identi
ty link) in its arguments
myModel <- glm(formula, data = mydata, family = gaussian(identity))
```

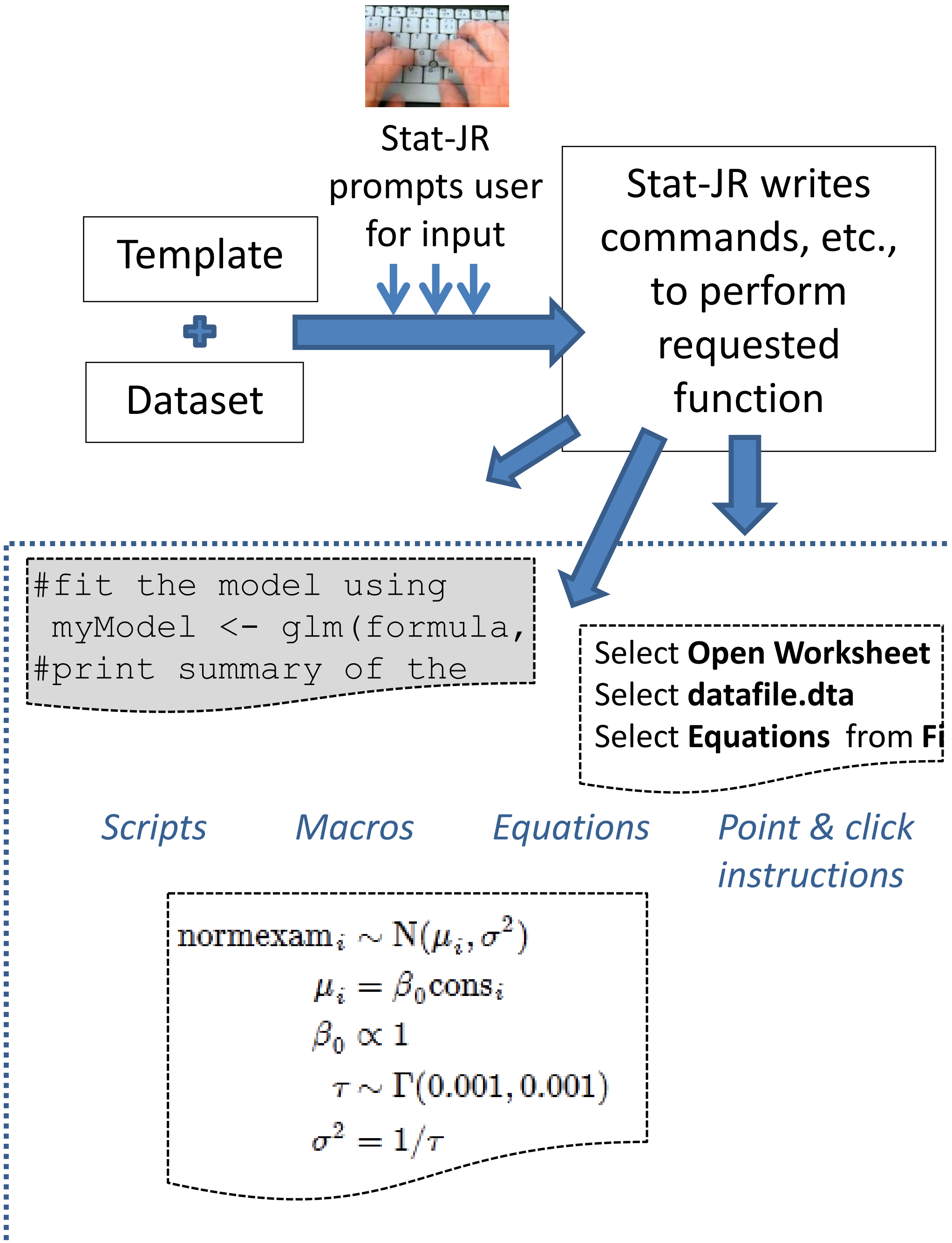

Stat-JR
prompts user
for input
Template
Dataset

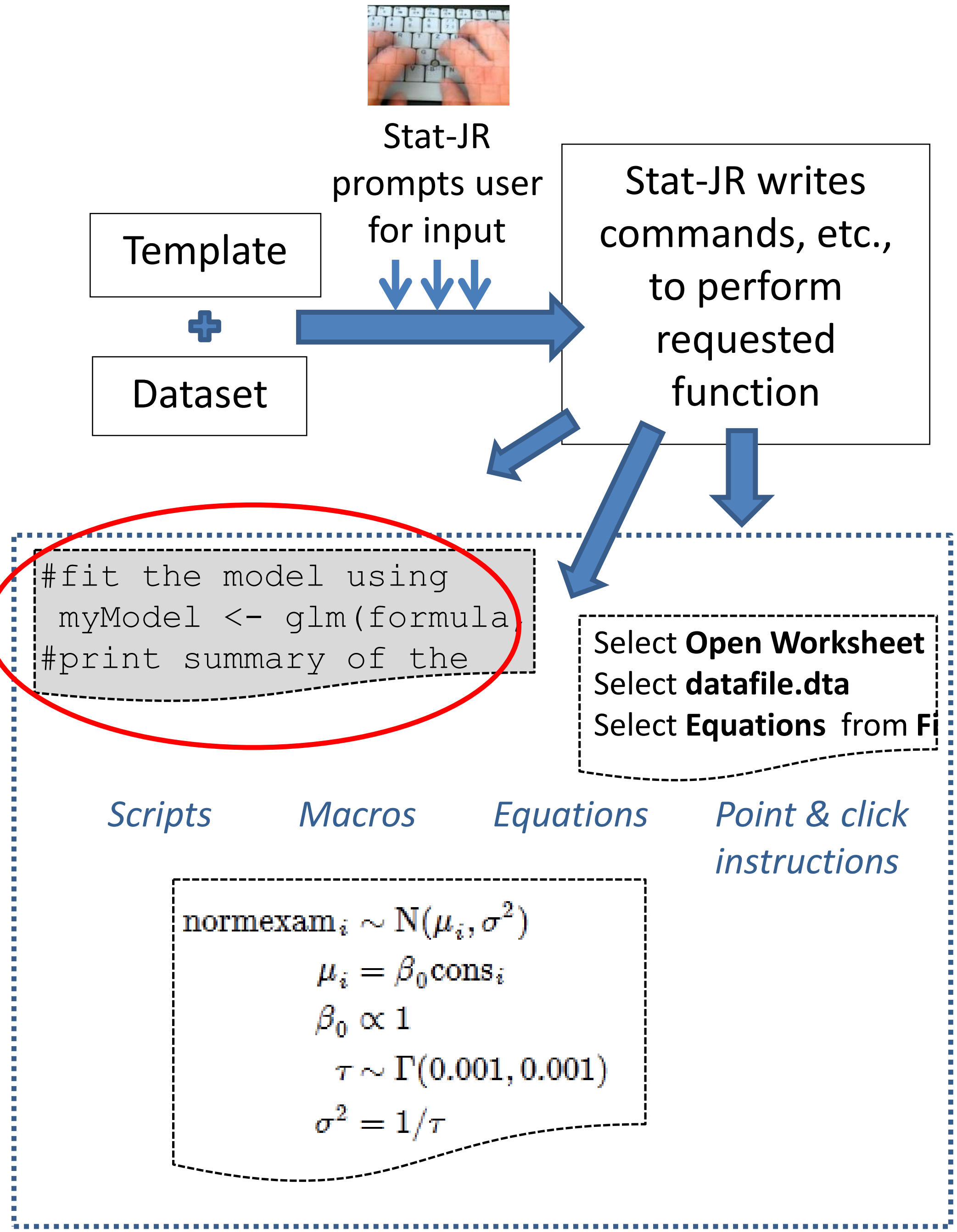

Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function

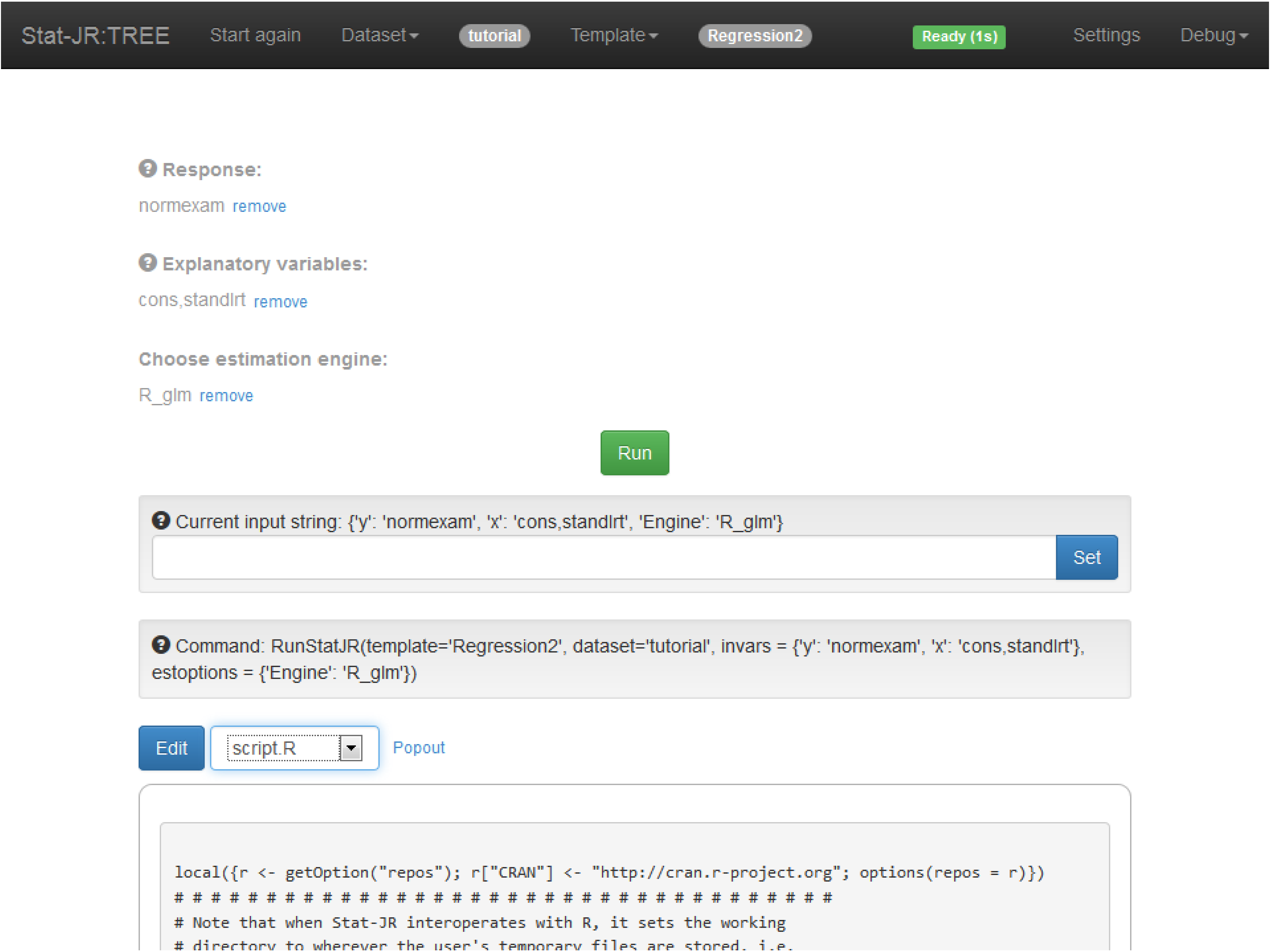


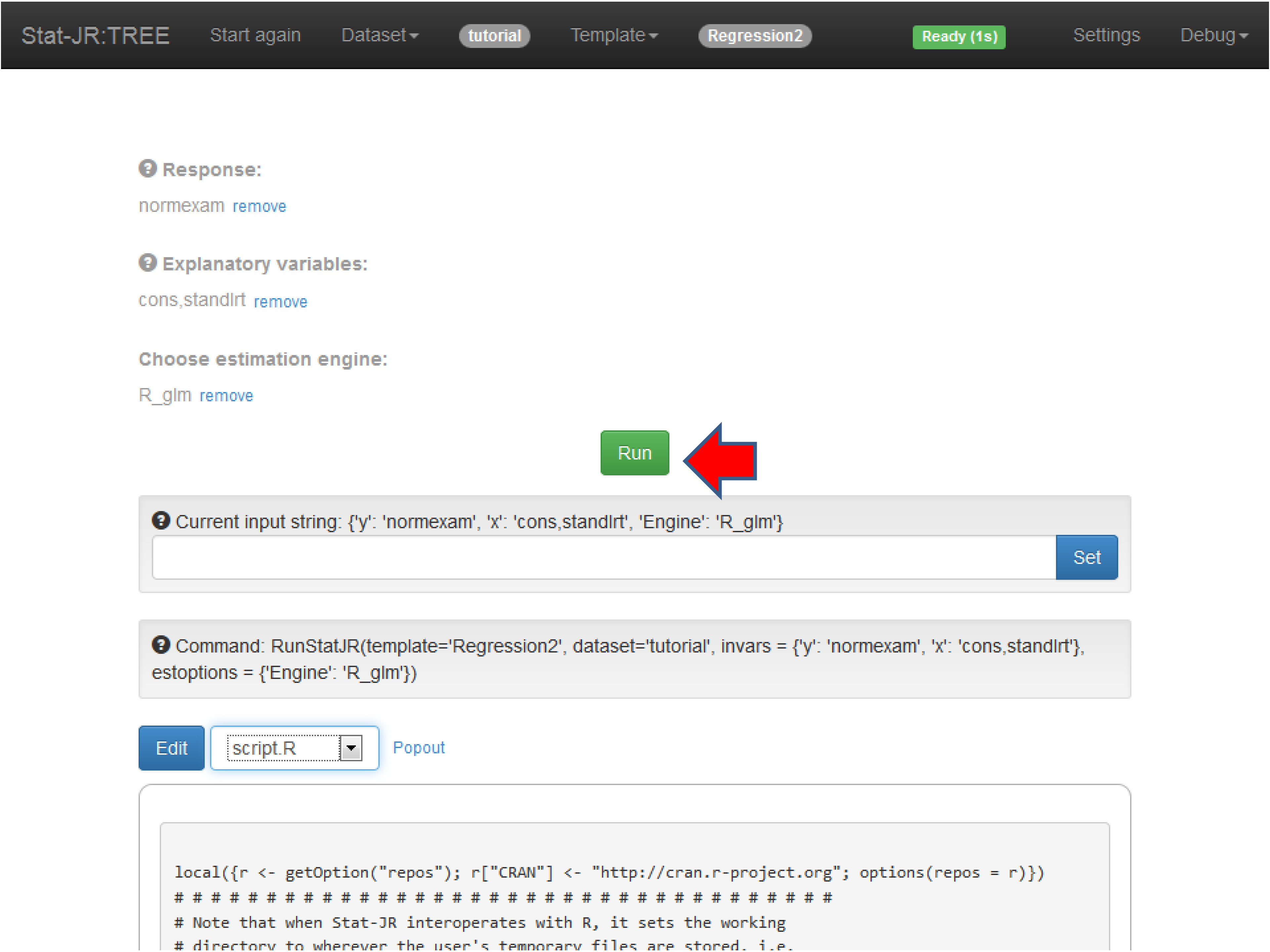

*Scripts* *Macros*

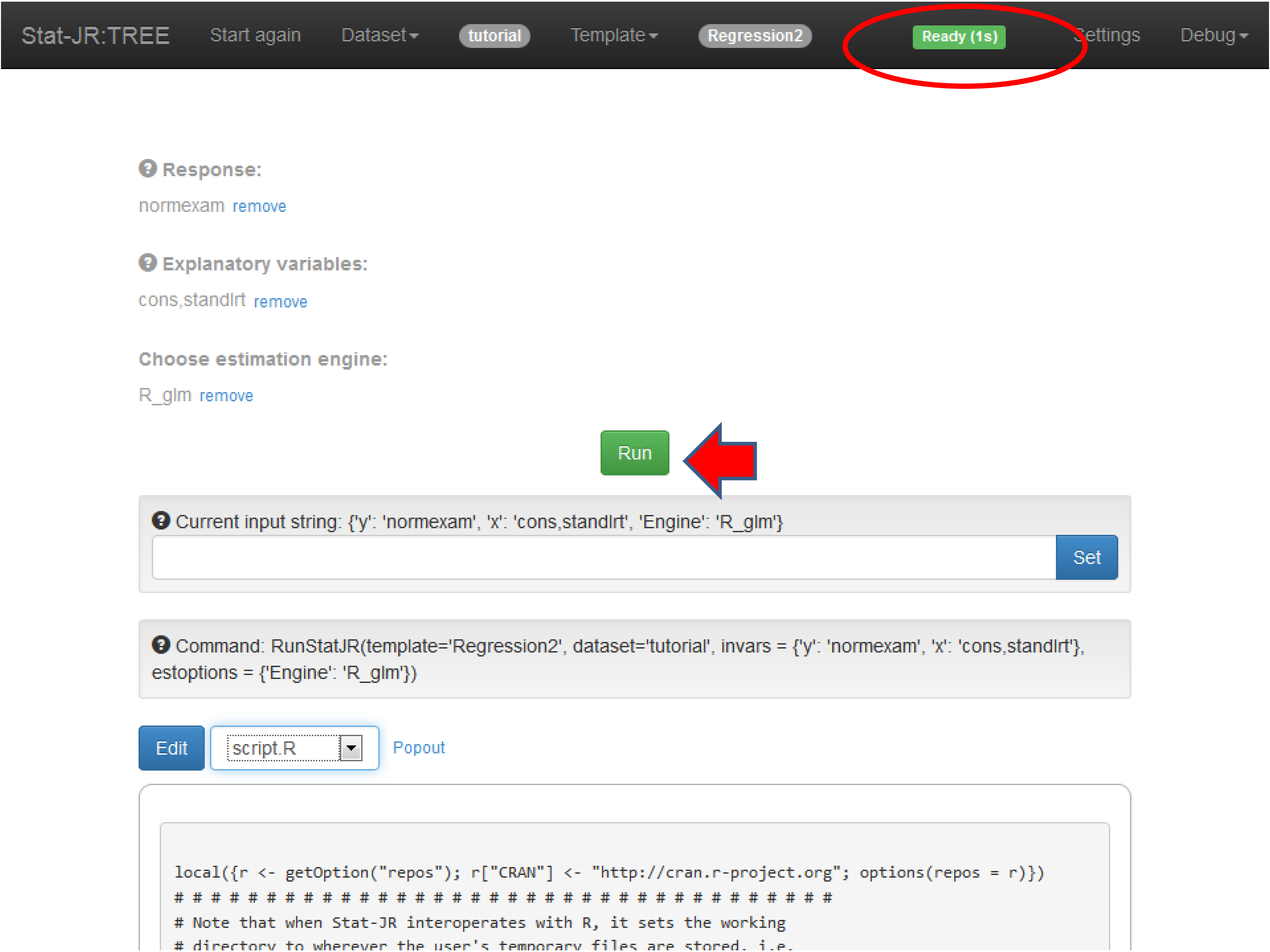

Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
#fit the model using
myModel <- glm(formula,
#print summary of the
Scripts
Macros
Equations
$\mathrm{normexam}_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$


Template
Dataset
Stat-JR prompts user for input
Stat-JR writes commands, etc., to perform requested function
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$

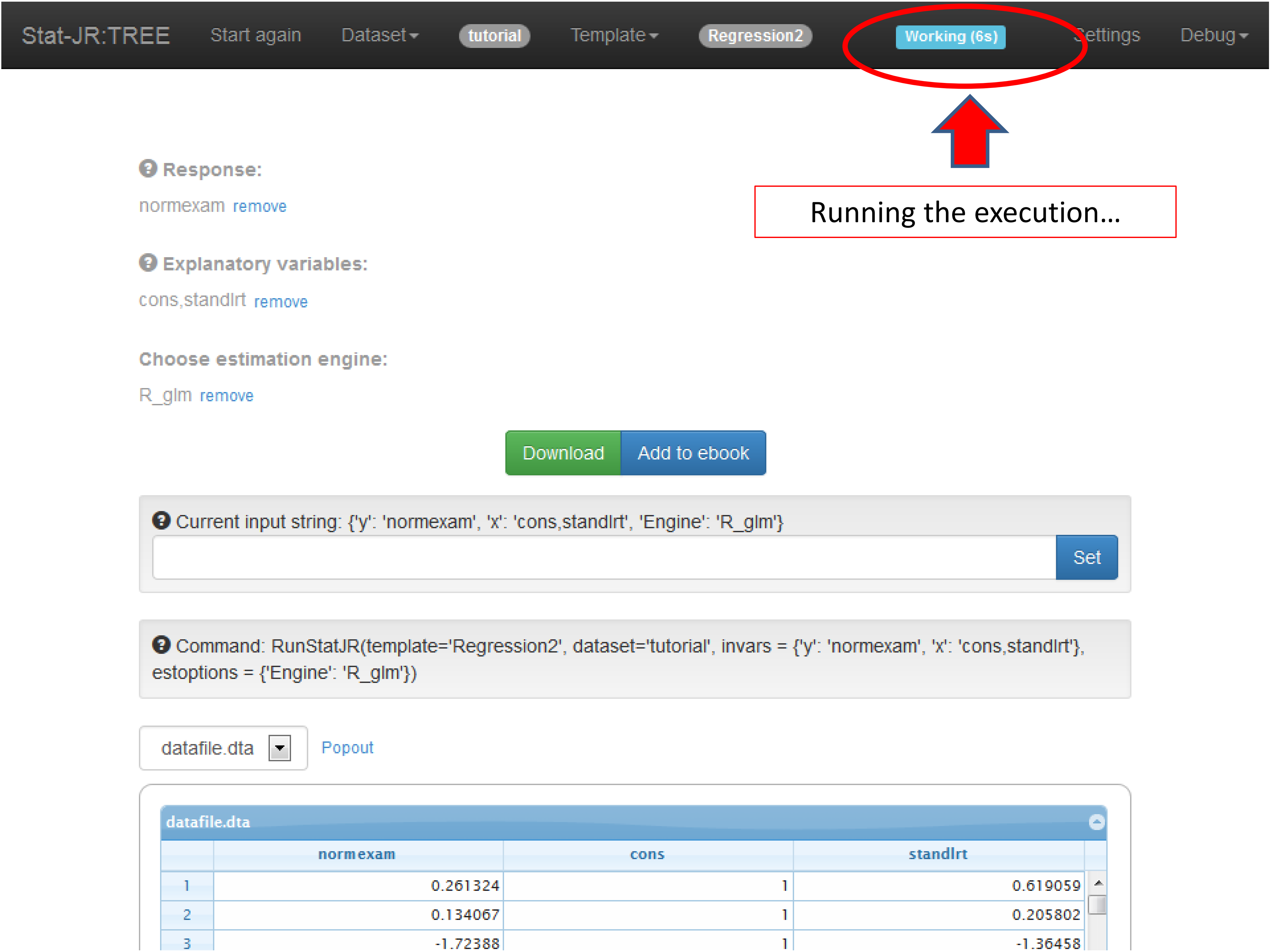

Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
#fit the model using
myModel <- glm(formula
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
script.R
Popout
local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Note that when Stat-JR interoperates with R, it sets the working

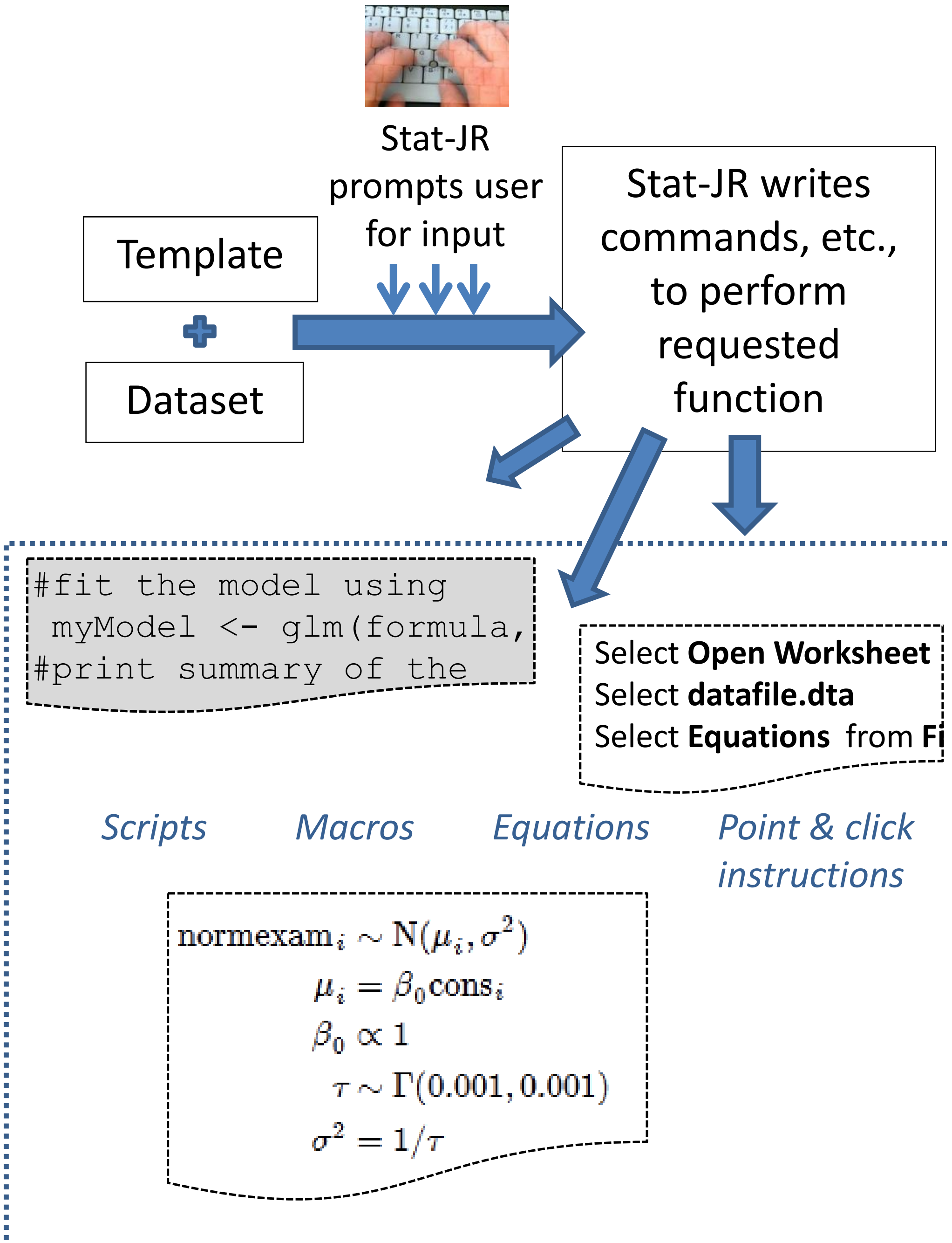

Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
script.R
Popout
local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Note that when Stat-JR interoperates with R, it sets the working

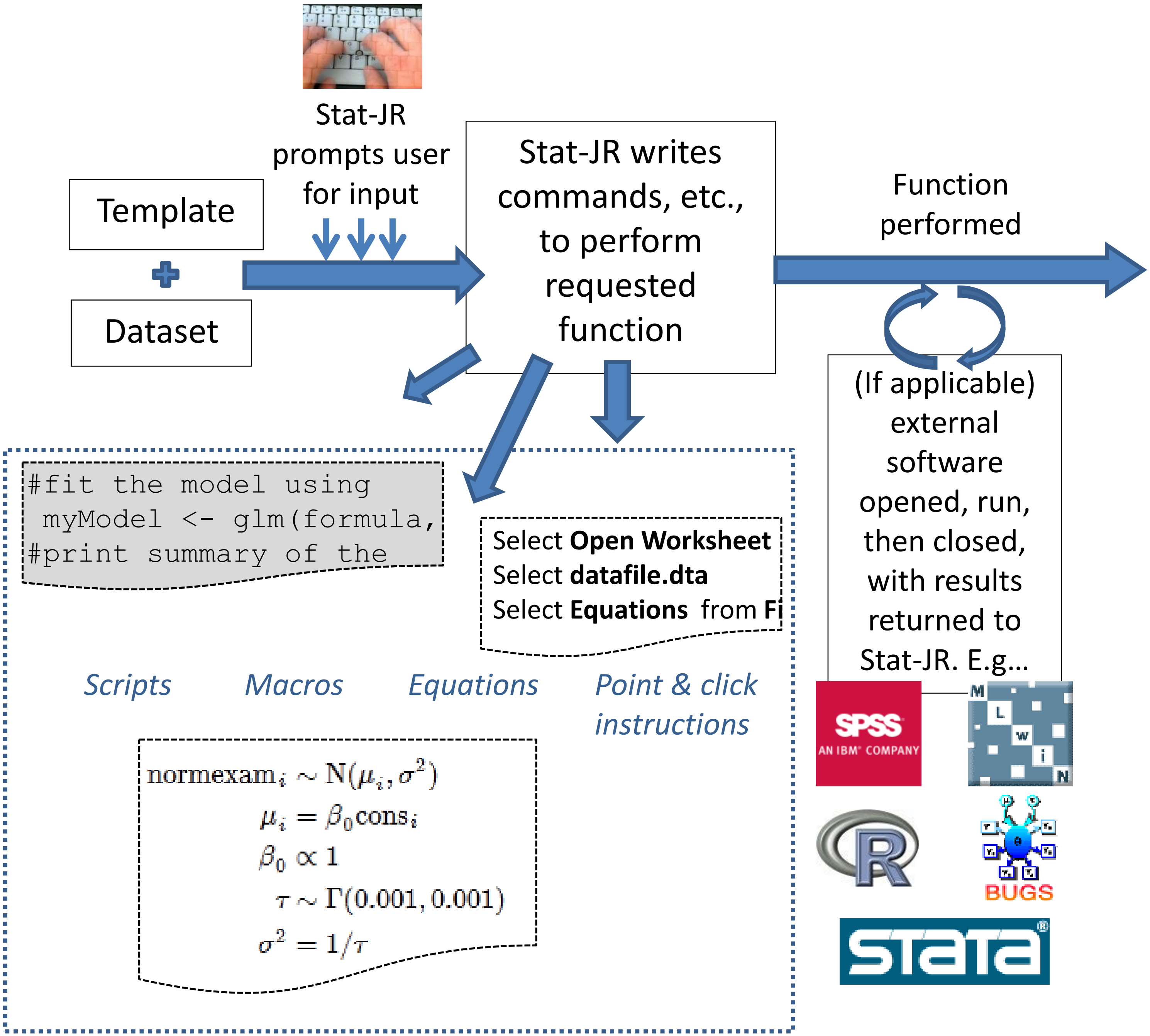


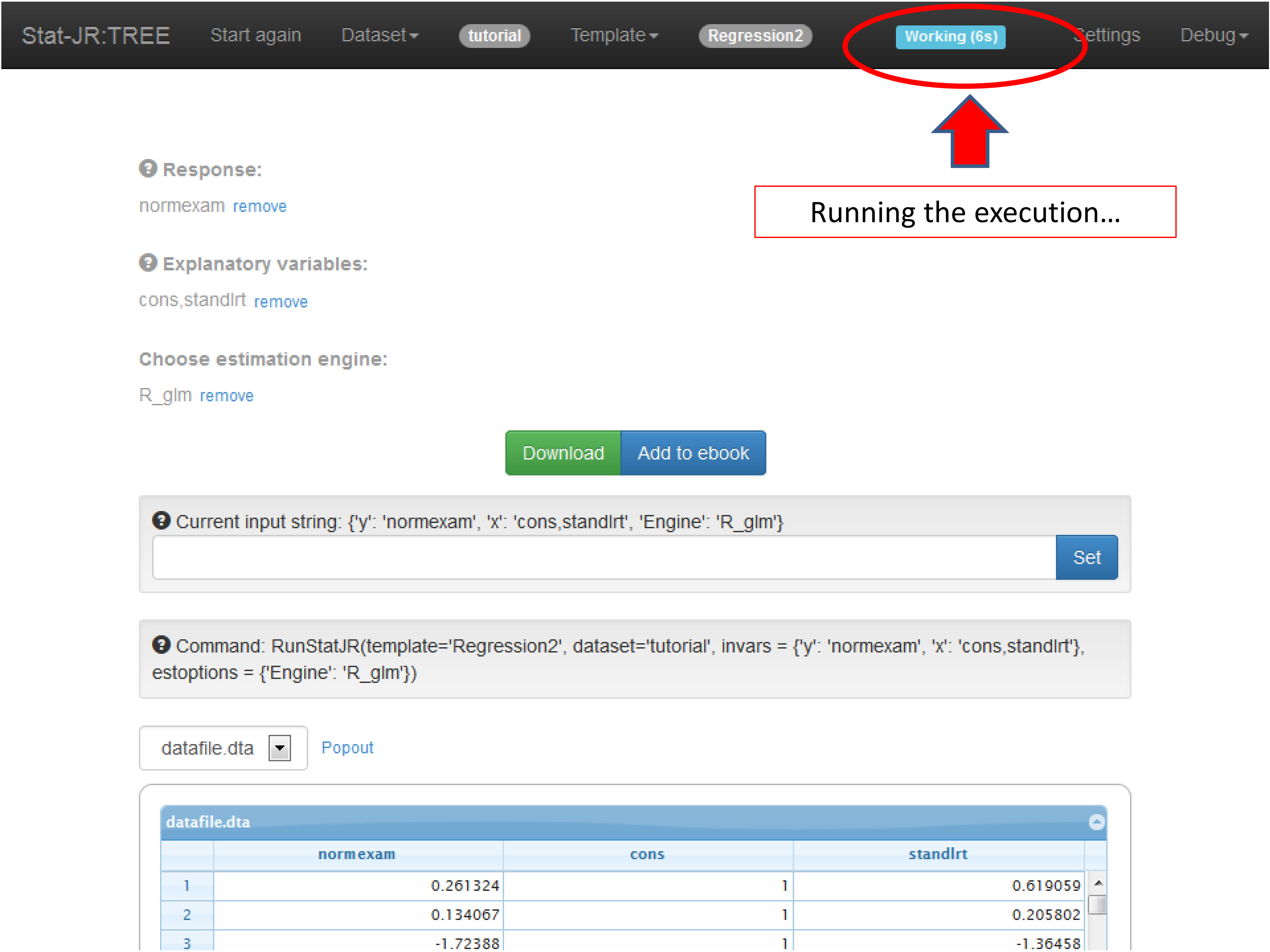
Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (1s)
Settings
Debug
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Run
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
Edit
script.R
Popout
local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Note that when Stat-JR interoperates with R, it sets the working
# directory to wherever the user's temporary files are stored, i.e.



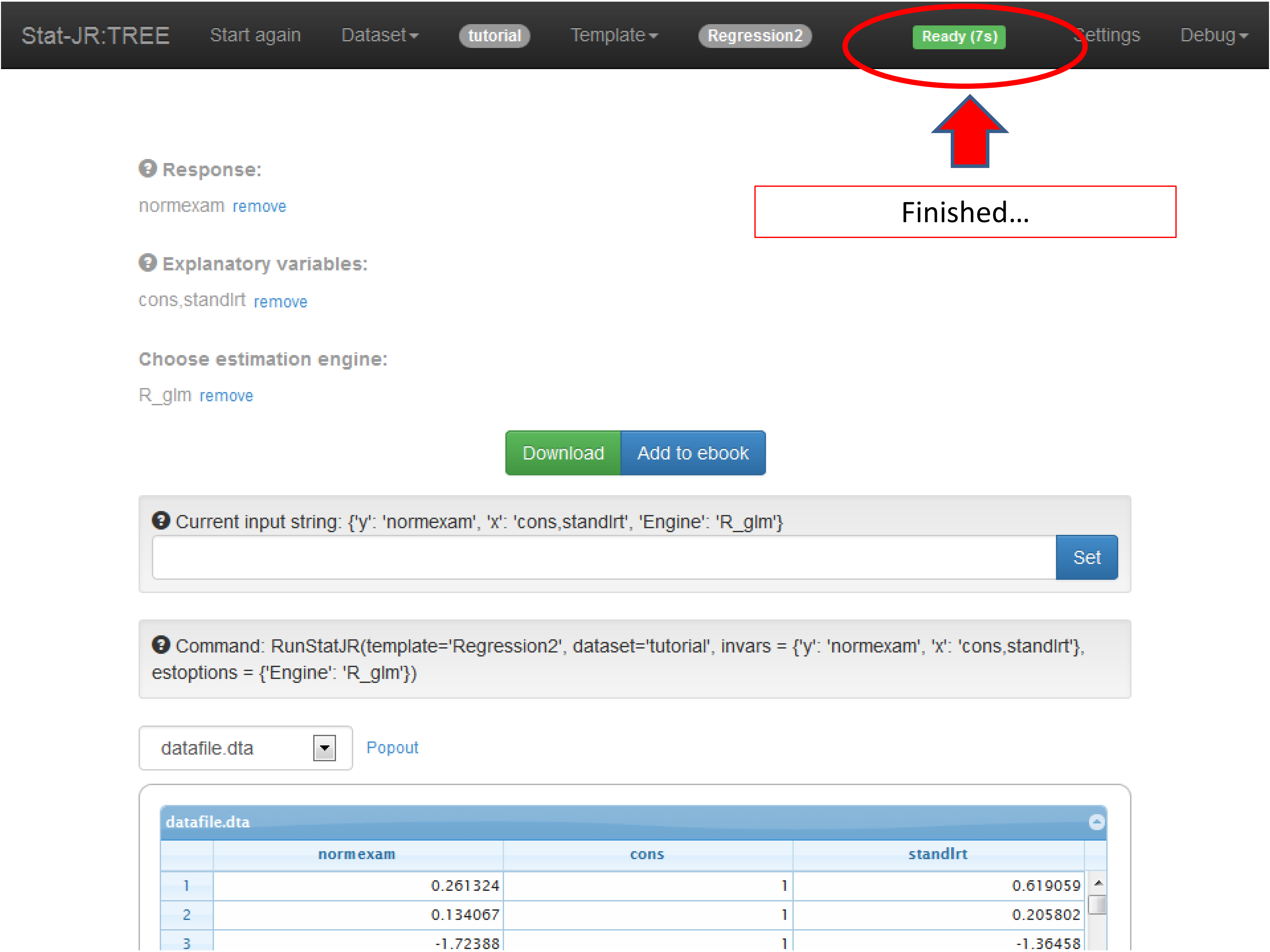
Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Working (6s)
Settings
Debug
Running the execution…
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Download
Add to ebook
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
datafile.dta
Popout
datafile.dta
normexam
cons
standlrt
1
0.261324
1
0.619059
2
0.134067
1
0.205802
3
-1.72388
1
-1.36458


Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam_i ~ N(μ_i, σ^2)
μ_i = β_0 cons_i
β_0 ∝ 1
τ ~ Γ(0.001, 0.001)
σ^2 = 1/τ

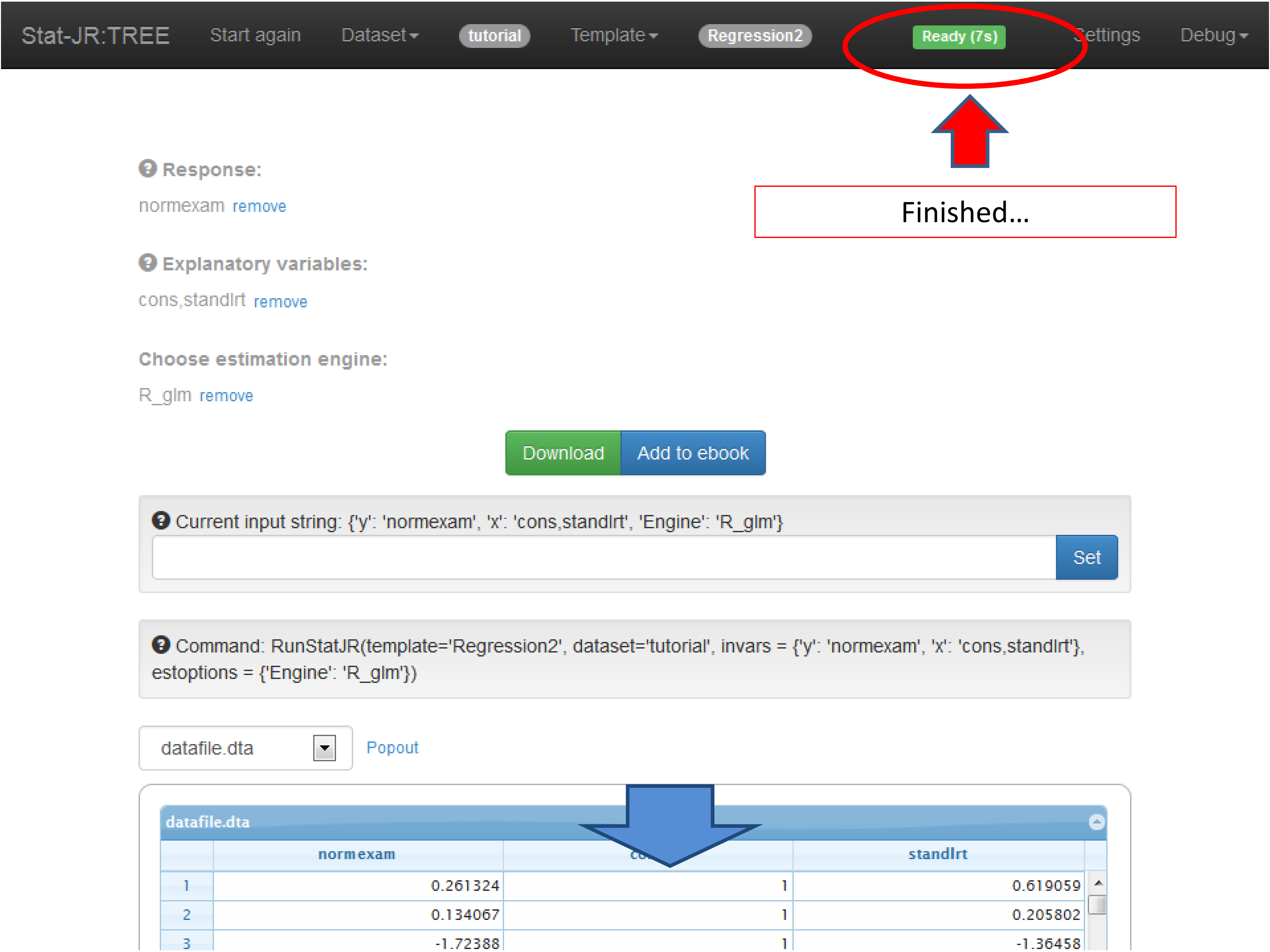

Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g...
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim$ N$(\mu_i, \sigma^2)$
$\mu_i = \beta_0$cons$_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
SPSS AN IBM COMPANY
MLwiN
R
BUGS
STATA



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Working (6s)
Settings
Debug
Running the execution…
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Download
Add to ebook
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
datafile.dta
Popout
datafile.dta
normexam
cons
standlrt
1 0.261324 1 0.619059
2 0.134067 1 0.205802
3 -1.72388 1 -1.36458



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (7s)
Settings
Debug
Finished...
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Download
Add to ebook
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
datafile.dta
Popout
datafile.dta
normexam
cons
standlrt
1 0.261324 1 0.619059
2 0.134067 1 0.205802
3 -1.72388 1 -1.36458



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (7s)
Settings
Debug
Finished...
Response:
normexam remove
Explanatory variables:
cons,standlrt remove
Choose estimation engine:
R_glm remove
Download
Add to ebook
Current input string: {'y': 'normexam', 'x': 'cons,standlrt', 'Engine': 'R_glm'}
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
datafile.dta
Popout
datafile.dta
normexam
standlrt
1 0.261324 1 0.619059
2 0.134067 1 0.205802
3 -1.72388 1 -1.36458

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

datafile.dta Popout

datafile.dta

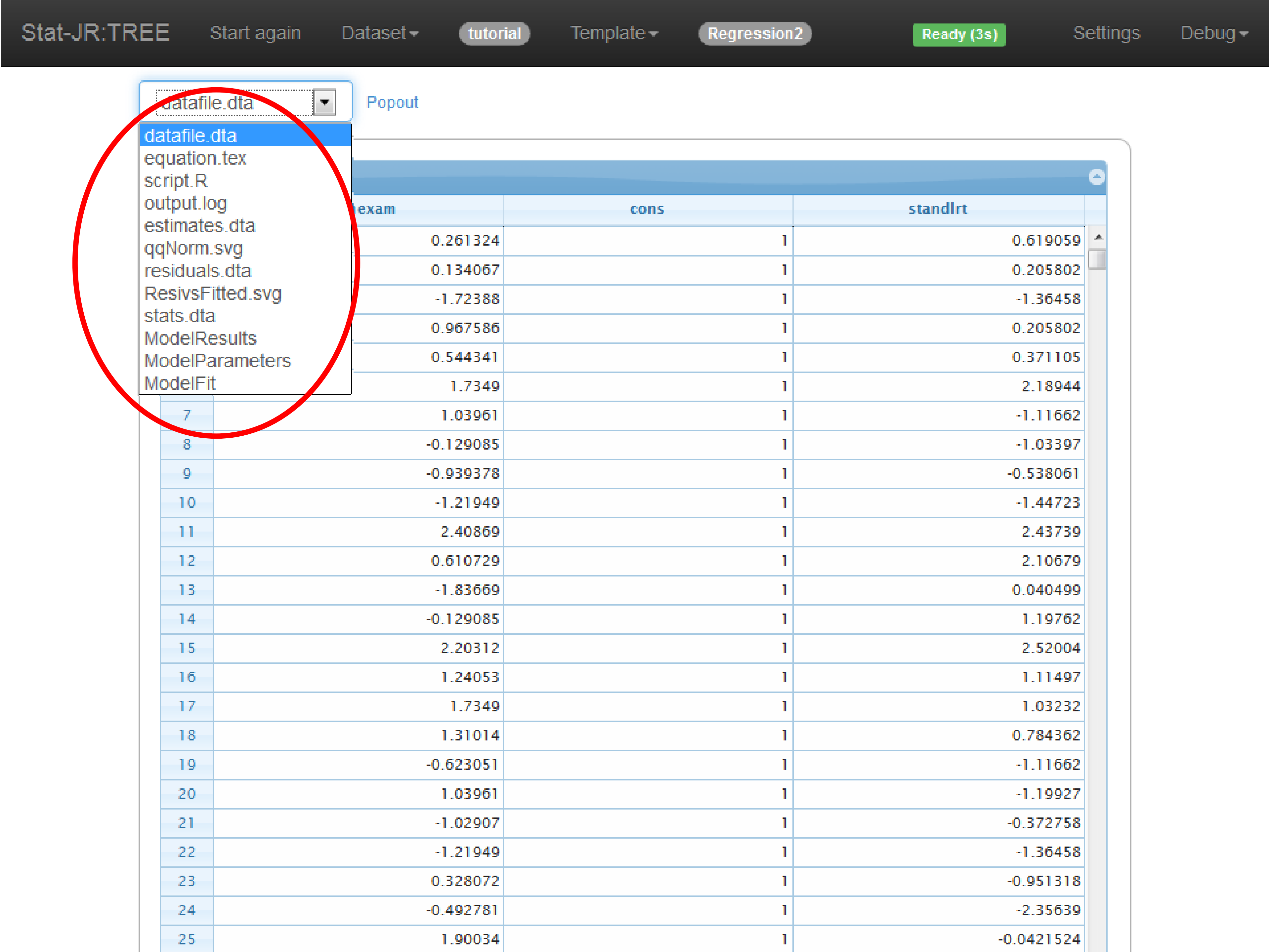
| | normexam | cons | standlrt |
|---|---|---|---|
| 1 | 0.261324 | 1 | 0.619059 |
| 2 | 0.134067 | 1 | 0.205802 |
| 3 | -1.72388 | 1 | -1.36458 |
| 4 | 0.967586 | 1 | 0.205802 |
| 5 | 0.544341 | 1 | 0.371105 |
| 6 | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

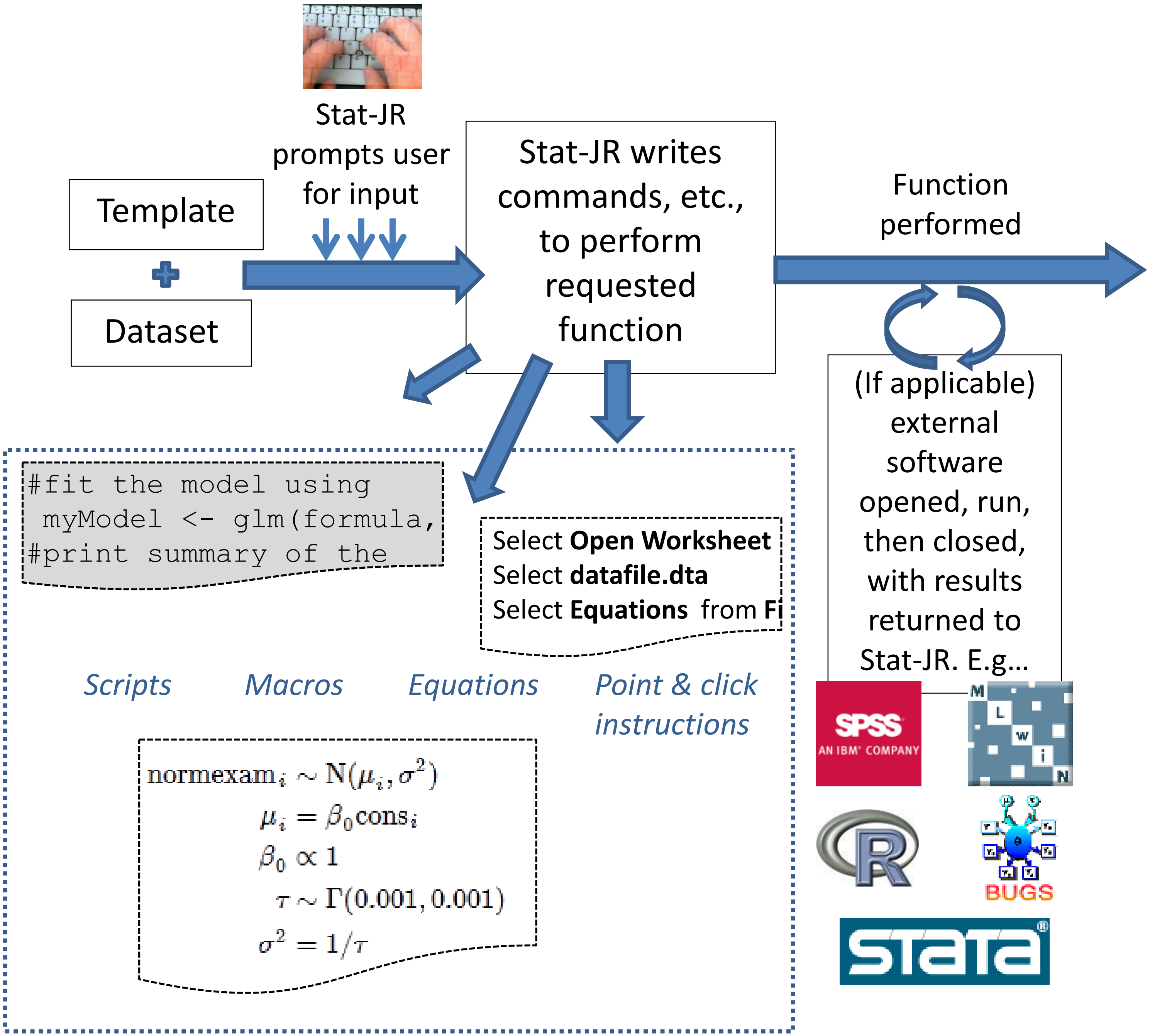
datafile.dta Popout

datafile.dta
equation.tex
script.R
output.log
estimates.dta
qqNorm.svg
residuals.dta
ResivsFitted.svg
stats.dta
ModelResults
ModelParameters
ModelFit

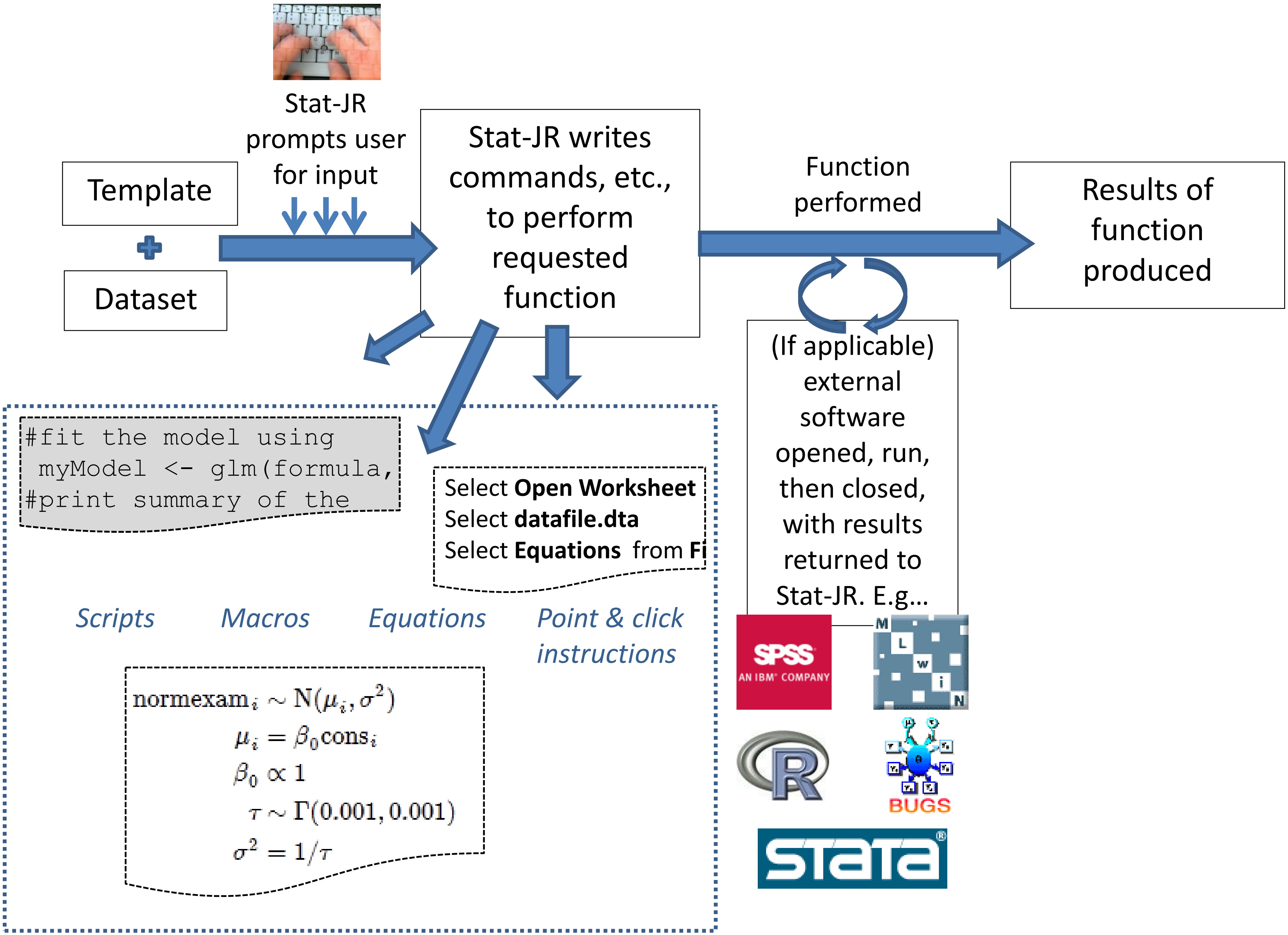
| | exam | cons | standlrt |
|---|---|---|---|
| | 0.261324 | 1 | 0.619059 |
| | 0.134067 | 1 | 0.205802 |
| | -1.72388 | 1 | -1.36458 |
| | 0.967586 | 1 | 0.205802 |
| | 0.544341 | 1 | 0.371105 |
| | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

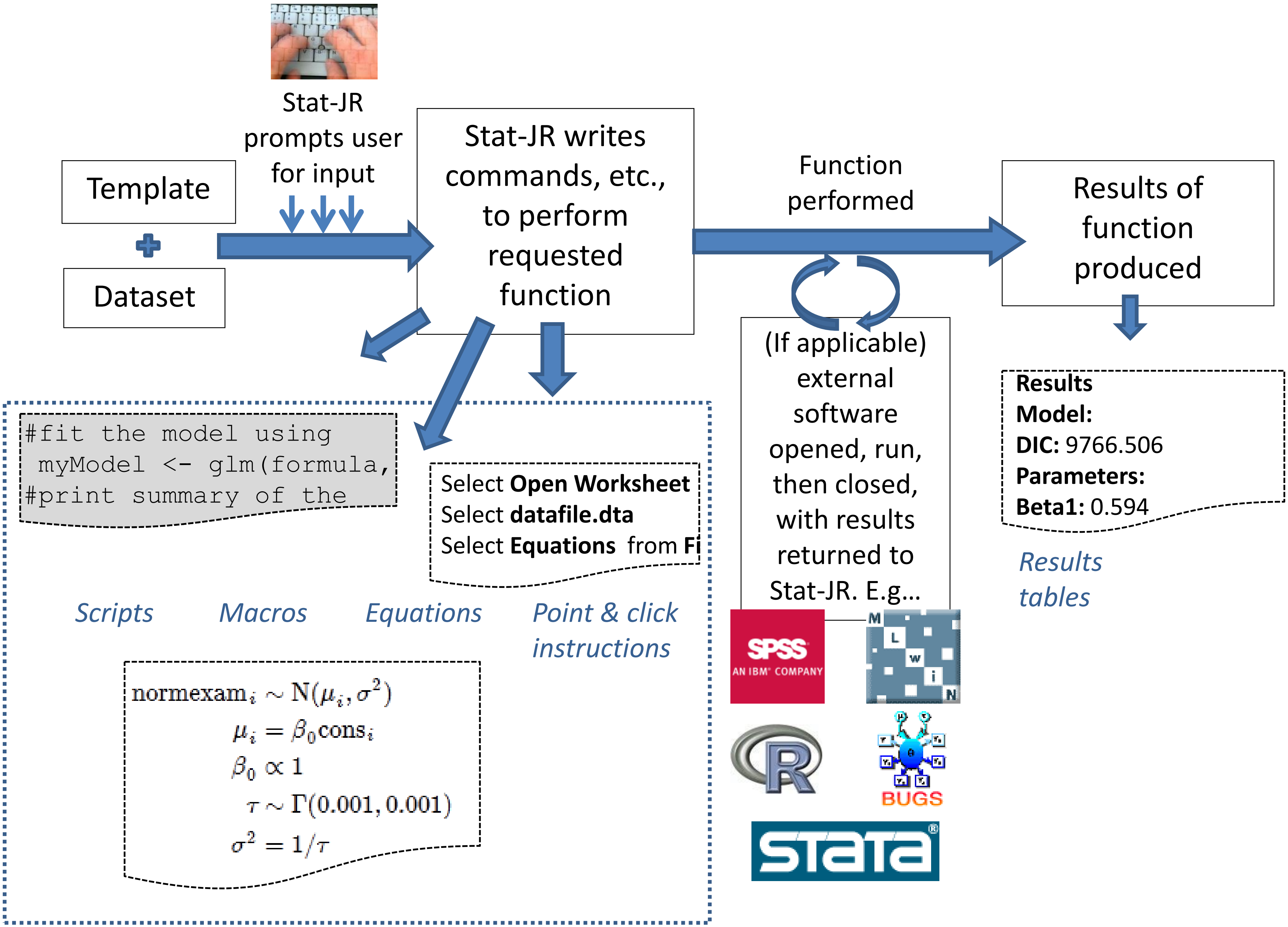


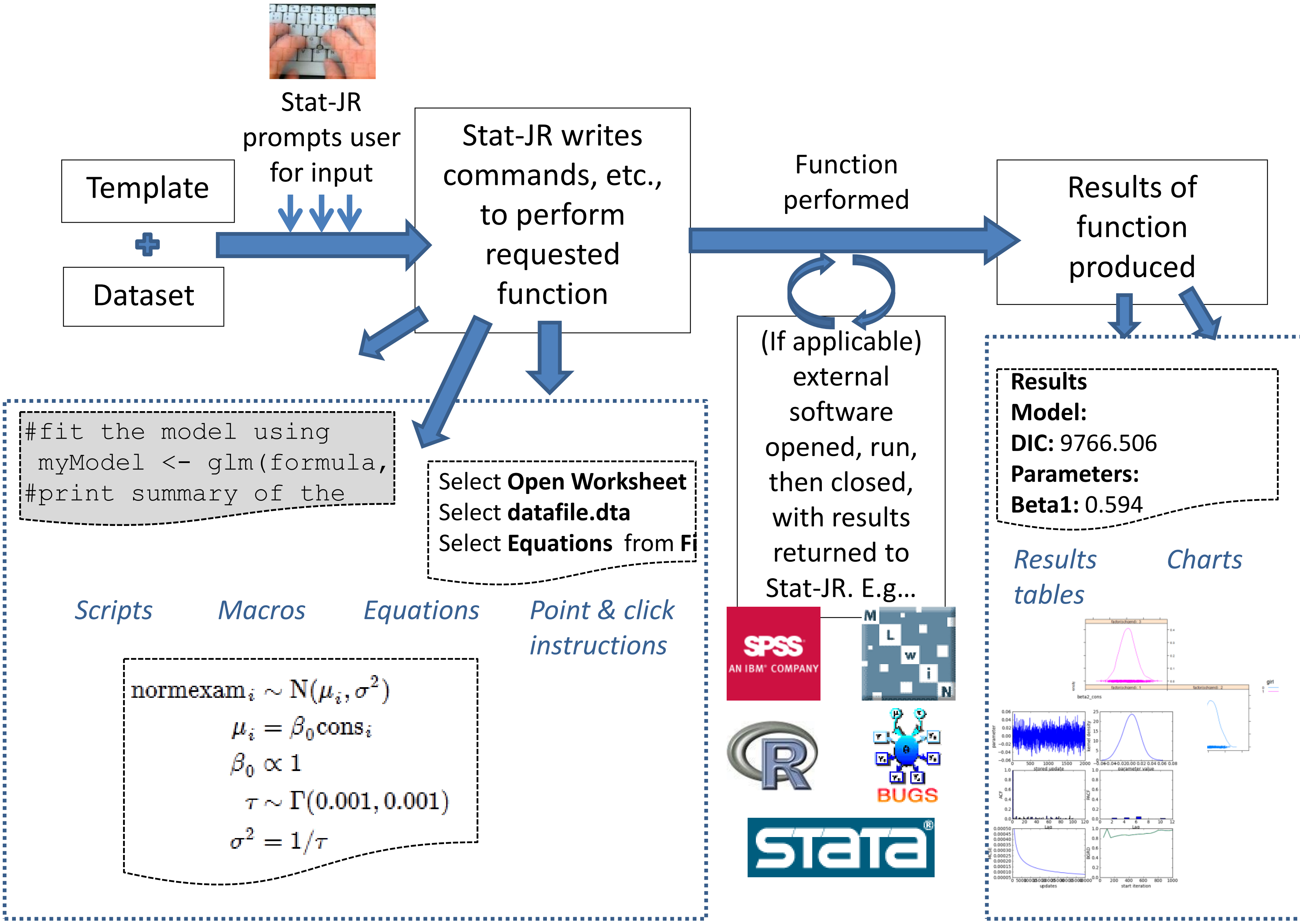

| | exam | cons | standlrt |
|---|---|---|---|
| | 0.261324 | 1 | 0.619059 |
| | 0.134067 | 1 | 0.205802 |
| | -1.72388 | 1 | -1.36458 |
| | 0.967586 | 1 | 0.205802 |
| | 0.544341 | 1 | 0.371105 |
| | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |



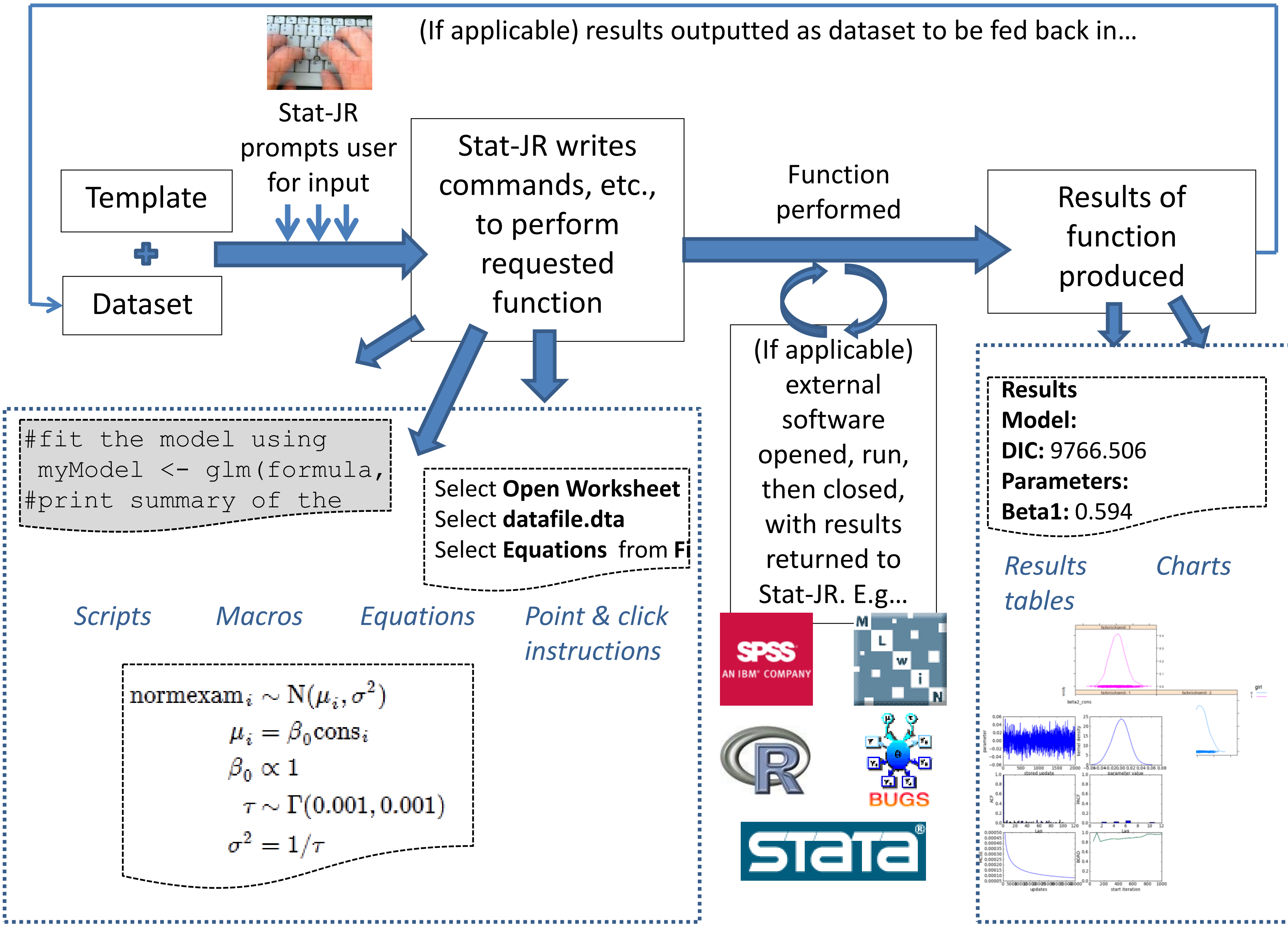
Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g...
SPSS AN IBM COMPANY
MLwiN
R
BUGS
STATA
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$


Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g...
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim$ N$(\mu_i, \sigma^2)$
$\mu_i = \beta_0$cons$_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
SPSS
AN IBM COMPANY
BUGS


Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g...
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim$ N($\mu_i, \sigma^2$)
$\mu_i = \beta_0$cons$_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
Results
Model:
DIC: 9766.506
Parameters:
Beta1: 0.594
Results tables
SPSS AN IBM COMPANY
MLwiN
BUGS
stata



Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g...
SPSS AN IBM COMPANY
MLwiN
R
BUGS
STATA
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
$\mathrm{normexam}_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
Results
Model:
DIC: 9766.506
Parameters:
Beta1: 0.594
Results tables
Charts


(If applicable) results outputted as dataset to be fed back in…
Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g…
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i \sim \mathrm{N}(\mu_i, \sigma^2)$
$\mu_i = \beta_0 \mathrm{cons}_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
SPSS AN IBM COMPANY
MLwiN
R
BUGS
STATA
Results
Model:
DIC: 9766.506
Parameters:
Beta1: 0.594
Results tables
Charts

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

datafile.dta

Popout

- datafile.dta
- equation.tex
- script.R
- output.log
- estimates.dta
- qqNorm.svg
- residuals.dta
- ResivsFitted.svg
- stats.dta
- ModelResults
- ModelParameters
- ModelFit

| | exam | cons | standlrt |
|---|---|---|---|
| | 0.261324 | 1 | 0.619059 |
| | 0.134067 | 1 | 0.205802 |
| | -1.72388 | 1 | -1.36458 |
| | 0.967586 | 1 | 0.205802 |
| | 0.544341 | 1 | 0.371105 |
| | 1.7349 | 1 | 2.18944 |
| 7 | 1.03961 | 1 | -1.11662 |
| 8 | -0.129085 | 1 | -1.03397 |
| 9 | -0.939378 | 1 | -0.538061 |
| 10 | -1.21949 | 1 | -1.44723 |
| 11 | 2.40869 | 1 | 2.43739 |
| 12 | 0.610729 | 1 | 2.10679 |
| 13 | -1.83669 | 1 | 0.040499 |
| 14 | -0.129085 | 1 | 1.19762 |
| 15 | 2.20312 | 1 | 2.52004 |
| 16 | 1.24053 | 1 | 1.11497 |
| 17 | 1.7349 | 1 | 1.03232 |
| 18 | 1.31014 | 1 | 0.784362 |
| 19 | -0.623051 | 1 | -1.11662 |
| 20 | 1.03961 | 1 | -1.19927 |
| 21 | -1.02907 | 1 | -0.372758 |
| 22 | -1.21949 | 1 | -1.36458 |
| 23 | 0.328072 | 1 | -0.951318 |
| 24 | -0.492781 | 1 | -2.35639 |
| 25 | 1.90034 | 1 | -0.0421524 |

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

output.log Popout

```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
> local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
> # Note that when Stat-JR interoperates with R, it sets the working
> # directory to wherever the user's temporary files are stored, i.e.
> # workdir = tempdir(). The data to be modelled, this script, and the
> # files exported from R, are all saved there.
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
>
> # test to see if foreign package is already installed, if not, then install it
> if (!require(foreign)) {
+     install.packages("foreign")
```

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

output.log Popout

```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
> local({r <- getOption("repos"); r["CRAN"] <- "http://cran.r-project.org"; options(repos = r)})
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
> # Note that when Stat-JR interoperates with R, it sets the working
> # directory to wherever the user's temporary files are stored, i.e.
> # workdir = tempdir(). The data to be modelled,          ipt, and the
> # files exported from R, are all saved the
> # # # # # # # # # # # # # # # # # # # # #           # # # # # # # #
>
> # test to see if foreign package is already installed, if not, then install it
> if (!require(foreign)) {
+     install.packages("foreign")
```

Stat-JR:TREE   Start again   Dataset   tutorial   Template   Regression2   Ready (3s)   Settings   Debug

```
>
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
> # Below we specify the model formula, formatted as y ~ x1 + x2 + ...
> # Since Stat-JR assumes users have included the intercept in their list
> # of explanatory variables, -1 removes the intercept which the glm
> # function otherwise adds by default.
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
>
> formula <- normexam ~ cons + standlrt - 1
> # fit the model using the glm function, specifying the formula, data, and distribution (with iden
tity link) in its arguments
> myModel <- glm(formula, data = mydata, family = gaussian(identity))
> # print summary of the model fit
> summary(myModel)

Call:
glm(formula = formula, family = gaussian(identity), data = mydata)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.65615  -0.51848   0.01264   0.54399   2.97399

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
cons     -0.001191   0.012642  -0.094    0.925
standlrt  0.595057   0.012730  46.744   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.6487385)

    Null deviance: 4049.4  on 4059  degrees of freedom
Residual deviance: 2631.9  on 4057  degrees of freedom
AIC: 9766.5

Number of Fisher Scoring iterations: 2
```

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

```
>
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
> # Below we specify the model formula, formatted as y ~ x1 + x2 + ...
> # Since Stat-JR assumes users have included the intercept in their list
> # of explanatory variables, -1 removes the intercept which the glm
> # function otherwise adds by default.
> # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
>
> formula <- normexam ~ cons + standlrt - 1
> # fit the model using the glm function, specifying the formula, data, and distribution (with iden
tity link) in its arguments
> myModel <- glm(formula, data = mydata, family = gaussian(identity))
> # print summary of the model fit
> summary(myModel)

Call:
glm(formula = formula, family = gaussian(identity), data = mydata)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.65615  -0.51848   0.01264   0.54399   2.97399

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
cons     -0.001191   0.012642  -0.094    0.925
standlrt  0.595057   0.012730  46.744   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to [illegible]6487385)

    Null deviance: 4049.4  on 4059  degrees of fr[illegible]
Residual deviance: 2631.9  on 4057  degrees of freedom
AIC: 9766.5

Number of Fisher Scoring iterations: 2
```

Stat-JR:TREE Start again Dataset tutorial Template Regression2 Ready (3s) Settings Debug

Set

Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})

ModelResults Popout

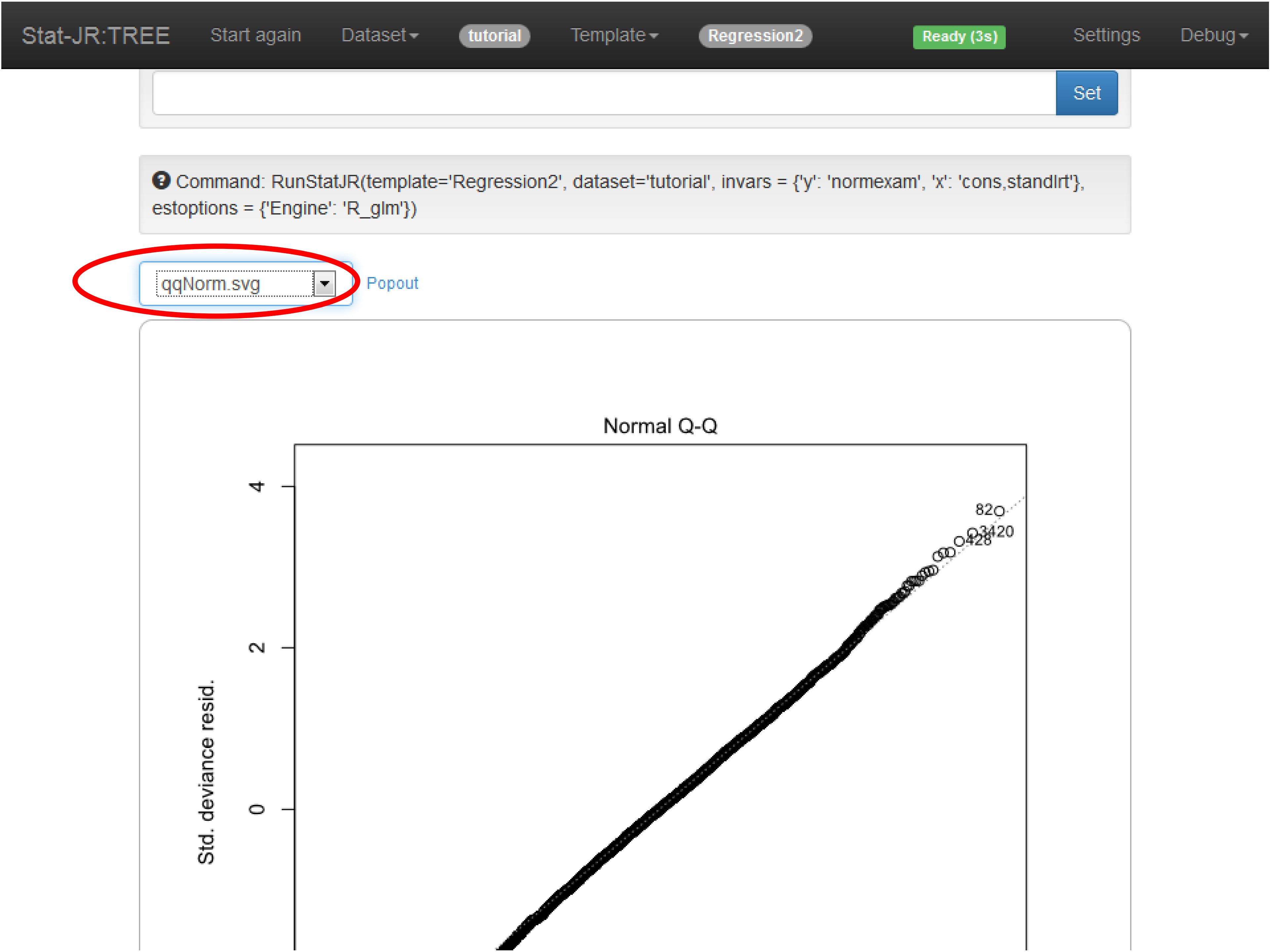
# Results

## Parameters:

| parameter | est | se |
|---|---|---|
| cons | -0.00119111914973 | 0.0126422981796 |
| standlrt | 0.595056780156 | 0.0127300927553 |

## Model:

| Statistic | Value |
|---|---|
| deviance | 2631.93206193 |
| nulldeviance | 4049.43302581 |
| aic | 9766.50937651 |
| converged | 1 |
| iter | 2 |


Stat-JR:TREE
Start again
Dataset
tutorial
Template
Regression2
Ready (3s)
Settings
Debug
Set
Command: RunStatJR(template='Regression2', dataset='tutorial', invars = {'y': 'normexam', 'x': 'cons,standlrt'}, estoptions = {'Engine': 'R_glm'})
qqNorm.svg
Popout
Normal Q-Q
Std. deviance resid.
4
2
0
82
3420
428

# Choice of interface

Three different ways to interact with Stat-JR:

1. **Point-and-click menu-driven** interface (TREE)

2. **eBook** interface (DEEP)

3. **Command line** interface (runStatJR)

# Choice of interface

Three different ways to interact with Stat-JR:

1. **Point-and-click menu-driven** interface (TREE)
2. **eBook** interface (DEEP)
3. **Command line** interface (runStatJR)

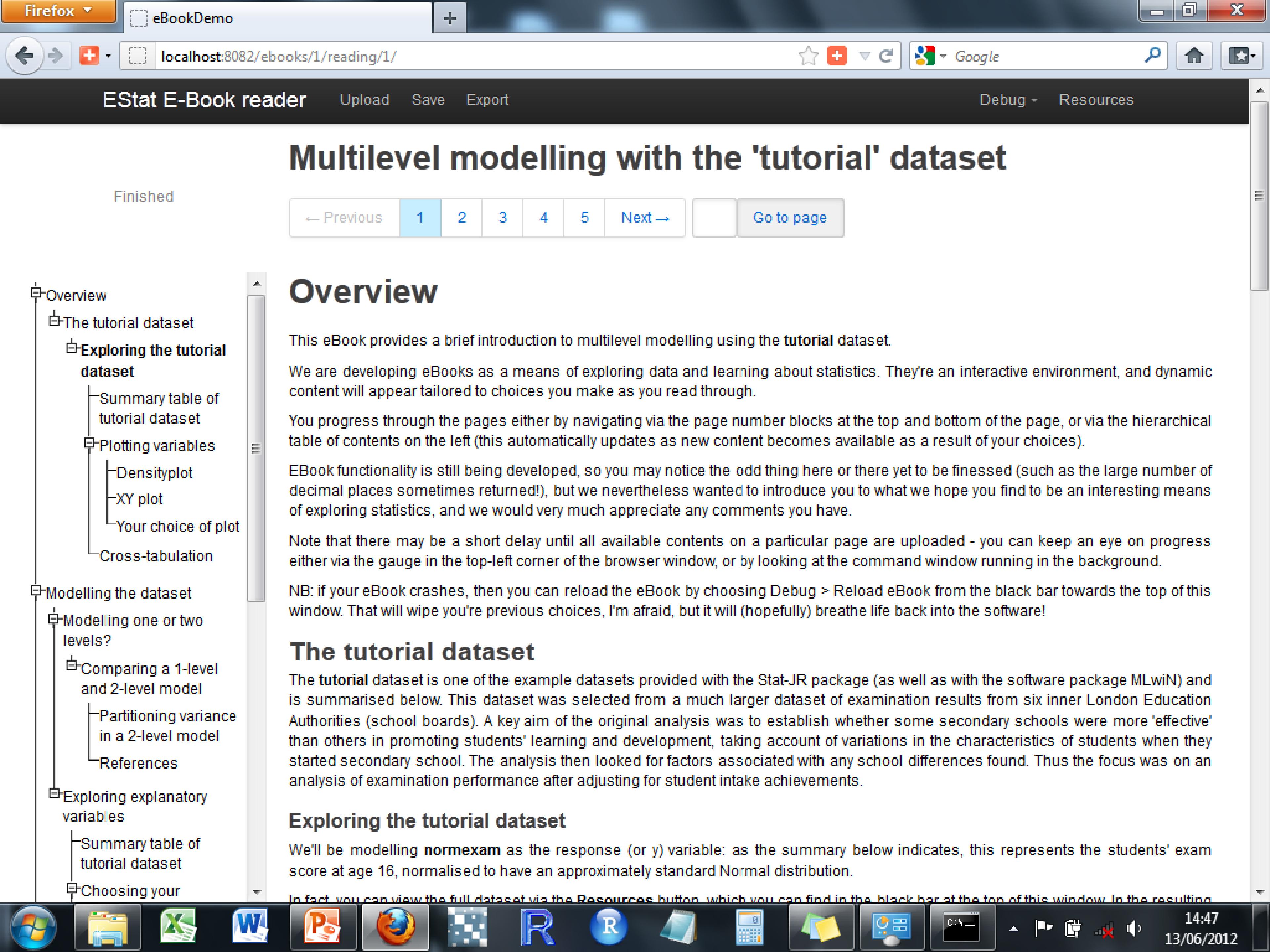
# Stat-JR's eBook interface: **DEEP**

**D**ocuments with

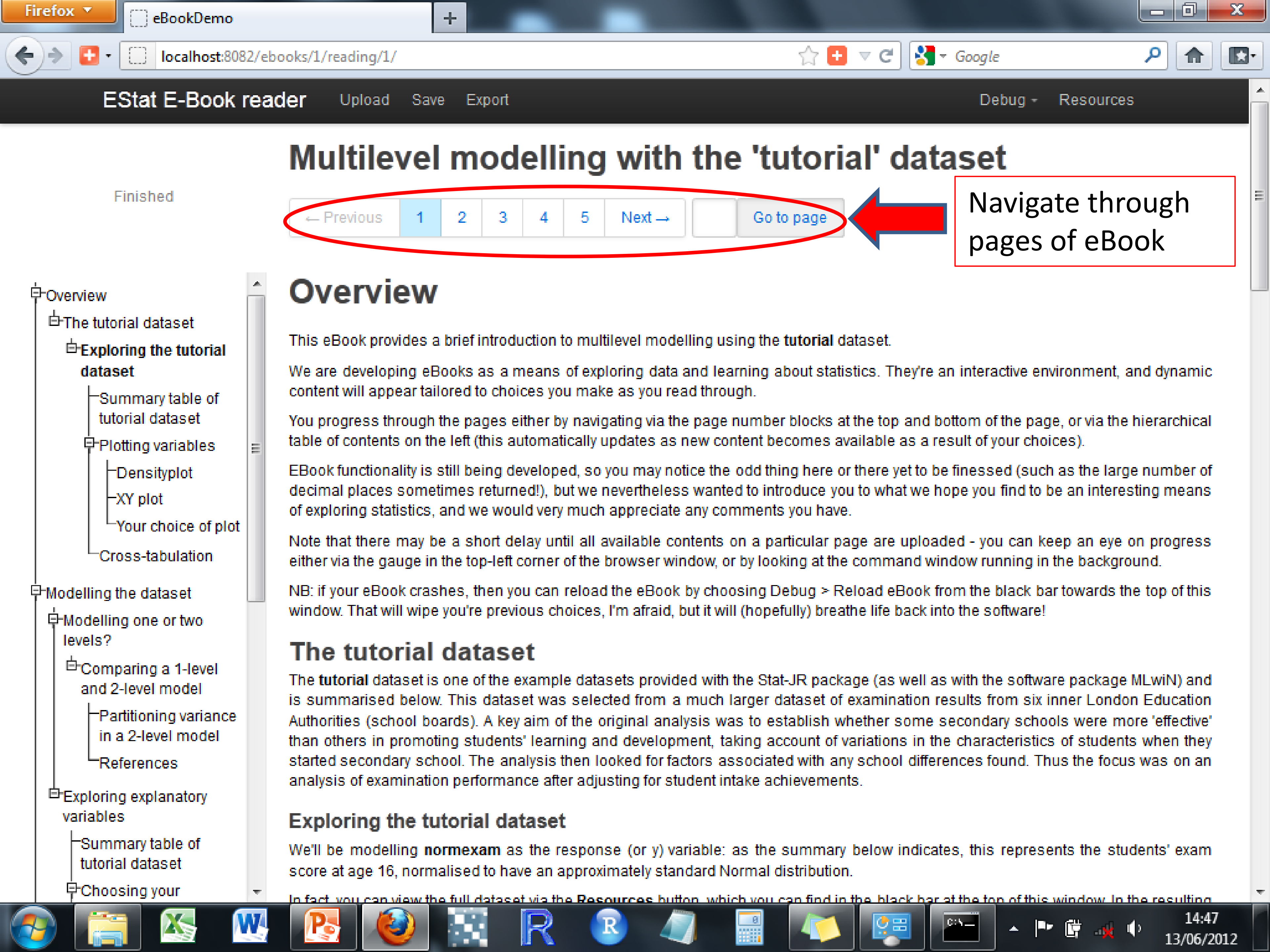
**E**mbedded

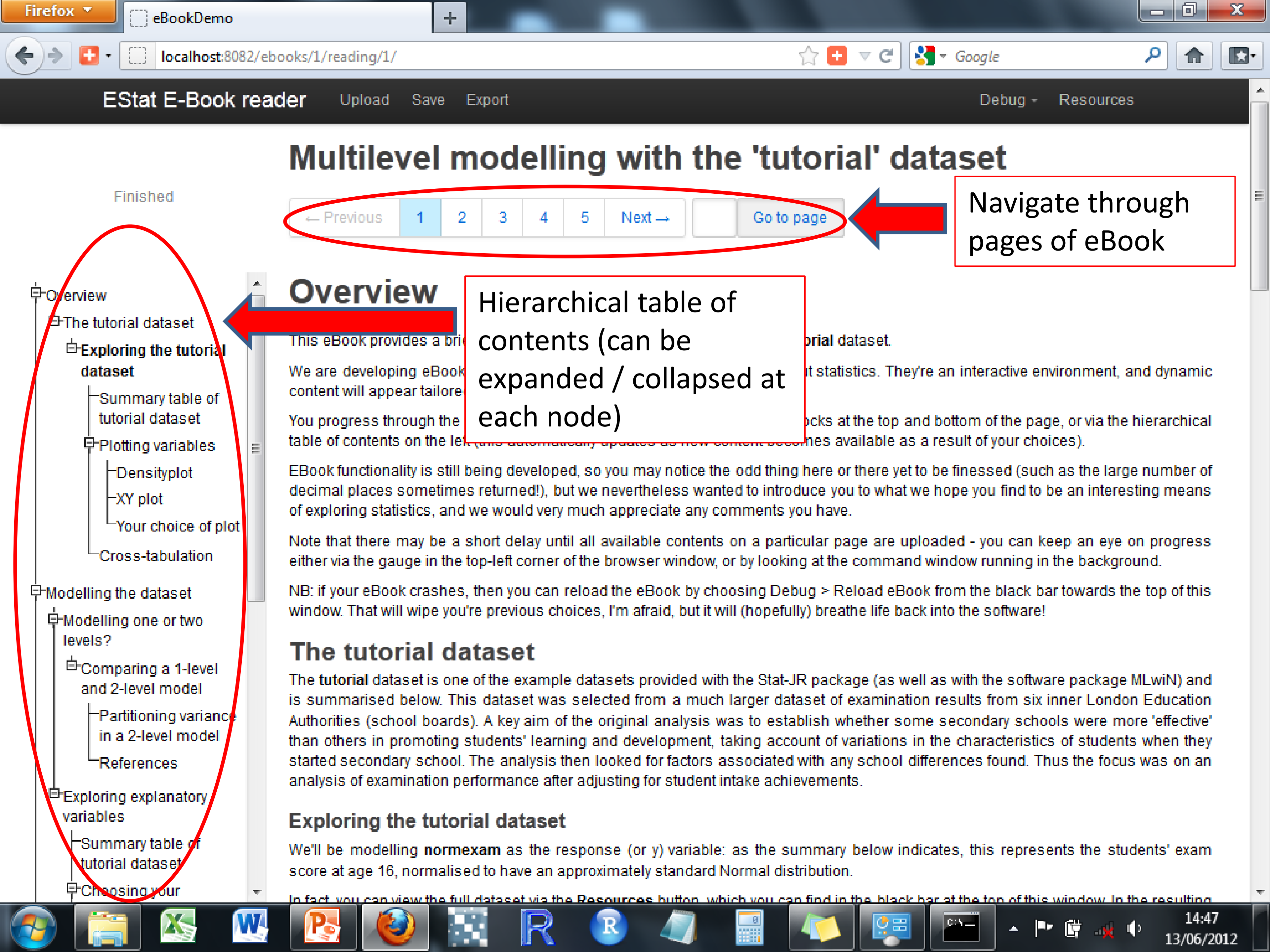
**E**xecution and

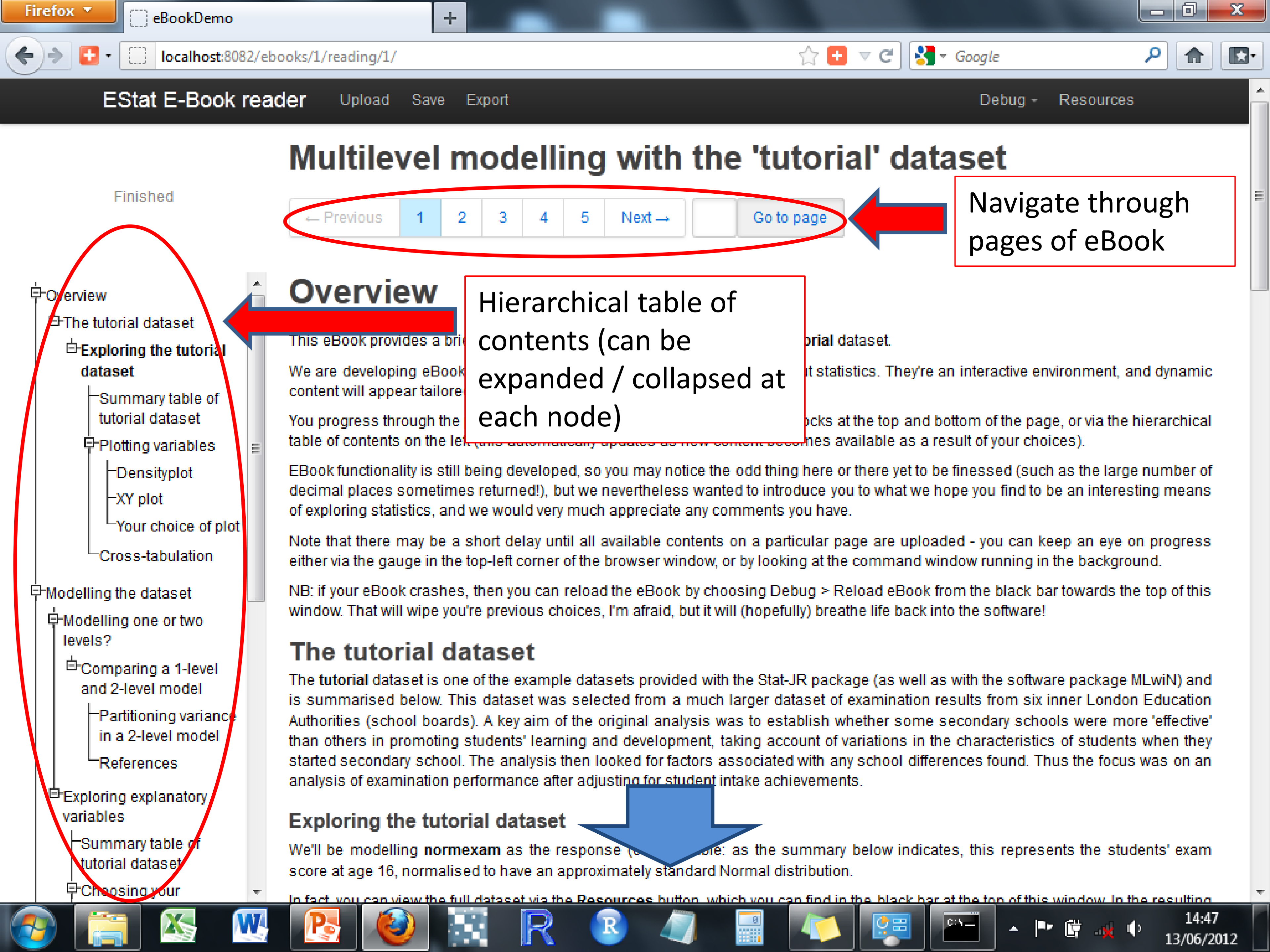
**P**rovenance



Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Overview
This eBook provides a brief introduction to multilevel modelling using the tutorial dataset.
We are developing eBooks as a means of exploring data and learning about statistics. They're an interactive environment, and dynamic content will appear tailored to choices you make as you read through.
You progress through the pages either by navigating via the page number blocks at the top and bottom of the page, or via the hierarchical table of contents on the left (this automatically updates as new content becomes available as a result of your choices).
EBook functionality is still being developed, so you may notice the odd thing here or there yet to be finessed (such as the large number of decimal places sometimes returned!), but we nevertheless wanted to introduce you to what we hope you find to be an interesting means of exploring statistics, and we would very much appreciate any comments you have.
Note that there may be a short delay until all available contents on a particular page are uploaded - you can keep an eye on progress either via the gauge in the top-left corner of the browser window, or by looking at the command window running in the background.
NB: if your eBook crashes, then you can reload the eBook by choosing Debug > Reload eBook from the black bar towards the top of this window. That will wipe you're previous choices, I'm afraid, but it will (hopefully) breathe life back into the software!
The tutorial dataset
The tutorial dataset is one of the example datasets provided with the Stat-JR package (as well as with the software package MLwiN) and is summarised below. This dataset was selected from a much larger dataset of examination results from six inner London Education Authorities (school boards). A key aim of the original analysis was to establish whether some secondary schools were more 'effective' than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started secondary school. The analysis then looked for factors associated with any school differences found. Thus the focus was on an analysis of examination performance after adjusting for student intake achievements.
Exploring the tutorial dataset
We'll be modelling normexam as the response (or y) variable: as the summary below indicates, this represents the students' exam score at age 16, normalised to have an approximately standard Normal distribution.
14:47
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Navigate through pages of eBook
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Overview
This eBook provides a brief introduction to multilevel modelling using the tutorial dataset.
We are developing eBooks as a means of exploring data and learning about statistics. They're an interactive environment, and dynamic content will appear tailored to choices you make as you read through.
You progress through the pages either by navigating via the page number blocks at the top and bottom of the page, or via the hierarchical table of contents on the left (this automatically updates as new content becomes available as a result of your choices).
EBook functionality is still being developed, so you may notice the odd thing here or there yet to be finessed (such as the large number of decimal places sometimes returned!), but we nevertheless wanted to introduce you to what we hope you find to be an interesting means of exploring statistics, and we would very much appreciate any comments you have.
Note that there may be a short delay until all available contents on a particular page are uploaded - you can keep an eye on progress either via the gauge in the top-left corner of the browser window, or by looking at the command window running in the background.
NB: if your eBook crashes, then you can reload the eBook by choosing Debug > Reload eBook from the black bar towards the top of this window. That will wipe you're previous choices, I'm afraid, but it will (hopefully) breathe life back into the software!
The tutorial dataset
The tutorial dataset is one of the example datasets provided with the Stat-JR package (as well as with the software package MLwiN) and is summarised below. This dataset was selected from a much larger dataset of examination results from six inner London Education Authorities (school boards). A key aim of the original analysis was to establish whether some secondary schools were more 'effective' than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started secondary school. The analysis then looked for factors associated with any school differences found. Thus the focus was on an analysis of examination performance after adjusting for student intake achievements.
Exploring the tutorial dataset
We'll be modelling normexam as the response (or y) variable: as the summary below indicates, this represents the students' exam score at age 16, normalised to have an approximately standard Normal distribution.
14:47
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Navigate through pages of eBook
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Overview
Hierarchical table of contents (can be expanded / collapsed at each node)
This eBook provides a bri
rial dataset.
We are developing eBook
t statistics. They're an interactive environment, and dynamic
content will appear tailore
You progress through the
ocks at the top and bottom of the page, or via the hierarchical
table of contents on the le
es available as a result of your choices).
EBook functionality is still being developed, so you may notice the odd thing here or there yet to be finessed (such as the large number of decimal places sometimes returned!), but we nevertheless wanted to introduce you to what we hope you find to be an interesting means of exploring statistics, and we would very much appreciate any comments you have.
Note that there may be a short delay until all available contents on a particular page are uploaded - you can keep an eye on progress either via the gauge in the top-left corner of the browser window, or by looking at the command window running in the background.
NB: if your eBook crashes, then you can reload the eBook by choosing Debug > Reload eBook from the black bar towards the top of this window. That will wipe you're previous choices, I'm afraid, but it will (hopefully) breathe life back into the software!
The tutorial dataset
The tutorial dataset is one of the example datasets provided with the Stat-JR package (as well as with the software package MLwiN) and is summarised below. This dataset was selected from a much larger dataset of examination results from six inner London Education Authorities (school boards). A key aim of the original analysis was to establish whether some secondary schools were more 'effective' than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started secondary school. The analysis then looked for factors associated with any school differences found. Thus the focus was on an analysis of examination performance after adjusting for student intake achievements.
Exploring the tutorial dataset
We'll be modelling normexam as the response (or y) variable: as the summary below indicates, this represents the students' exam score at age 16, normalised to have an approximately standard Normal distribution.
14:47
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Navigate through pages of eBook
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Overview
Hierarchical table of contents (can be expanded / collapsed at each node)
This eBook provides a bri
orial dataset.
We are developing eBook
t statistics. They're an interactive environment, and dynamic
content will appear tailore
You progress through the
ocks at the top and bottom of the page, or via the hierarchical
table of contents on the le
mes available as a result of your choices).
EBook functionality is still being developed, so you may notice the odd thing here or there yet to be finessed (such as the large number of decimal places sometimes returned!), but we nevertheless wanted to introduce you to what we hope you find to be an interesting means of exploring statistics, and we would very much appreciate any comments you have.
Note that there may be a short delay until all available contents on a particular page are uploaded - you can keep an eye on progress either via the gauge in the top-left corner of the browser window, or by looking at the command window running in the background.
NB: if your eBook crashes, then you can reload the eBook by choosing Debug > Reload eBook from the black bar towards the top of this window. That will wipe you're previous choices, I'm afraid, but it will (hopefully) breathe life back into the software!
The tutorial dataset
The tutorial dataset is one of the example datasets provided with the Stat-JR package (as well as with the software package MLwiN) and is summarised below. This dataset was selected from a much larger dataset of examination results from six inner London Education Authorities (school boards). A key aim of the original analysis was to establish whether some secondary schools were more 'effective' than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started secondary school. The analysis then looked for factors associated with any school differences found. Thus the focus was on an analysis of examination performance after adjusting for student intake achievements.
Exploring the tutorial dataset
We'll be modelling normexam as the response
le: as the summary below indicates, this represents the students' exam score at age 16, normalised to have an approximately standard Normal distribution.
14:47
13/06/2012

Firefox eBookDemo

localhost:8082/ebooks/1/reading/1/

EStat E-Book reader Upload Save Export Debug Resources

# Multilevel modelling with the 'tutorial' dataset

Finished

← Previous 1 2 3 4 5 Next → Go to page



### Summary table of tutorial dataset

| Column name | n | Missing | Min | Max | Description |
|---|---|---|---|---|---|
| **school** | 4059 | 0 | 1 | 65 | Numeric school identifier |
| **student** | 4059 | 0 | 1 | 198 | Numeric student identifier |
| **normexam** | 4059 | 0 | -3.67 | 3.67 | Students' exam score at age 16, normalised to have approximately a standard Normal distribution. |
| **cons** | 4059 | 0 | 1 | 1 | A column of ones. If included as an explanatory variable in a regression model, its coefficient is the intercept. |
| **standlrt** | 4059 | 0 | -2.93 | 3.02 | Students' score at age 11 on the London Reading Test (LRT), standardised using Z-scores. |
| **girl** | 4059 | 0 | 0 | 1 | Students' gender: 0=boy; 1=girl |
| **schgend** | 4059 | 0 | 1 | 3 | School gender: 1=mixed; 2=boys' school; 3=girls' school |
| **avslrt** | 4059 | 0 | -0.76 | 0.64 | Average LRT score in school |
| **schav** | 4059 | 0 | 1 | 3 | Average LRT score in school, coded into 3 categories: 1=bottom 25%; 2=middle 50%; 3=top 25% |
| **vrband** | 4059 | 0 | 1 | 3 | Students' score in test of verbal reasoning at age 11, coded into 3 categories: 1=top 25%; 2=middle 50%; 3=bottom 25% |

### Plotting variables

Here you can graphically-explore the **tutorial** dataset.

In the first two sections, below, you can produce a densityplot and XY plot, respectively; here you can re-specify your choice of variables

14:48
13/06/2012

Firefox | eBookDemo | +

localhost:8082/ebooks/1/reading/1/

EStat E-Book reader   Upload   Save   Export   Debug   Resources

# Multilevel modelling with the 'tutorial' dataset

Finished

← Previous | 1 | 2 | 3 | 4 | 5 | Next → | | Go to page



### Summary table of tutorial dataset

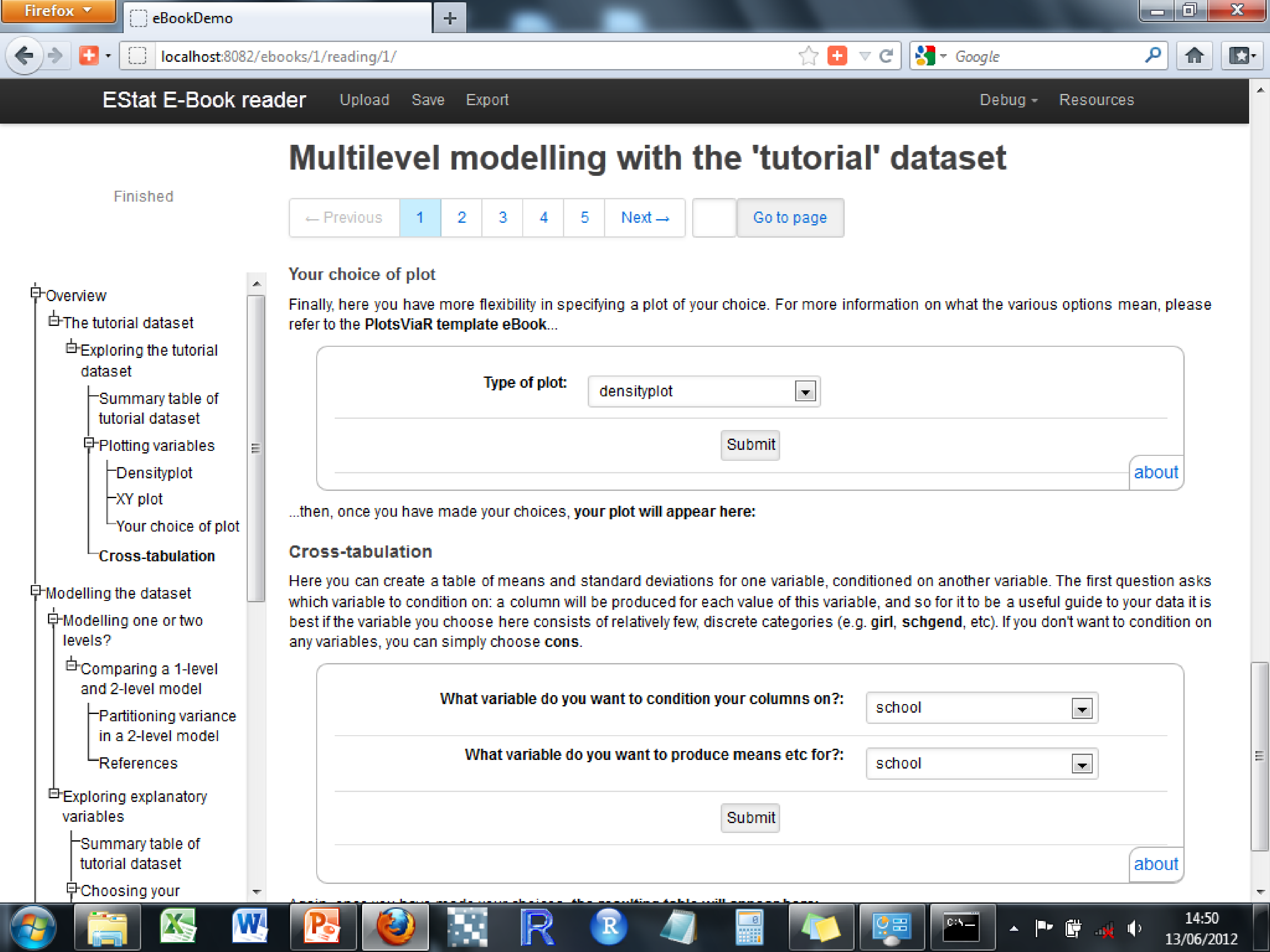
| Column name | n | Missing | Min | Max | Description |
|---|---|---|---|---|---|
| **school** | 4059 | 0 | 1 | 65 | Numeric school identifier |
| **student** | 4059 | 0 | 1 | 198 | Numeric student identifier |
| **normexam** | 4059 | 0 | -3.67 | 3.67 | Students' exam score at age 16, normalised to have approximately a standard Normal distribution. |
| **cons** | 4059 | 0 | 1 | 1 | A column of ones. If included as an explanatory variable in a regression model, its coefficient is the intercept. |
| **standlrt** | 4059 | 0 | -2.93 | 3.02 | Students' score at age 11 on the London Reading Test (LRT), standardised using Z-scores. |
| **girl** | 4059 | 0 | 0 | 1 | Students' gender: 0=boy; 1=girl |
| **schgend** | 4059 | 0 | 1 | 3 | School gender: 1=mixed; 2=boys' school; 3=girls' school |
| **avslrt** | 4059 | 0 | -0.76 | 0.64 | Average LRT score in school |
| **schav** | 4059 | 0 | 1 | 3 | Average LRT score in school, coded into 3 categories: 1=bottom 25%; 2=middle 50%; 3=top 25% |
| **vrband** | 4059 | 0 | 1 | 3 | Students' score in test of verbal reasoning at age 11, coded into 3 categories: 1=top 25%; 2=middle 50%; 3=bottom 25% |

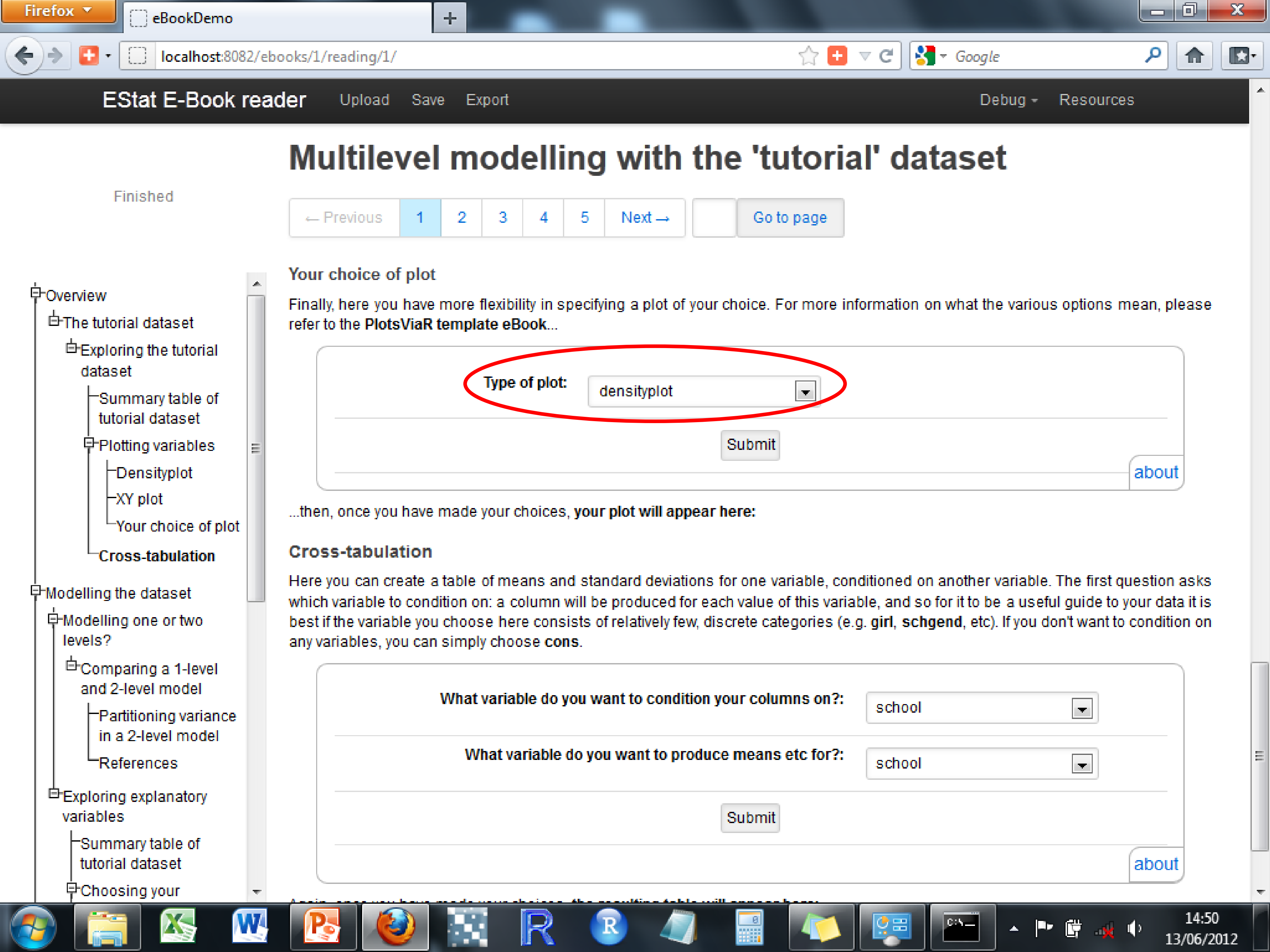
### Plotting variables

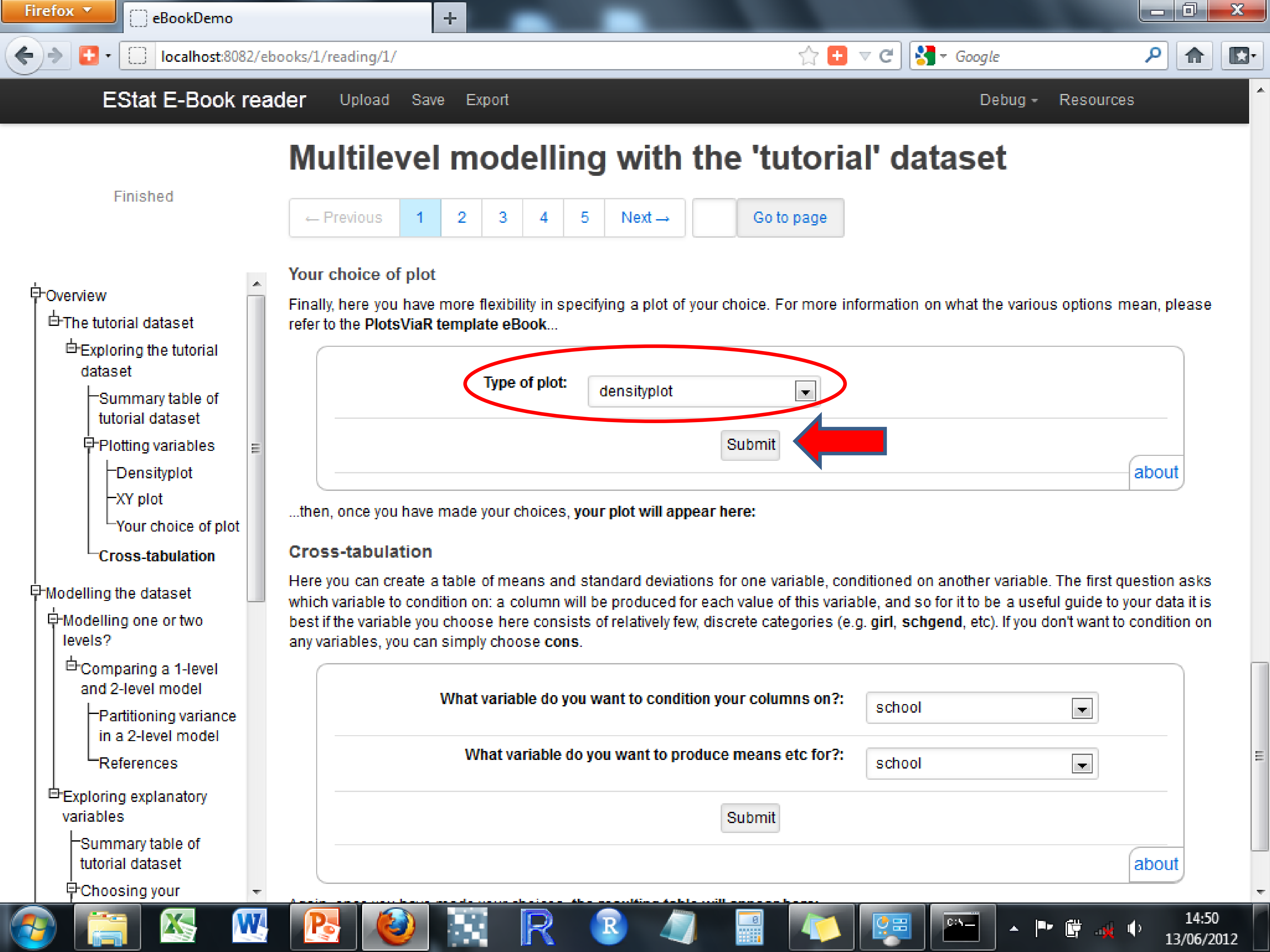
Here you can graphically-explore the **tutorial** dataset.

In the first two sections, below, you can produce a densityplot and XY plot, respectively; here you can re-specify your choice of variables

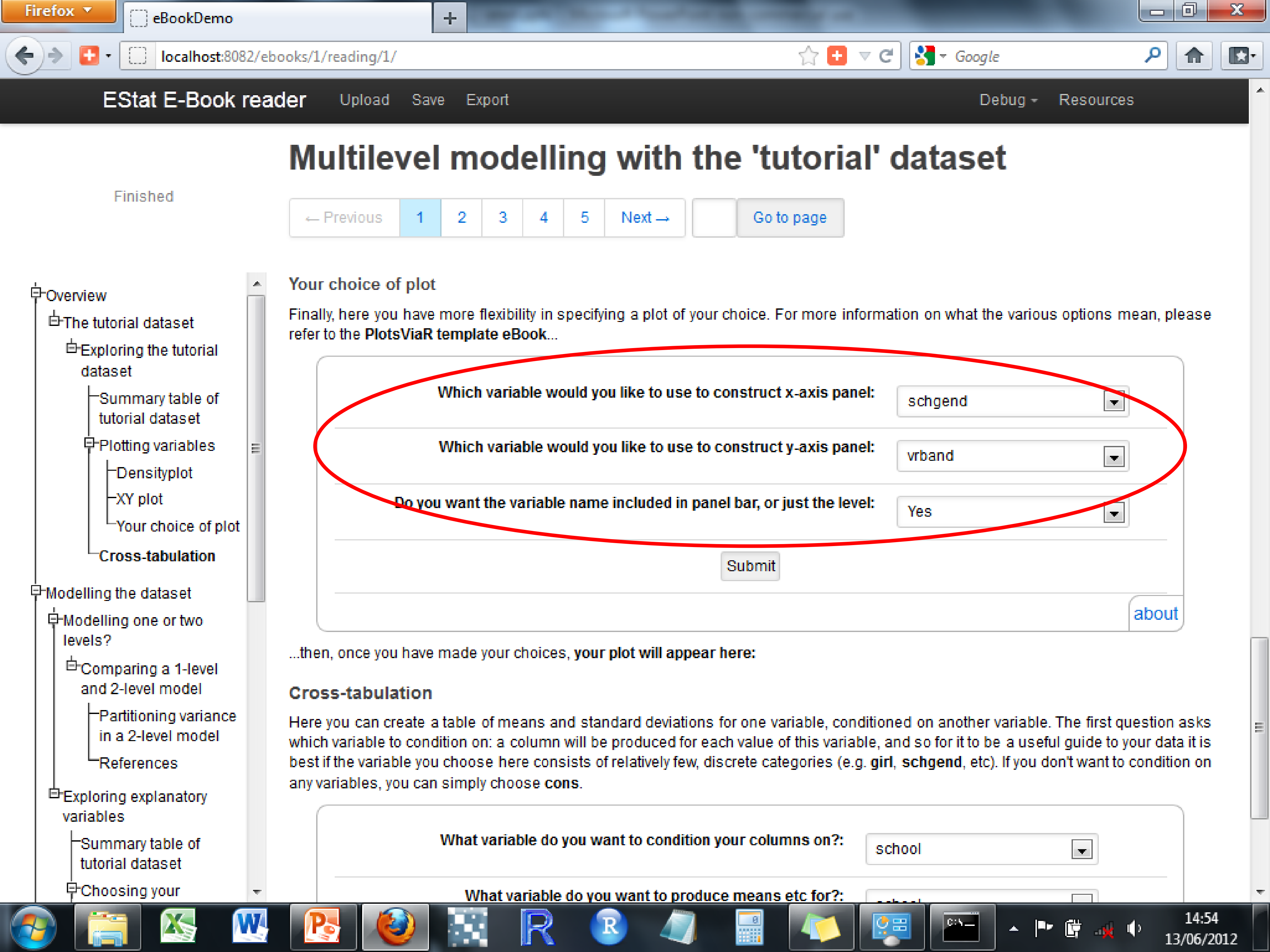
14:48
13/06/2012

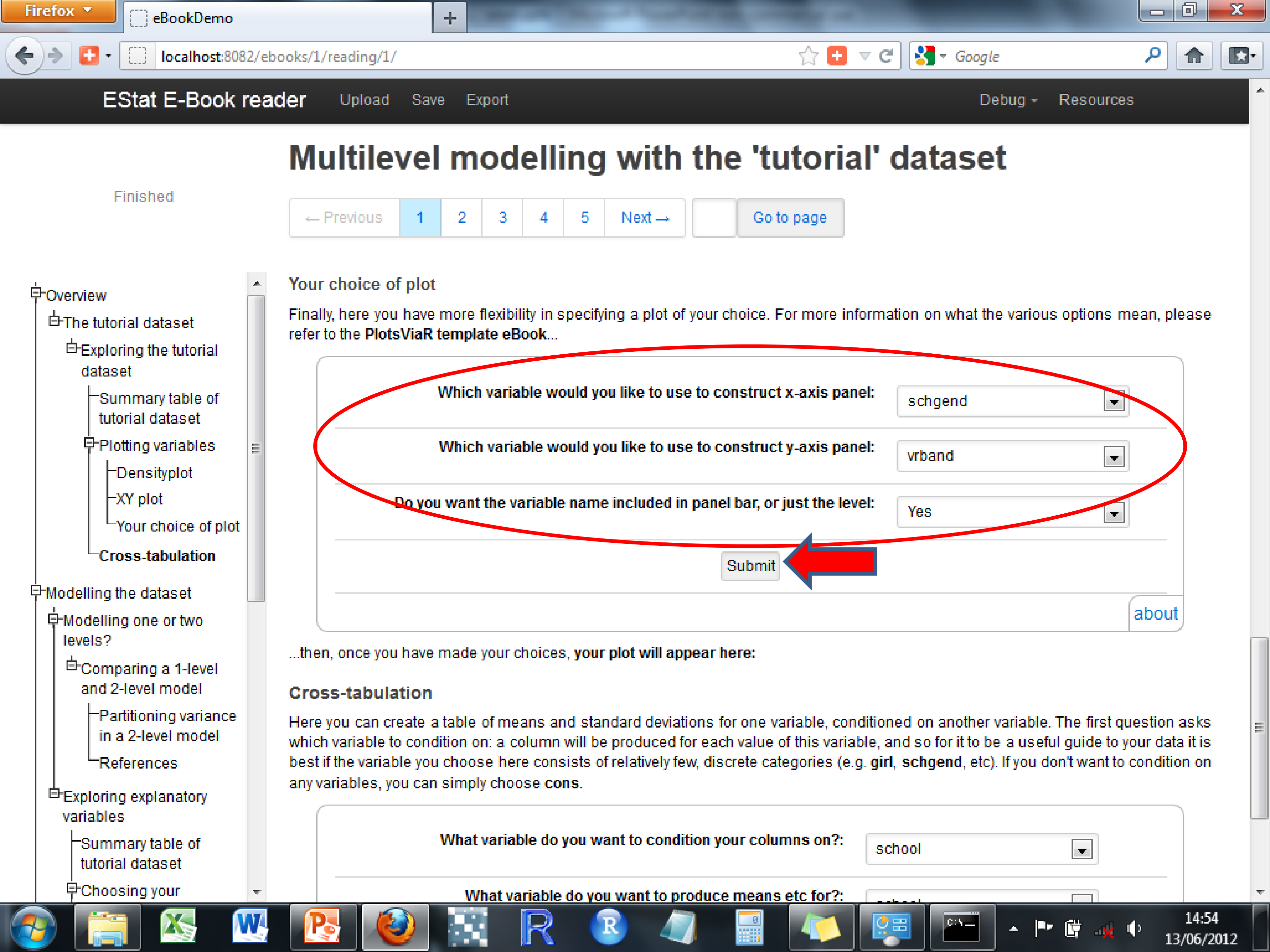

Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Type of plot:
densityplot
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
school
Submit
about
14:50
13/06/2012

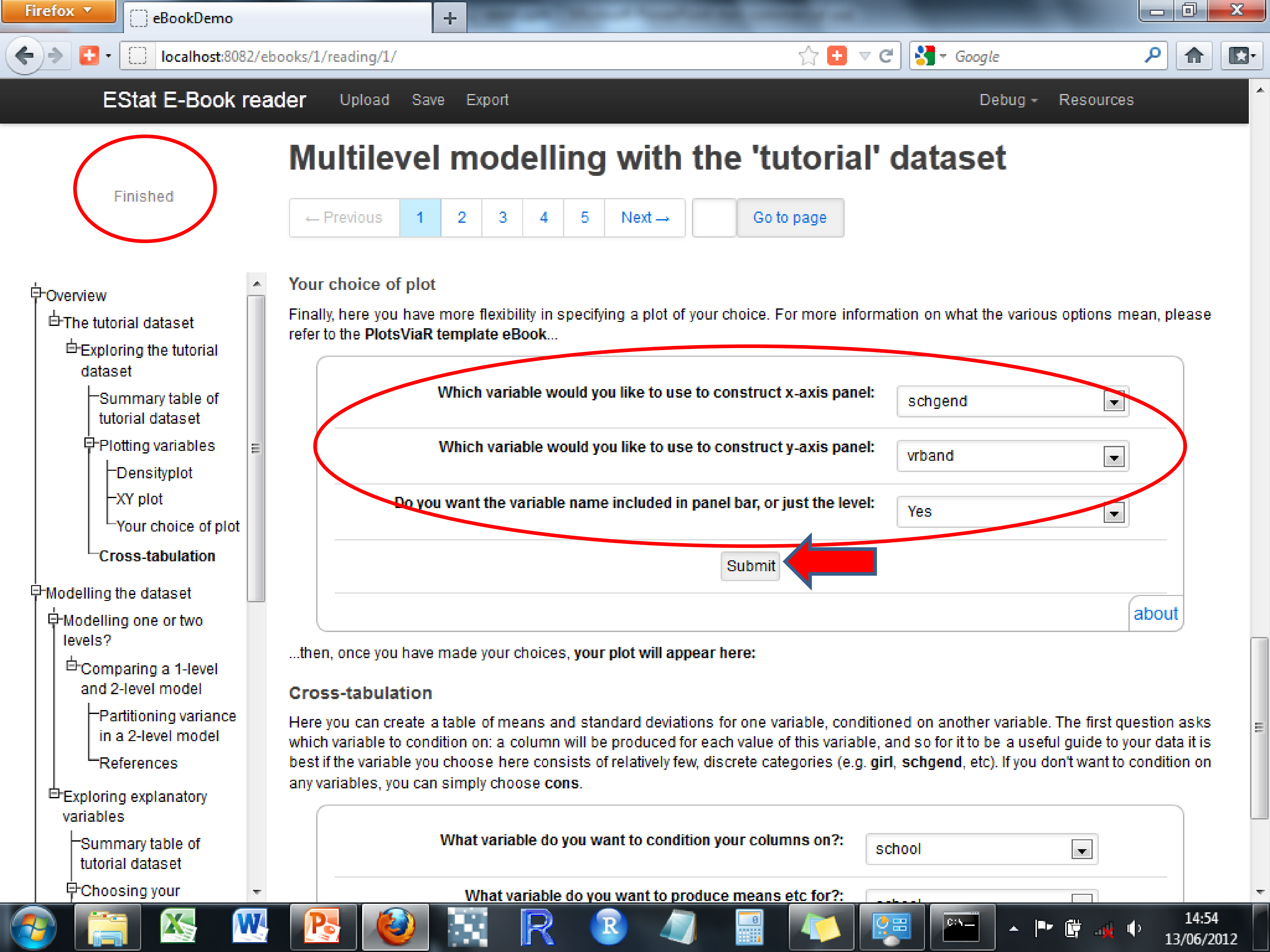

Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Type of plot:
densityplot
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
school
Submit
about
14:50
13/06/2012

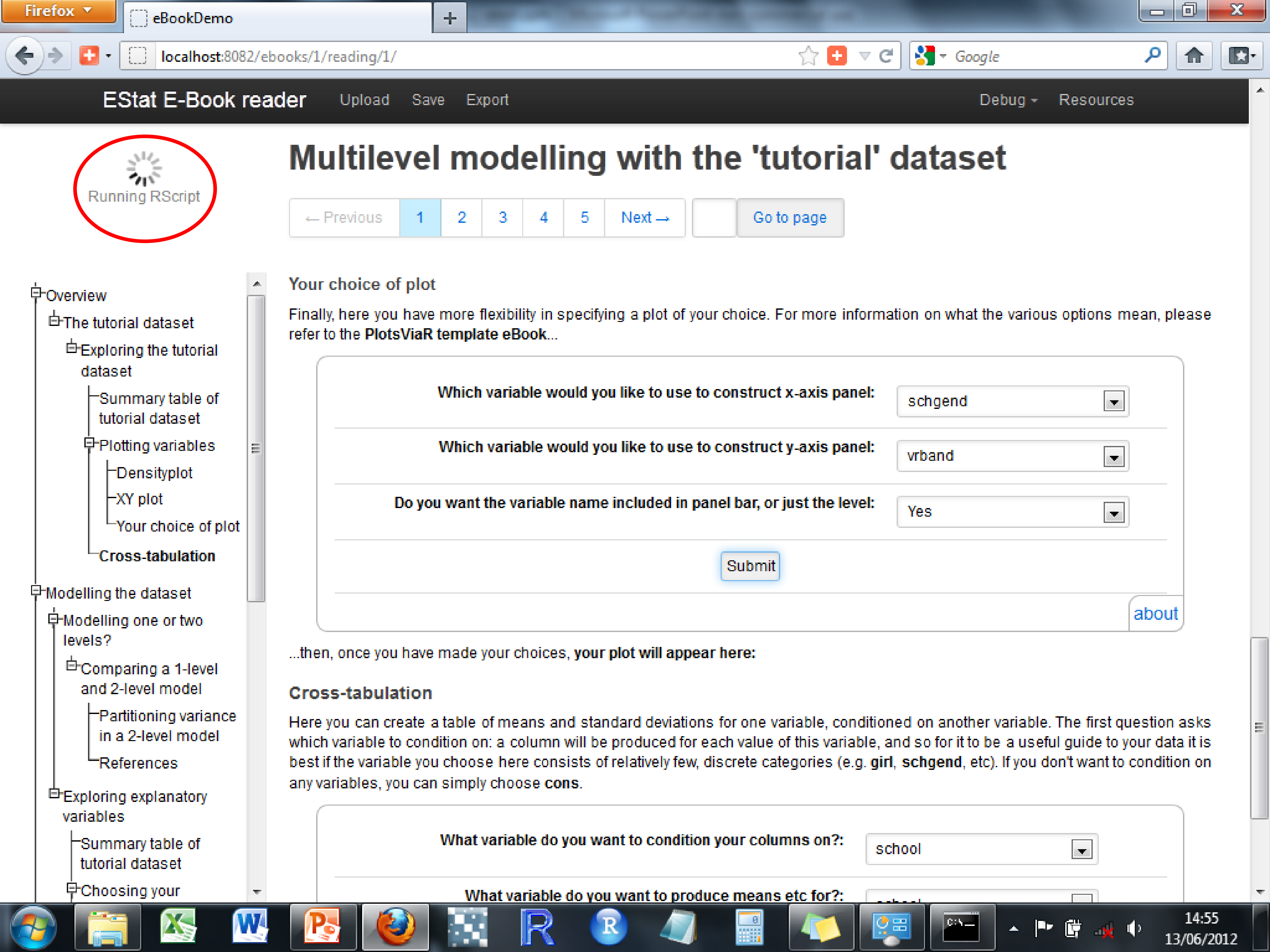


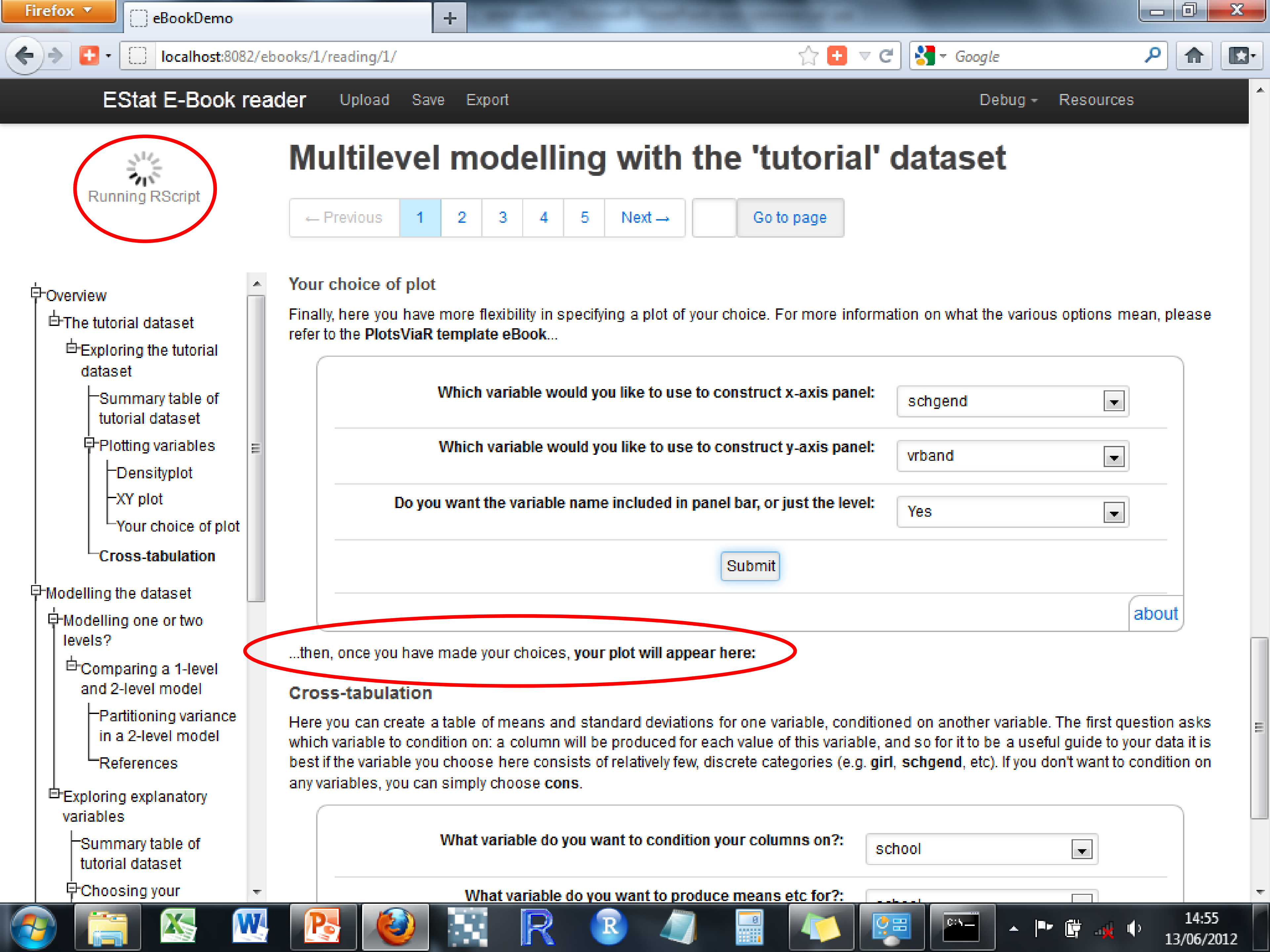
Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Type of plot:
densityplot
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
school
Submit
about
14:50
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
14:54
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
14:54
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Finished
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the **PlotsViaR template eBook**...
**Which variable would you like to use to construct x-axis panel:**
schgend
**Which variable would you like to use to construct y-axis panel:**
vrband
**Do you want the variable name included in panel bar, or just the level:**
Yes
Submit
about
...then, once you have made your choices, **your plot will appear here:**
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. **girl**, **schgend**, etc). If you don't want to condition on any variables, you can simply choose **cons**.
**What variable do you want to condition your columns on?:**
school
**What variable do you want to produce means etc for?:**
14:54
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Running RScript
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
14:55
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Running RScript
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
...then, once you have made your choices, your plot will appear here:
Cross-tabulation
Here you can create a table of means and standard deviations for one variable, conditioned on another variable. The first question asks which variable to condition on: a column will be produced for each value of this variable, and so for it to be a useful guide to your data it is best if the variable you choose here consists of relatively few, discrete categories (e.g. girl, schgend, etc). If you don't want to condition on any variables, you can simply choose cons.
What variable do you want to condition your columns on?:
school
What variable do you want to produce means etc for?:
14:55
13/06/2012

Firefox eBookDemo

localhost:8082/ebooks/1/reading/1/

EStat E-Book reader Upload Save Export Debug Resources

Finished

# Multilevel modelling with the 'tutorial' dataset

← Previous 1 2 3 4 5 Next → Go to page



## Your choice of plot

Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the **PlotsViaR template eBook**...

**Which variable would you like to use to construct x-axis panel:** schgend

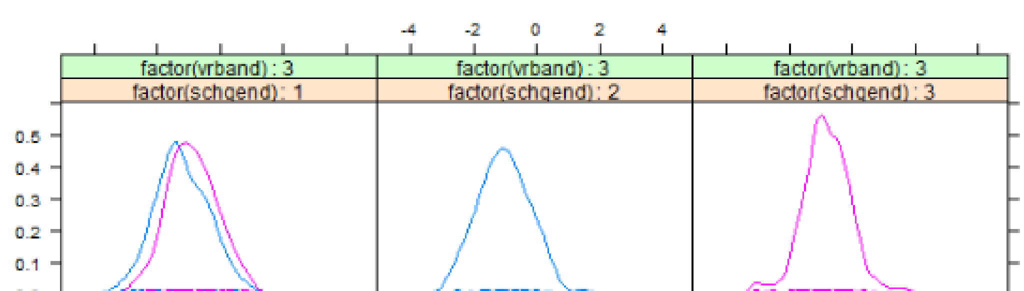
**Which variable would you like to use to construct y-axis panel:** vrband

**Do you want the variable name included in panel bar, or just the level:** Yes

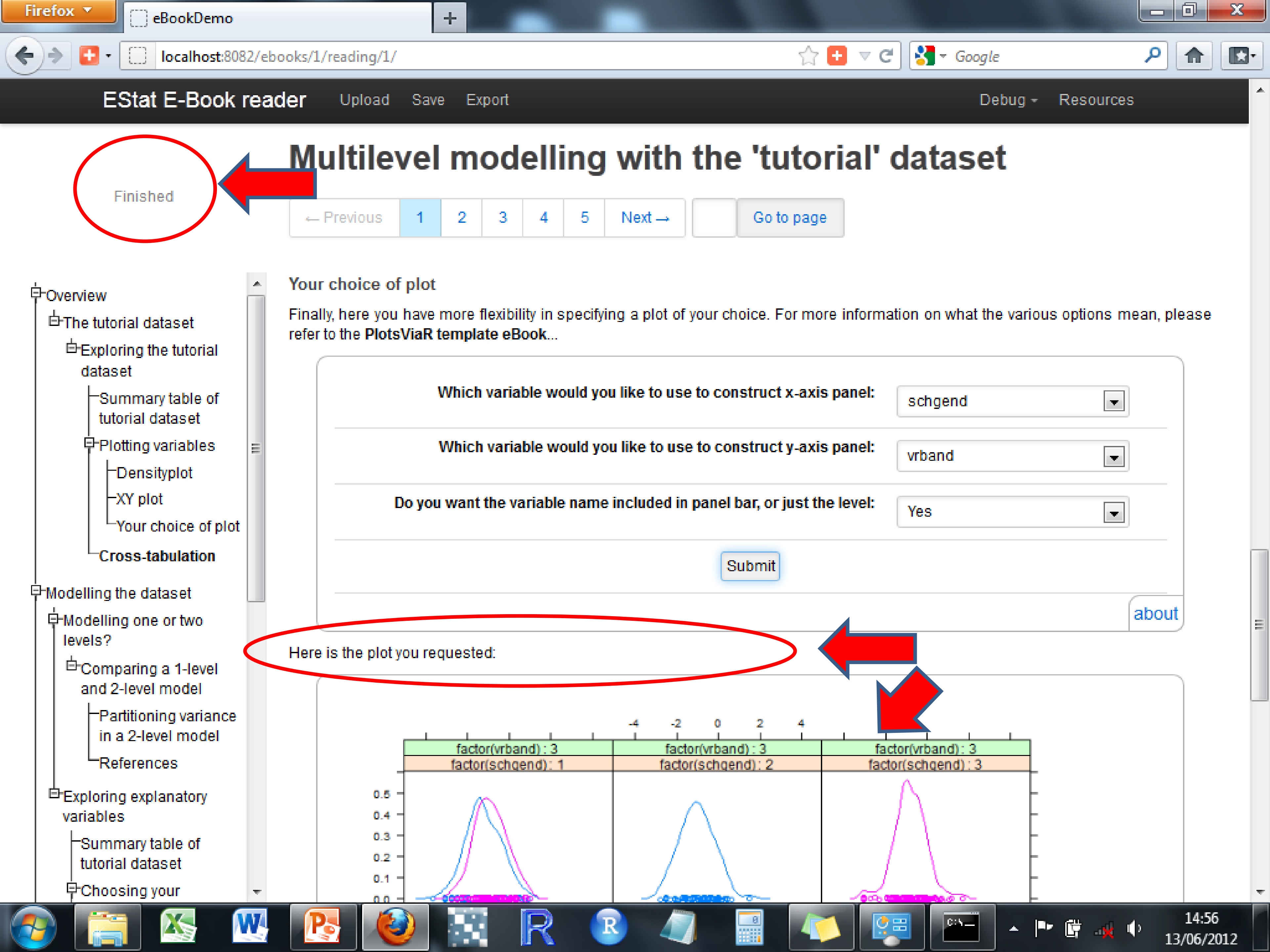
Submit

about

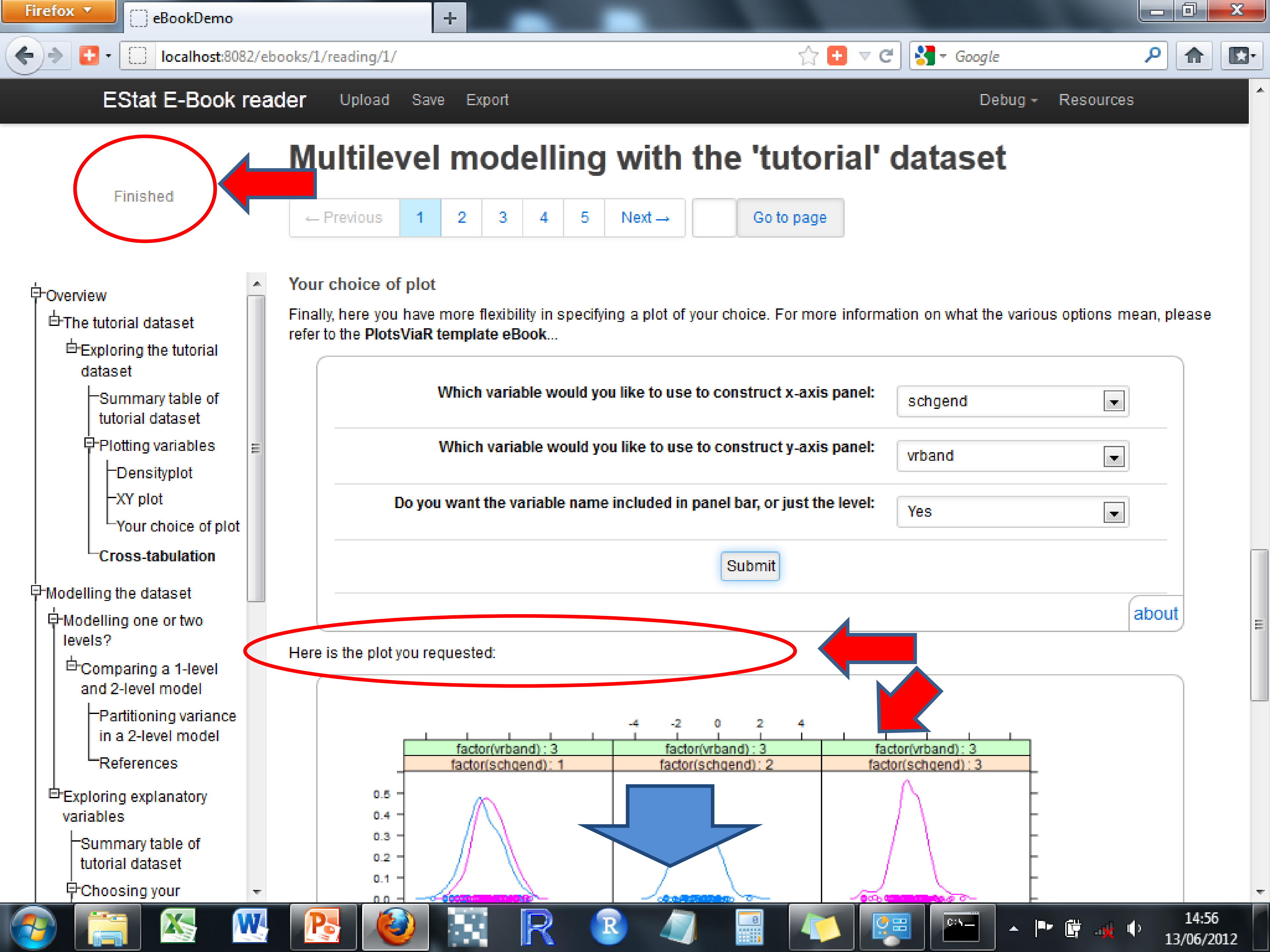
Here is the plot you requested:

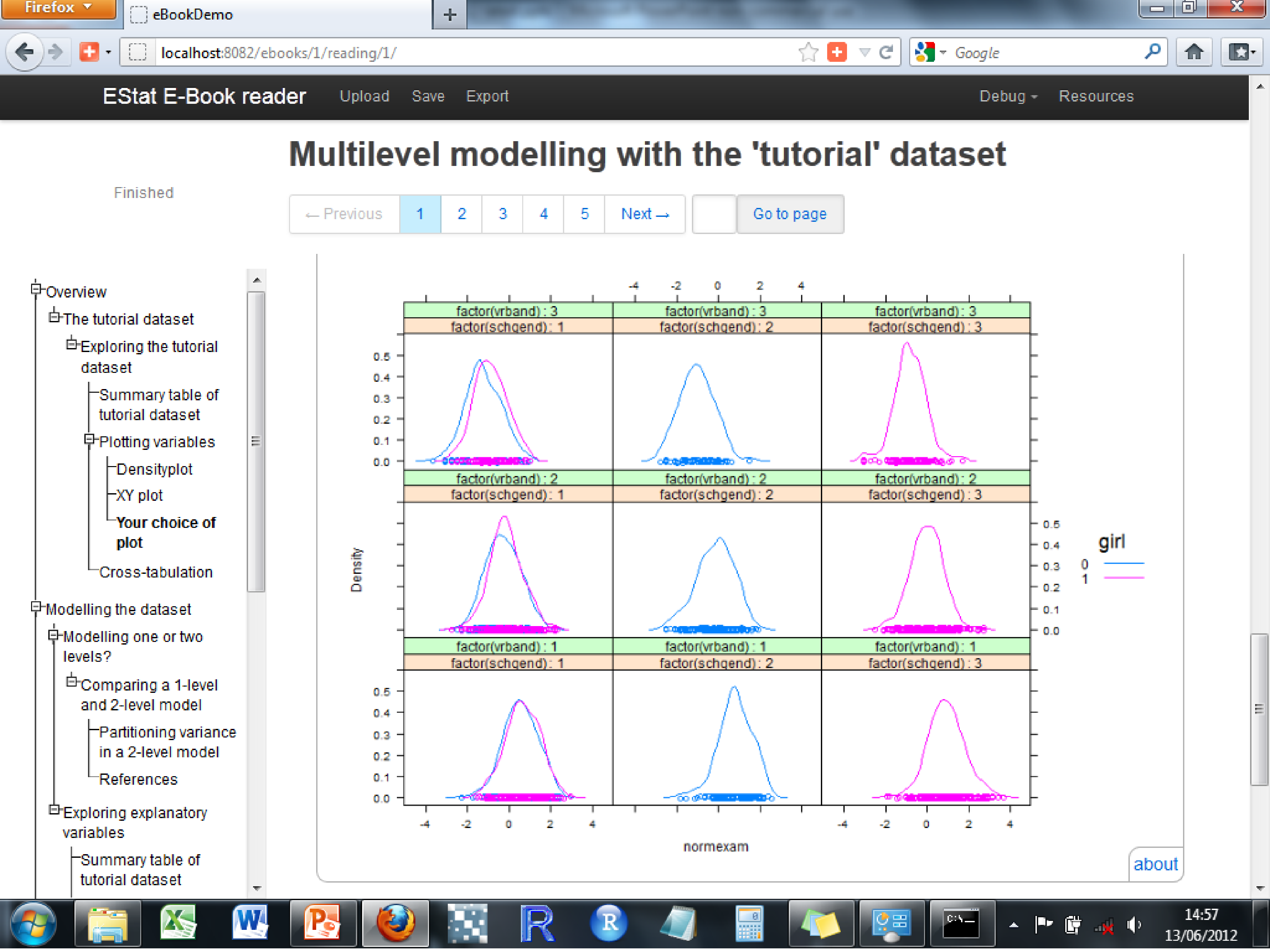



14:56
13/06/2012



Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Finished
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
Here is the plot you requested:
factor(vrband) : 3
factor(schgend) : 1
factor(vrband) : 3
factor(schgend) : 2
factor(vrband) : 3
factor(schgend) : 3
14:56
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Finished
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
Choosing your
Your choice of plot
Finally, here you have more flexibility in specifying a plot of your choice. For more information on what the various options mean, please refer to the PlotsViaR template eBook...
Which variable would you like to use to construct x-axis panel:
schgend
Which variable would you like to use to construct y-axis panel:
vrband
Do you want the variable name included in panel bar, or just the level:
Yes
Submit
about
Here is the plot you requested:
factor(vrband) : 3
factor(schgend) : 1
factor(vrband) : 3
factor(schgend) : 2
factor(vrband) : 3
factor(schgend) : 3
-4
-2
0
2
4
0.5
0.4
0.3
0.2
0.1
0.0
14:56
13/06/2012


eBookDemo
localhost:8082/ebooks/1/reading/1/
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Plotting variables
Densityplot
XY plot
Your choice of plot
Cross-tabulation
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
Summary table of tutorial dataset
factor(vrband) : 3
factor(vrband) : 2
factor(vrband) : 1
factor(schgend) : 1
factor(schgend) : 2
factor(schgend) : 3
Density
normexam
girl
0
1
about
14:57
13/06/2012

# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Stat-JR prompts user for input

Template

+

Dataset

Stat-JR writes commands, etc., to perform requested function

Function performed

Results of function produced

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

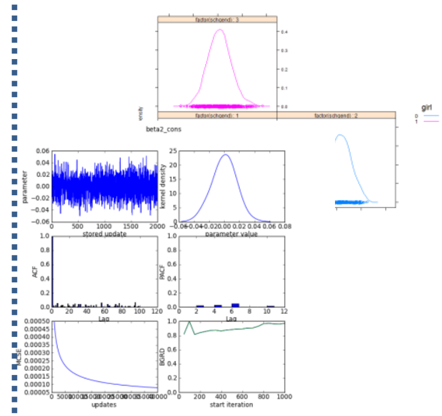
Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2)\\
\mu_i &= \beta_0 \mathrm{cons}_i\\
\beta_0 &\propto 1\\
\tau &\sim \Gamma(0.001, 0.001)\\
\sigma^2 &= 1/\tau
\end{aligned}
$$

*Scripts* *Macros* *Equations* *Point & click instructions*

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Template

+

Dataset

Stat-JR prompts user for input

Stat-JR writes commands, etc., to perform requested function

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

*Scripts* *Macros* *Equations* *Point & click instructions*

$$
\begin{aligned}
\text{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \text{cons}_i \\
\beta_0 &\propto 1 \\
\tau &\sim \Gamma(0.001, 0.001) \\
\sigma^2 &= 1/\tau
\end{aligned}
$$

Function performed

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

SPSS AN IBM® COMPANY

MLwiN

R

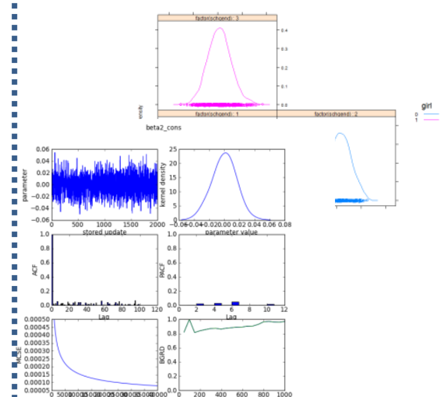
BUGS

STATA

Results of function produced

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Stat-JR prompts user for input

Template

+

Dataset

Stat-JR writes commands, etc., to perform requested function

Function performed

Results of function produced

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

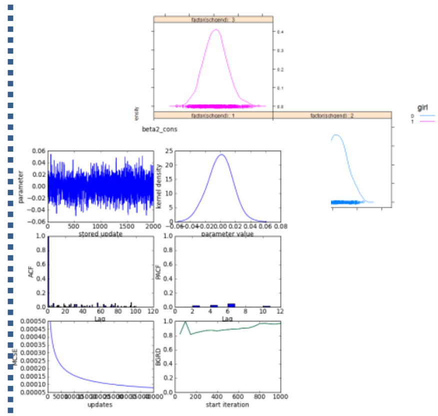
Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

*Scripts* *Macros* *Equations* *Point & click instructions*

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \mathrm{cons}_i \\
\beta_0 &\propto 1 \\
\tau &\sim \Gamma(0.001, 0.001) \\
\sigma^2 &= 1/\tau
\end{aligned}
$$

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



# Multilevel modelling with the 'tutorial' dataset

Finished
← Previous 1 2 3 4 5 Next → Go to page



## Modelling the dataset

### Modelling one or two levels?

If a dataset has a hierarchical or clustered structure, such as students nested within schools, and an analysis neglects to model that structure appropriately, it can compromise the conclusions drawn from it in very important ways. Whilst there are a variety of ways to analyse such a structure, multilevel modelling is an efficient and informative way to do so (e.g. see: Goldstein, 2011; Steele, 2008).

In some situations, however, a given hierarchy may be irrelevant: in a hypothetical example, if we had 4059 dogs nested within 65 towns of habitation, then (for argument's sake) it is perhaps unlikely that the inferences we draw from a model exploring the association between 'speed' and 'breed' would change if we took account of the town in which the dog lived.

In other situations such a structure may matter much more, though: e.g. the exam performance of a pupil may, on average, be more likely to be similar to the exam performance of another pupil in the same school than the exam performance of another pupil in a different school, conceivably due to differences between schools in their pupil selection procedures, socioeconomic status of the catchment area, teaching methods and personnel, and so on. So here we may be missing a very important piece of information, violating the model's assumptions, if we simply ignore the fact that some of the pupils in our sample go to the same school, whereas others do not.

#### Comparing a 1-level and 2-level model

So, if we want to investigate, for instance, the association between exam scores at age 16 (**normexam**, in this example) and those gained earlier at age 11 (**standlrt**), how can we tell whether it's important for an analysis to take into account a multilevel structure of students nested within schools?

Well, one way we can investigate that is by fitting a single-level (or 1-level) model, ignoring the fact that pupils went to certain schools, and then compare the results of that analysis to a 2-level model which allows for random effects for both students and schools.

We'll use Stat-JR's in-house estimation engine, **eSTAT**, to fit both models. eSTAT uses **MCMC** estimation, and so, for each, we'll run 1 chain for a burn-in of 1,000 and then 2,000 main iterations (otherwise, we'll choose 1 as the value of the random seed, and of the thinning factor too).

Below you can choose the **explanatory variables** you'd like to include in the two models (**normexam** has been pre-selected as the **response variable**); to vary **only** the random effects structure between the two models, choose the **same** explanatory variables for each. For example, if you chose **cons** and **standlrt** for each model, this would fit the model(s) we referred to just above.

# Multilevel modelling with the 'tutorial' dataset





# Modelling the dataset

## Modelling one or two levels?

If a dataset has a hierarchical or clustered structure, such as students nested within schools, and an analysis neglects to model that structure appropriately, it can compromise the conclusions drawn from it in very important ways. Whilst there are a variety of ways to analyse such a structure, multilevel modelling is an efficient and informative way to do so (e.g. see: Goldstein, 2011; Steele, 2008).

In some situations, however, a given hierarchy may be irrelevant: in a hypothetical example, if we had 4059 dogs nested within 65 towns of habitation, then (for argument's sake) it is perhaps unlikely that the inferences we draw from a model exploring the association between 'speed' and 'breed' would change if we took account of the town in which the dog lived.

In other situations such a structure may matter much more, though: e.g. the exam performance of a pupil may, on average, be more likely to be similar to the exam performance of another pupil in the same school than the exam performance of another pupil in a different school, conceivably due to differences between schools in their pupil selection procedures, socioeconomic status of the catchment area, teaching methods and personnel, and so on. So here we may be missing a very important piece of information, violating the model's assumptions, if we simply ignore the fact that some of the pupils in our sample go to the same school, whereas others do not.

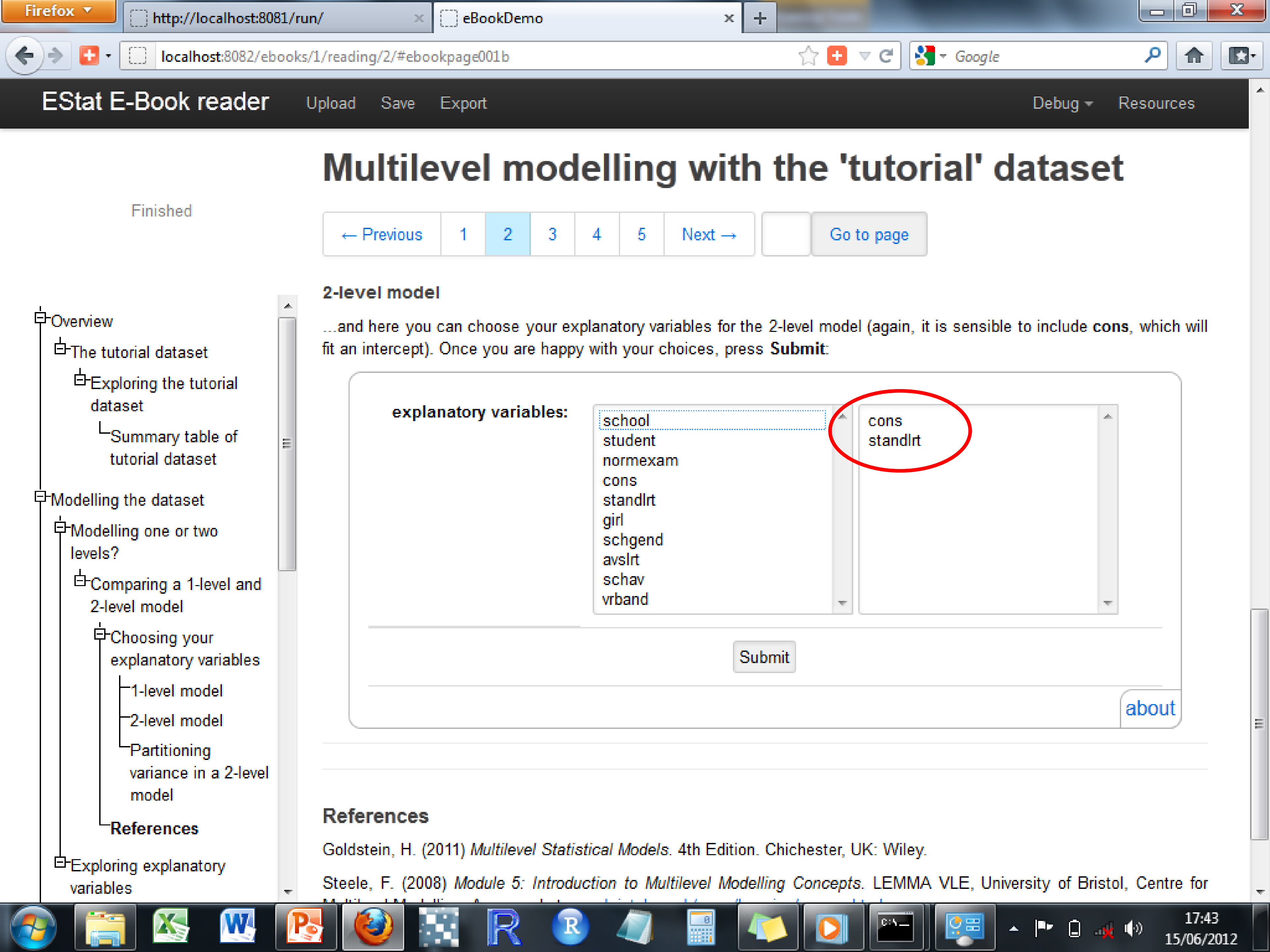
### Comparing a 1-level and 2-level model

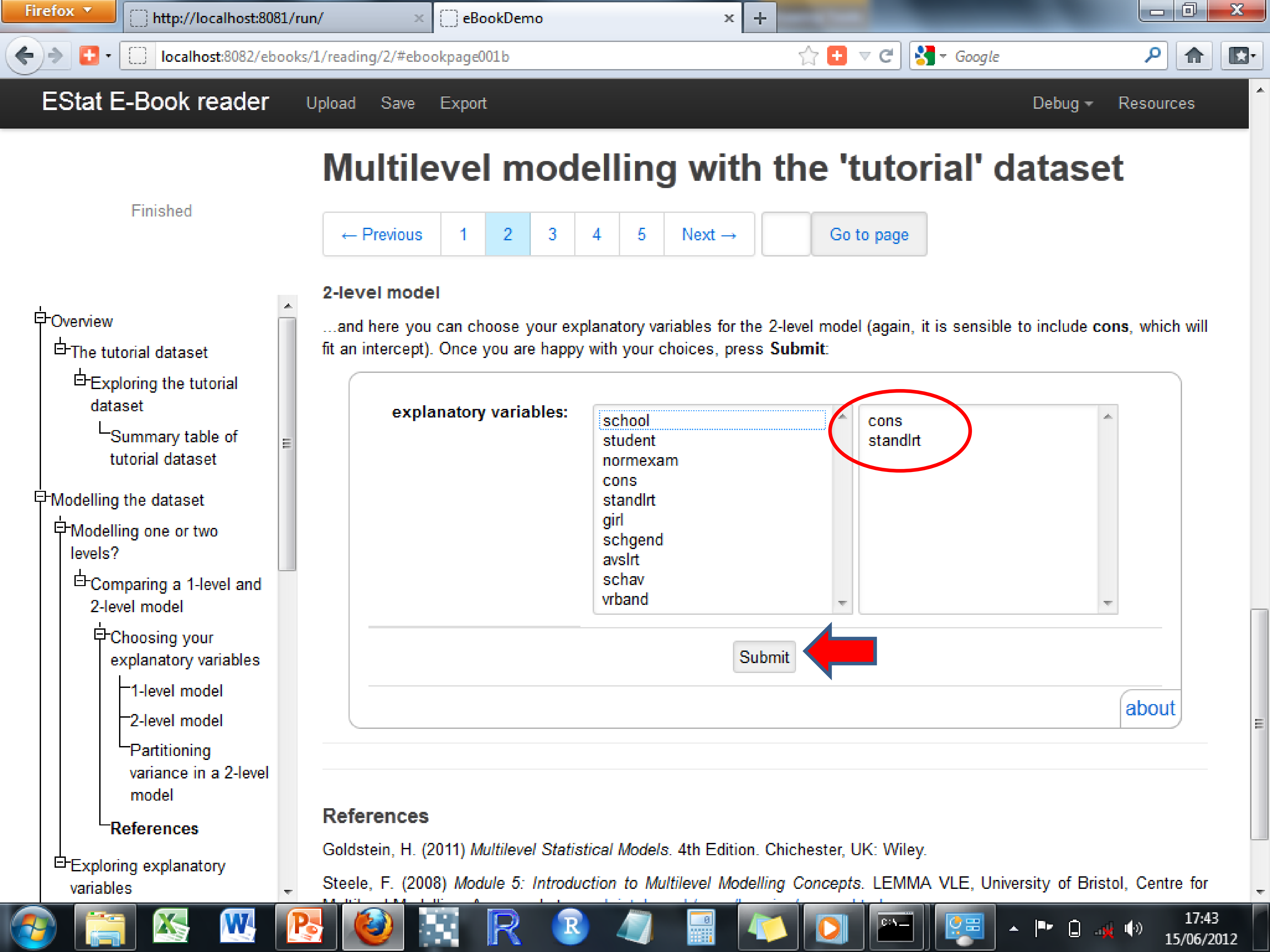
So, if we want to investigate, for instance, the association between exam scores at age 16 (**normexam**, in this example) and those gained earlier at age 11 (**standlrt**), how can we tell whether it's important for an analysis to take into account a multilevel structure of students nested within schools?

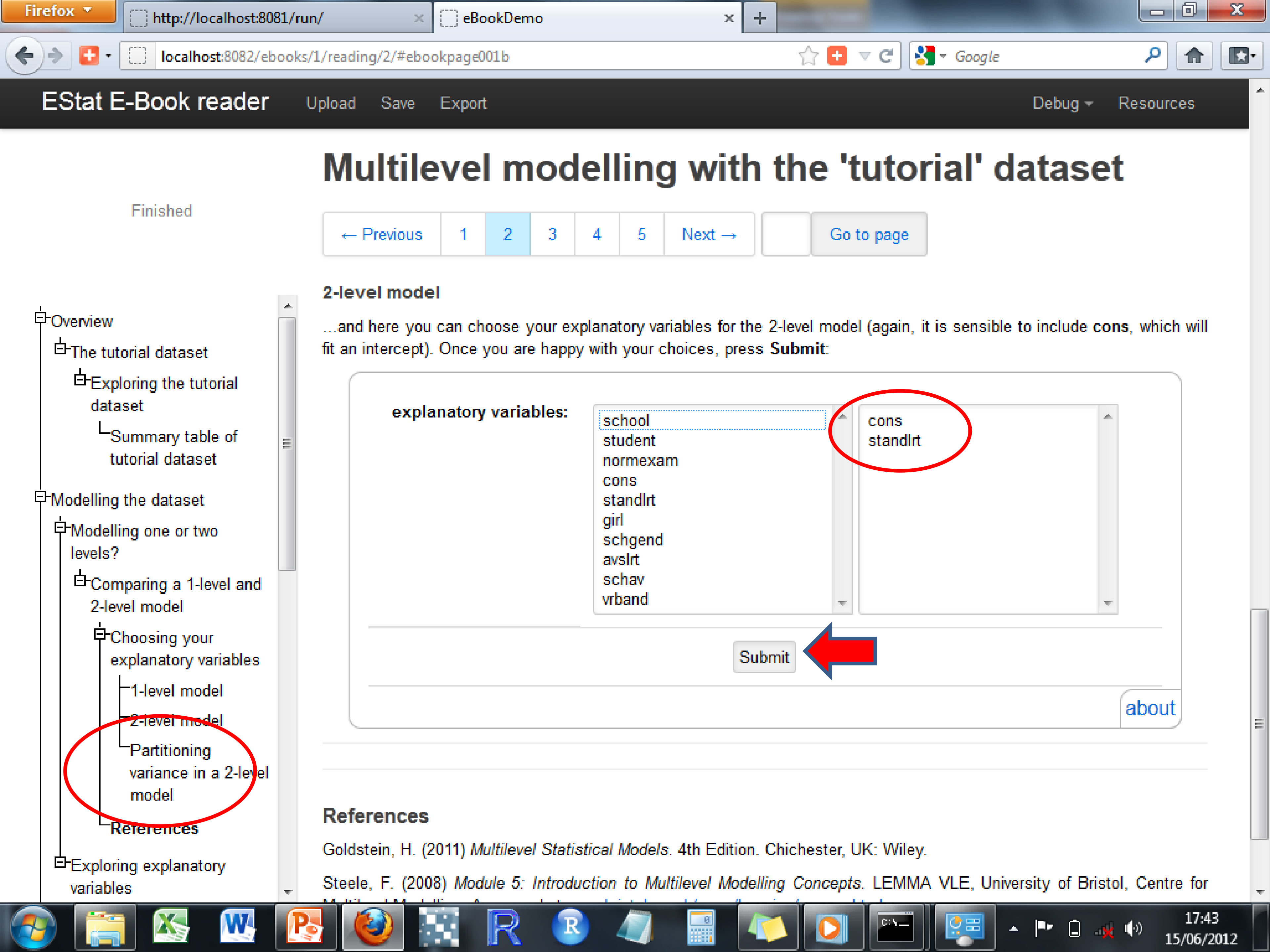
Well, one way we can investigate that is by fitting a single-level (or 1-level) model, ignoring the fact that pupils went to certain schools, and then compare the results of that analysis to a 2-level model which allows for random effects for both students and schools.

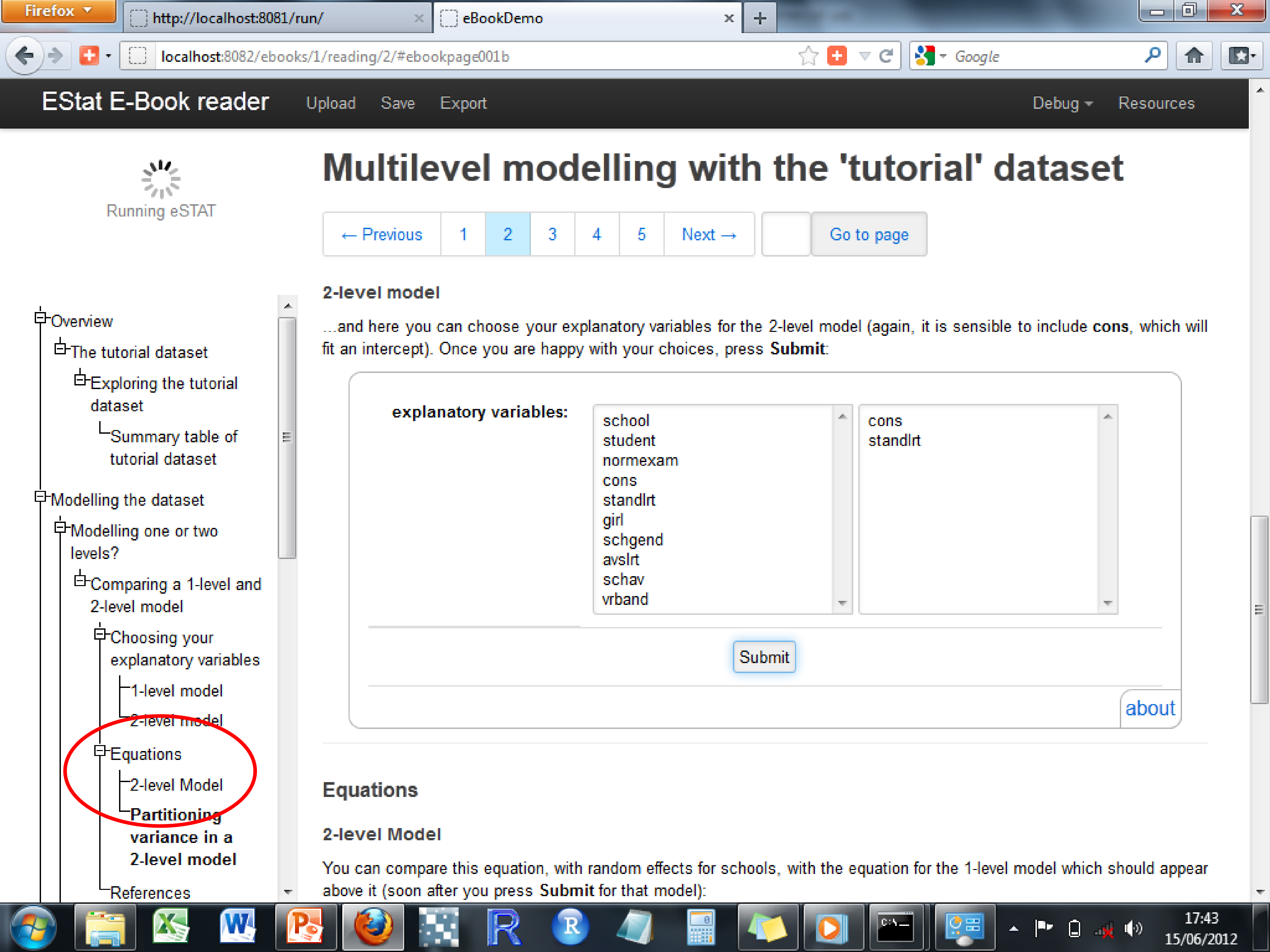
We'll use Stat-JR's in-house estimation engine, **eSTAT**, to fit both models. eSTAT uses **MCMC** estimation, and so, for each, we'll run 1 chain for a burn-in of 1,000 and then 2,000 mai otherwise, we'll choose 1 as the value of the random seed, and of the thinning factor too).

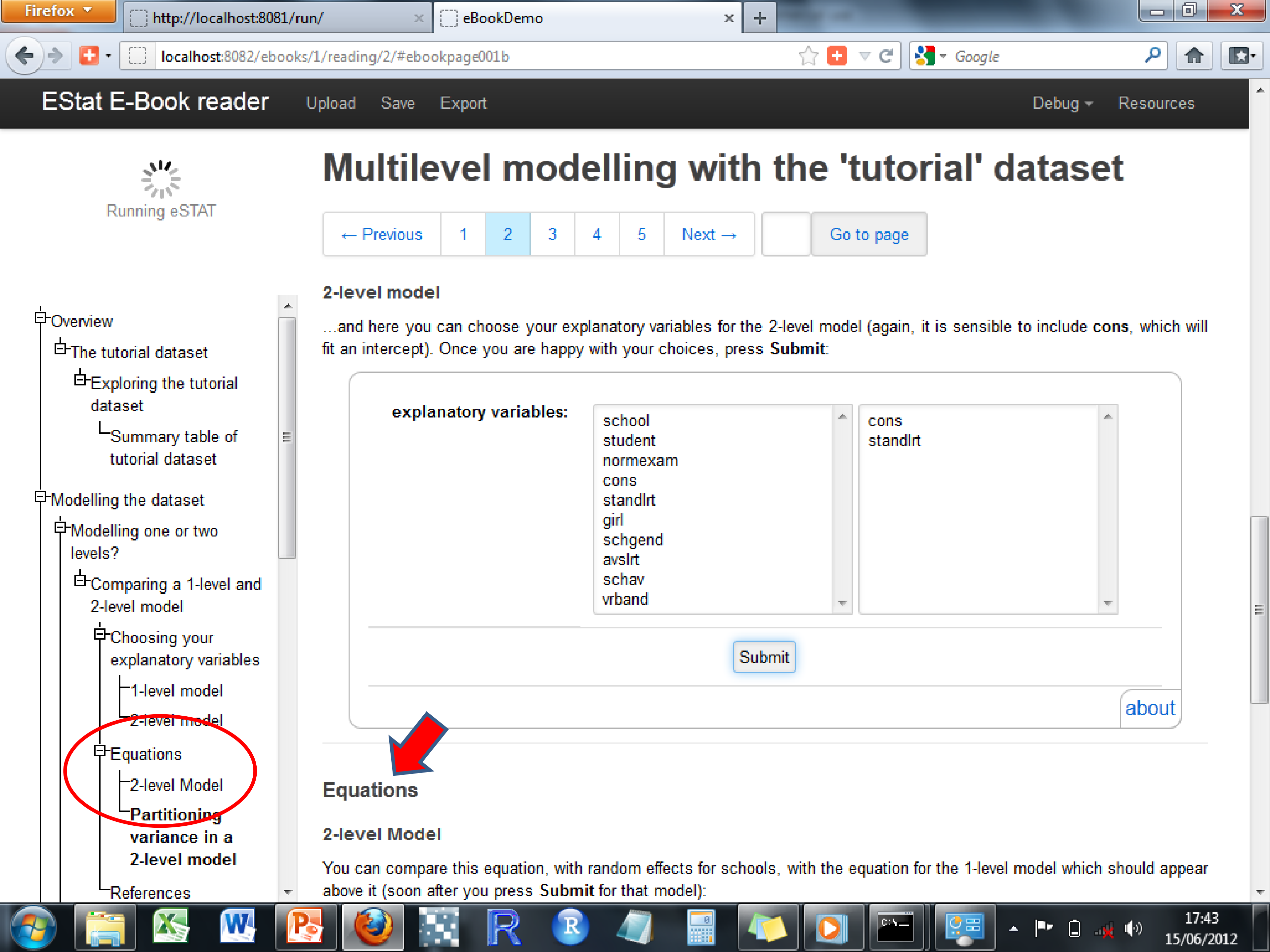
Below you can choose the **explanatory variable** include in the two models (**normexam** has been pre-selected as the **response variable**); to vary **only** the random effects stru ture between the two models, choose the **same** explanatory variables for each. For example, if you chose **cons** and **standlrt** for each model, this would fit the model(s) we referred to just above.

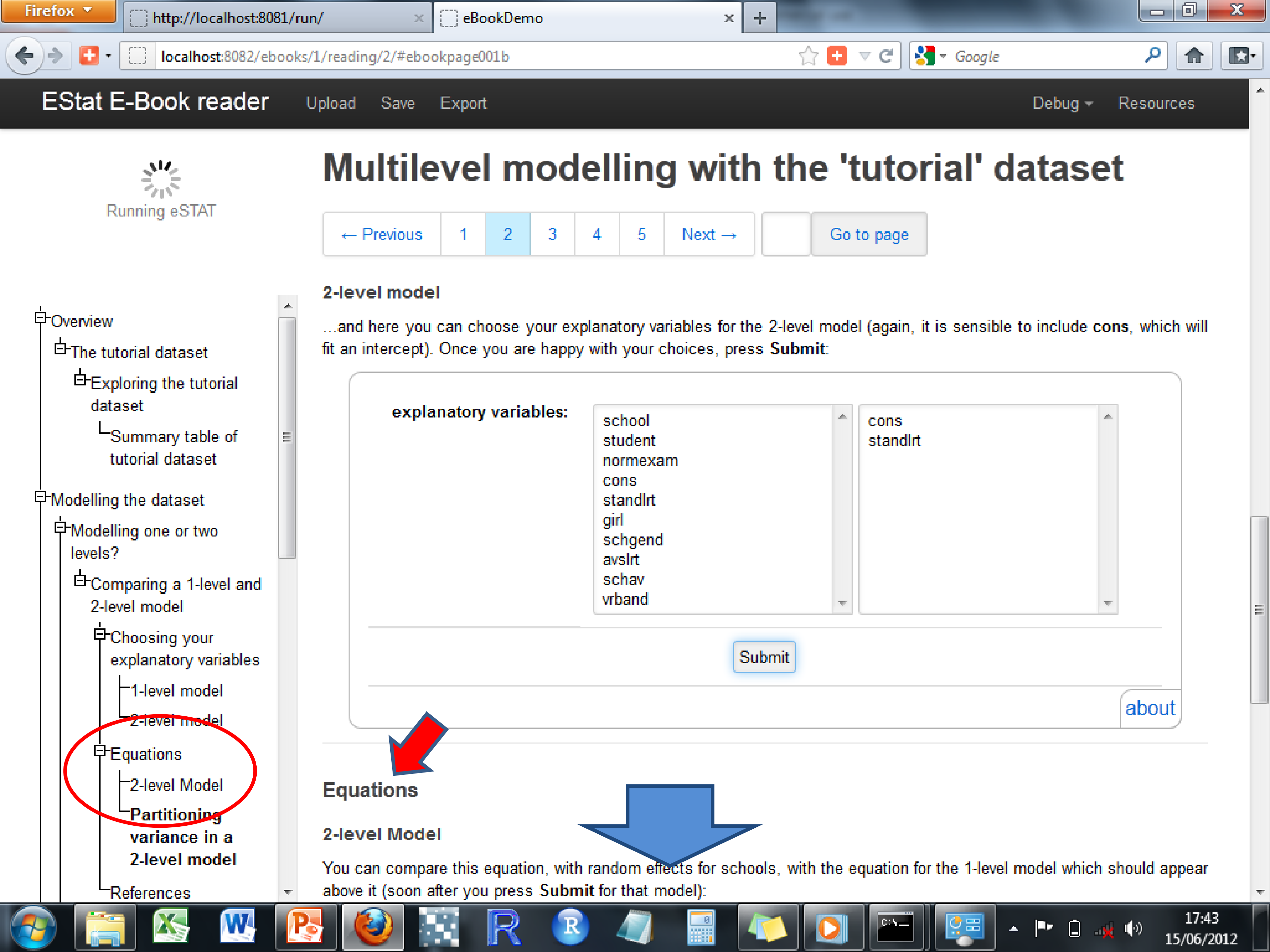

Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include cons, which will fit an intercept). Once you are happy with your choices, press Submit:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
References
Goldstein, H. (2011) Multilevel Statistical Models. 4th Edition. Chichester, UK: Wiley.
Steele, F. (2008) Module 5: Introduction to Multilevel Modelling Concepts. LEMMA VLE, University of Bristol, Centre for
17:43
15/06/2012


Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include cons, which will fit an intercept). Once you are happy with your choices, press Submit:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
References
Goldstein, H. (2011) Multilevel Statistical Models. 4th Edition. Chichester, UK: Wiley.
Steele, F. (2008) Module 5: Introduction to Multilevel Modelling Concepts. LEMMA VLE, University of Bristol, Centre for
17:43
15/06/2012


Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Partitioning variance in a 2-level model
References
Exploring explanatory variables
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include cons, which will fit an intercept). Once you are happy with your choices, press Submit:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
References
Goldstein, H. (2011) Multilevel Statistical Models. 4th Edition. Chichester, UK: Wiley.
Steele, F. (2008) Module 5: Introduction to Multilevel Modelling Concepts. LEMMA VLE, University of Bristol, Centre for
17:43
15/06/2012


Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Running eSTAT
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Equations
2-level Model
Partitioning variance in a 2-level model
References
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include **cons**, which will fit an intercept). Once you are happy with your choices, press **Submit**:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
Equations
2-level Model
You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press **Submit** for that model):
17:43
15/06/2012


Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Running eSTAT
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Equations
2-level Model
Partitioning variance in a 2-level model
References
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include cons, which will fit an intercept). Once you are happy with your choices, press Submit:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
Equations
2-level Model
You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press Submit for that model):
17:43
15/06/2012


Firefox
http://localhost:8081/run/
eBookDemo
localhost:8082/ebooks/1/reading/2/#ebookpage001b
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Running eSTAT
Multilevel modelling with the 'tutorial' dataset
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
Modelling the dataset
Modelling one or two levels?
Comparing a 1-level and 2-level model
Choosing your explanatory variables
1-level model
2-level model
Equations
2-level Model
Partitioning variance in a 2-level model
References
2-level model
...and here you can choose your explanatory variables for the 2-level model (again, it is sensible to include cons, which will fit an intercept). Once you are happy with your choices, press Submit:
explanatory variables:
school
student
normexam
cons
standlrt
girl
schgend
avslrt
schav
vrband
cons
standlrt
Submit
about
Equations
2-level Model
You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press Submit for that model):
17:43
15/06/2012

# Multilevel modelling with the 'tutorial' dataset





## Equations

### 2-level Model

You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press **Submit** for that model):

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \mathrm{cons}_i + \beta_1 \mathrm{standlrt}_i + u_{\mathrm{school}[i]} \\
u_{\mathrm{school}[i]} &\sim \mathrm{N}(0, \sigma_u^2) \\
\beta_0 &\propto 1 \\
\beta_1 &\propto 1 \\
\tau &\sim \Gamma(0.001, 0.001) \\
\sigma^2 &= 1/\tau \\
\tau_u &\sim \Gamma(0.001, 0.001) \\
\sigma_u^2 &= 1/\tau_u
\end{aligned}
$$

about

## Results

Firefox | http://localhost:8081/run/ | eBookDemo

localhost:8082/ebooks/1/reading/2/#ebookpage001b

EStat E-Book reader   Upload   Save   Export   Debug   Resources

# Multilevel modelling with the 'tutorial' dataset

Finished

← Previous | 1 | 2 | 3 | 4 | 5 | Next → | Go to page



## Equations

### 2-level Model

You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press **Submit** for that model):

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \mathrm{cons}_i + \beta_1 \mathrm{standlrt}_i + u_{\mathrm{school}[i]} \\
u_{\mathrm{school}[i]} &\sim \mathrm{N}(0, \sigma_u^2) \\
\beta_0 &\propto 1 \\
\beta_1 &\propto 1 \\
\tau &\sim \Gamma(0.001, 0.001) \\
\sigma^2 &= 1/\tau \\
\tau_u &\sim \Gamma(0.001, 0.001) \\
\sigma_u^2 &= 1/\tau_u
\end{aligned}
$$

about

## Results

17:47 15/06/2012

# Multilevel modelling with the 'tutorial' dataset





## Equations

### 2-level Model

You can compare this equation, with random effects for schools, with the equation for the 1-level model which should appear above it (soon after you press **Submit** for that model):

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2)\\
\mu_i &= \beta_0 \mathrm{cons}_i + \beta_1 \mathrm{standlrt}_i + u_{\mathrm{school}[i]}\\
u_{\mathrm{school}[i]} &\sim \mathrm{N}(0, \sigma_u^2)\\
\beta_0 &\propto 1\\
\beta_1 &\propto 1\\
\tau &\sim \Gamma(0.001, 0.001)\\
\sigma^2 &= 1/\tau\\
\tau_u &\sim \Gamma(0.001, 0.001)\\
\sigma_u^2 &= 1/\tau_u
\end{aligned}
$$

about

## Results

# Multilevel modelling with the 'tutorial' dataset

← Previous 1 2 3 4 5 Next → Go to page



## Results

### Deviance statistic and DIC diagnostic

Below you'll see the DIC diagnostic, along with the parameters from which it is derived, for both models *(with apologies for the number of decimal places!)*

So, according to the DIC, which model appears to be better?

| Statistic | Description | 1-level model | 2-level model |
|---|---|---|---|
| $\bar{D}$ | Average deviance across all (post-burnin) iterations. | 9763.49 | 9209.21 |
| $D(\bar{\theta})$ | Deviance at the expected value of the unknown parameters ($\theta$ ). | 9760.51 | 9148.95 |
| $pD$ | The effective number of parameters, computed as $\bar{D}$ - $D(\bar{\theta})$. | 2.98 | 60.25 |
| $DIC$ | Deviance Information Criterion (Spiegelhalter et al, 2002); this is computed as $D(\bar{\theta}) + 2pD$: i.e. it is a measure of goodness of model fit which adjusts for model complexity, allowing models to be compared: **a smaller DIC suggests a better model**. | 9766.47 | 9269.46 |

### Summary of parameter estimates: tables

These tables contain the parameter estimates, providing, for each, values for the following:

(NB: for this, and all other models fitted in this eBook, you can view other outputs, such as MCMC diagnostic plots, via the **Resources** button in the black bar at the top).

- **Mean**: posterior mean estimate;
- **SD**: posterior standard deviation;
- **ESS**: effective sample size.

1-LEVEL MODEL

Firefox eBookDemo

EStat E-Book reader Upload Save Export Debug Resources

# Multilevel modelling with the 'tutorial' dataset

Finished

← Previous 1 2 3 4 5 Next → Go to page



## Results

### Deviance statistic and DIC diagnostic

Below you'll see the DIC diagnostic, along with the parameters from which it is derived, for both models *(with apologies for the number of decimal places!)*

So, according to the DIC, which model appears to be better?

| Statistic | Description | 1-level model | 2-level model |
|---|---|---|---|
| $\bar{D}$ | Average deviance across all (post-burnin) iterations. | 9763.49 | 9209.21 |
| $D(\bar{\theta})$ | Deviance at the expected value of the unknown parameters ($\theta$ ). | 9760.51 | 9148.95 |
| $pD$ | The effective number of parameters, computed as $\bar{D}$ - $D(\bar{\theta})$. | 2.98 | 60.25 |
| $DIC$ | Deviance Information Criterion (Spiegelhalter et al, 2002); this is computed as $D(\bar{\theta}) + 2pD$: i.e. it is a measure of goodness of model fit which adjusts for model complexity, allowing models to be compared: **a smaller DIC suggests a better model**. | 9766.47 | 9269.46 |

### Summary of parameter estimates: tables

These tables contain the parameter estimates, providing, for each, values for the following:

(NB: for this, and all other models fitted in this eBook, you can view other outputs, such as MCMC diagnostic plots, via the **Resources** button in the black bar at the top).

- **Mean**: posterior mean estimate;
- **SD**: posterior standard deviation;
- **ESS**: effective sample size.

1-LEVEL MODEL

13:31 07/01/2013

Firefox eBookDemo

EStat E-Book reader Upload Save Export Debug Resources

# Multilevel modelling with the 'tutorial' dataset

Finished





## Results

### Deviance statistic and DIC diagnostic

Below you'll see the DIC diagnostic, along with the parameters from which it is derived, for both models *(with apologies for the number of decimal places!)*

So, according to the DIC, which model appears to be better?

| Statistic | Description | 1-level model | 2-level model |
|---|---|---|---|
| $\bar{D}$ | Average deviance across all (post-burnin) iterations. | 9763.49 | 9209.21 |
| $D(\bar{\theta})$ | Deviance at the expected value of the unknown parameters ($\theta$ ). | 9760.51 | 9148.95 |
| $pD$ | The effective number of parameters, computed as $\bar{D}$ - $D(\bar{\theta})$. | 2.98 | 60.25 |
| $DIC$ | Deviance Information Criterion (Spiegelhalter et al, 2002); this is computed as $D(\bar{\theta}) + 2pD$: i.e. it is a measure of goodness of model fit which adjusts for model complexity, allowing models to be compared: **a smaller DIC suggests a better model**. | 9766.47 | 9269.46 |

### Summary of parameter estimates: tables

These tables contain the parameter estimates, providing, for each, values for the following:

(NB: for this, and all other models fitted in this eBook, you can view other outputs, such as MCMC diagnostic plots, via the **Resources** button in the black bar at the top).

- **Mean**: posterior mean estimate;
- **SD**: posterior standard deviation;
- **ESS**: effective sample size.

1-LEVEL MODEL

13:31 07/01/2013

Firefox eBookDemo

EStat E-Book reader Upload Save Export Debug Resources

# Multilevel modelling with the 'tutorial' dataset

Finished





## Partitioning variance in a 2-level model

For some multilevel models, such as the 2-level random intercept model you fitted here, it is quite straightforward to calculate how much of the unexplained variance is attributable to each level, a parameter which may be of interest to the researcher.

So, let's pluck a few statistics out of the results tables which appear above, and see how we would do this.

In the 2-level model, the parameter $\sigma^2$ (i.e. 'sigma2', as it appears in the results table of parameter estimates) is the variance attributable to **differences between pupils** *within* **a school**, whereas $\sigma^2_u$ ('sigma2_u') is the variance attributable to **differences** *between* **schools**.

Therefore, to calculate what proportion of residual variance is attributed to level 2 (known as the **Variance Partition Coefficient**: in this instance, the residual variance attributable to differences between schools), we simply divide $\sigma^2_u$ by the total variance, i.e.:

$$\sigma^2_u/(\sigma^2_u+\sigma^2) = \text{Variance Partition Coefficient (VPC)}$$

Here, then, for our 2-level model we have (with rounding):

0.097 / ( 0.097 + 0.567 ) = **0.146**.

So, the proportion of the unexplained variance attributable to differences between schools in the 2-level model you specified is 0.146.

You may find it interesting to see how these parameter estimates change if you run the 2-level model with different explanatory variables.

For example, if you keep *only* **cons** in, and therefore fit what some call a **variance components model** what does the total variance add up to? (Approximately!) Why might that be?

If you add a range of different explanatory variables (in addition to **cons**), how does the proportion of variance attributable to level 2 change? Does the addition of school-level, or pupil-level, explanatory variables have any bearing on this?

## References

Goldstein, H. (2011) *Multilevel Statistical Models*. 4th Edition. Chichester, UK: Wiley.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A. (2002) Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, 64, 583-639.

Steele, F. (2008) *Module 5: Introduction to Multilevel Modelling Concepts*. LEMMA VLE, University of Bristol, Centre for Multilevel Modelling. Accessed at www.bristol.ac.uk/cmm/learning/course.html.



13:31 07/01/2013

Firefox eBookDemo

EStat E-Book reader Upload Save Export Debug Resources

# Multilevel modelling with the 'tutorial' dataset

Finished





### Partitioning variance in a 2-level model

For some multilevel models, such as the 2-level random intercept model you fitted here, it is quite straightforward to calculate how much of the unexplained variance is attributable to each level, a parameter which may be of interest to the researcher.

So, let's pluck a few statistics out of the results tables which appear above, and see how we would do this.

In the 2-level model, the parameter $\sigma^2$ (i.e. 'sigma2', as it appears in the results table of parameter estimates) is the variance attributable to **differences between pupils** *within* **a school**, whereas $\sigma^2_u$ ('sigma2_u') is the variance attributable to **differences** *between* **schools**.

Therefore, to calculate what proportion of residual variance is attributed to level 2 (known as the **Variance Partition Coefficient**: in this instance, the residual variance attributable to differences between schools), we simply divide $\sigma^2_u$ by the total variance, i.e.:

$$\sigma^2_u/(\sigma^2_u + \sigma^2) = \text{Variance Partition Coefficient (VPC)}$$

Here, then, for our 2-level model we have (with rounding):

0.097 / ( 0.097 + 0.567 ) = **0.146**.

So, the proportion of the unexplained variance attributable to differences between schools in the 2-level model you specified is 0.146.

You may find it interesting to see how these parameter estimates change if you run the 2-level model with different explanatory variables.

For example, if you keep *only* **cons** in, and therefore fit what some call a **variance components model** what does the total variance add up to? (Approximately!) Why might that be?

If you add a range of different explanatory variables (in addition to **cons**), how does the proportion of variance attributable to level 2 change? Does the addition of school-level, or pupil-level, explanatory variables have any bearing on this?

### References


Goldstein, H. (2011) *Multilevel Statistical Models*. 4th Edition. Chichester, UK: Wiley.

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., van der Linde, A. (2002) Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, 64, 583-639.

Steele, F. (2008) *Module 5: Introduction to Multilevel Modelling Concepts*. LEMMA VLE, University of Bristol, Centre for Multilevel Modelling. Accessed at www.bristol.ac.uk/cmm/learning/course.html.




13:31
07/01/2013

# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Template

+

Dataset

Stat-JR prompts user for input

Stat-JR writes commands, etc., to perform requested function

Function performed

Results of function produced

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Template
+
Dataset

Stat-JR prompts user for input

Stat-JR writes commands, etc., to perform requested function

Function performed

Results of function produced

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

*Scripts* *Macros* *Equations* *Point & click instructions*

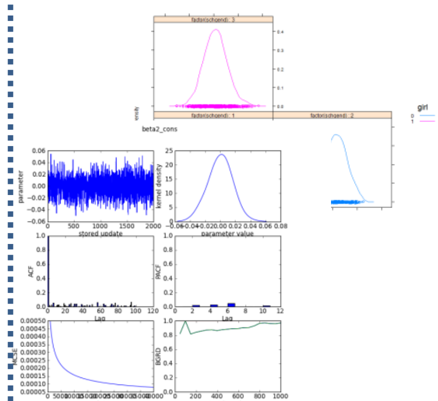
$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2)\\
\mu_i &= \beta_0 \mathrm{cons}_i\\
\beta_0 &\propto 1\\
\tau &\sim \Gamma(0.001, 0.001)\\
\sigma^2 &= 1/\tau
\end{aligned}
$$

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

SPSS AN IBM COMPANY
MLwiN
R
BUGS
STATA

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



(If applicable) results outputted as dataset to be fed back in…

# Stat-JR: to re-cap…

Stat-JR prompts user for input

Template

+

Dataset

Stat-JR writes commands, etc., to perform requested function

Function performed

Results of function produced

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

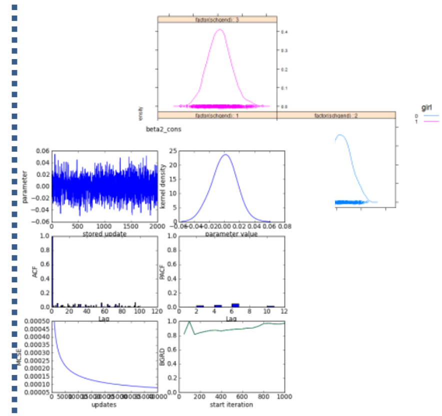
*Scripts* *Macros* *Equations* *Point & click instructions*

$$
\begin{aligned}
\mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2)\\
\mu_i &= \beta_0 \mathrm{cons}_i\\
\beta_0 &\propto 1\\
\tau &\sim \Gamma(0.001, 0.001)\\
\sigma^2 &= 1/\tau
\end{aligned}
$$

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Template

+

Dataset

Stat-JR prompts user for input

Stat-JR writes commands, etc., to perform requested function

Function performed

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

Results of function produced

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from **Fi**

*Scripts* *Macros* *Equations* *Point & click instructions*

$\text{normexam} \sim \mathrm{N}(\mu, \sigma^2)$

$\beta$

$, 0.001)$

$\sigma^2 = 1/\tau$

SPSS AN IBM COMPANY

MLwiN

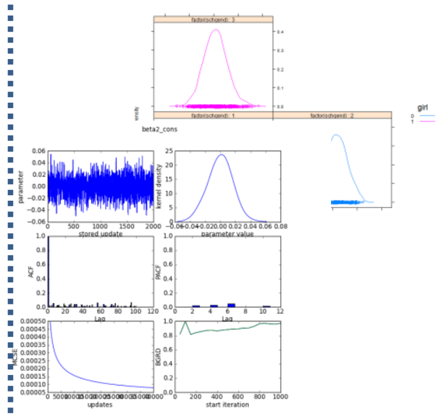
R

BUGS

STATA

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*



# Stat-JR: to re-cap…

(If applicable) results outputted as dataset to be fed back in…

Template + Dataset → Stat-JR prompts user for input → Stat-JR writes commands, etc., to perform requested function → Function performed → Results of function produced

(If applicable) external software opened, run, then closed, with results returned to Stat-JR.

```
myModel<- glm(normexam~
Summary(myModel)
plot(myModel,1)
```

Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from Fi

*Scripts* *Macros* *Equations* *Point & click instructions*

$$\mathrm{normexam} \sim \mathrm{N}(\mu, \sigma^2)$$

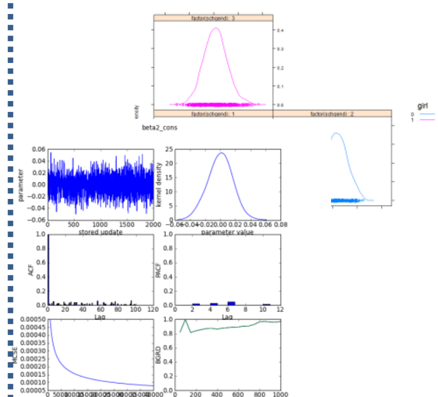
$$\beta$$

$$, 0.001)$$

$$\sigma^2 = 1/\tau$$

**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594

*Results tables* *Charts*

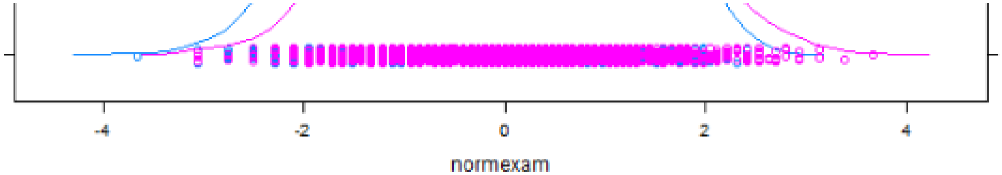


# PlotsViaR Template documentation

Finished








about

### R script

And now looking at the R script, you can see that **groups** is now listed as an argument in the **densityplot** function to indicate that we're grouping the data by a particular variable (the one you chose!) - you can see that the title (and the position) of the legend has also been specified via **auto.key** (which you could naturally change by directly editing the script and re-running in R).

```
library(foreign)
mydata<-read.dta("datafile.dta")
summary(mydata)
PACKages<-as.character(data.frame(installed.packages())$Package)

test<-("lattice" %in% PACKages)
if (!test){
        install.packages("lattice",repos="http://cran.r-project.org")
}
library(lattice)

png("Plot1.png",width=733,height=550)
densityplot(~normexam,groups=girl,auto.key=list(space='right',title="girl"),data=mydata)
dev.off()
```

eBookDemo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Staff Mailstore | SquirrelMail | Bristol University | Centre for Multilev... | eBookDemo

localhost:8082/ebooks/3/reading/2/#page2

gruzin wordsworth ballet

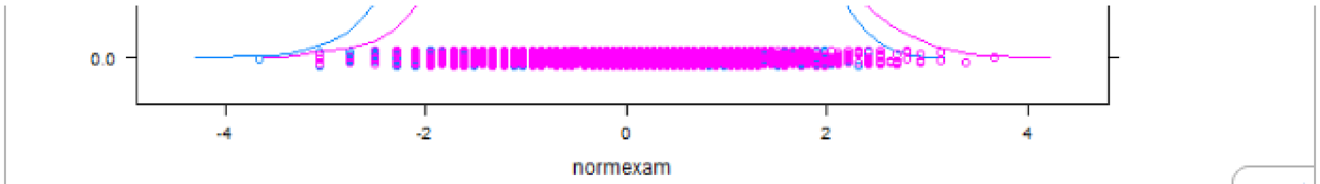
EStat E-Book reader   Upload   Save   Export   Debug   Resources

# PlotsViaR Template documentation

Finished

← Previous | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 15 | 16 | Next → | Go to page






about

### R script

And now looking at the R script, you can see that **groups** is now listed as an argument in the **densityplot** function to indicate that we're grouping the data by a particular variable (the one you chose!) - you can see that the title (and the position) of the legend has also been specified via **auto.key** (which you could naturally change by directly editing the script and re-running in R).

```
library(foreign)
mydata<-read.dta("datafile.dta")
summary(mydata)
PACKages<-as.character(data.frame(installed.packages())$Package)

test<-("lattice" %in% PACKages)
if (!test){
        install.packages("lattice",repos="http://cran.r-project.org")
}
library(lattice)

png("Plot1.png",width=733,height=550)
densityplot(~normexam,groups=girl,auto.key=list(space='right',title="girl"),data=mydata)
dev.off()
```

13:51
31/08/2012

eBookDemo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Staff Mailstore | SquirrelMail | Bristol University | Centre for Multilev... | eBookDemo

localhost:8082/ebooks/3/reading/2/#page2

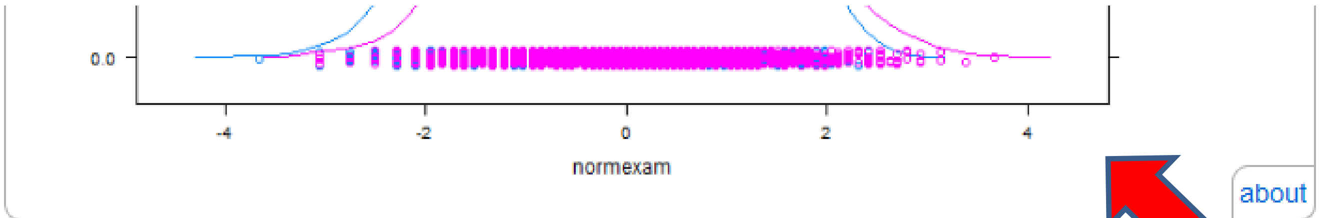
EStat E-Book reader  Upload  Save  Export  Debug  Resources

# PlotsViaR Template documentation

Finished








about

### R script

And now looking at the R script, you can see that **groups** is now listed as an argument in the **densityplot** function to indicate that we're grouping the data by a particular variable (the one you chose!) - you can see that the title (and the position) of the legend has also been specified via **auto.key** (which you could naturally change by directly editing the script and re-running in R).

```
library(foreign)
mydata<-read.dta("datafile.dta")
summary(mydata)
PACKages<-as.character(data.frame(installed.packages())$Package)

test<-("lattice" %in% PACKages)
if (!test){
        install.packages("lattice",repos="http://cran.r-project.org")
}
library(lattice)

png("Plot1.png",width=733,height=550)
densityplot(~normexam,groups=girl,auto.key=list(space='right',title="girl"),data=mydata)
dev.off()
```

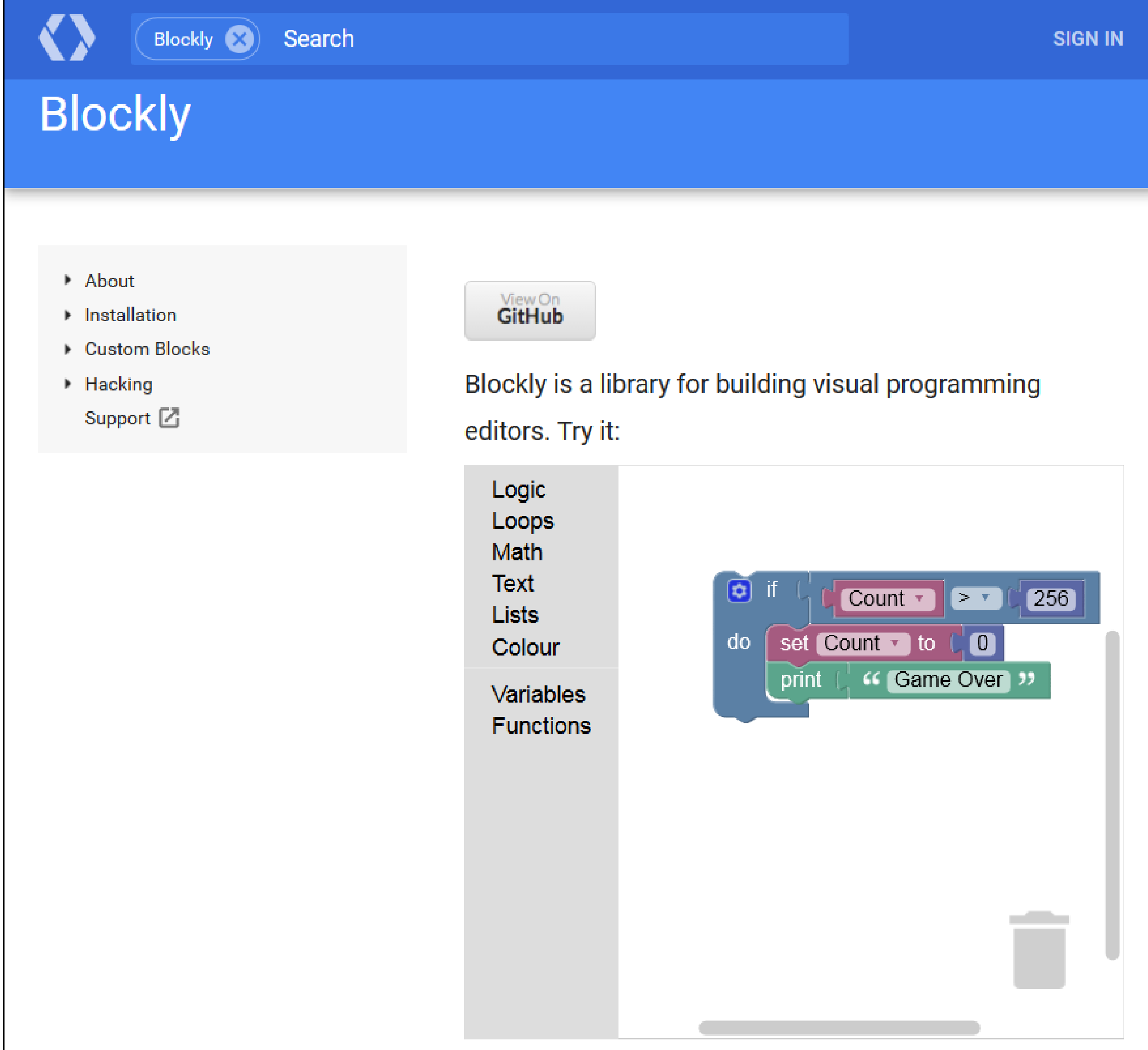
13:51
31/08/2012

# Further developments to Stat-JR as part of current grant

1. **Point-and-click menu-driven** interface (TREE)
2. **eBook** interface (DEEP)
3. **Command line** interface (runStatJR)

# Further developments to Stat-JR as part of current grant

1. **Point-and-click menu-driven** interface (TREE)
2. **eBook** interface (DEEP)
3. **Command line** interface (runStatJR)
4. **Workflow** system


Blockly
Search
SIGN IN
Blockly
About
Installation
Custom Blocks
Hacking
Support
View On
GitHub
Blockly is a library for building visual programming editors. Try it:
Logic
Loops
Math
Text
Lists
Colour
Variables
Functions
if
Count
>
256
do
set
Count
to
0
print
“
Game Over
”

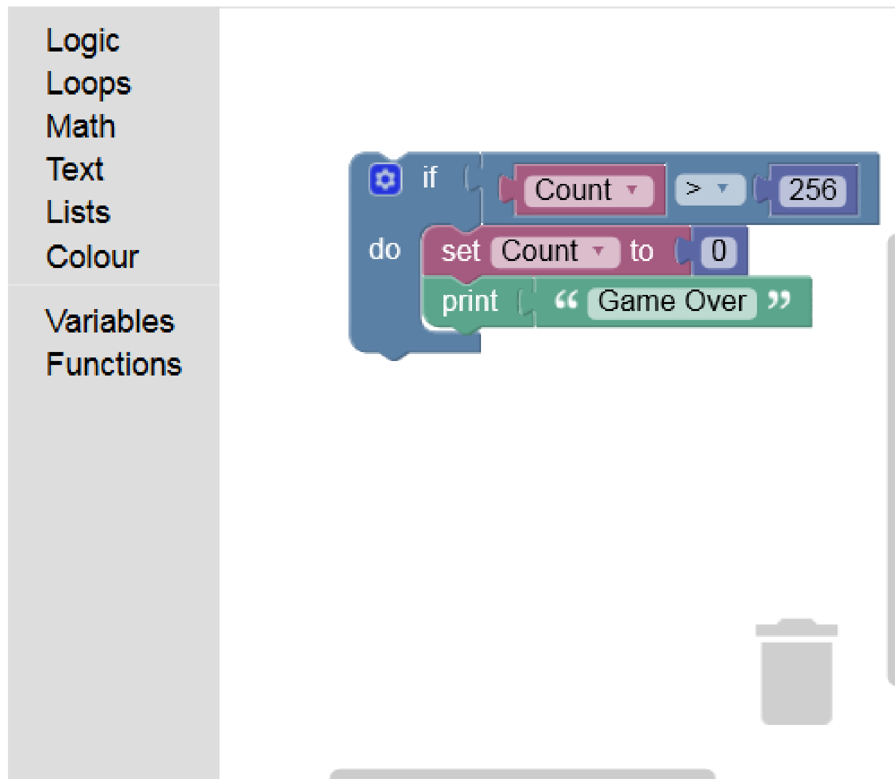
Blockly Search SIGN IN

# Blockly

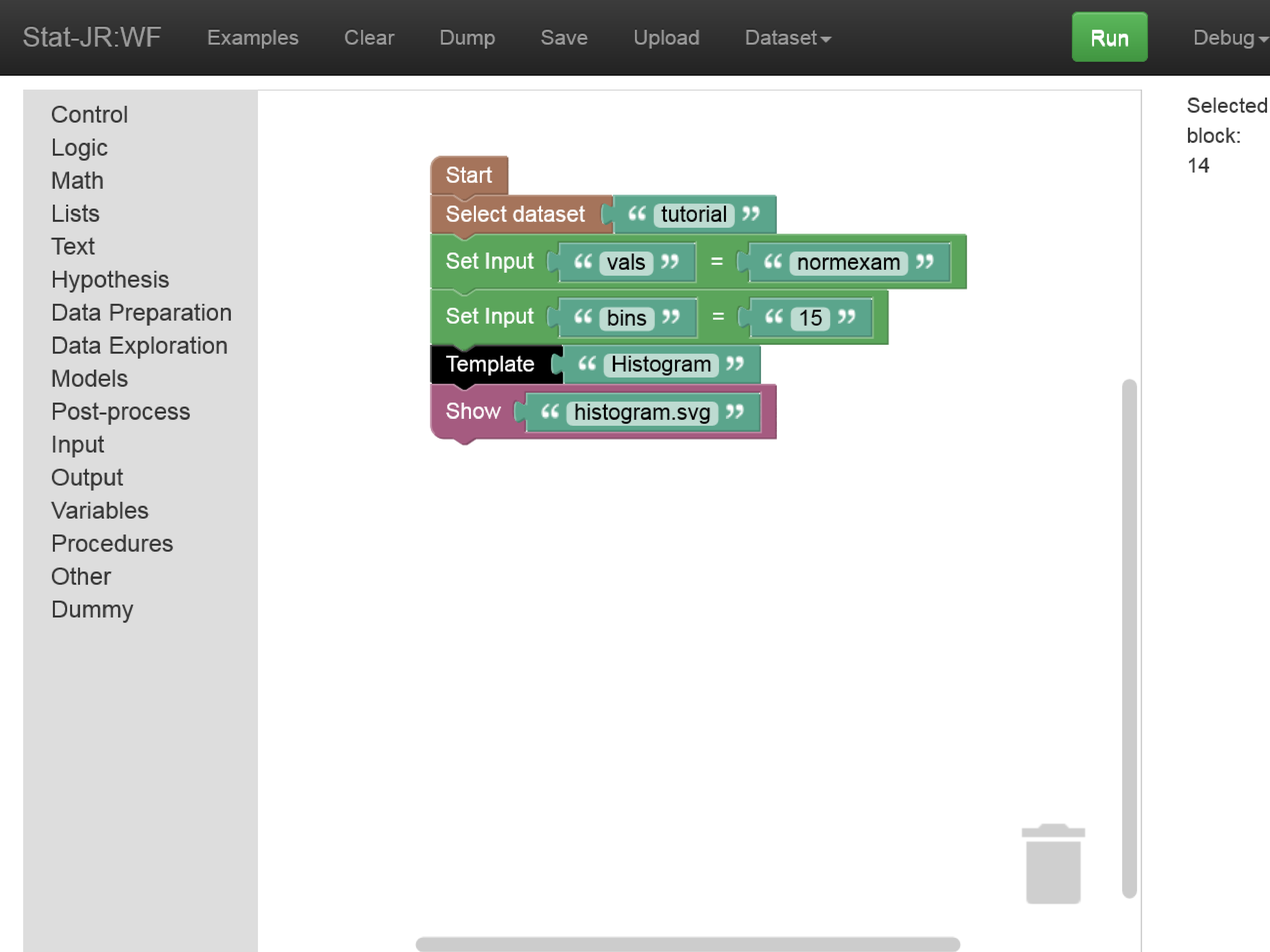
- About
- Installation
- Custom Blocks
- Hacking
- Support

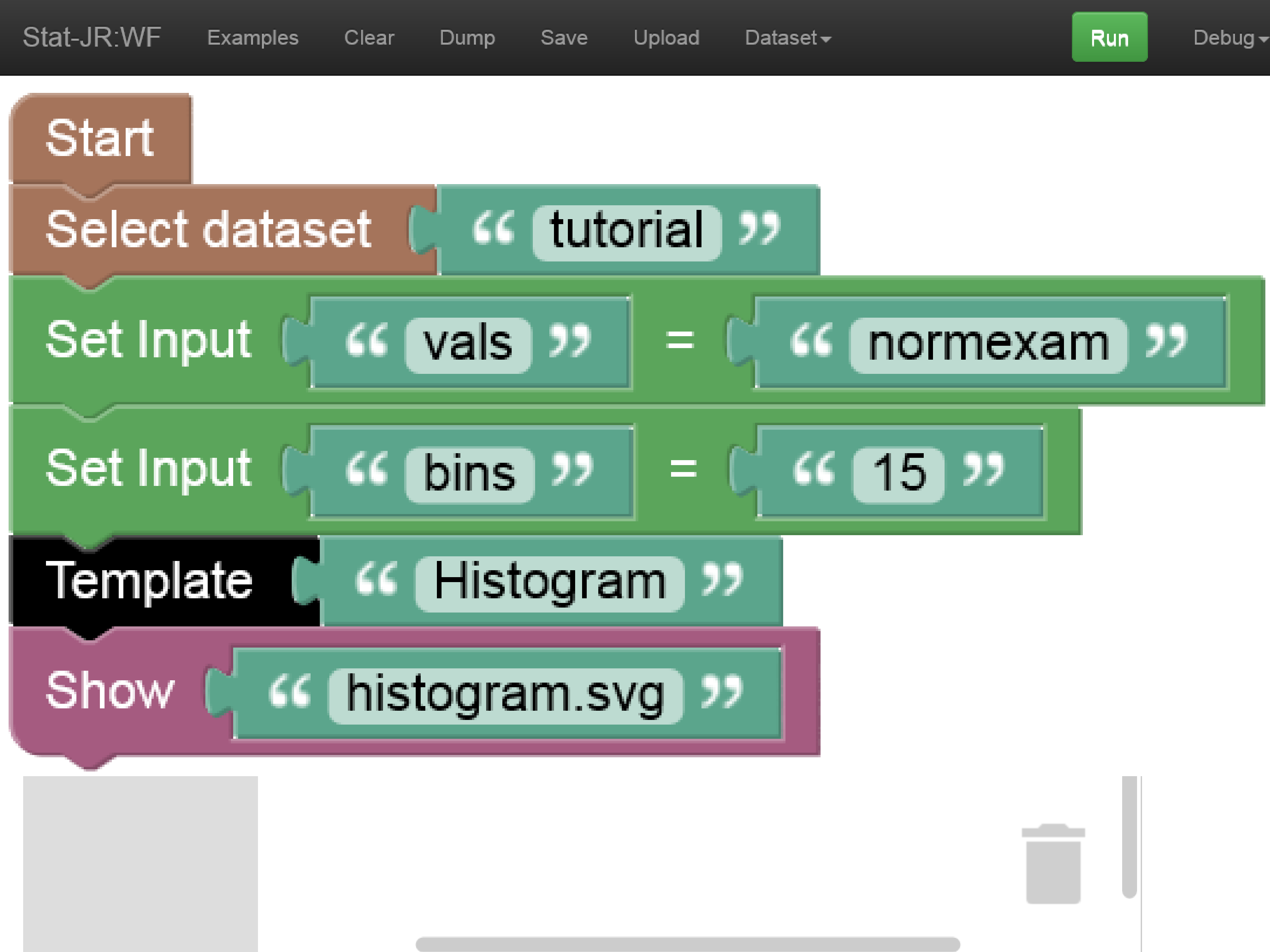
View On GitHub

Blockly is a library for building visual programming editors. Try it:

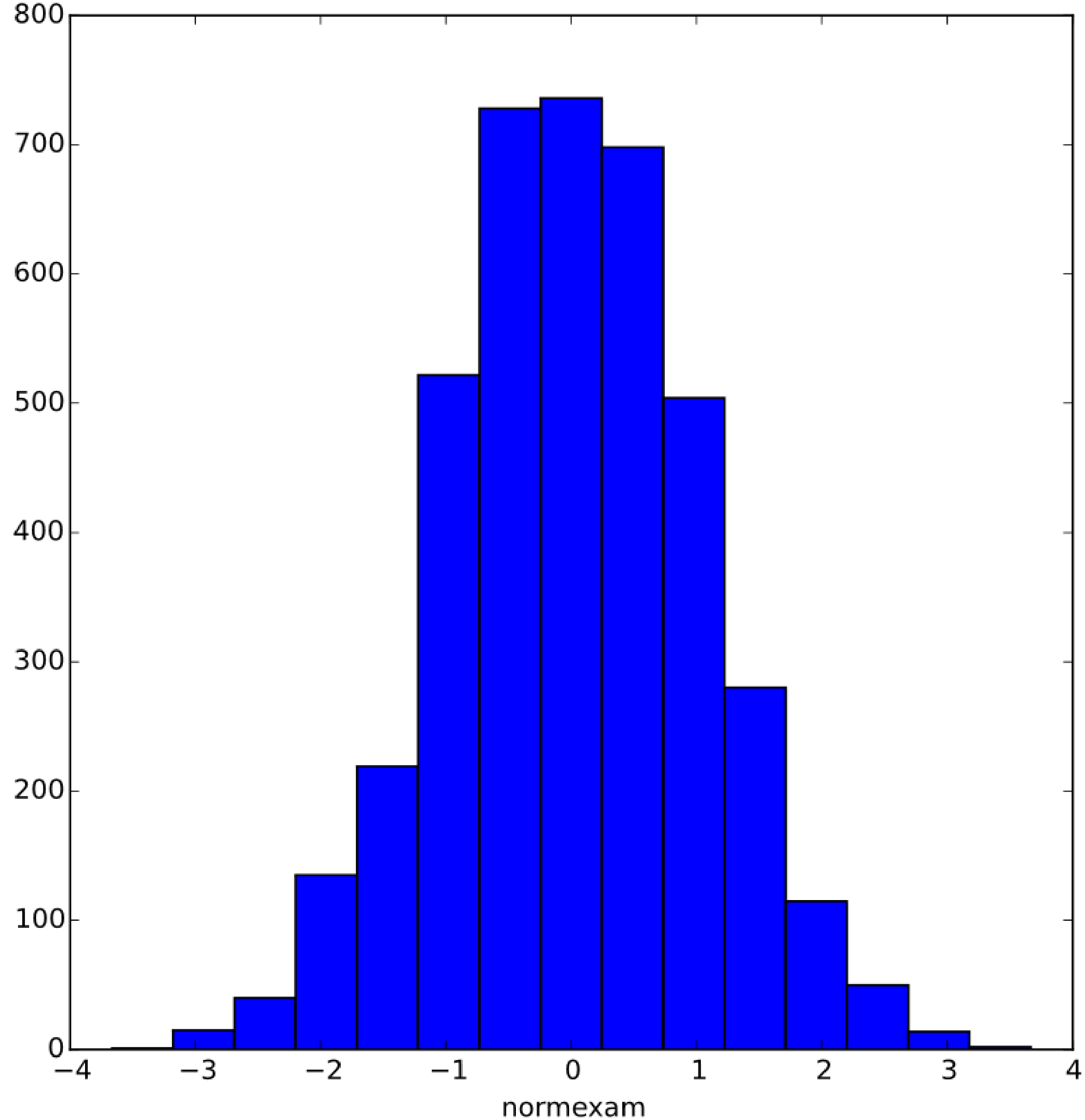



Our workflow system is written using **Blockly**…
…a visual block programming editor developed by Google (similar to Scratch)


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Start
Select dataset
tutorial
Set Input
vals
=
normexam
Set Input
bins
=
15
Template
Histogram
Show
histogram.svg
Selected block:
14


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Start
Select dataset
tutorial
Set Input
vals
=
normexam
Set Input
bins
=
15
Template
Histogram
Show
histogram.svg

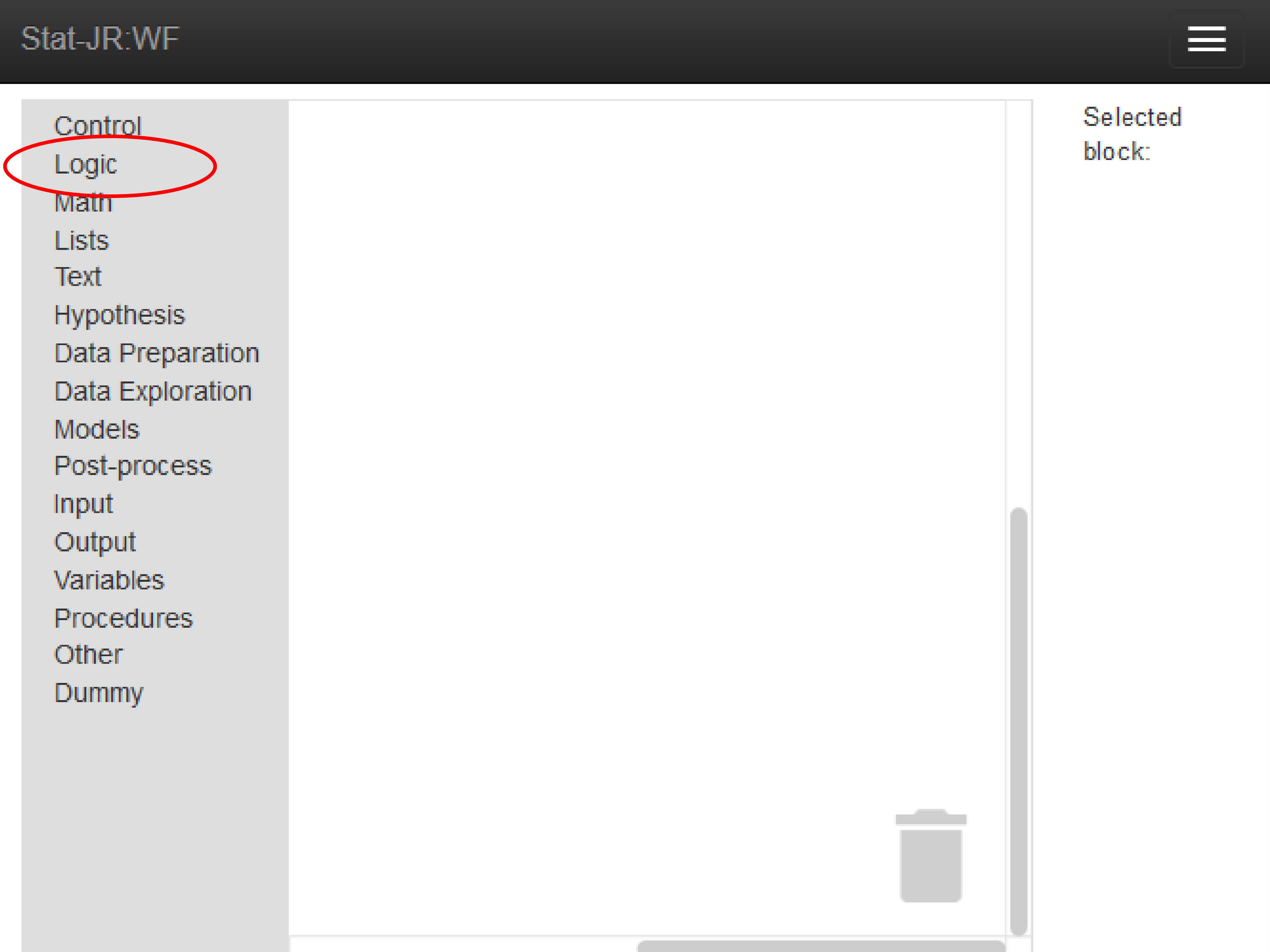
Stat-JR:WF Examples Clear Dump Save Upload Dataset Run Debug



Stat-JR:WF

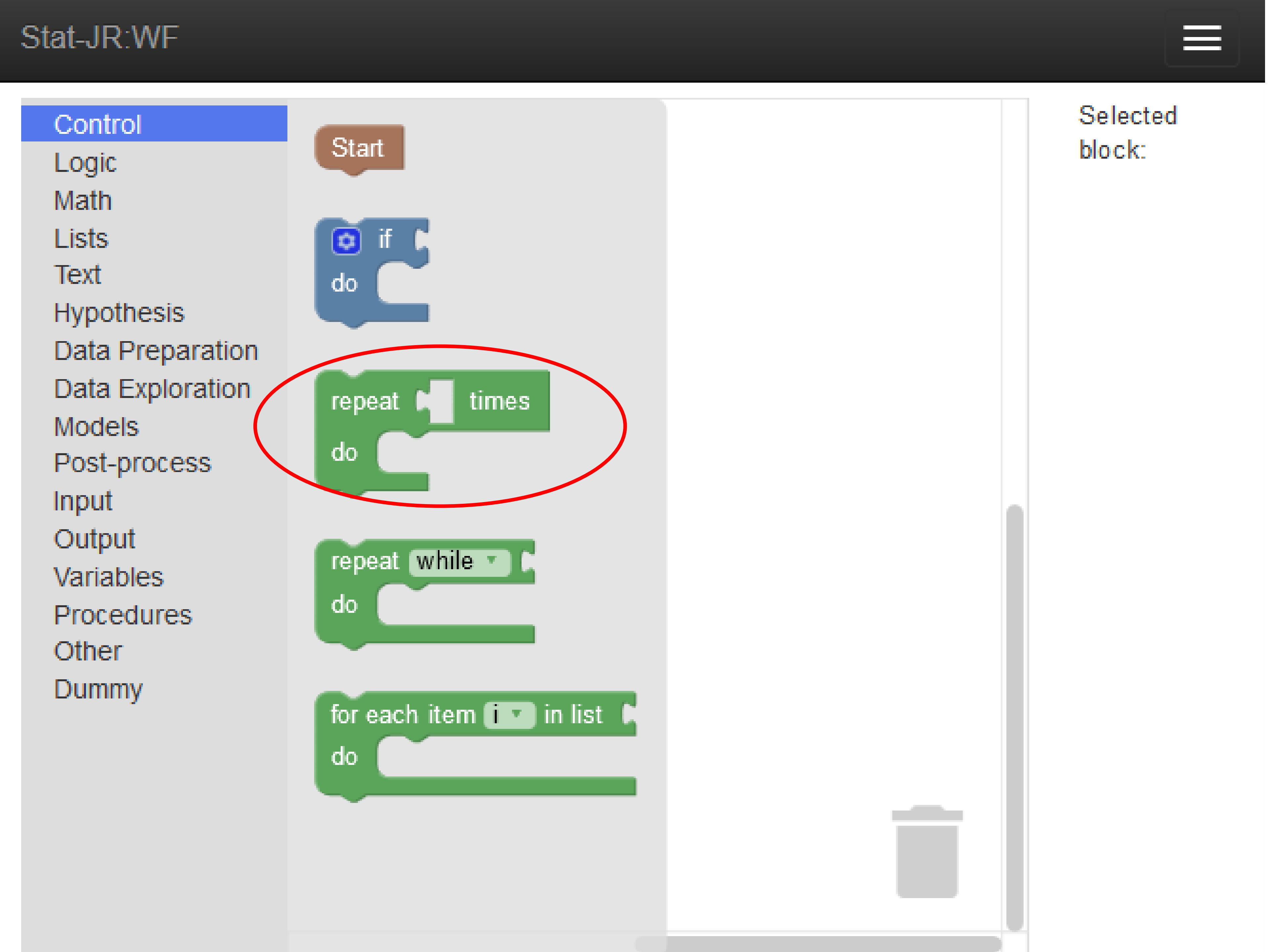
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy

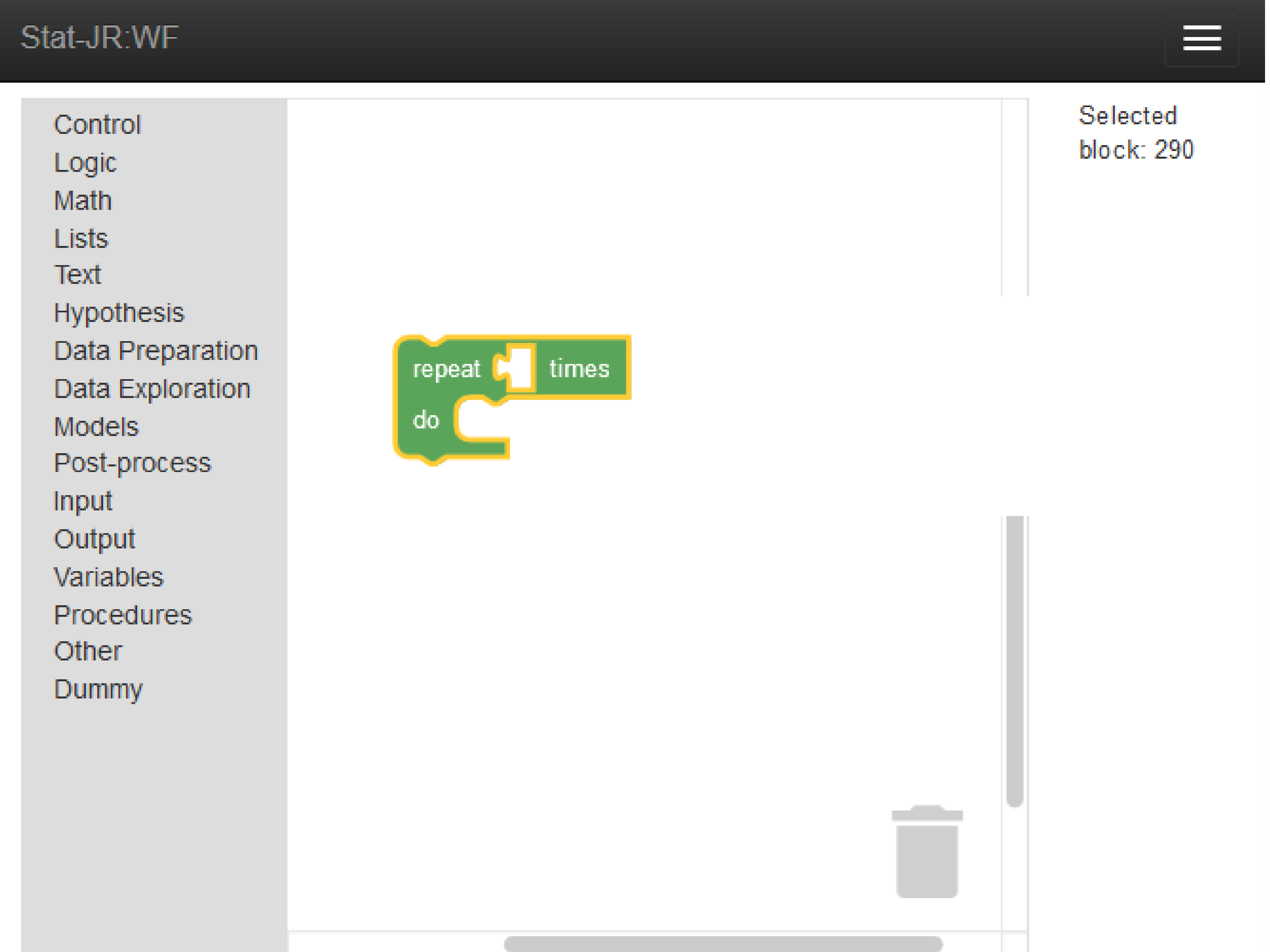
Selected block:

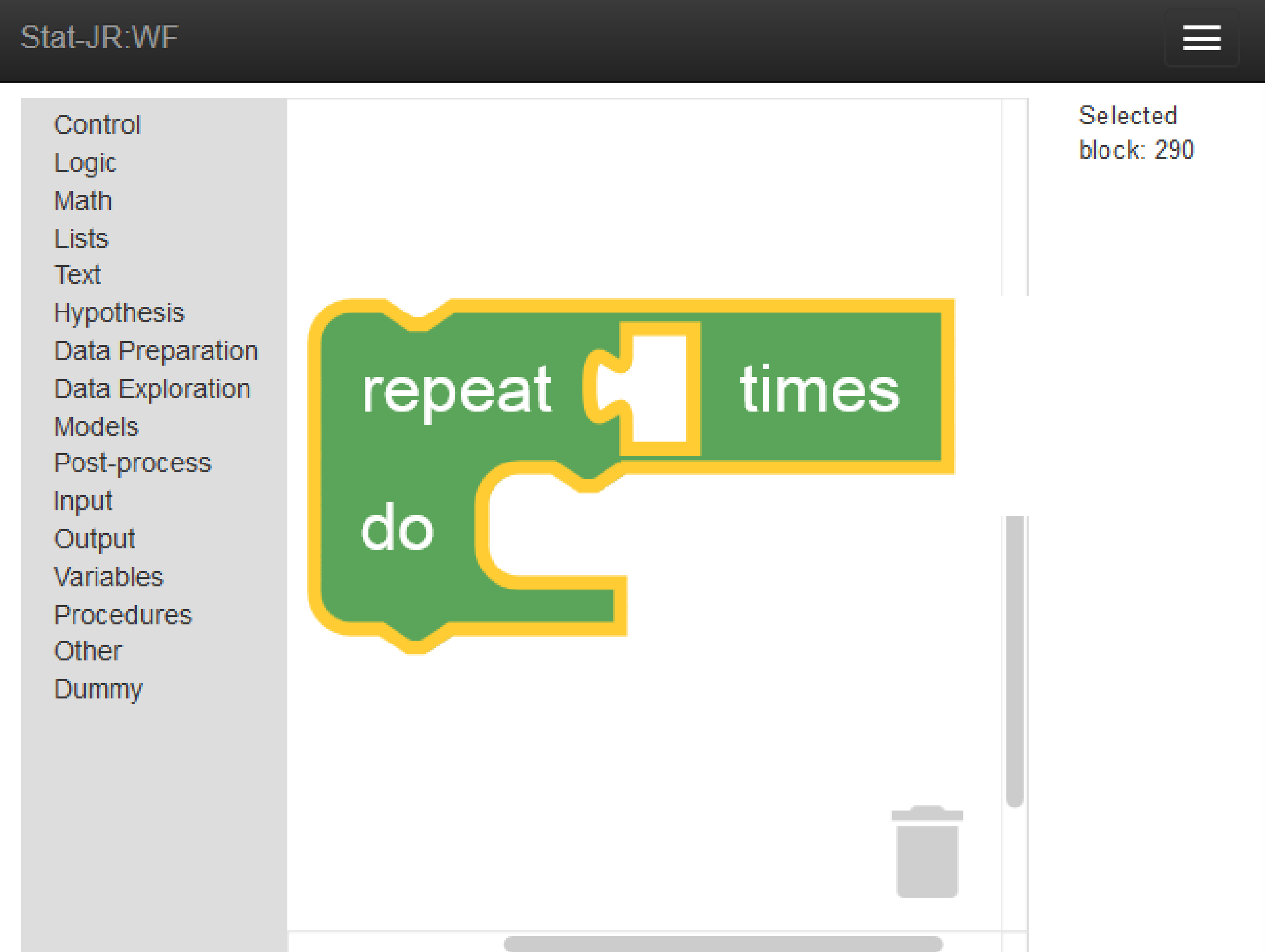

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:

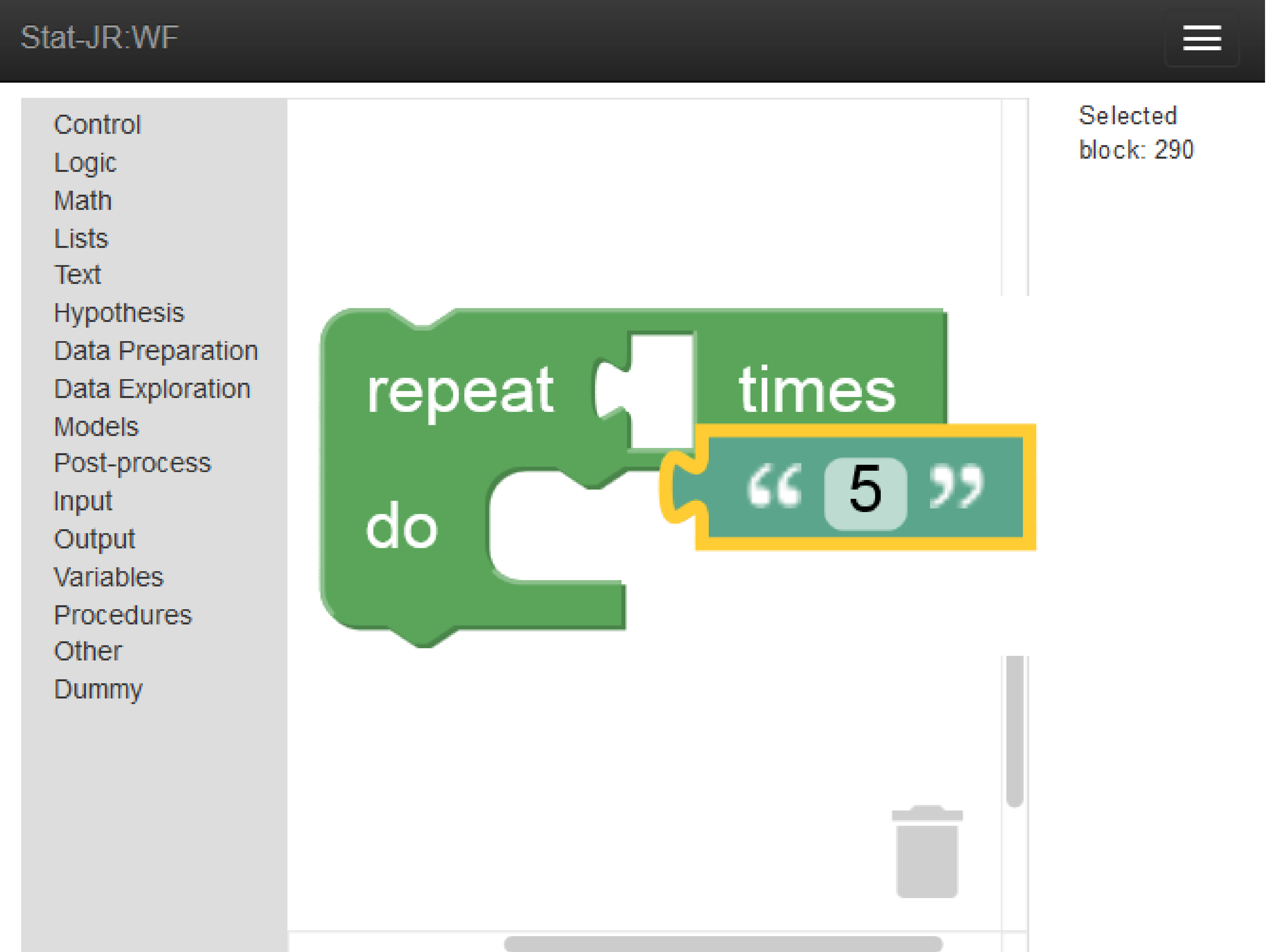

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
and
not
true
=
Ask yes/no
Selected block: 186

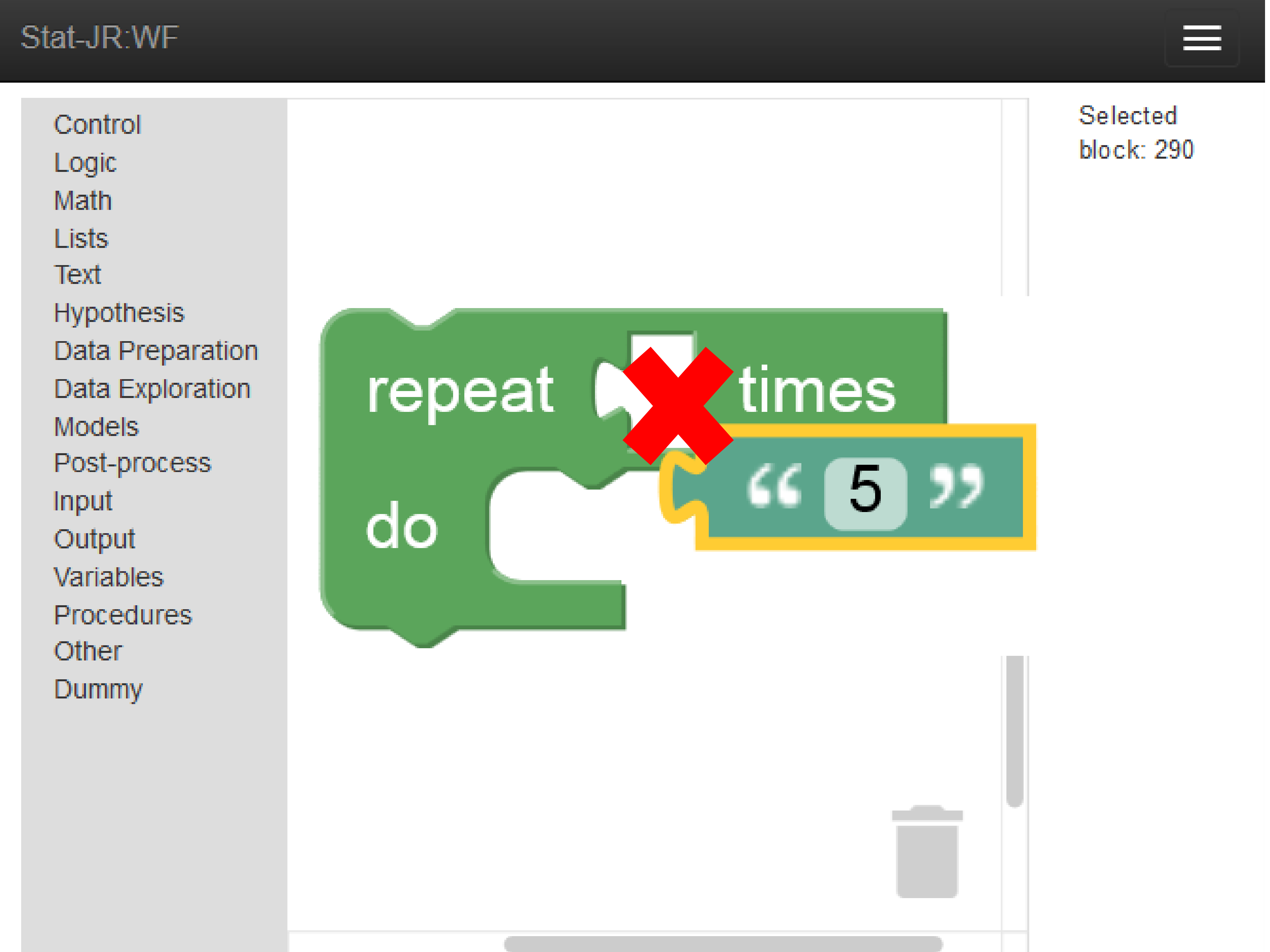

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Start
if
do
repeat
times
do
repeat
while
do
for each item
i
in list
do
Selected block:

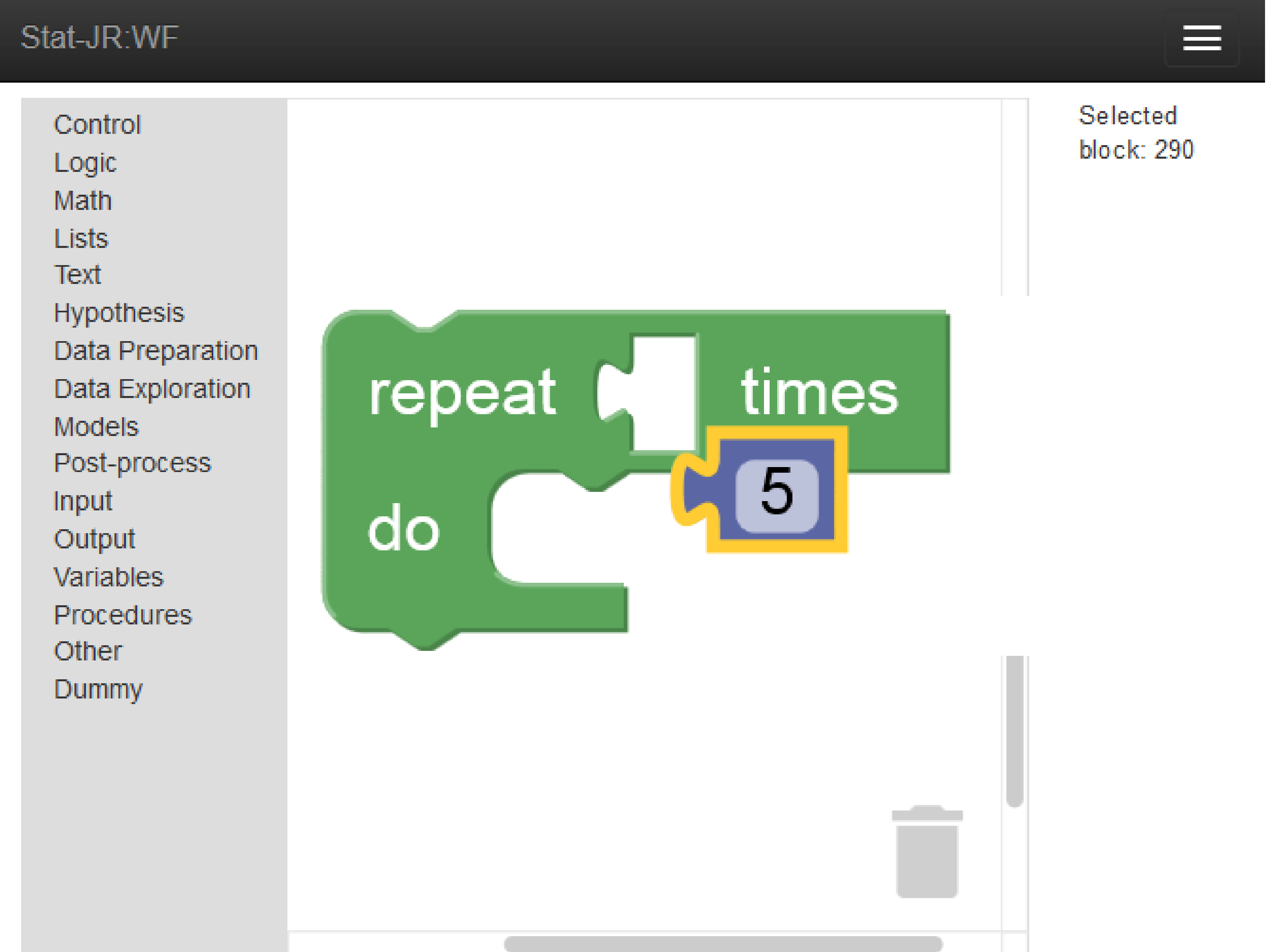

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Start
if
do
repeat
times
do
repeat
while
do
for each item
i
in list
do
Selected
block:

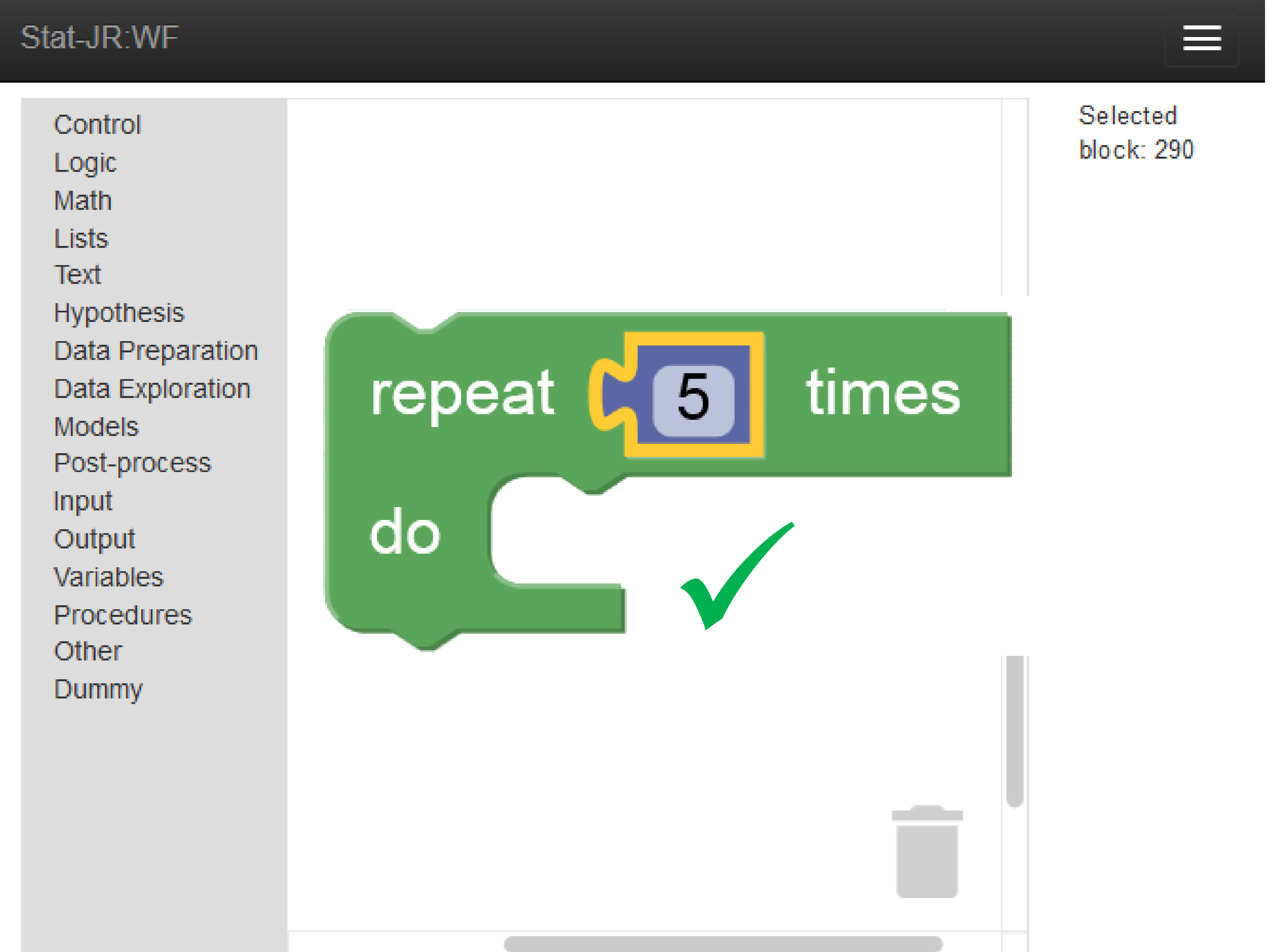

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
times
do
Selected block: 290

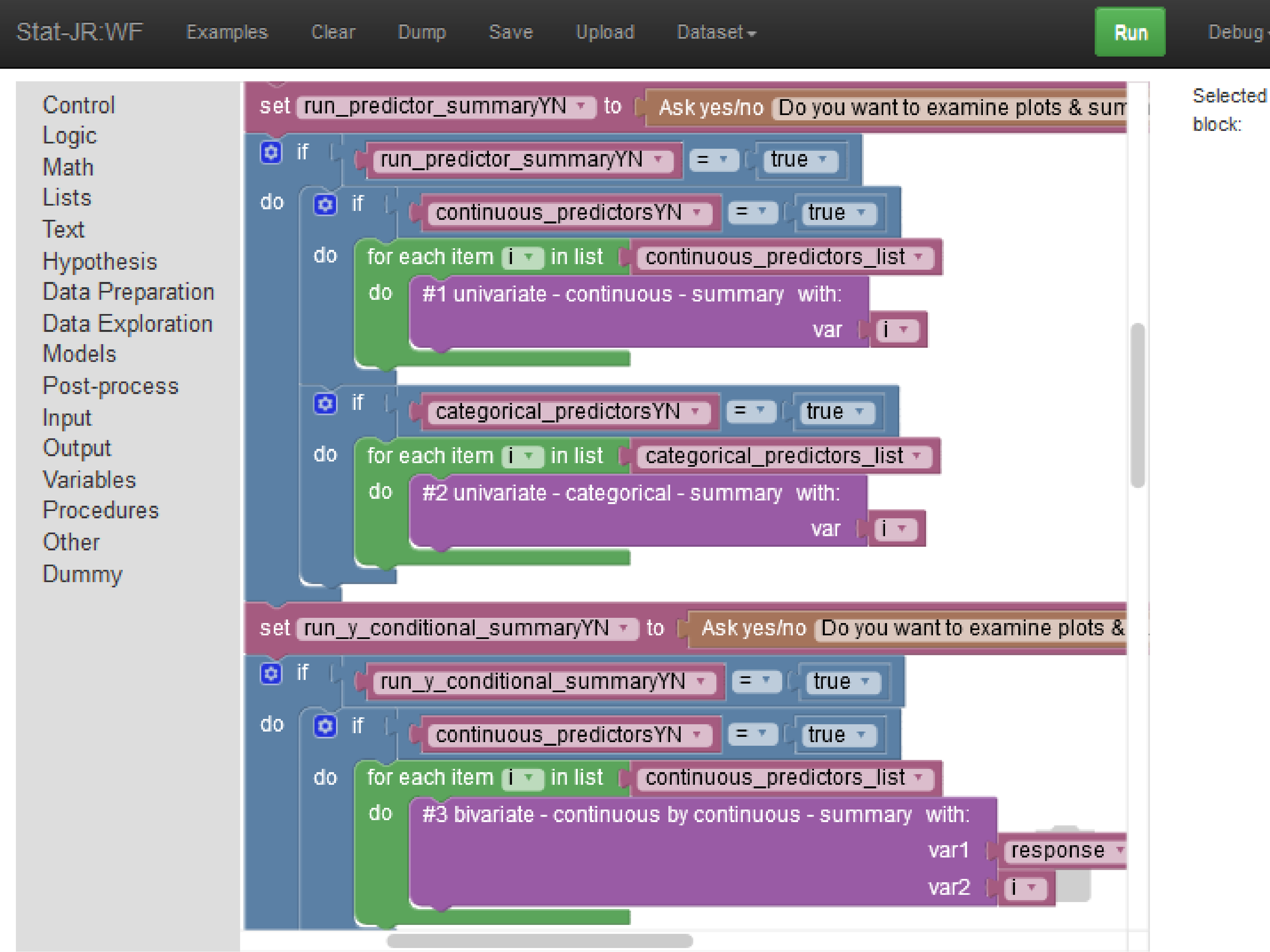

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
times
do
Selected block: 290

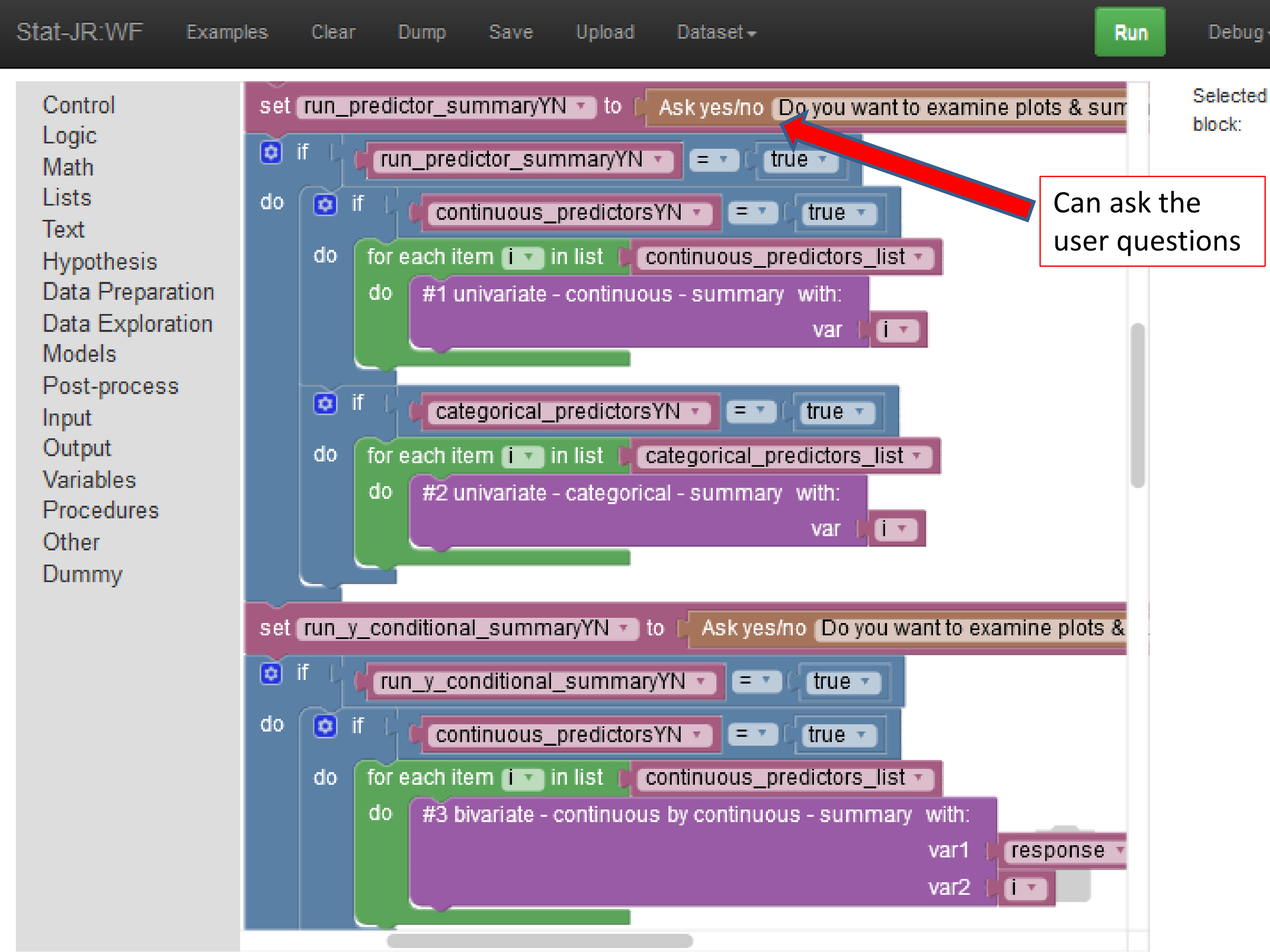

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
times
do
“ 5 ”
Selected block: 290


Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
times
do
“ 5 ”
Selected block: 290

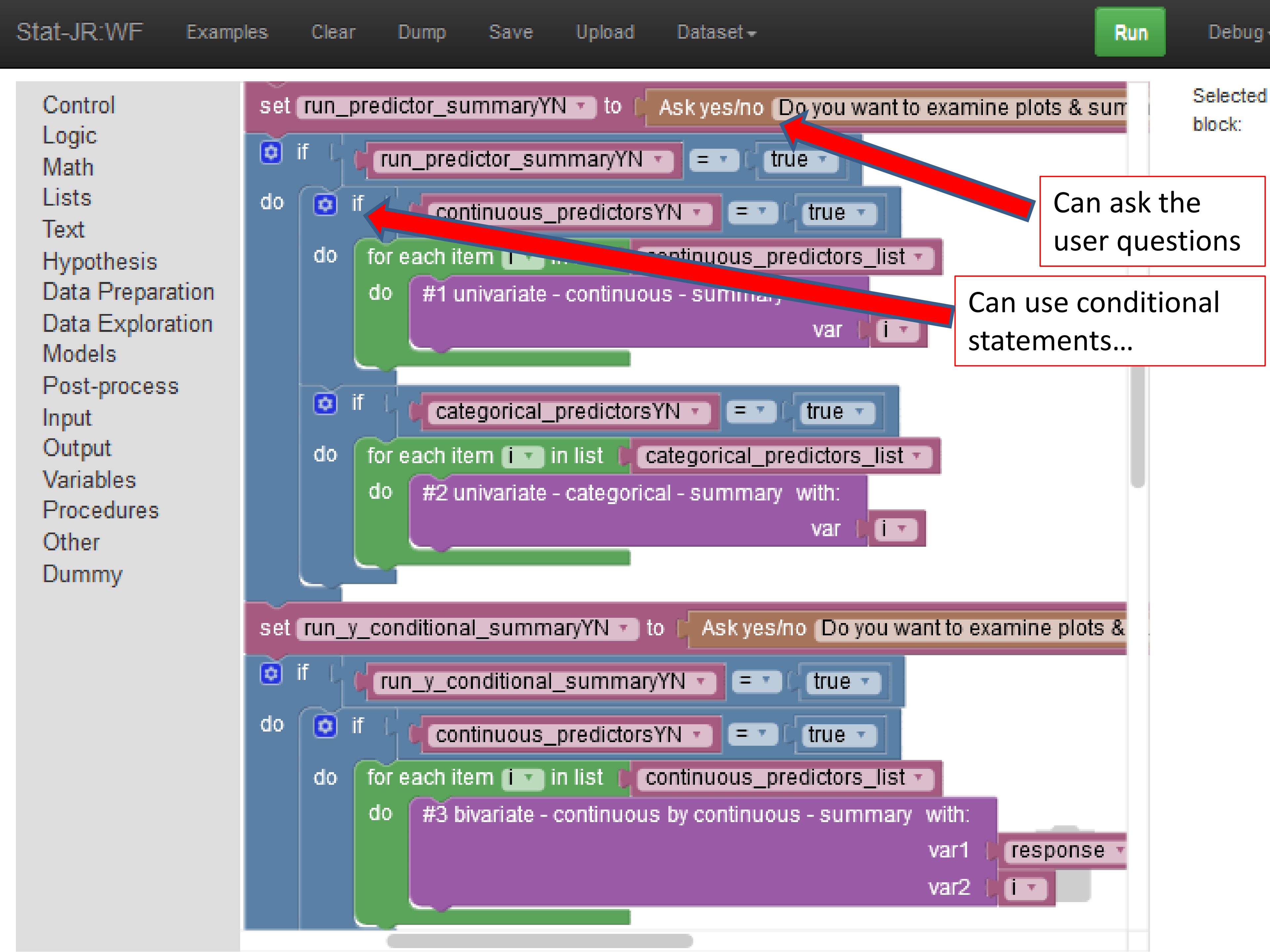

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
times
5
do
Selected block: 290

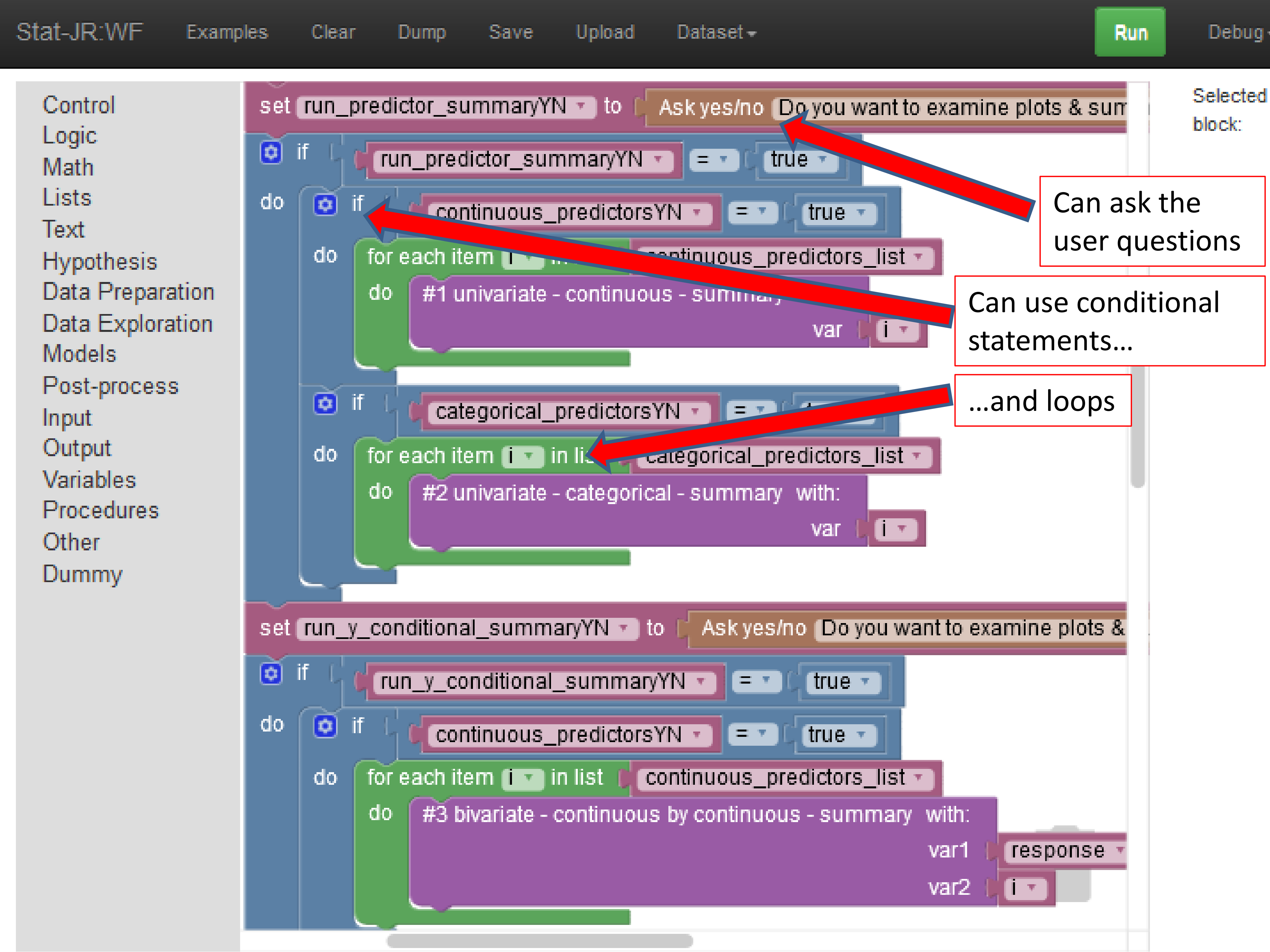

Stat-JR:WF
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
repeat
5
times
do
Selected block: 290

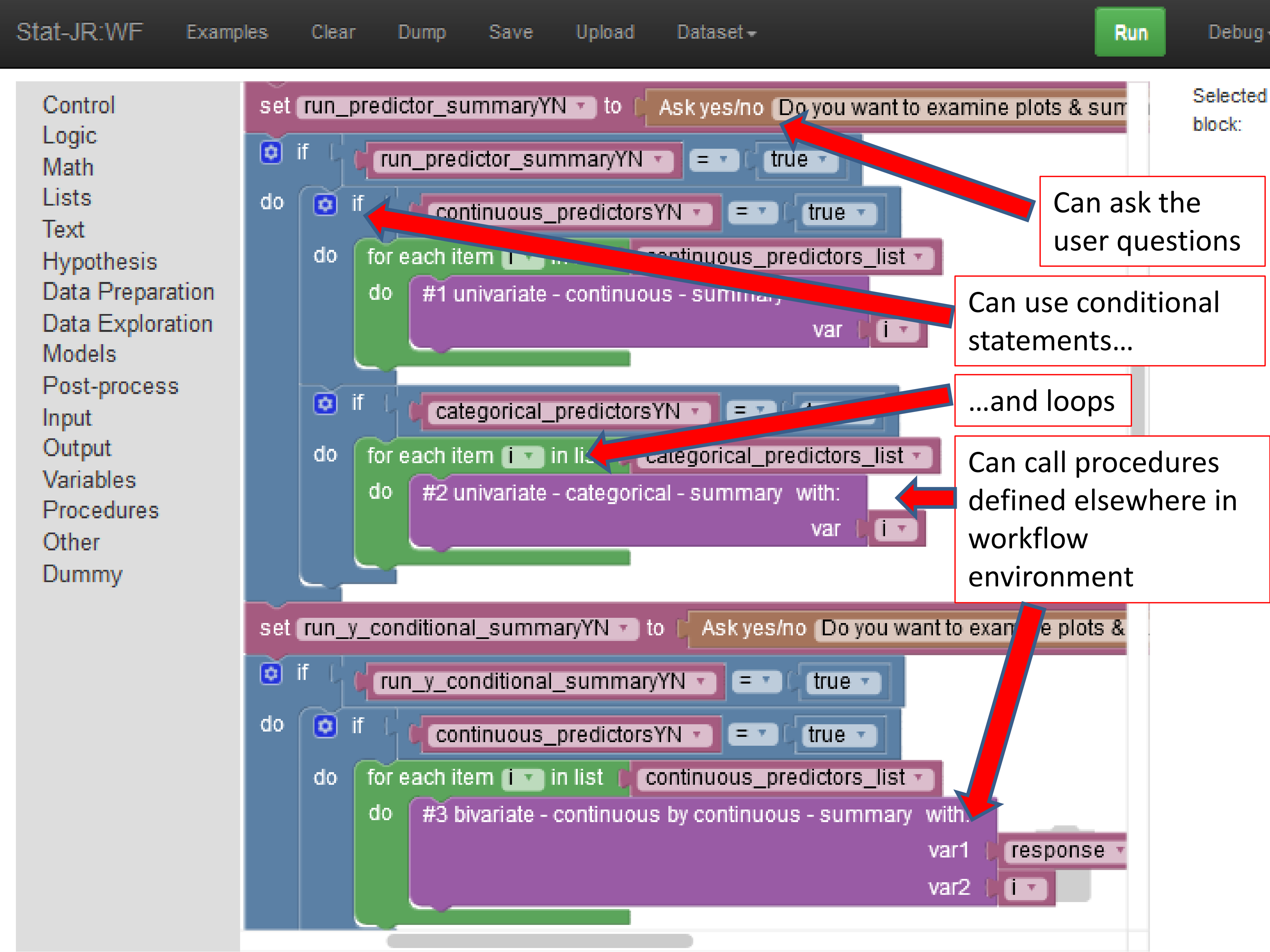

Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i

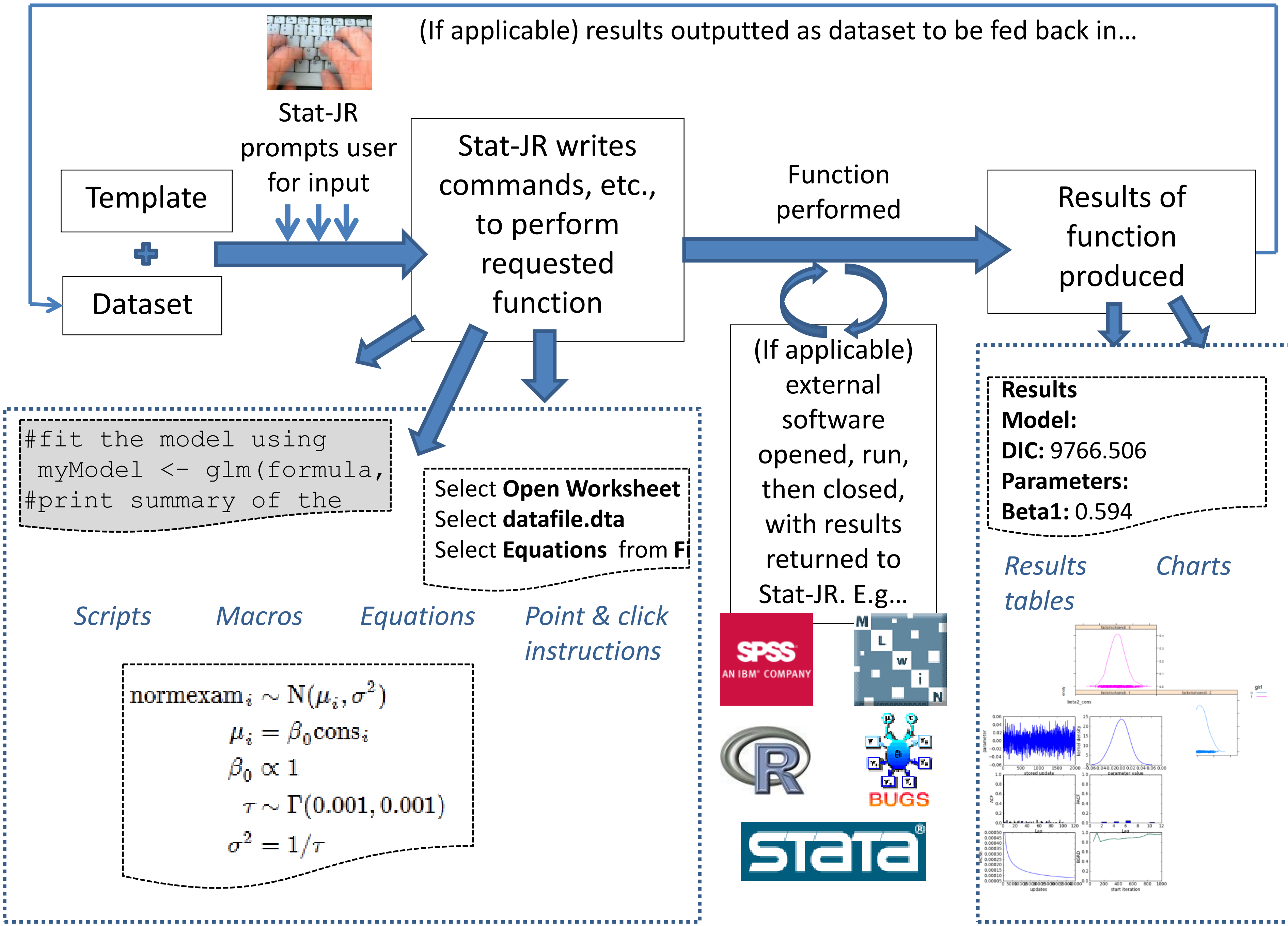


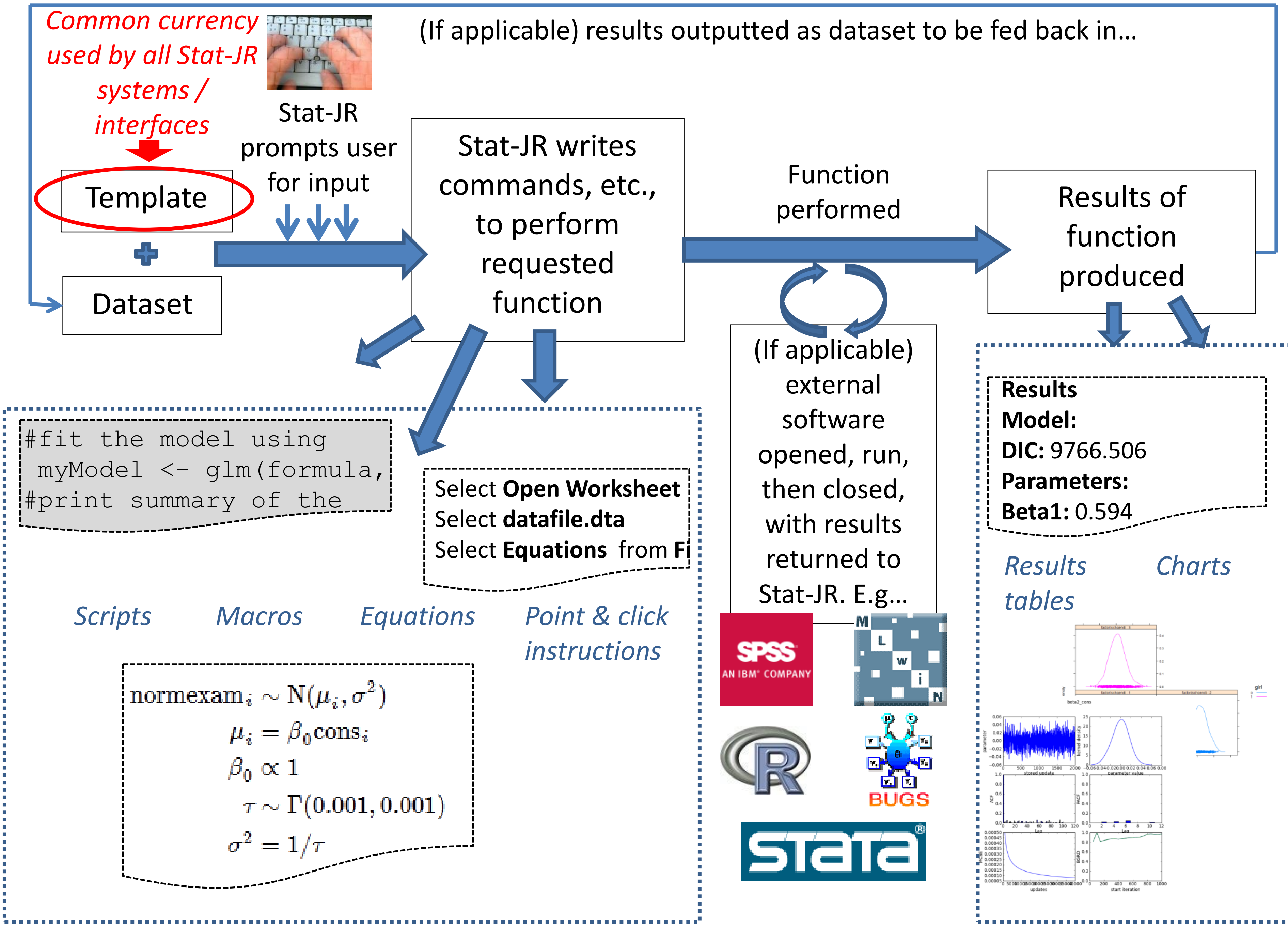
Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Can ask the user questions


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Can ask the user questions
Can use conditional statements...


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Can ask the user questions
Can use conditional statements…
…and loops


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Can ask the user questions
Can use conditional statements…
…and loops
Can call procedures defined elsewhere in workflow environment


(If applicable) results outputted as dataset to be fed back in…
Stat-JR prompts user for input
Template
Dataset
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
#fit the model using
myModel <- glm(formula,
#print summary of the
Select **Open Worksheet**
Select **datafile.dta**
Select **Equations** from Fi
*Scripts*
*Macros*
*Equations*
*Point & click instructions*
$$\begin{aligned} \mathrm{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\ \mu_i &= \beta_0 \mathrm{cons}_i \\ \beta_0 &\propto 1 \\ \tau &\sim \Gamma(0.001, 0.001) \\ \sigma^2 &= 1/\tau \end{aligned}$$
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g…
SPSS
AN IBM COMPANY
BUGS
**Results**
**Model:**
**DIC:** 9766.506
**Parameters:**
**Beta1:** 0.594
*Results tables*
*Charts*



Common currency used by all Stat-JR systems / interfaces
(If applicable) results outputted as dataset to be fed back in…
Template
Dataset
Stat-JR prompts user for input
Stat-JR writes commands, etc., to perform requested function
Function performed
Results of function produced
(If applicable) external software opened, run, then closed, with results returned to Stat-JR. E.g…
#fit the model using
myModel <- glm(formula,
#print summary of the
Select Open Worksheet
Select datafile.dta
Select Equations from Fi
Scripts
Macros
Equations
Point & click instructions
normexam$_i$ ~ N($\mu_i$, $\sigma^2$)
$\mu_i = \beta_0$cons$_i$
$\beta_0 \propto 1$
$\tau \sim \Gamma(0.001, 0.001)$
$\sigma^2 = 1/\tau$
SPSS AN IBM COMPANY
MLwiN
BUGS
STATA
Results
Model:
DIC: 9766.506
Parameters:
Beta1: 0.594
Results tables
Charts

Stat-JR:TREE Start again Dataset tutorial Template Histogram Ready (1s) Settings Debug

**Values:**

normexam remove

**Number of bins:**

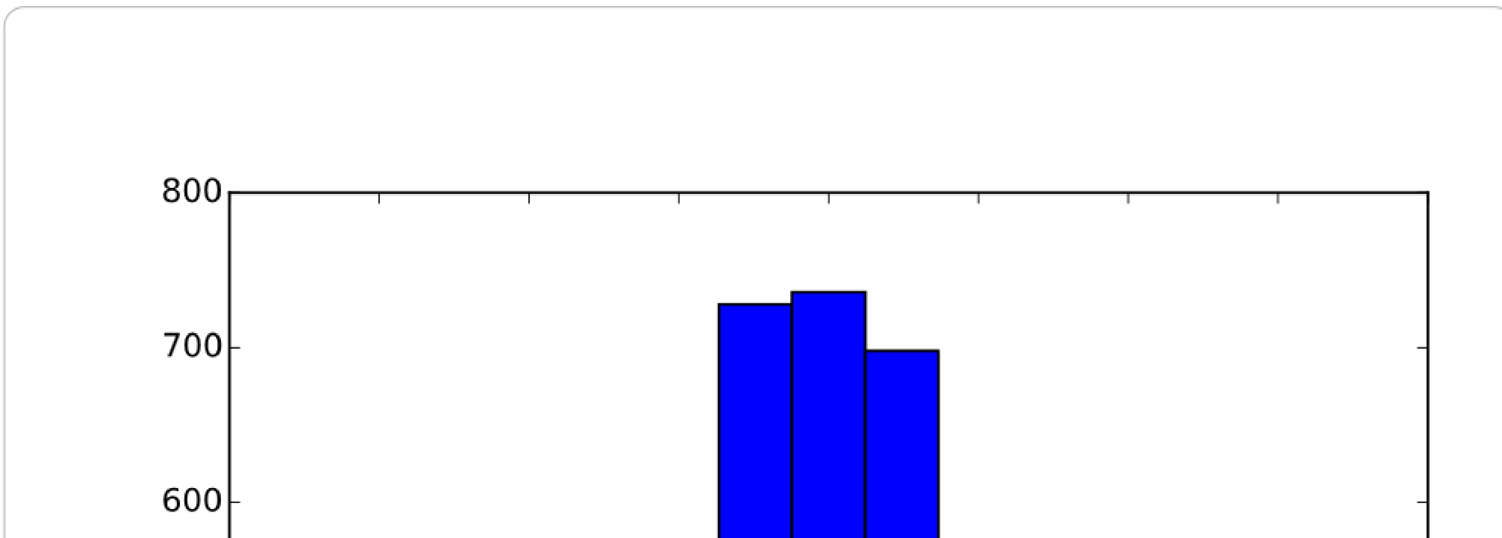
15 remove

Download Add to ebook

Current input string: {'vals': 'normexam', 'bins': '15'}

Command: RunStatJR(template='Histogram', dataset='tutorial', invars = {'vals': 'normexam', 'bins': '15'}, estoptions = {})

histogram.svg Popout



Stat-JR:TREE Start again Dataset tutorial Template Histogram Ready (1s) Settings Debug

**Values:**

normexam remove

**Number of bins:**

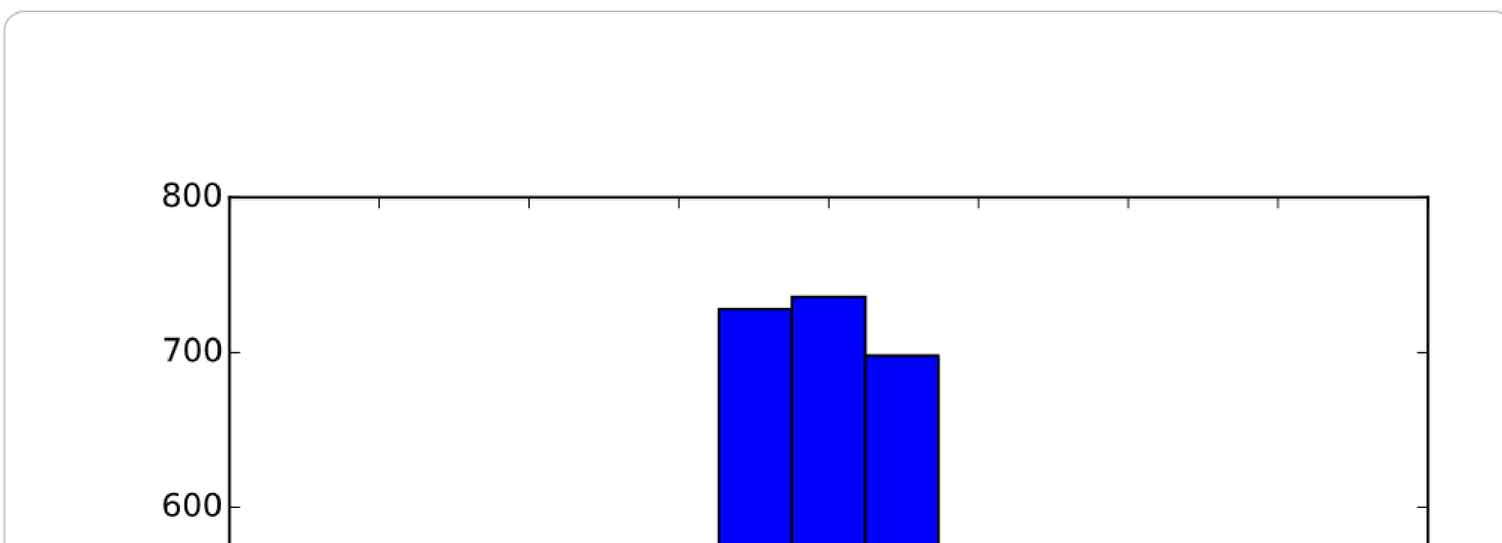
15 remove

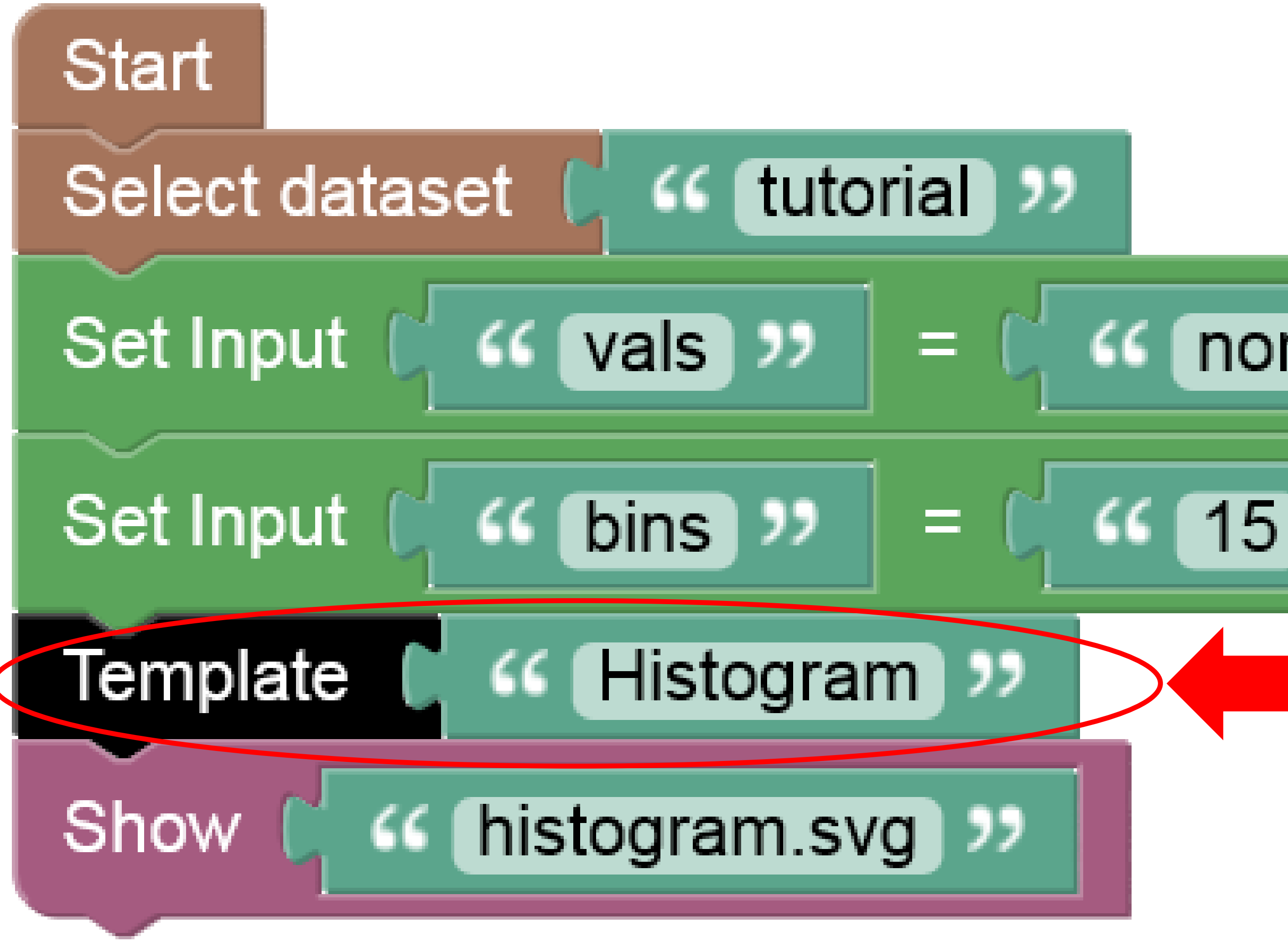
Download | Add to ebook

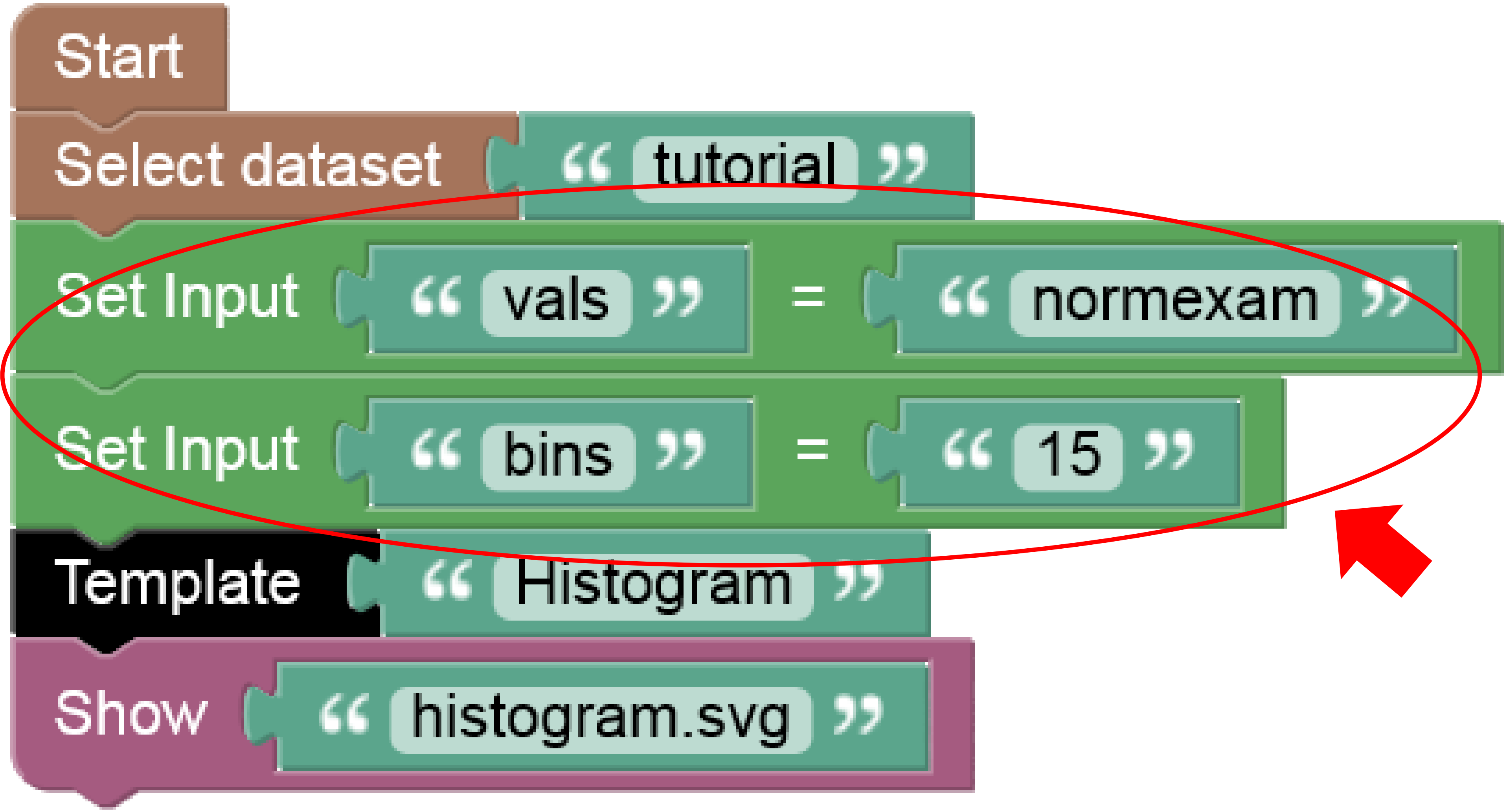
Current input string: {'vals': 'normexam', 'bins': '15'}

Command: RunStatJR(template='Histogram', dataset='tutorial', invars = {'vals': 'normexam', 'bins': '15'}, estoptions = {})

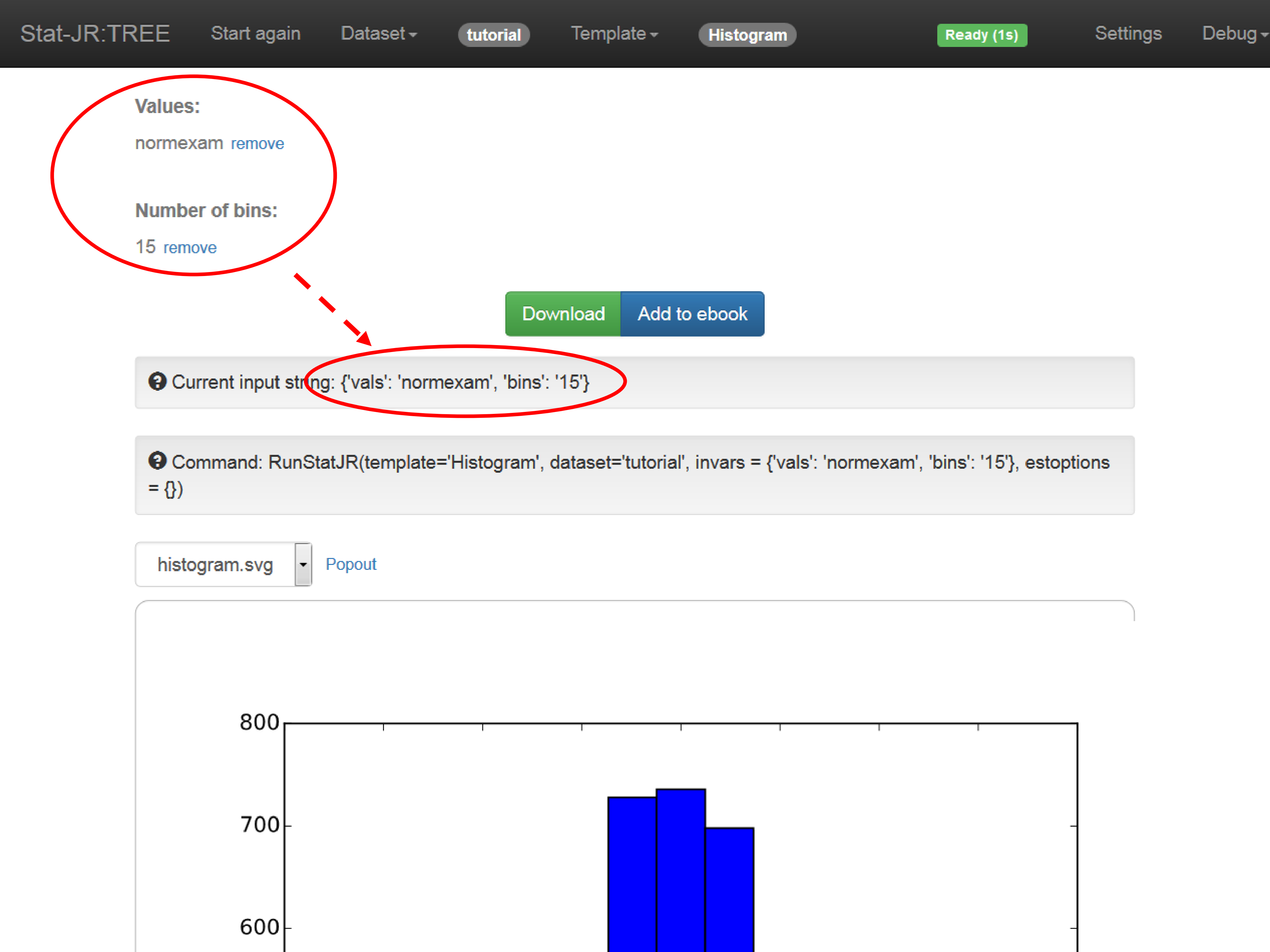
histogram.svg Popout




Start
Select dataset
“ tutorial ”
Set Input
“ vals ” = “ normexam ”
Set Input
“ bins ” = “ 15 ”
Template
“ Histogram ”
Show
“ histogram.svg ”



Start
Select dataset
“ tutorial ”
Set Input
“ vals ”
=
“ normexam ”
Set Input
“ bins ”
=
“ 15 ”
Template
“ Histogram ”
Show
“ histogram.svg ”



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Histogram
Ready (1s)
Settings
Debug
Values:
normexam remove
Number of bins:
15 remove
Download
Add to ebook
Current input string: {'vals': 'normexam', 'bins': '15'}
Command: RunStatJR(template='Histogram', dataset='tutorial', invars = {'vals': 'normexam', 'bins': '15'}, estoptions = {})
histogram.svg
Popout
800
700
600

Stat-JR:TREE Start again Dataset tutorial Template Histogram Ready (1s) Settings Debug

**Values:**

normexam remove

**Number of bins:**

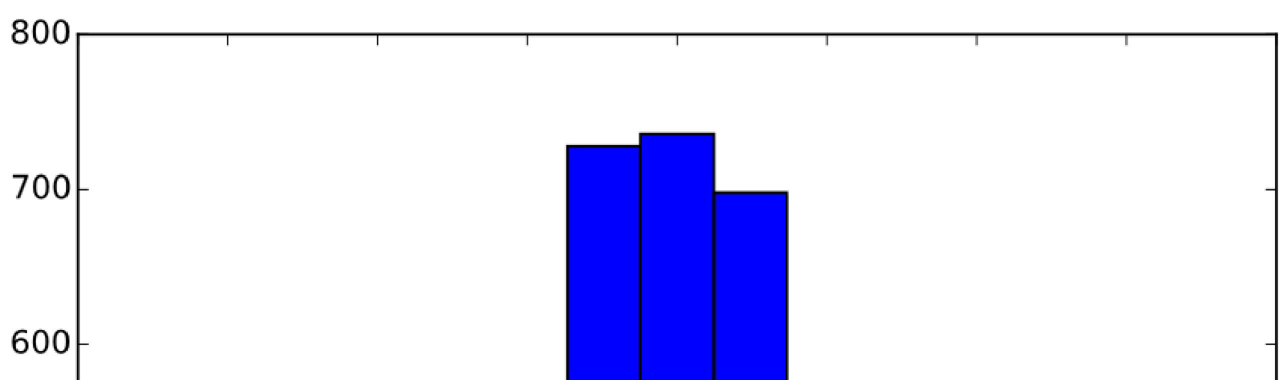
15 remove

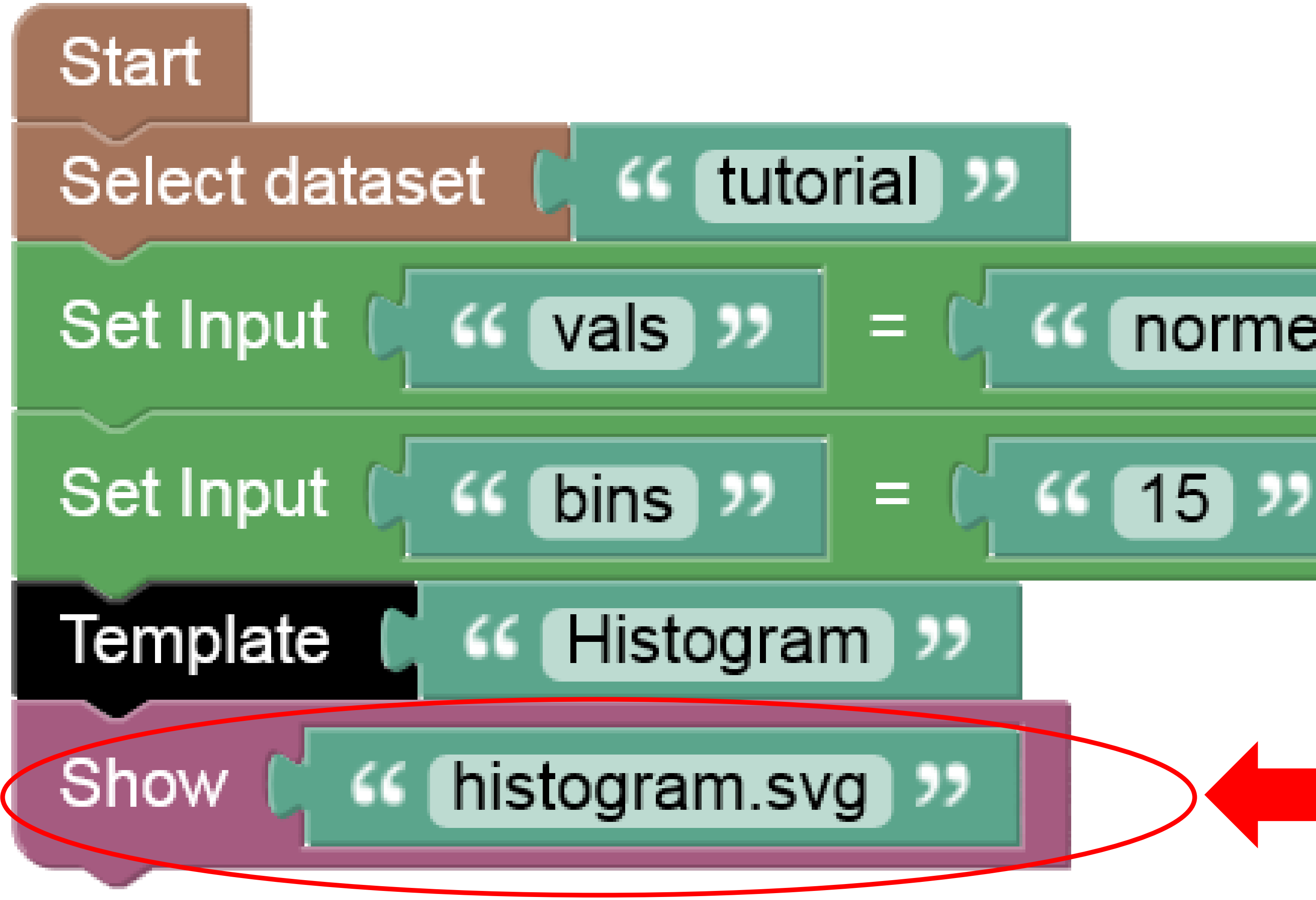
Download Add to ebook

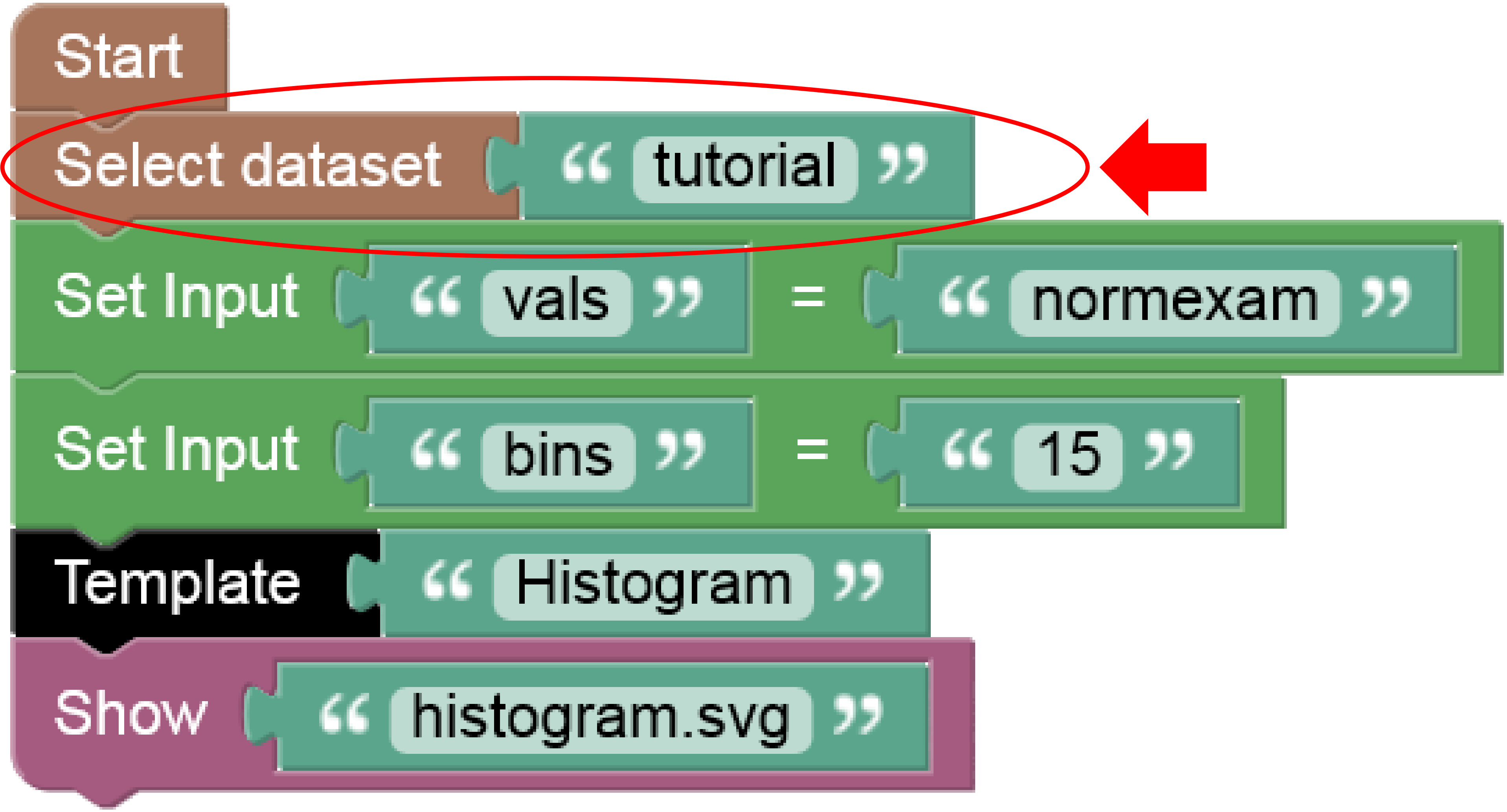
Current input string: {'vals': 'normexam', 'bins': '15'}

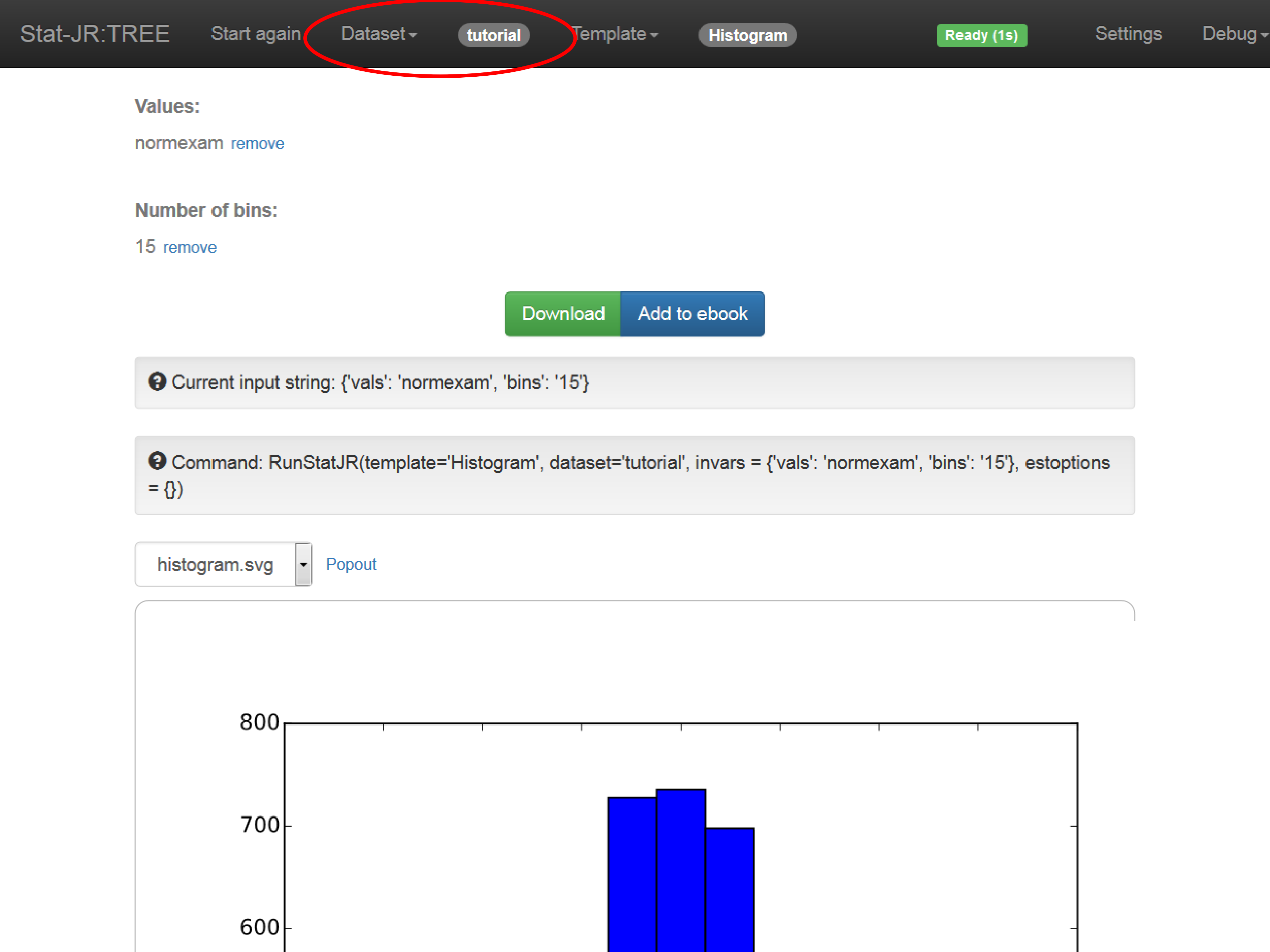
Command: RunStatJR(template='Histogram', dataset='tutorial', invars = {'vals': 'normexam', 'bins': '15'}, estoptions = {})

histogram.svg Popout




Start
Select dataset
“ tutorial ”
Set Input
“ vals ”
=
“ normexam ”
Set Input
“ bins ”
=
“ 15 ”
Template
“ Histogram ”
Show
“ histogram.svg ”


Start
Select dataset
“ tutorial ”
Set Input
“ vals ”
=
“ normexam ”
Set Input
“ bins ”
=
“ 15 ”
Template
“ Histogram ”
Show
“ histogram.svg ”



Stat-JR:TREE
Start again
Dataset
tutorial
Template
Histogram
Ready (1s)
Settings
Debug
Values:
normexam remove
Number of bins:
15 remove
Download
Add to ebook
Current input string: {'vals': 'normexam', 'bins': '15'}
Command: RunStatJR(template='Histogram', dataset='tutorial', invars = {'vals': 'normexam', 'bins': '15'}, estoptions = {})
histogram.svg
Popout
800
700
600

```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'

tags = [ 'Plots' ]
engines = ['Python_script']

inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''

pythonscript = '''
from io import BytesIO

from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')

buf.seek(0)
outputs['histogram.svg'] = ImageOutput(buf.getvalue())
buf.close()
'''
```

```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'

tags = [ 'Plots' ]
engines = ['Python_script']

inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''

pythonscript = '''
from io import BytesIO

from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')

buf.seek(0)
outputs['histogram.svg'] = ImageOutput(buf.getvalue())
buf.close()
'''
```

*...can also find this information by looking in template code itself...*

```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'

tags = [ 'Plots' ]
engines = ['Python script']

inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''

pythonscript = '''
from io import BytesIO

from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')

buf.seek(0)
outputs['histogram.svg'] = ImageOutput(buf.getvalue())
buf.close()
'''
```

*…can also find this information by looking in template code itself…*

```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'
```

```
inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''
```

```
from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')

buf.seek(0)
outputs['histogram.svg'] = ImageOutput(buf.getvalue())
buf.close()
'''
```

```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'
```

```
inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''
```

```
from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')

buf.seek(0)
outputs['histogram.svg'] = ImageOutput(buf.getvalue())
buf.close()
'''
```

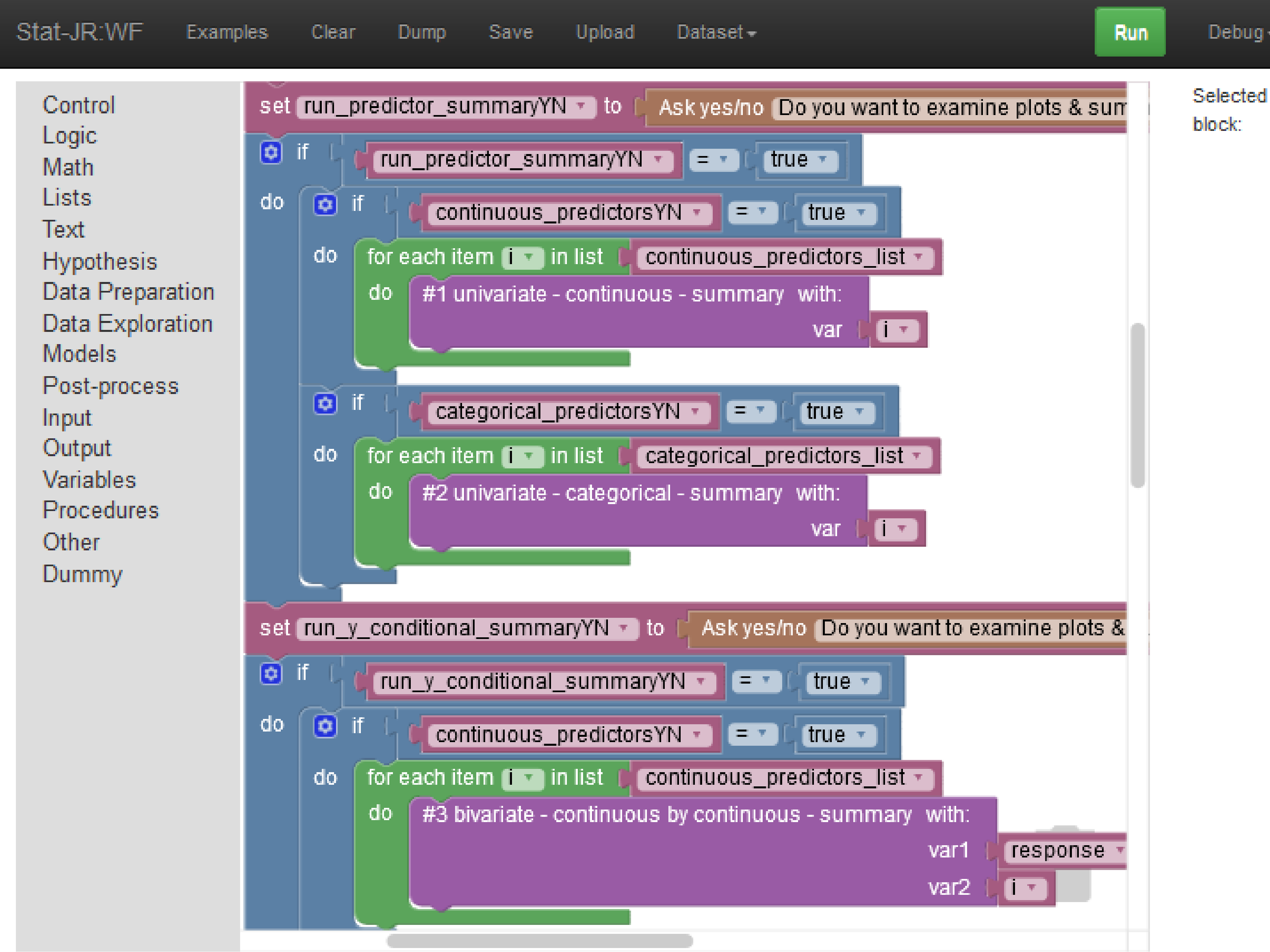
```
# Copyright (c) 2013, University of Bristol and University of Southampton.

from EStat.Templating import Template

class TemplateHistogram(Template):
'Produces a histogram from a column of data, with the number of bins chosen by the user.'

__version__ = '1.0.0'
```

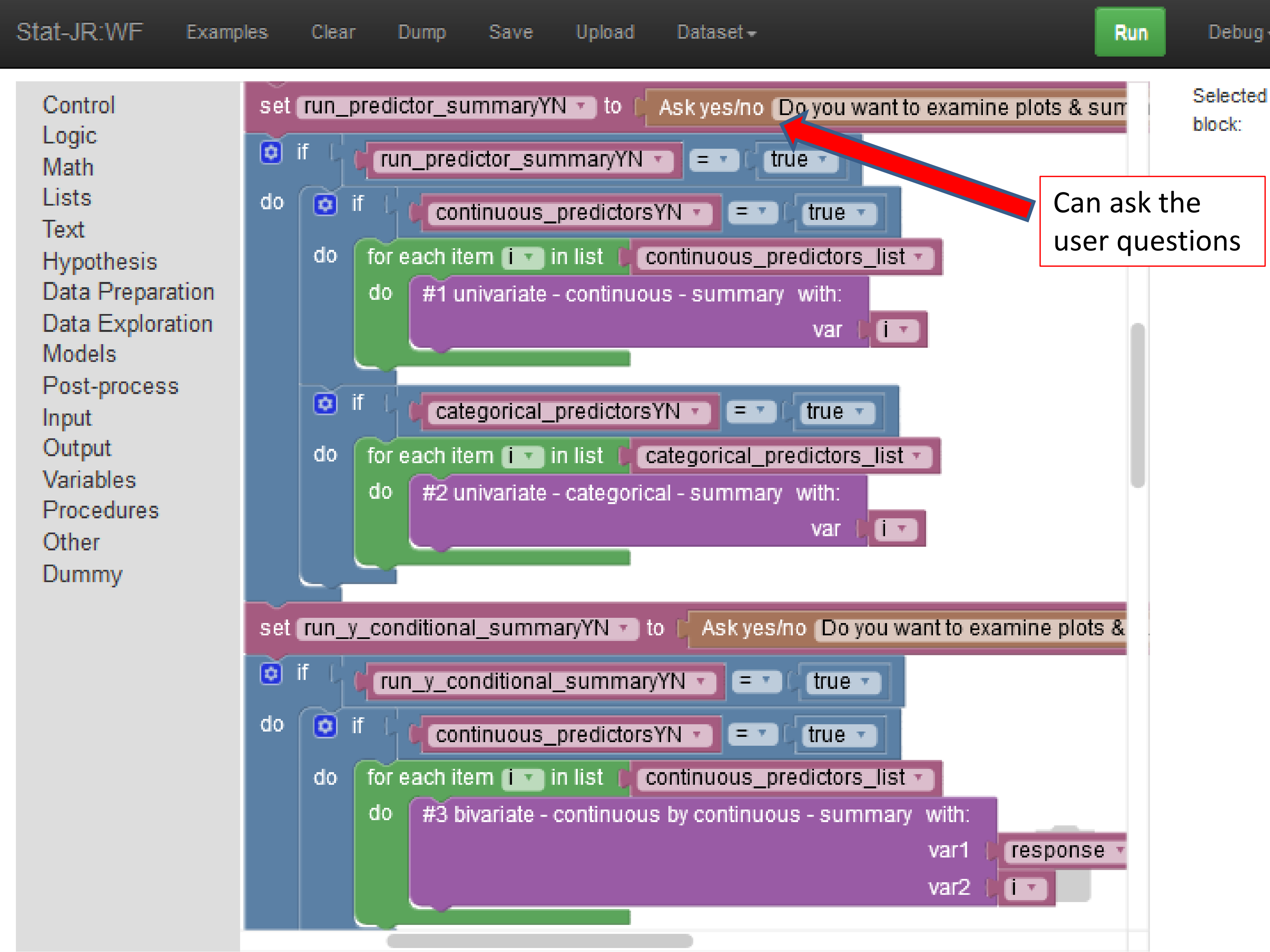
```
inputs = '''
vals = DataVector('Values: ')
bins = Integer('Number of bins: ')
'''
```

```
from matplotlib.figure import Figure
import matplotlib.lines as lines
from matplotlib.backends.backend_agg import FigureCanvasAgg

import EStat
from EStat.Templating import *
fig = Figure(figsize=(8,8))
ax = fig.add_subplot(100 + 10 + 1, xlabel = vals)
ax.hist(datafile.variables[vals]['data'], bins)

canvas = FigureCanvasAgg(fig)
buf = BytesIO()
canvas.print_figure(buf, dpi=80, format='svg')
```

```
outputs['histogram.svg'] = ImageOutput…
```

```
...
```


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i



Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Selected block:
Can ask the user questions

# Practical 1…

**Accessories > Command Prompt**

cd /d c:\Users\your_username

mkdir .statjr

**Open E:\tree**

**(Best in Chrome, so open Chrome & copy address across from Internet Explorer)**

**Open Settings (in black bar at top of browser)**

**Find “Location of G++ compiler (Windows only)”**

**Change this to:**

E:\MinGW\bin

**Press Set**

## eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes
  - statistical analysis assistant

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes
  - statistical analysis assistant

## A key to assist in your choice of statistical test

Starting at step 1 in the list above move through the key following the path that best describes your data. If you are unsure about any of the terms used then consult the glossary or the relevant sections of the next two chapters. This is not a true dichotomous key and at several points there are more than two routes or end points.

There may be several end points appropriate to your data that result from this key. For example you may wish to know the correct display method for the data and then the correct measure of dispersion to use. If this is the case, go through the key twice.

All the tests and techniques mentioned in the key are described in later chapters.

Italics indicate instructions about what you should do.

Numbers in brackets indicate that the point in the key is something of a compromise destination.

There are several points where rather arbitrary numbers are used to determine which path you should take. For example, I use 30 different observations as the arbitrary level at which to split continuous and discontinuous data. If your data set falls close to this level you should not feel constrained to take one path if you feel more comfortable with the other.

---

| | | |
|---|---|---|
| 1 | Testing a clear hypothesis and associated null hypothesis (e.g. $H_1$ = blood glucose level is related to age and $H_0$ = blood glucose is not related to age). | 25 |
| | Not testing any hypothesis but simply want to present, summarize | 2 |

Dytham, C (2010) Choosing & Using Statistics: A Biologist's Guide. Hoboken, NJ: Wiley-Blackwell

| | | |
|---|---|---|
| 1 | Testing a clear hypothesis and associated null hypothesis (e.g. $H_1$ = blood glucose level is related to age and $H_0$ = blood glucose is not related to age). | 25 |
| | Not testing any hypothesis but simply want to present, summarize or explore data. | 2 |
| 2 | Methods to summarize and display the data required. | 3 |
| | Data exploration for the purpose of understanding and getting a feel for the data or perhaps to help with formulation of hypotheses. For example, you may wish to find possible groups within the data (e.g. 10 morphological variables have been taken from a large number of carabid beetles; the multivariate test may establish whether they can be divided into separate taxa). | 60 |
| 3 | There is only one collected variable under consideration (e.g. the only variable measured is brain volume although it may have been measured from several different populations). | 4 |
| | There is more than one measured variable (e.g. you have measured the number of algaeper millilitre *and* the water pH in the same sample). | 24 |
| 4 | The data are discrete; there are fewer than 30 different values (e.g. number of species in a sample). | 5 |

Dytham, C (2010) Choosing & Using Statistics: A Biologist's Guide. Hoboken, NJ: Wiley-Blackwell

The data are continuous; there are more than 29 different values (e.g. bee wing length measured to the nearest 0.01 mm). 16
(*Note*: the distinction between the above is rather arbitrary.)

5 There is only one group or sample (e.g. all measurements taken from the same river on the same day). 6
There is more than one group or sample (e.g. you have measured the number of antenna segments in a species of beetle and have divided the sample according to sex to give two groups). 15

6 A graphical representation of the data is required. 7
A numerical summary or description of the data required. 11

7 A display of the whole distribution is required. 8
Crude display of position and spread of data is required: *use a box and whisker display to show medians, range and inter-quartile range, page 49 (also known as a box plot).*

8 Values have real meaning (e.g. number of mammals caught per night). 10
Values are arbitrary labels that have no real sequence (e.g. different vegetation-type classifications in an area of forest). 9

9 There are fewer than 10 different values or classifications: *draw a pie chart, page 52. Ensure that each segment is labelled clearly and that adjacent shading patterns are as distinct as possible. Avoid using three-dimensional or shadow effects, dark shading or colour. Do not add the proportion in figures to the 'piece' of the pie as this information is redundant.*
There are 10 or more different values or classifications: *amalgamate values until there are fewer than 10 or divide the sample to produce two sets each with fewer than 10 values. Ten is a level above which it is difficult to distinguish different sections of the pie or to have sufficiently distinct shading patterns.*

10 There are more than 20 different values: *amalgamate values to produce around 12 classes (almost certainly done automatically by*

Dytham, C (2010) Choosing & Using Statistics: A Biologist's Guide. Hoboken, NJ: Wiley-Blackwell

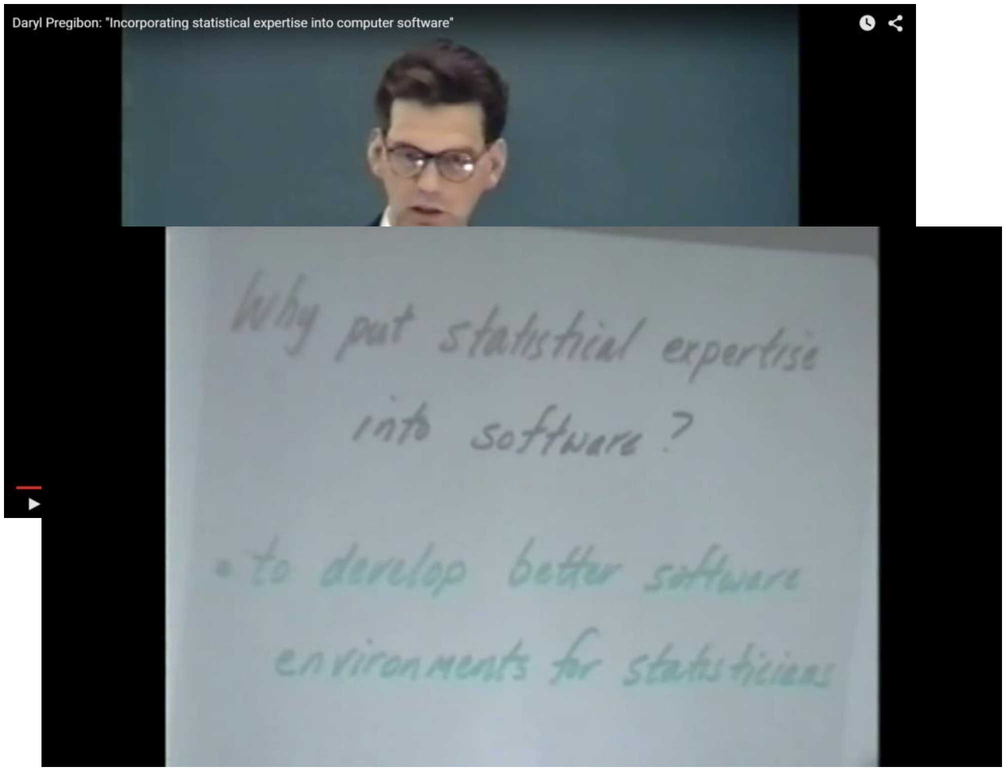
# Building a Statistical Analysis Assistant

# Building a Statistical Analysis Assistant

From 1987…

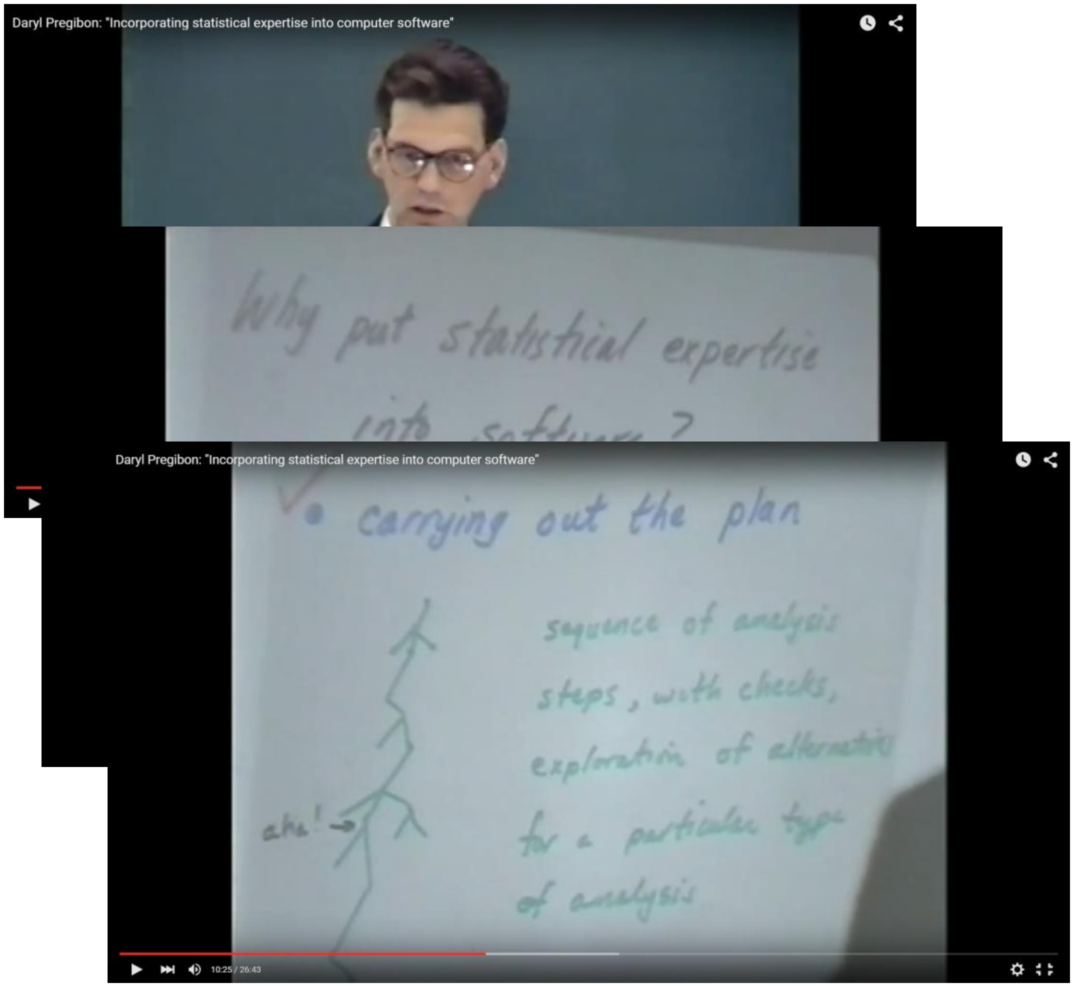
# Building a Statistical Analysis Assistant

From 1987…



# Building a Statistical Analysis Assistant

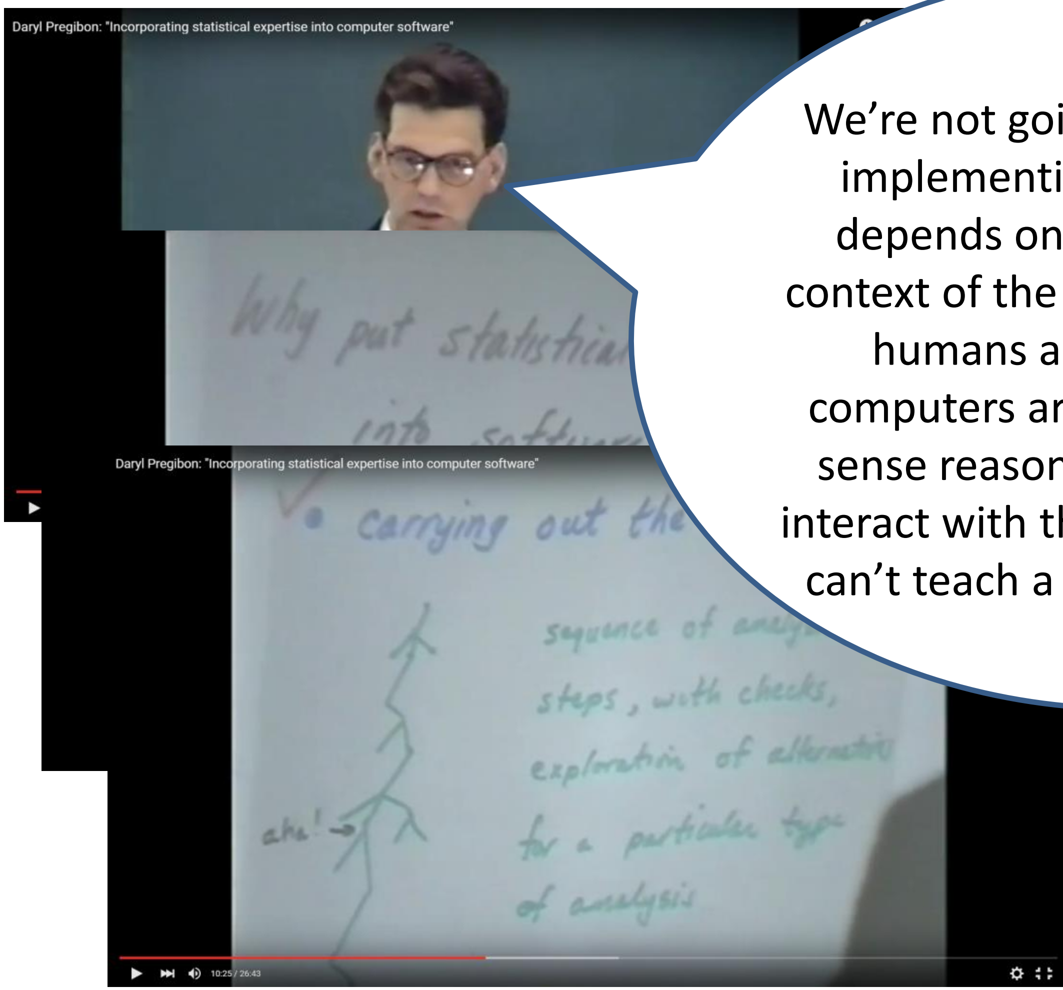
## From 1987…




Pregibon, D. (1987) “Incorporating Statistical Expertise into Computer Software”. 2nd International Tampere Conference in Stats.

# Building a Statistical Analysis Assistant

## From 1987…

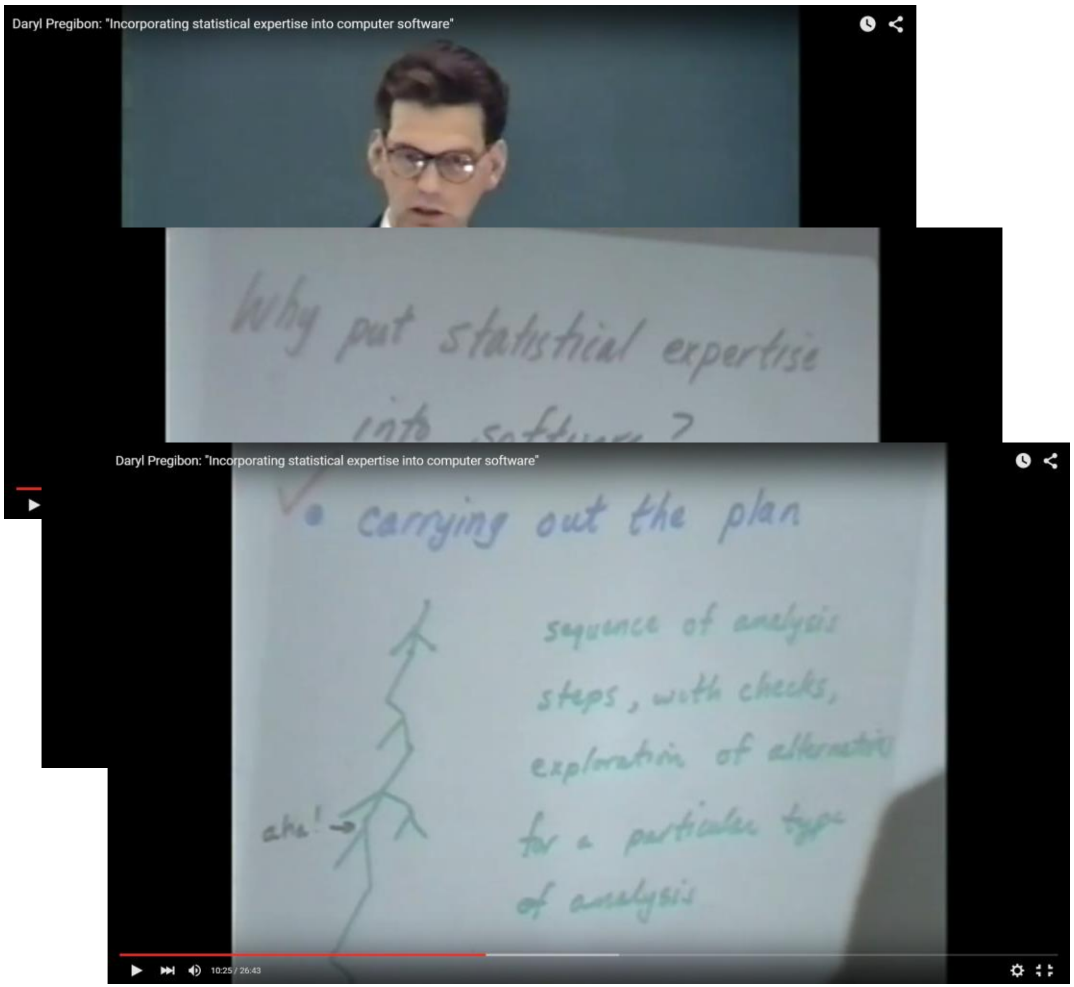



We're not going to be successful in implementing expertise [which depends on understanding] the context of the problem… this is what humans are so good at, and computers are so bad at: common sense reasoning, knowing how to interact with the subject matter – we can't teach a computer that much.

Pregibon, D. (1987) "Incorporating Statistical Expertise into Computer Software". 2nd International Tampere Conference in Stats.

# Building a Statistical Analysis Assistant

## From 1987… …to 2015






As stated in the open letter, the intelligence part of "AI", as it is currently practiced, relates to "statistical and economic notions of rationality – colloquially, the ability to make good decisions, plans, or inferences". And, according to *Superintelligence* author Nick Bostrom, in this context "the ideal is that of the perfect Bayesian agent, one that makes probabilistically optimal use of available information".

The Bayesian ideal "constitutes a kind of optimality notion," Bostrom said. "It specifies what an ideally rational agent would do. Such a thing might be computationally intractable to achieve, but it creates an ideal against which one could measure practicable systems; one can see how far they deviate from this ideal, and one can consider for some possible change to any actual system whether it would take it closer to the Bayesian ideal or further away."

Now, consider again the AI tasked with creating a million paperclips. One might think that it would stop manufacturing the objects once its goal had been reached. But there is a problem, according to Bostrom: "If the AI is a sensible Bayesian agent, it would never assign exactly zero probability to the hypothesis that it has not yet achieved its goal – this, after all, being an empirical hypothesis against which the AI can have only uncertain perceptual evidence.

"The AI should therefore continue to make paperclips in order to reduce the (perhaps astronomically small) probability that it has somehow failed to make at least a million of them, all appearances notwithstanding. There is nothing to be lost by continuing paperclip production and there is always at least some microscopic probability increment of achieving its final goal to be attained."

This might be frustrating more than frightening, at least to begin with. Worry only sets in when the AI starts devoting more and more of Earth's dwindling resources to the creation of ever more paperclips.

Are such concerns overblown? Perhaps. But Google's Peter Norvig (another signatory to the open letter) thinks Bostrom is right to be worried about unintended consequences. "If we build systems that are game-theoretic or utility maximisers, we won't get what we're hoping to get. It's the 'three wishes from the genie problem.' The genie technically delivers, but it isn't what we really want."

# The Automatic Statistician

A Q&A with **Zoubin Ghahramani** discussing his Google-backed project to create an artificial intelligence for data science

Last year, Zoubin Ghahramani won $750 000 in research funding from Google. The money came as a no-strings-attached donation to support a project being led by Ghahramani, professor of information engineering in the University of Cambridge Machine Learning Group.

The project is called the Automatic Statistician. It aims to produce an artificial intelligence for statistics and data science, one that can automate the process of statistical model selection, data analysis and reporting.

**What inspired the Automatic Statistician project, and how has the idea evolved since its initial conception?**

I've had a long-standing interest in probabilistic modelling, and in particular in Bayesian model selection. One of the hardest things when confronted with new data is to know what kinds of models to apply. Each individual researcher may have a small set of modelling tools at his or her disposal and limited patience to implement and test different models. For almost a decade, I had this idea that it would be really interesting and useful to develop an online tool where one could upload data and a computer could try to develop and test models for the data, reporting back what it had discovered.

A nice thing about probabilistic models is that they are compositional, in the sense that you can build more complex models out of simpler ones, like building complex objects out of Lego bricks. Statistics provides excellent methods to determine the appropriate complexity of a model given the data available; I'm thinking of tools from Bayesian nonparametrics, and ideas such as the marginal likelihood (Bayesian model evidence), and cross-validation. Putting these ideas together, along with the explosion of interest in so-called "big data" and the "data sciences", and the demand for expertise in data analysis, created the perfect storm for the Automatic Statistician. What really got it going was working with three brilliant PhD students: James Lloyd, David Duvenaud, and Roger Grosse.

The project started by looking at automatically determining the kernel (or covariance function) for Gaussian process nonlinear regression by composing together simpler kernels. We then moved on to looking

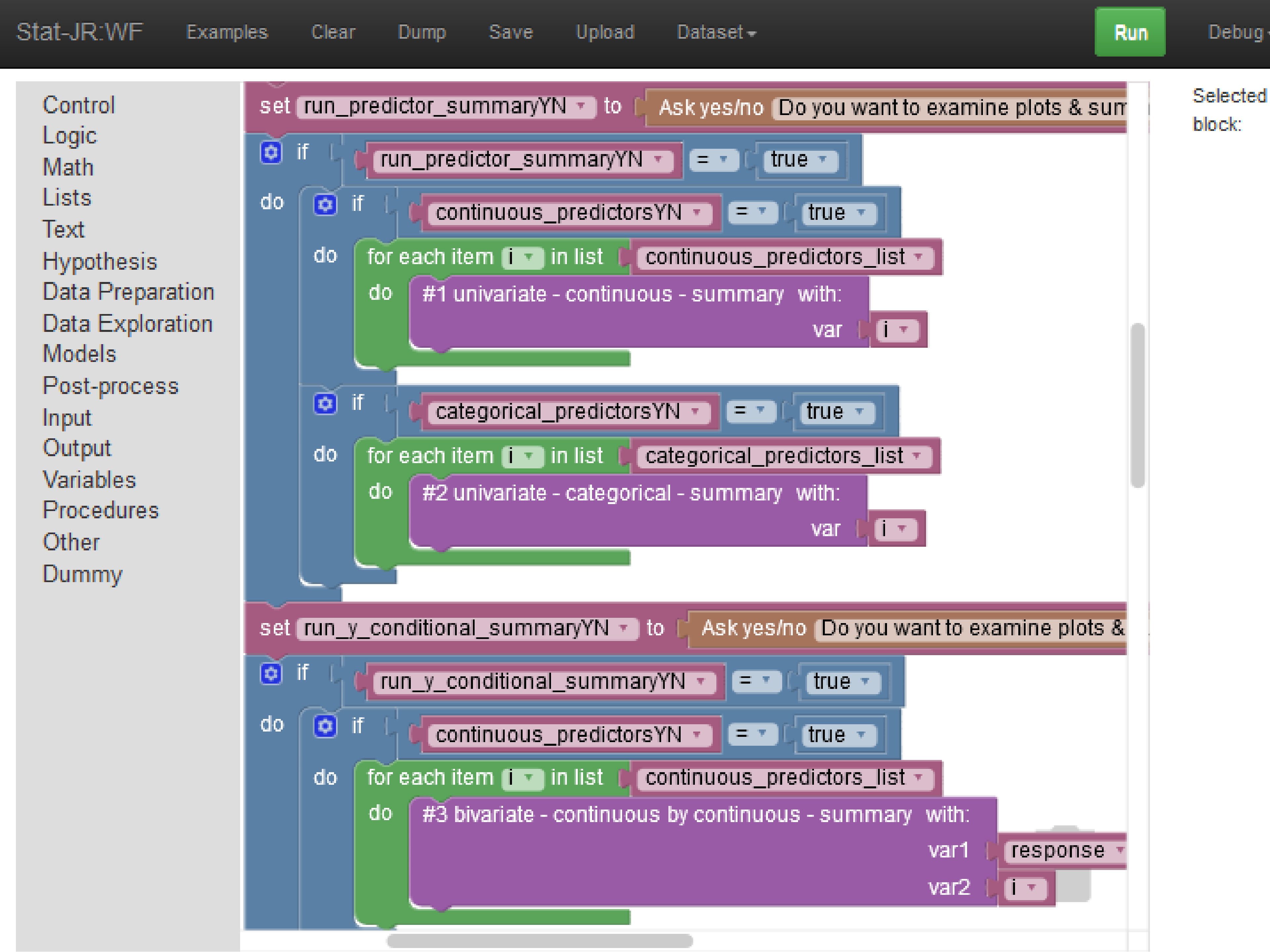
Zoubin Ghahramani (right), with research associate James Lloyd: "We'd really like to provide a tool that is useful to many people. It should help data scientists and statisticians become more productive"

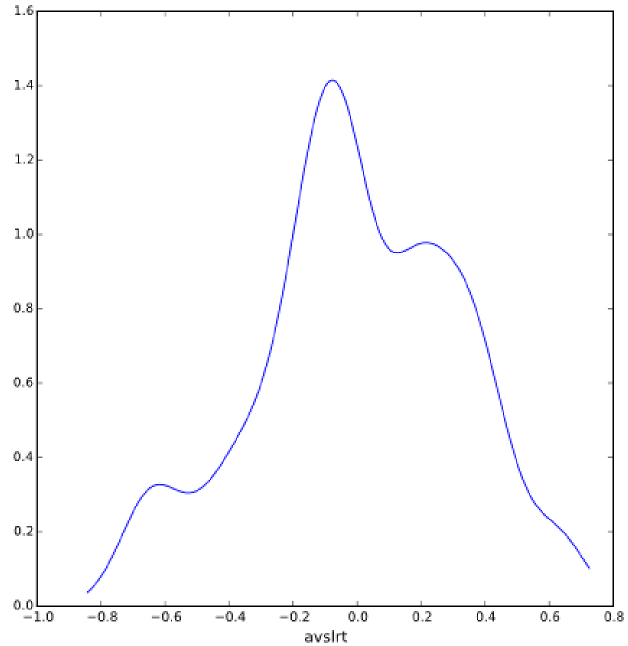




Pregibon, D. (1987) "Incorporating Statistical Expertise into Computer Software". 2nd International Tampere Conference in Stats.

Ghahramani, Z (2015) The Automatic Statistician. *Significance, 12(1)*, 14-15.

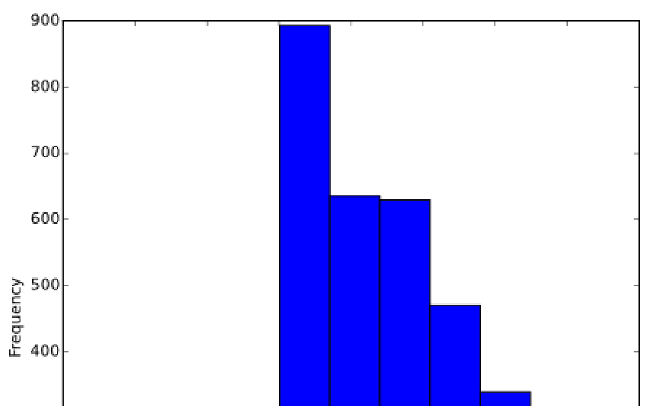

Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i




Block 32 None

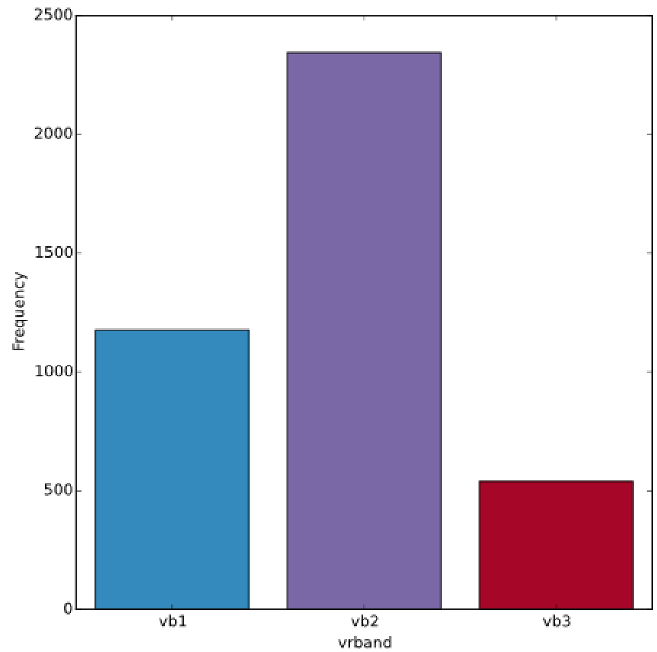
Block 33 None

Block 34 None



Block 56 None

Block 57 None




Block 58 ProcedureCall(#2 univariate - categorical - summary, Variable(i))

Block 59 None

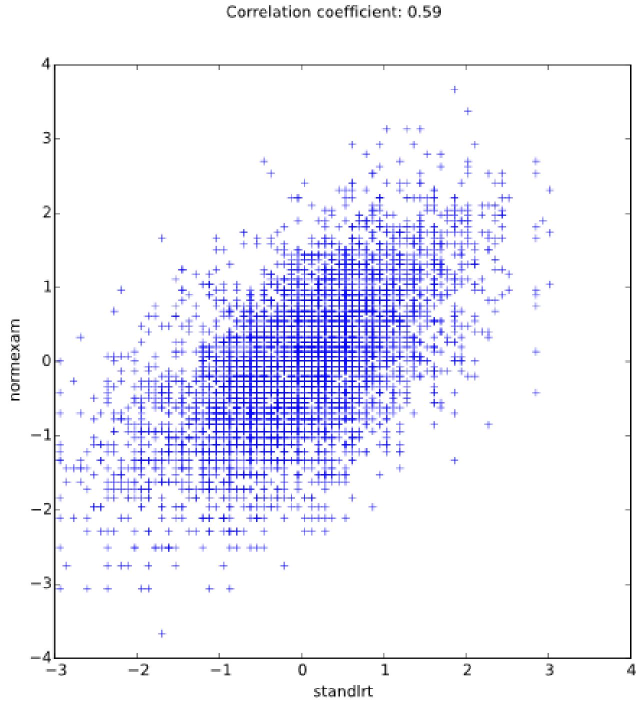
Block 60 None

Block 61 None

Block 62 None

Correlation coefficient: 0.59

Block 62 None



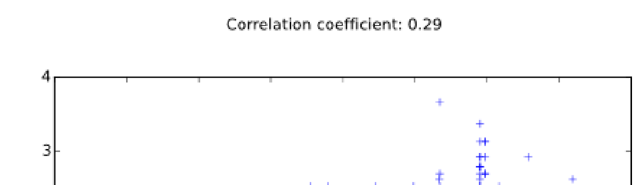

Block 63 ProcedureCall(#3 bivariate - continuous by continuous - summary, Variable(response),Variable(i))

Block 64 None

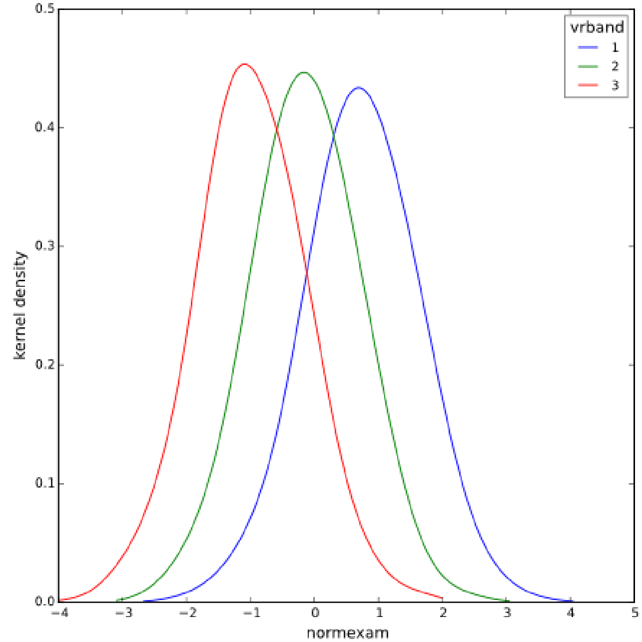
Block 65 None

Block 66 None

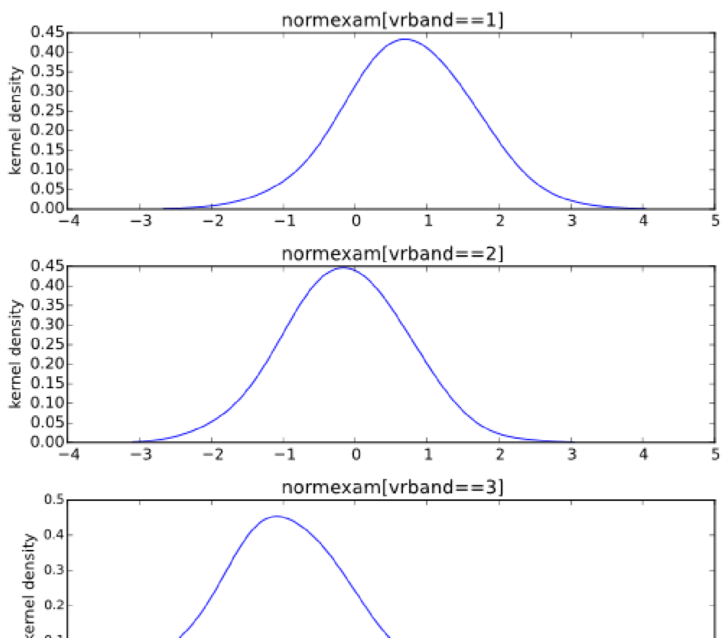
Block 67 None



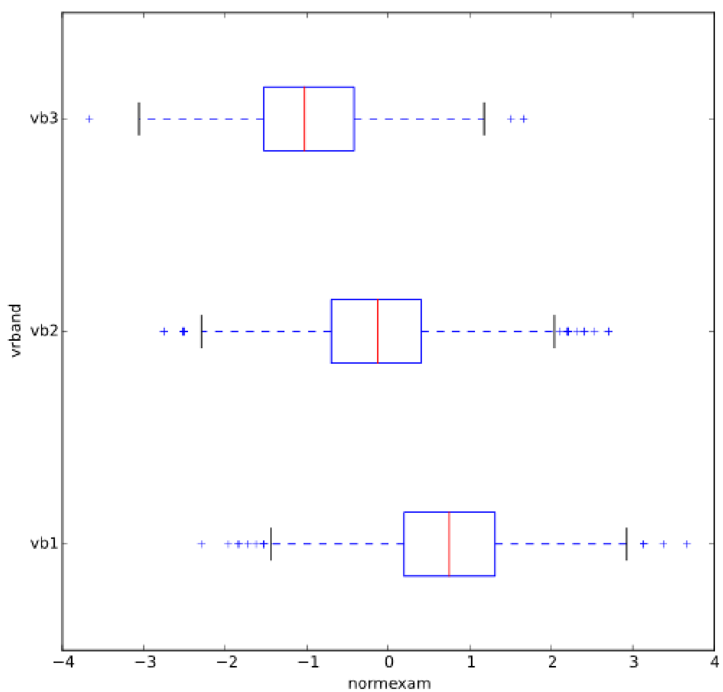
Stat-JR:WF




Block 91 None



Block 95 None




Block 96 None

Block 97 None

| Variable summarised: normexam | vrband == 1 | vrband == 2 | vrband == 3 |
|---|---|---|---|
| N | 1176 | 2344 | 539 |
| Number of missing values | 0 | 0 | 0 |
| Mean | 0.735631883144 | -0.141659572721 | -0.989824473858 |
| Median | 0.747227728367 | -0.129084676504 | -1.02906680107 |
| Min | -2.29173 | -2.75266 | -3.66607 |
| Max | 3.66609 | 2.7018 | 1.66181 |
| SD | 0.856097 | 0.827098 | 0.831329 |
| IQR | 1.115993 | 1.102174 | 1.106852 |

Block 98 ProcedureCall(#4 bivariate - continuous by categorical - summary, Variable(response),Variable(i))

Variable transformation explanatory text

# Performing the transformations

Here's how the transformations you asked for were calculated.

## Log transformation

Since **standlrt**$_i$ has values equal to or less than zero, a constant was first added prior to the log-transformation (it's not possible to get log values for negative numbers nor for zero). Here a constant was added to bring the minimum value up to 1 (and thus the log-transformed minimum value is 0), although we could have chosen other constants. In the outputted dataset, the transformed variable appears as **loge_standlrt_plus_cons**.

$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 1 + |\mathbf{standlrt}_{min}|)$$
$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 1 + 2.935)$$
$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 3.935)$$

## Square root transformation

Since **standlrt**$_i$ has negative values, a constant was first added prior to the square-root-transformation (there's no square-root for negative values). Here we've just added the absolute value of the minimum value of **standlrt**, so that the mimimum value prior to square-root-transformation is now zero (but we could have used a different constant). In the outputted dataset, the transformed variable appears as **sqrt_standlrt_plus_cons**.

$$\mathbf{sqrt\_standlrt\_plus\_cons}_i = \sqrt{\mathbf{standlrt}_i + |\mathbf{standlrt}_{min}|}$$
$$\mathbf{sqrt\_standlrt\_plus\_cons}_i = \sqrt{\mathbf{standlrt}_i + 2.935}$$

Variable transformation explanatory text

# Performing the transformations

Here's how the transformations you asked for were calculated.

## Log transformation

Since **standlrt$_i$** has values equal to or less than zero, a constant was first added prior to the log-transformation (it's not possible to get log values for negative numbers nor for zero). Here a constant was added to bring the minimum value up to 1 (and thus the log-transformed minimum value is 0), although we could have chosen other constants. In the outputted dataset, the transformed variable appears as **loge_standlrt_plus_cons**.

$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 1 + |\mathbf{standlrt}_{min}|)$$
$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 1 + 2.935)$$
$$\mathbf{loge\_standlrt\_plus\_cons}_i = \ln(\mathbf{standlrt}_i + 3.935)$$

E.g. talking user through some transformations they may consider…

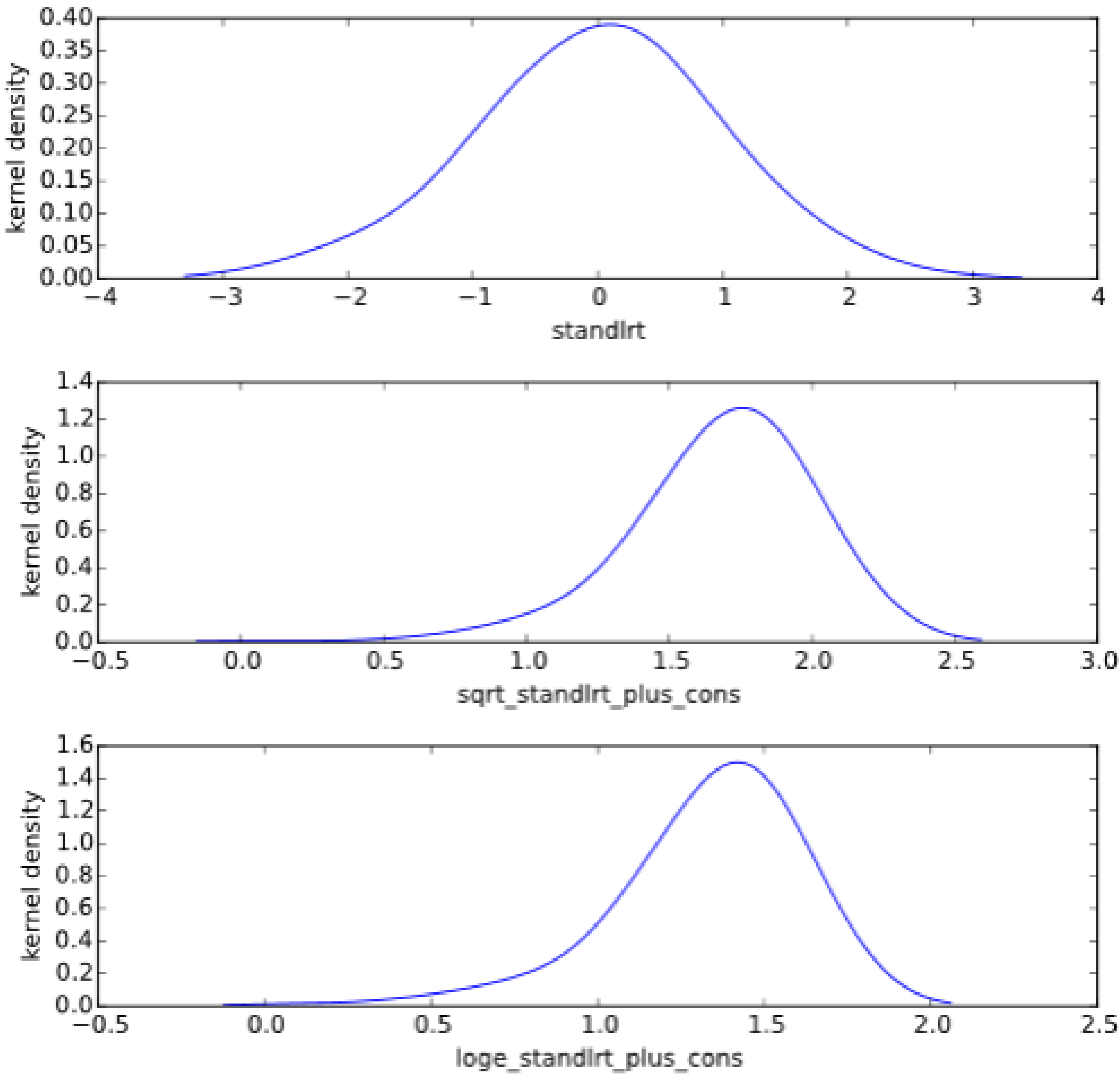
## Square root transformation

Since **standlrt$_i$** has negative values, a constant was first added prior to the square-root-transformation (there's no square-root for negative values). Here we've just added the absolute value of the minimum value of **standlrt**, so that the mimimum value prior to square-root-transformation is now zero (but we could have used a different constant). In the outputted dataset, the transformed variable appears as **sqrt_standlrt_plus_cons**.

$$\mathbf{sqrt\_standlrt\_plus\_cons}_i = \sqrt{\mathbf{standlrt}_i + |\mathbf{standlrt}_{min}|}$$
$$\mathbf{sqrt\_standlrt\_plus\_cons}_i = \sqrt{\mathbf{standlrt}_i + 2.935}$$

| Percentile | standlrt | standlrt with cons (for sqrt) | sqrt_standlrt_plus_cons | standlrt with cons (for log) | loge_standlrt_plus_cons |
|---|---|---|---|---|---|
| **0%** | -2.935 | 0 | 0 | 1.0 | 0 |
| **2.5%** | -2.108 | 0.909 | 0.909 | 0.602 | 0.602 |
| **25%** | -0.621 | 1.521 | 1.521 | 1.198 | 1.198 |
| **50%** | 0.0405 | 1.725 | 1.725 | 1.38 | 1.38 |
| **75%** | 0.619 | 1.885 | 1.885 | 1.516 | 1.516 |
| **97.5%** | 1.941 | 2.208 | 2.208 | 1.771 | 1.771 |
| **100%** | 3.016 | 5.951 | 2.439 | 6.951 | 1.939 |

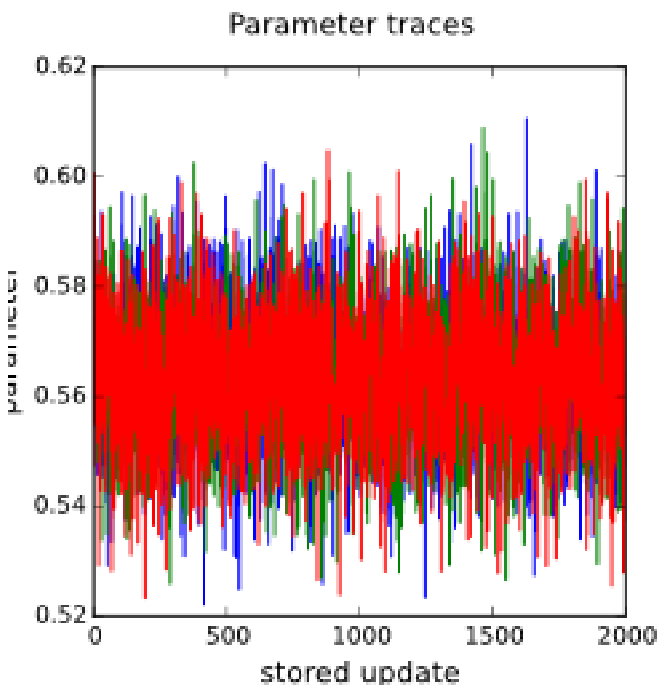
E.g. talking user through some transformations they may consider…

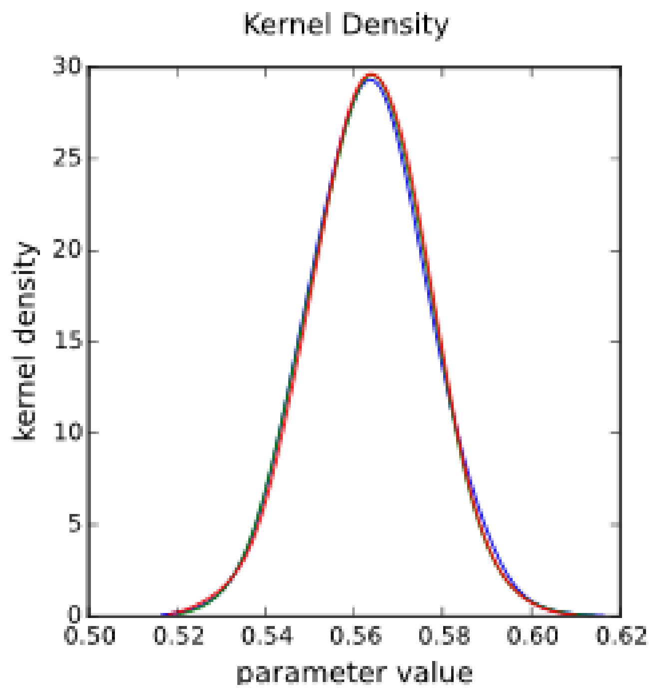



E.g. talking user through some transformations they may consider…

## Block 16 OutputObject(mcmctext)

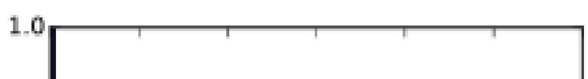
MCMC estimation methods are simulation based which means that rather than a point estimate (and accompanying standard error) for each parameter they instead produce a (dependent) chain of values from the posterior distribution of the parameter. In fact in Stat-JR several chains are run from differing starting values/ random number seeds and so for each parameter we have several chains of values that can be combined to summarise the parameter. For parameter beta_1 we can first look at the posterior mean which has value 0.563 and standard deviation of the chain which has value 0.0125 and plays the role of standard error for the parameter. We might also consider the posterior median which has value 0.563 as an alternative if the distribution is not symmetric. Here the median is close to the mean as the posterior is reasonably symmetric. We can use the quantiles of the distribution and so we see a 95% credible interval for beta_1 is 0.539 to 0.588. We can look at the 3 chains for the parameter beta_1and we can also look at kernel density plots (which are like smoothed histograms) of the 3 chains on a single plot:

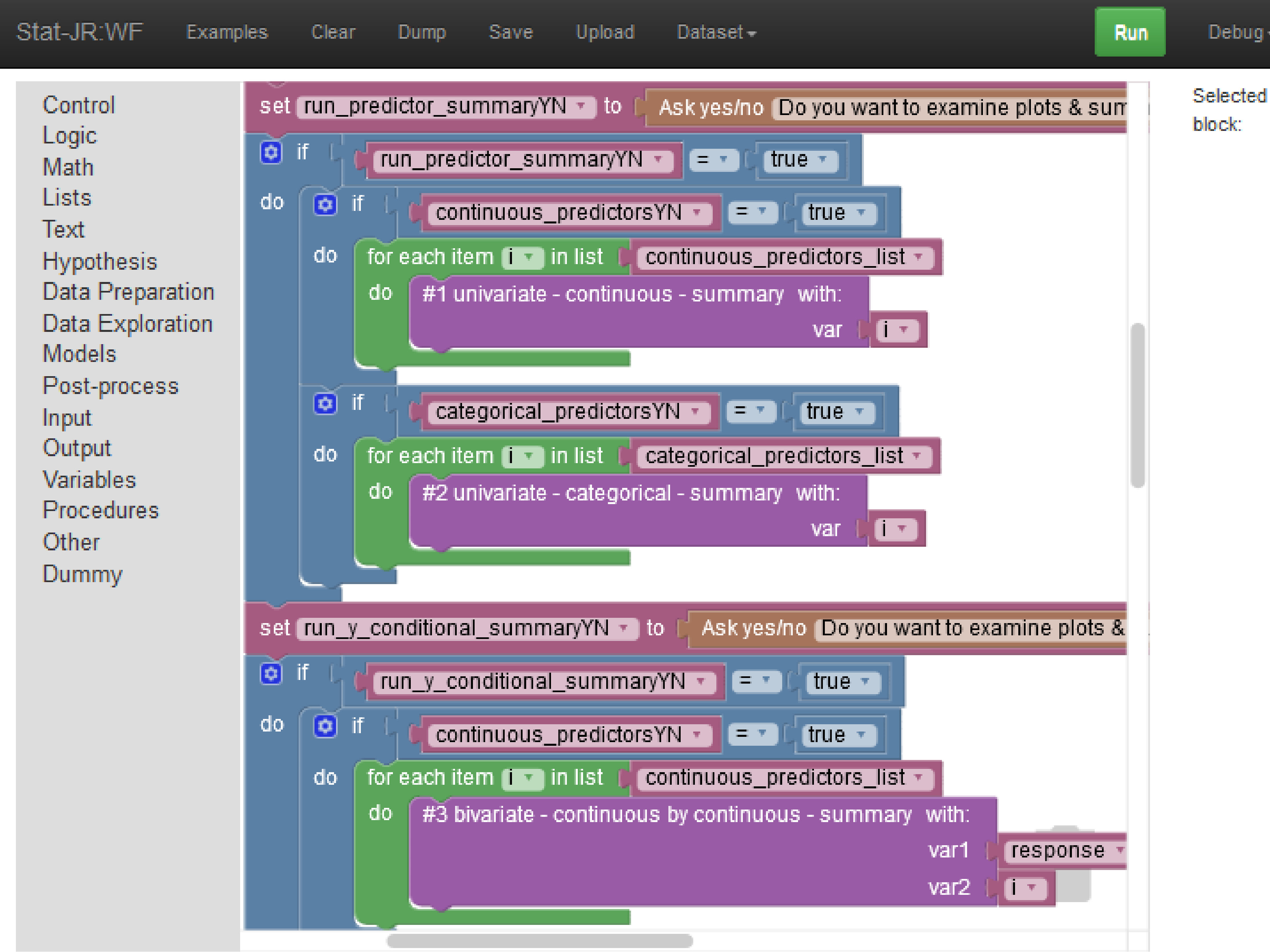


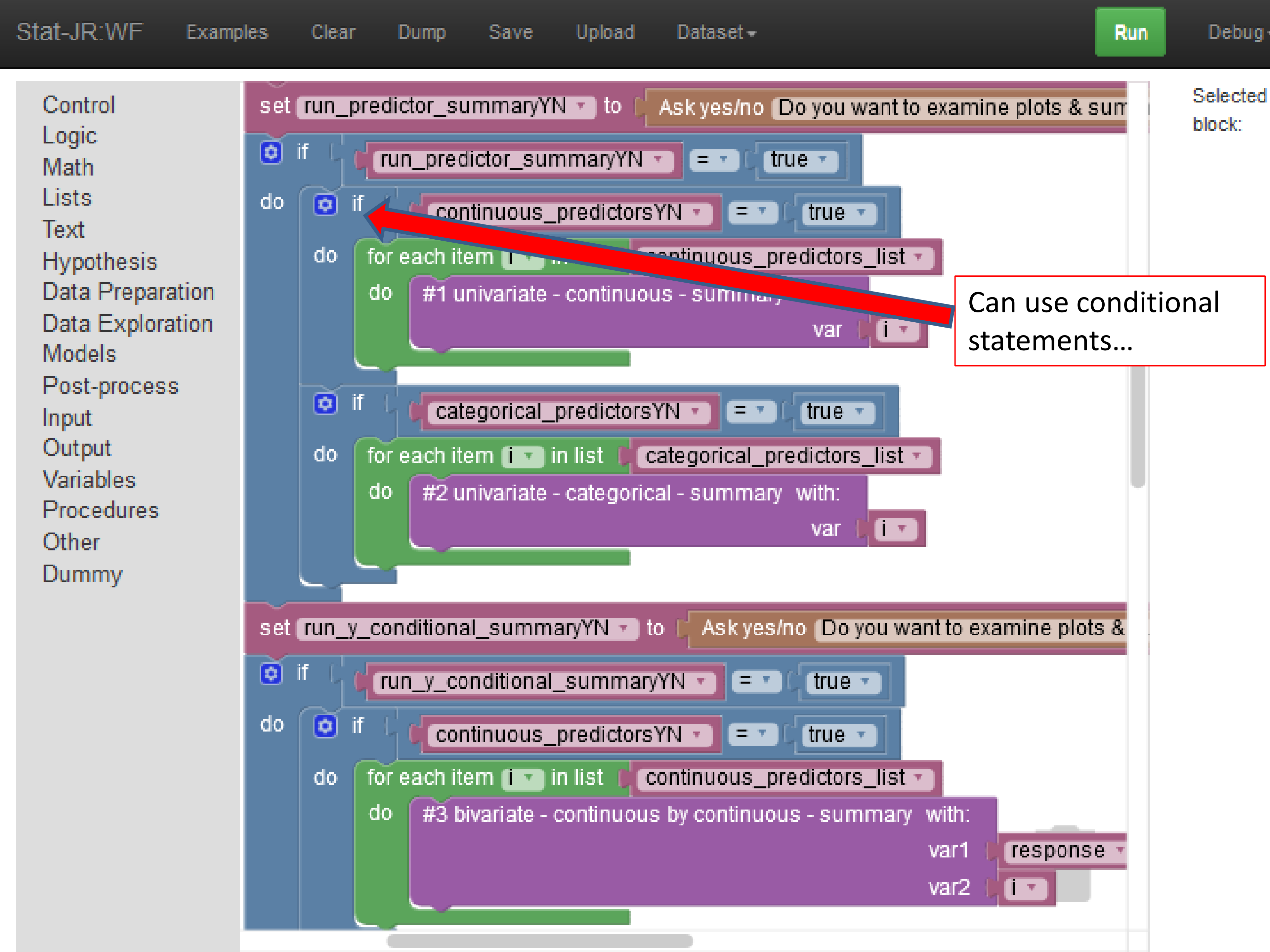



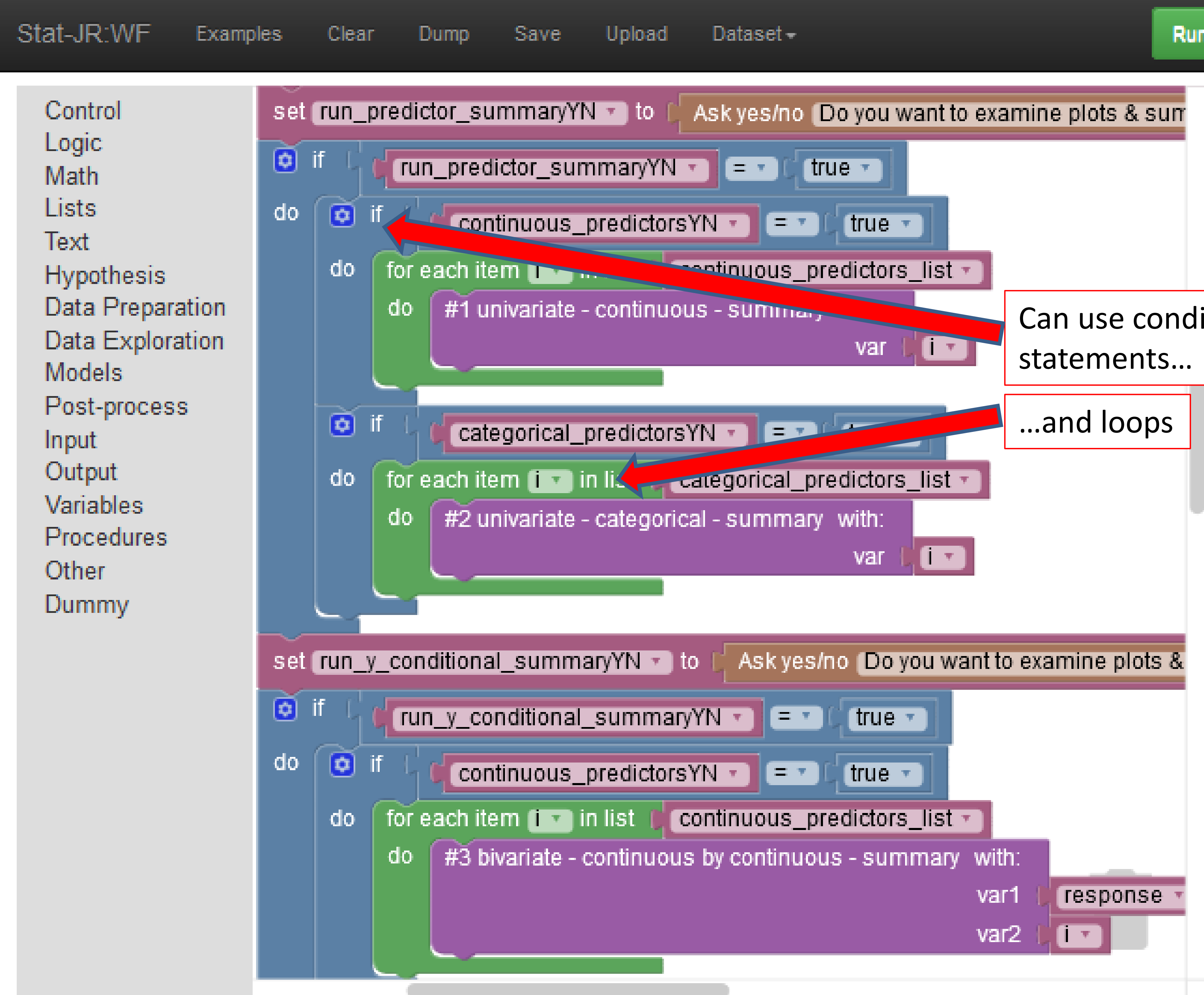

…and more complex examples, such as explaining elements of MCMC estimation

Due to the nature of MCMC algorithms updating parameters in separate steps there is some dependence in the parameter chains produced. One way of investigating this is to look at auto-correlation functions (acf) for the chains. Essentially an acf examines how correlated a chain of values is with a similar chain shifted by a number of iterations (the lag). We can plot such a function for a series of lags as shown below.




Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Can use conditional statements…




Can use conditional statements…

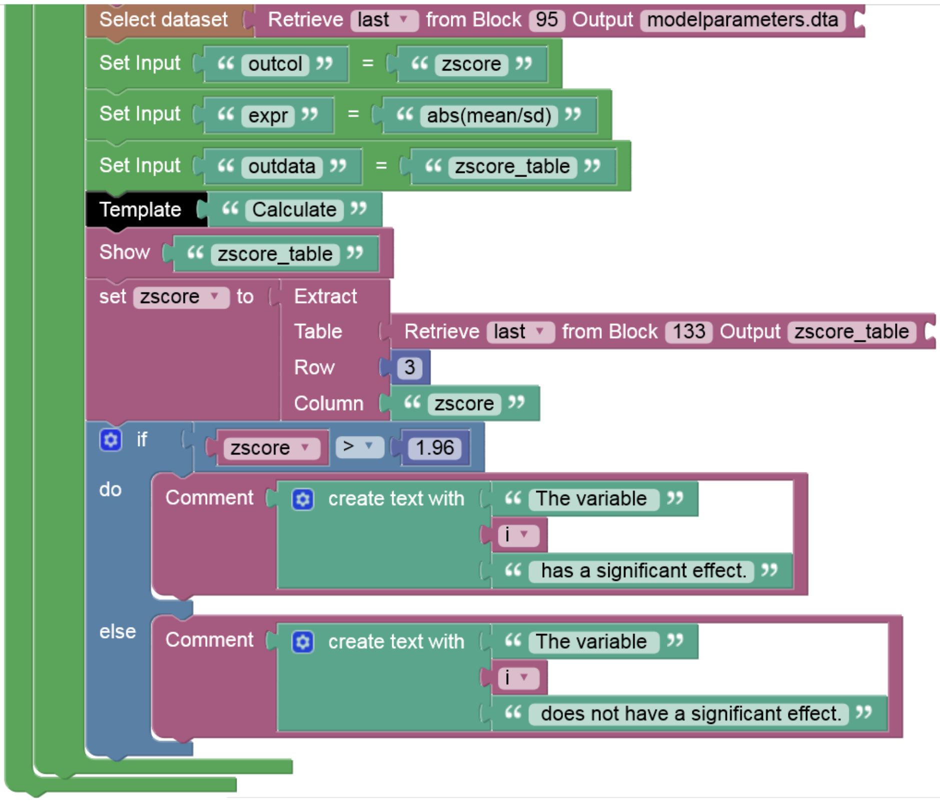
…and loops

# Practical 2…

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

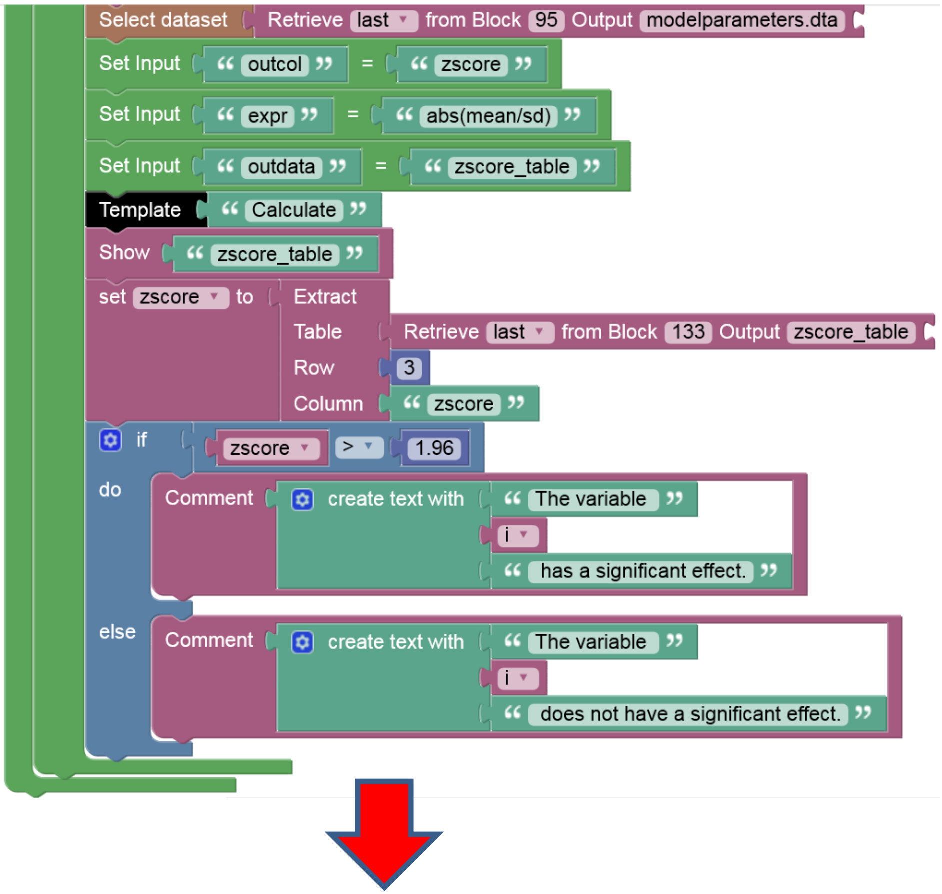
# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

Achieving many of the key operations via workflow blocks…



# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

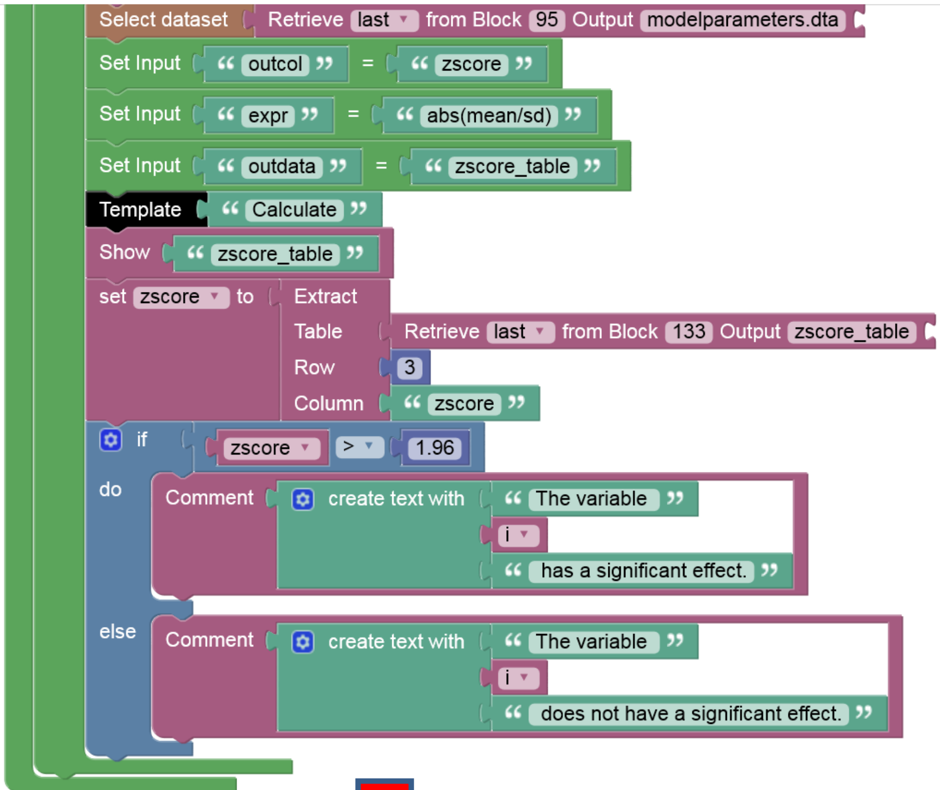
Achieving many of the key operations via workflow blocks…




The variable avslrt has a significant effect.

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

Achieving many of the key operations via workflow blocks…




The variable avslrt has a significant effect.

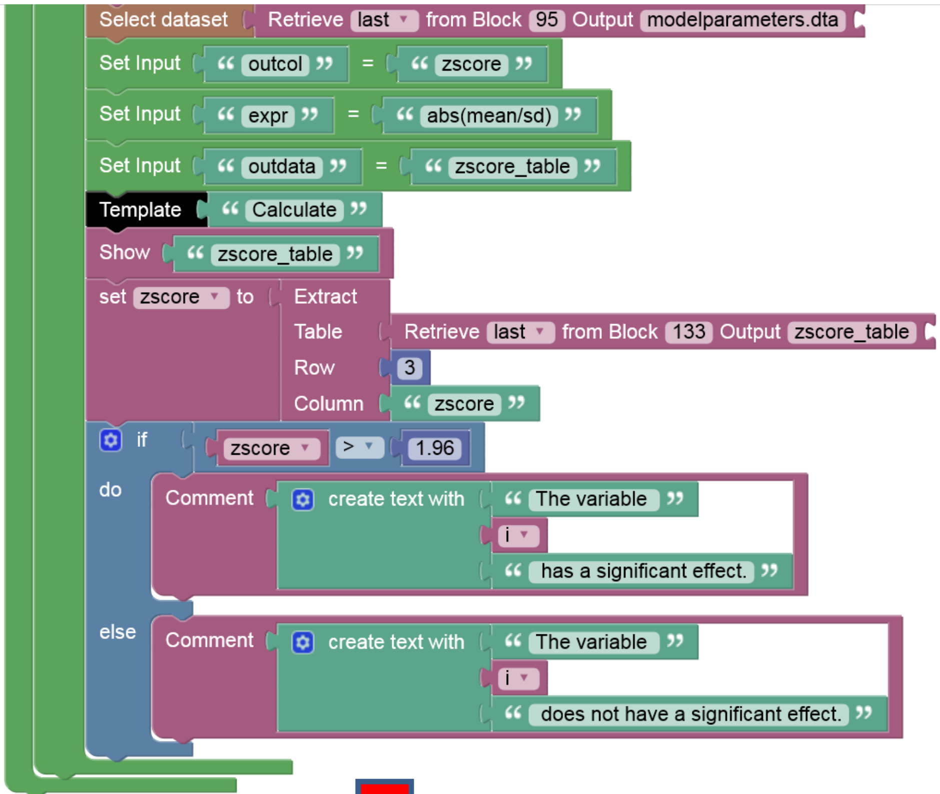
Putting it all in one template (or in a supertemplate)…

```
outvar = abs(mean/sd)

retval.nobs = len(localvars[localvars.keys()[0]])
for k in localvars.keys():
    retval.addvariable(k, data = localvars[k])

retval.addvariable(outcol, data = outvar)
outputs[str(outdata)] = retval

html = ""
for i in range(0, len(outvar)):
    if str(variable_name[i]) != "--":
        if outvar[i] > 1.96:
            html += "<p>The variable <strong>" +
str(variable_name[i]) + "</strong> has a significant
effect.</p>"
        else:
            html += "<p>The variable <strong>" +
str(variable_name[i]) + "</strong> does <em>not</em>
have a significant effect.</p>"

outputs['ztext'] = HTMLOutput(html, description = 'z
text')
'''
```

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

Achieving many of the key operations via workflow blocks…

Putting it all in one template (or in a supertemplate)…




```
outvar = abs(mean/sd)

retval.nobs = len(localvars[localvars.keys()[0]])
for k in localvars.keys():
    retval.addvariable(k, data = localvars[k])

retval.addvariable(outcol, data = outvar)
outputs[str(outdata)] = retval

html = ""
for i in range(0, len(outvar)):
    if str(variable_name[i]) != "--":
        if outvar[i] > 1.96:
            html += "<p>The variable <strong>" +
str(variable_name[i]) + "</strong> has a significant
effect.</p>"
        else:
            html += "<p>The variable <strong>" +
str(variable_name[i]) + "</strong> does <em>not</em>
have a significant effect.</p>"

outputs['ztext'] = HTMLOutput(html, description = 'z
text')
'''
```

The variable avslrt has a significant effect.

The variable **cons** does *not* have a significant effect.

The variable **standlrt** has a significant effect.

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

Achieving key operations via XPath queries in eBook html documents…

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat…

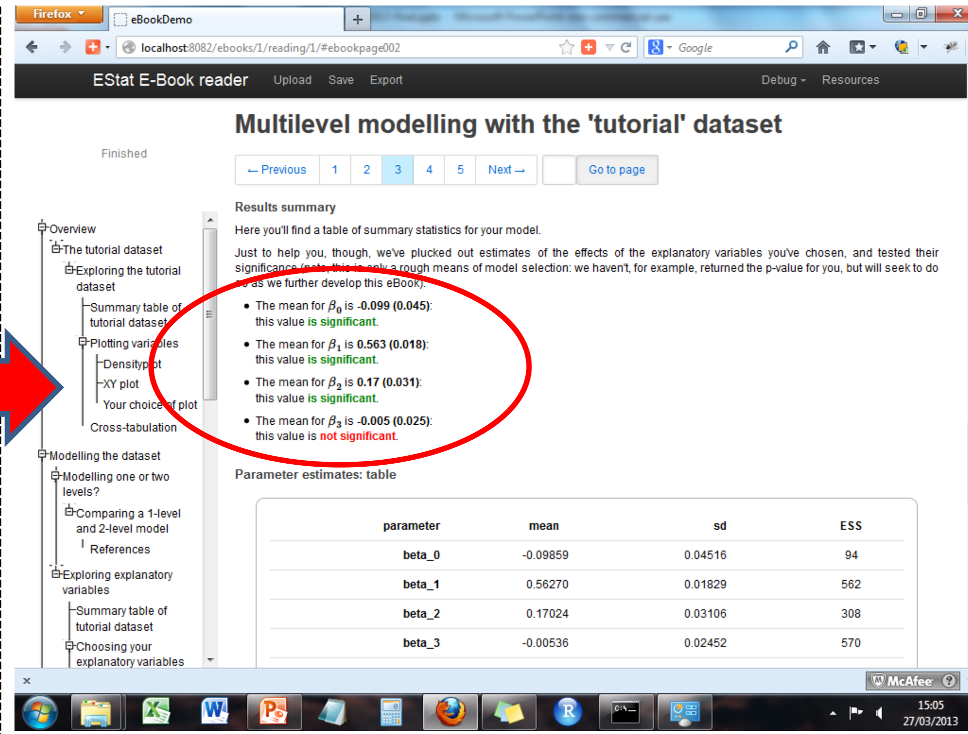
Achieving key operations via XPath queries in eBook html documents…

```
<ul>
<li class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="//row[@row='beta_0']/element[@col='Mean'] ">The mean for <strong>\(\beta_0\)</strong> is<strong>
<span class="deep_dynamic_output" data-deep-id="template2-out-summary" data-deep-expression="round(1000*(//row[@row='beta_0']/element[@col='Mean'])) div 1000"></span>
(<span class="deep_dynamic_output" data-deep-id="template2-out-summary" data-deep-expression="round(1000*(//row[@row='beta_0']/element[@col='Std'])) div 1000"></span>)</strong>:
<p class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="//row[@row='beta_0']/element[@col='Mean'] > //row[@row='beta_0']/element[@col='Std']*1.96 or
//row[@row='beta_0']/element[@col='Mean'] < //row[@row='beta_0']/element[@col='Std']*(-1.96) ">
this value <strong><font color="green">is significant</font></strong>.
</p>
<p class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="not(//row[@row='beta_0']/element[@col='Mean'] > //row[@row='beta_0']/element[@col='Std']*1.96 or
//row[@row='beta_0']/element[@col='Mean'] < //row[@row='beta_0']/element[@col='Std']*(-1.96)) ">
this value is <strong><font color="red">not significant</font></strong>.
</p>
</li>
```

# Using Stat-JR tools to add functionality and content: many ways to skin proverbial cat...

Achieving key operations via XPath queries in eBook html documents...

```
<ul>
<li class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="//row[@row='beta_0']/element[@col='Mean'] ">The mean for <strong>\(\beta_0\)</strong> is<strong>
<span class="deep_dynamic_output" data-deep-id="template2-out-summary" data-deep-expression="round(1000*(//row[@row='beta_0']/element[@col='Mean'])) div 1000"></span>
(<span class="deep_dynamic_output" data-deep-id="template2-out-summary" data-deep-expression="round(1000*(//row[@row='beta_0']/element[@col='Std'])) div 1000"></span>)</strong>:
<p class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="//row[@row='beta_0']/element[@col='Mean'] > //row[@row='beta_0']/element[@col='Std']*1.96 or
//row[@row='beta_0']/element[@col='Mean'] < //row[@row='beta_0']/element[@col='Std']*(-1.96) ">
this value <strong><font color="green">is significant</font></strong>.
</p>
<p class="deep_dynamic_hidden" data-deep-showon="template2-out-summary" data-deep-expression="not(//row[@row='beta_0']/element[@col='Mean'] > //row[@row='beta_0']/element[@col='Std']*1.96 or
//row[@row='beta_0']/element[@col='Mean'] < //row[@row='beta_0']/element[@col='Std']*(-1.96)) ">
this value is <strong><font color="red">not significant</font></strong>.
</p>
</li>
```



eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
    - library of case studies
    - library of methodological advice / notes
    - statistical analysis assistant

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
    - library of case studies
    - library of methodological advice / notes
    - statistical analysis assistant

# Working with a range of social scientists…

# Working with a range of social scientists…

- …we ask them to choose a quantitative research question they have investigated…

# Working with a range of social scientists…

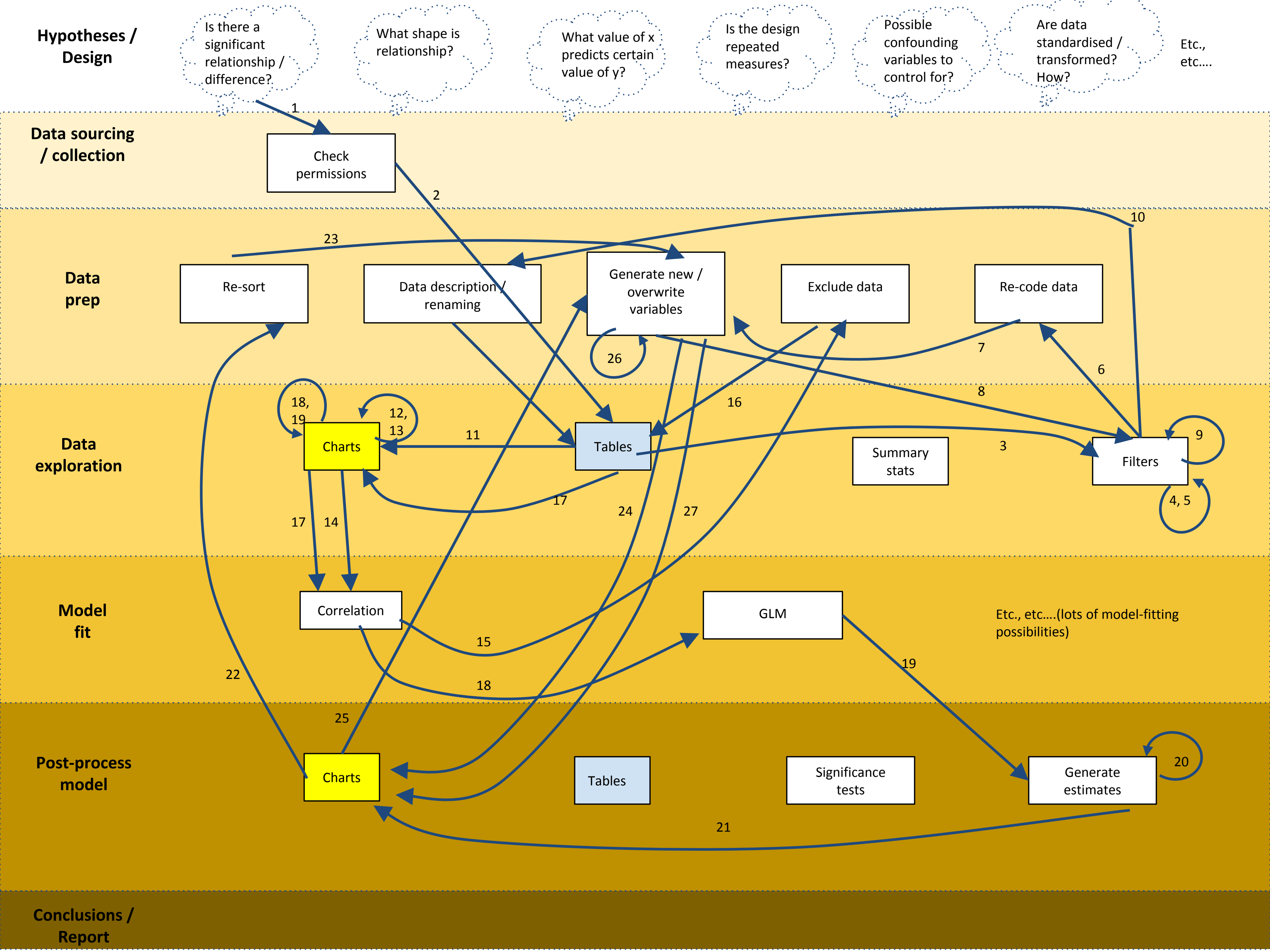
- …we ask them to choose a quantitative research question they have investigated…
- …and then work with them to develop it into an interactive case study using Stat-JR.

# Working with a range of social scientists…

- …we ask them to choose a quantitative research question they have investigated…
- …and then work with them to develop it into an interactive case study using Stat-JR.
- User will be able to interact with this resource: e.g. investigating alternative avenues the analyst may have taken.

# Working with a range of social scientists…

- …we ask them to choose a quantitative research question they have investigated…
- …and then work with them to develop it into an interactive case study using Stat-JR.
- User will be able to interact with this resource: e.g. investigating alternative avenues the analyst may have taken.
- Aim is to help demystify the process of quantitative research, and shed light on the day-to-day decisions working analysts make.



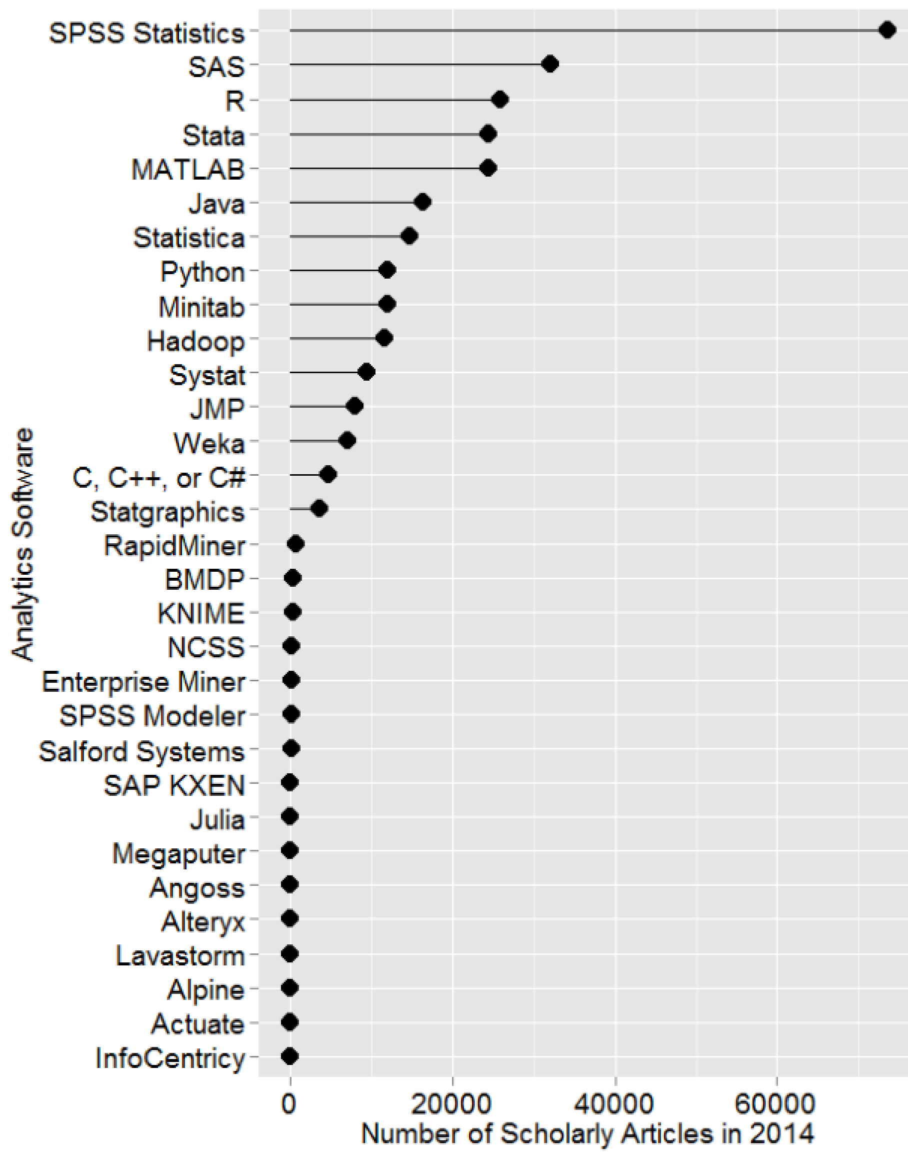
Hypotheses / Design
Is there a significant relationship / difference?
What shape is relationship?
What value of x predicts certain value of y?
Is the design repeated measures?
Possible confounding variables to control for?
Are data standardised / transformed? How?
Etc., etc....
Data sourcing / collection
Check permissions
Data prep
Re-sort
Data description / renaming
Generate new / overwrite variables
Exclude data
Re-code data
Data exploration
Charts
Tables
Summary stats
Filters
Model fit
Correlation
GLM
Etc., etc....(lots of model-fitting possibilities)
Post-process model
Charts
Tables
Significance tests
Generate estimates
Conclusions / Report
1
2
3
4, 5
6
7
8
9
10
11
12, 13
14
15
16
17
18
18, 19
19
20
21
22
23
24
25
26
27

- …as even this simple analysis shows, the path taken when investigating a quantitative research question can be a convoluted one!

- …as even this simple analysis shows, the path taken when investigating a quantitative research question can be a convoluted one!
- Workflow interface may help abstract main elements, and illustrate how its components fit thematically together.

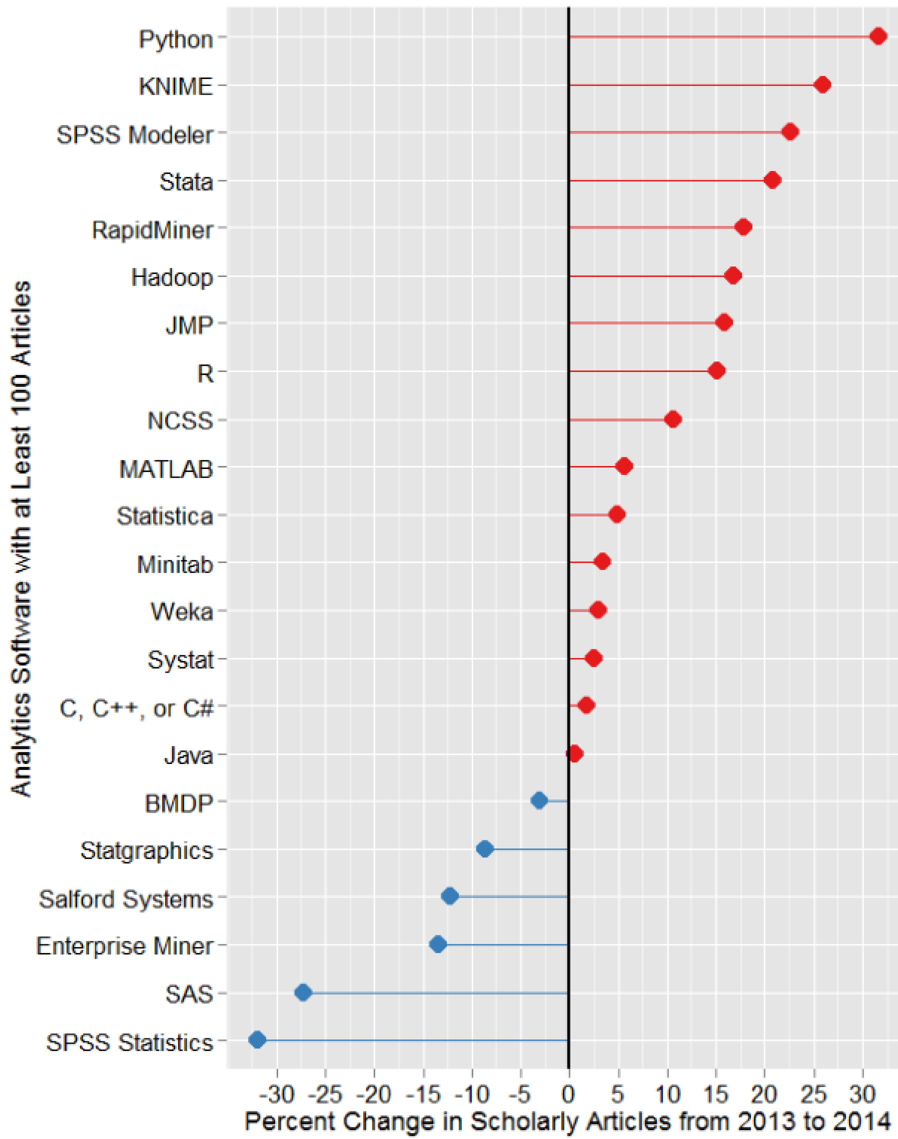
- …as even this simple analysis shows, the path taken when investigating a quantitative research question can be a convoluted one!
- Workflow interface may help abstract main elements, and illustrate how its components fit thematically together.
- Workflow environment will eventually be integrated into our eBook interface.

Number of scholarly articles found in the most recent complete year (2014) for each software package.



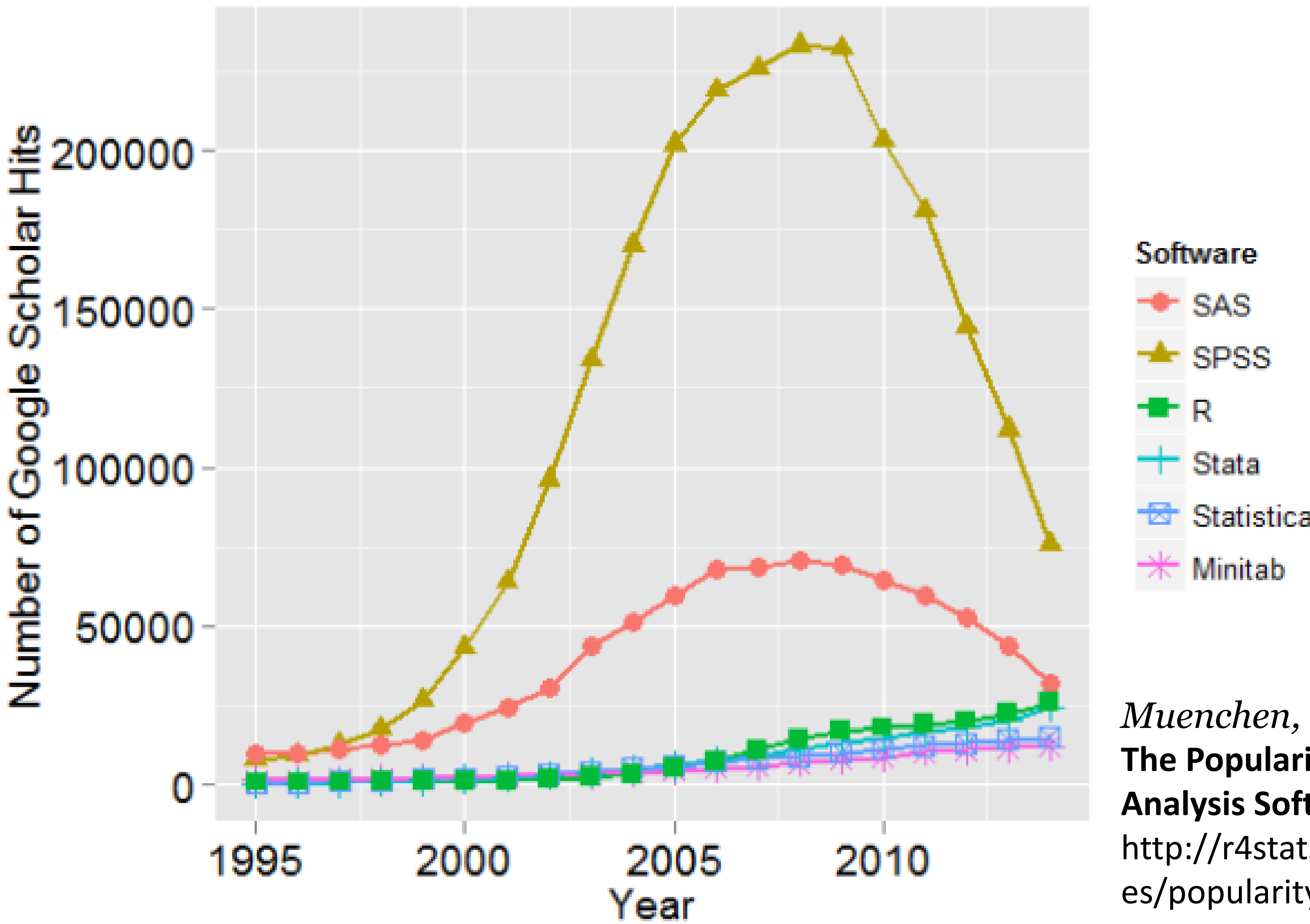

*Muenchen, RA.*
**The Popularity of Data Analysis Software**
http://r4stats.com/articles/popularity/

Change in the number of scholarly articles using each software in the most recent two complete years (2013 to 2014). Packages shown in red are “hot” and growing, while those shown in blue are “cooling down” or declining.



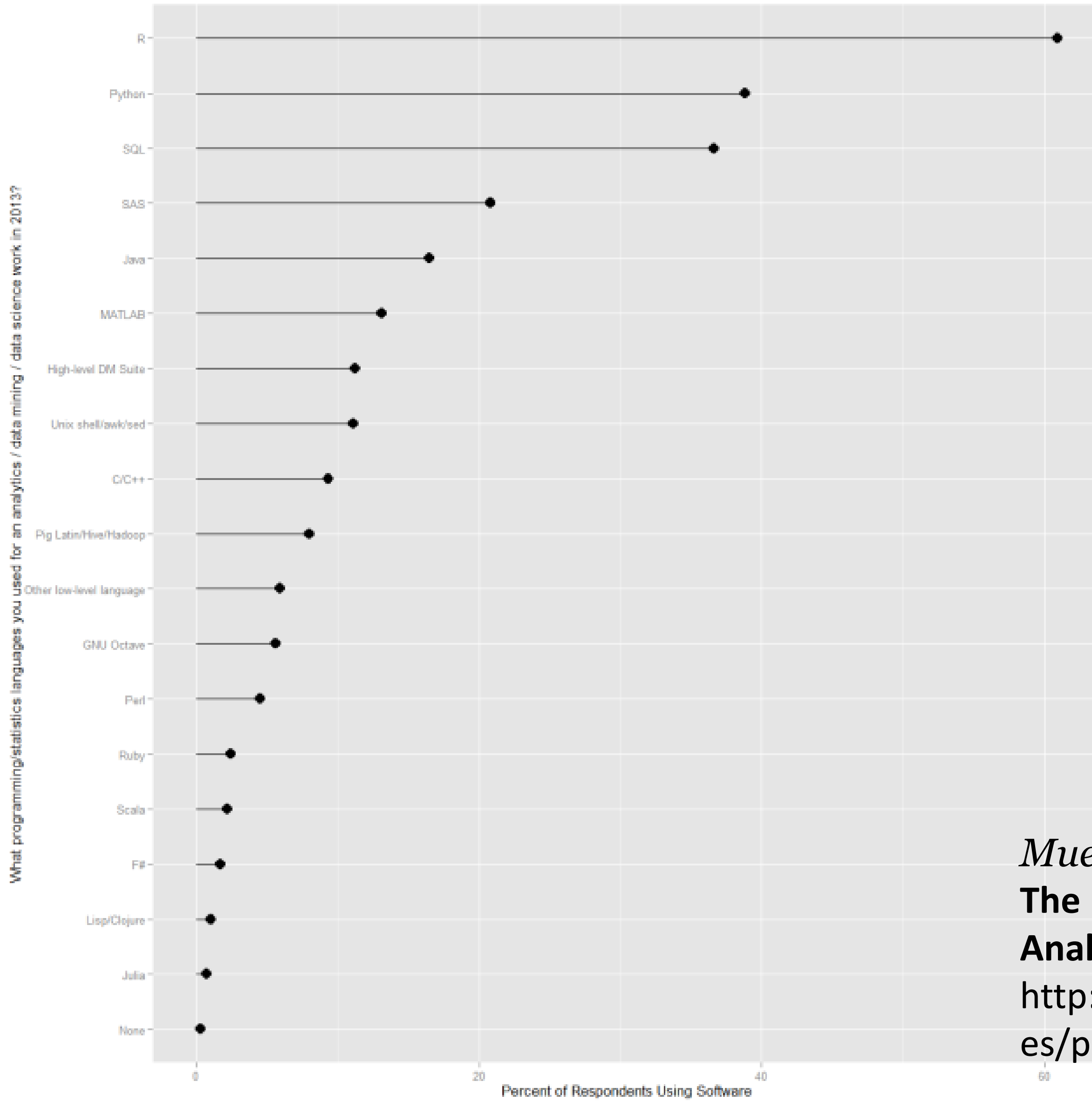

*Muenchen, RA.*
**The Popularity of Data Analysis Software**
http://r4stats.com/articles/popularity/

The number of scholarly articles found in each year by Google Scholar. Only the top six "classic" statistics packages are shown




*Muenchen, RA.*
**The Popularity of Data Analysis Software**
http://r4stats.com/articles/popularity/

KDnuggets poll on programming tools used for an analytics / data mining / data science work in 2013




*Muenchen, RA.*
**The Popularity of Data Analysis Software**
http://r4stats.com/articles/popularity/

# eBook project funded by ESRC

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
  - library of case studies
  - library of methodological advice / notes
  - statistical analysis assistant

# eBook project funded by ESRC

E·S·R·C ECONOMIC & SOCIAL RESEARCH COUNCIL

Research objectives include:

- Developing tools to support interactive eBooks / workflows for statistical analyses
- Using these tools to produce:
    - library of case studies
    - library of methodological advice / notes
    - statistical analysis assistant

# Module 3: Multiple Regression
# MLwiN Practicals

*Fiona Steele*[1]
Centre for Multilevel Modelling

## Contents

## P3.1 Regression with a Single Continuous Explanatory Variable

We will begin by looking at the relationship between attainment (SCORE) and cohort (COHORT90). Has attainment changed over time and, if so, is the trend linear?

### P3.1.1 Examining the data

To access the data files associated with this tutorial, you must have an account with LEMMA. To open the first data file,

From within the LEMMA Learning Environment

- Go to **Module 3: Multiple regression**, and scroll down to ***MLwiN* Datafiles**
- If you do not already have MLwiN to open the datafile with, click (**get *MLwiN***).
- Click "3.1.wsz"

When the worksheet is opened, the filename will appear in the title bar of the main window. The **Names** window will also appear, giving a summary of the data in the worksheet:

Names

Edit name | Toggle Categorical | Categories | Description | Copy | Paste | Delete | Data | Help

| Name | Cn | n | missing | min | max | categorical | description |
|---|---|---|---|---|---|---|---|
| CASEID | 1 | 33988 | 0 | 1 | 38192 | False | |
| SCORE | 2 | 33988 | 0 | 0 | 75 | False | |
| COHORT90 | 3 | 33988 | 0 | -6 | 8 | False | |
| FEMALE | 4 | 33988 | 0 | 0 | 1 | False | |
| SCLASS | 5 | 33988 | 0 | 1 | 4 | False | |
| C6 | 6 | 0 | 0 | 0 | 0 | False | |
| C7 | 7 | 0 | 0 | 0 | 0 | False | |

The MLwiN worksheet holds the data and other information in a series of columns, as on a spreadsheet. There are initially named c1, c2, etc. but we recommend that they be given meaningful names to show what their content relates to. This has already been done in the worksheet that you have loaded.

Each line in the body of the **Names** window summarises a column of data. In the present case only the first five of the 400 columns of the worksheet contain data. Each column contains 33988 values, one for each student represented in the data set. There are no missing values, and the minimum and maximum value in each column are shown. It is possible to define a variable as categorical (we shall do this later) and to add variable descriptions.

Module 3 (Practice): Multiple Regression


The following window appears:

Data

goto line 1 | view | Help | Font | Show value labels

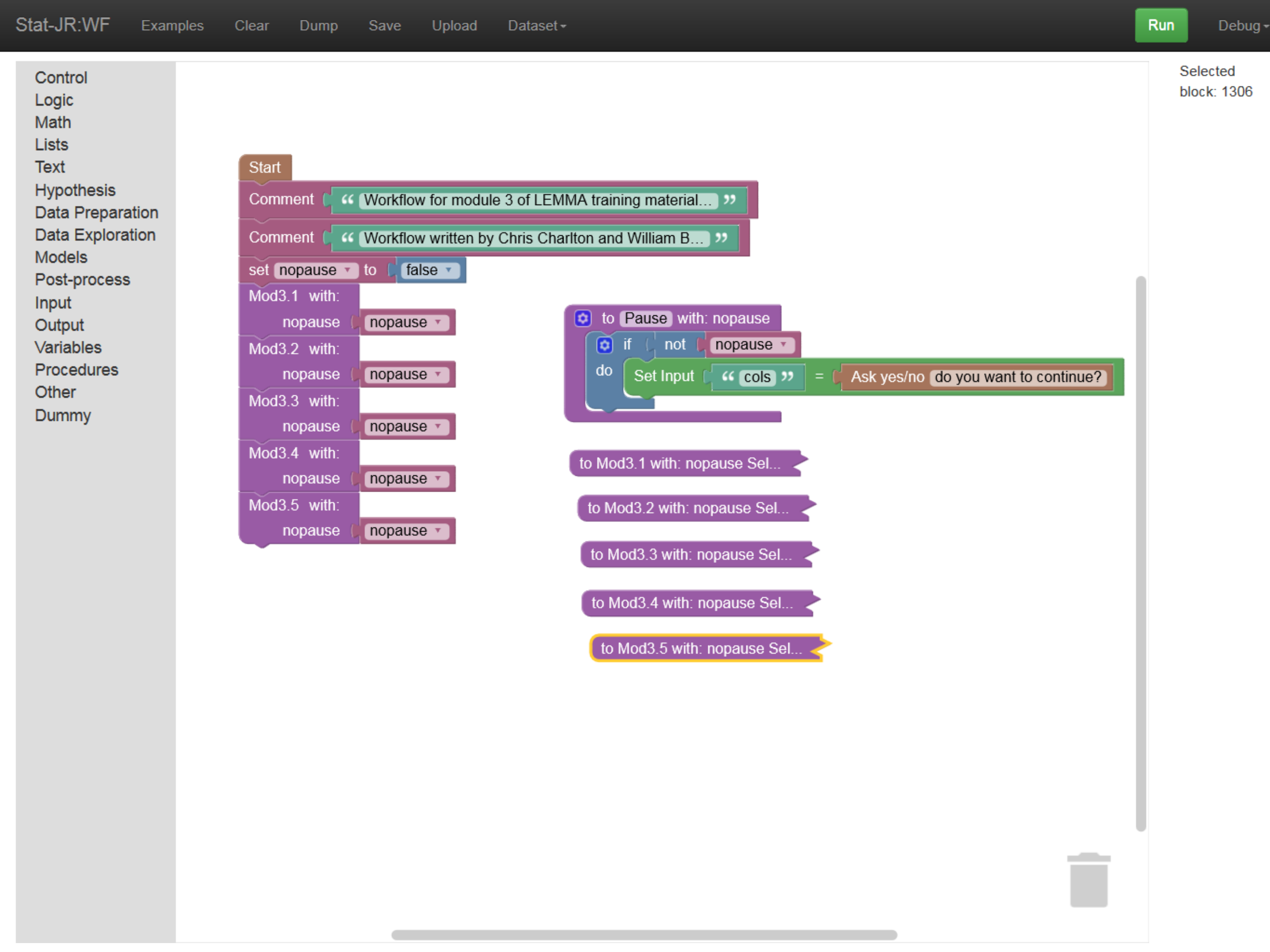
| | CASEID(33988) | SCORE(33988) | COHORT90(339 | FEMALE(33988) | SCLASS(33988) |
|---|---|---|---|---|---|
| 1 | 339.000 | 49.000 | -6.000 | 0.000 | 2.000 |
| 2 | 340.000 | 18.000 | -6.000 | 0.000 | 3.000 |
| 3 | 345.000 | 46.000 | -6.000 | 0.000 | 4.000 |
| 4 | 346.000 | 43.000 | -6.000 | 0.000 | 3.000 |
| 5 | 352.000 | 17.000 | -6.000 | 0.000 | 3.000 |
| 6 | 353.000 | 29.000 | -6.000 | 0.000 | 2.000 |
| 7 | 354.000 | 15.000 | -6.000 | 0.000 | 3.000 |
| 8 | 361.000 | 19.000 | -6.000 | 0.000 | 2.000 |
| 9 | 362.000 | 45.000 | -6.000 | 0.000 | 3.000 |
| 10 | 363.000 | 12.000 | -6.000 | 0.000 | 1.000 |
| 11 | 6824.000 | 0.000 | -4.000 | 0.000 | 1.000 |
| 12 | 6826.000 | 0.000 | -4.000 | 0.000 | 3.000 |
| 13 | 6827.000 | 20.000 | -4.000 | 0.000 | 2.000 |
| 14 | 6828.000 | 32.000 | -4.000 | 0.000 | 1.000 |
| 15 | 6829.000 | 0.000 | -4.000 | 0.000 | 2.000 |
| 16 | 6834.000 | 24.000 | -4.000 | 0.000 | 3.000 |

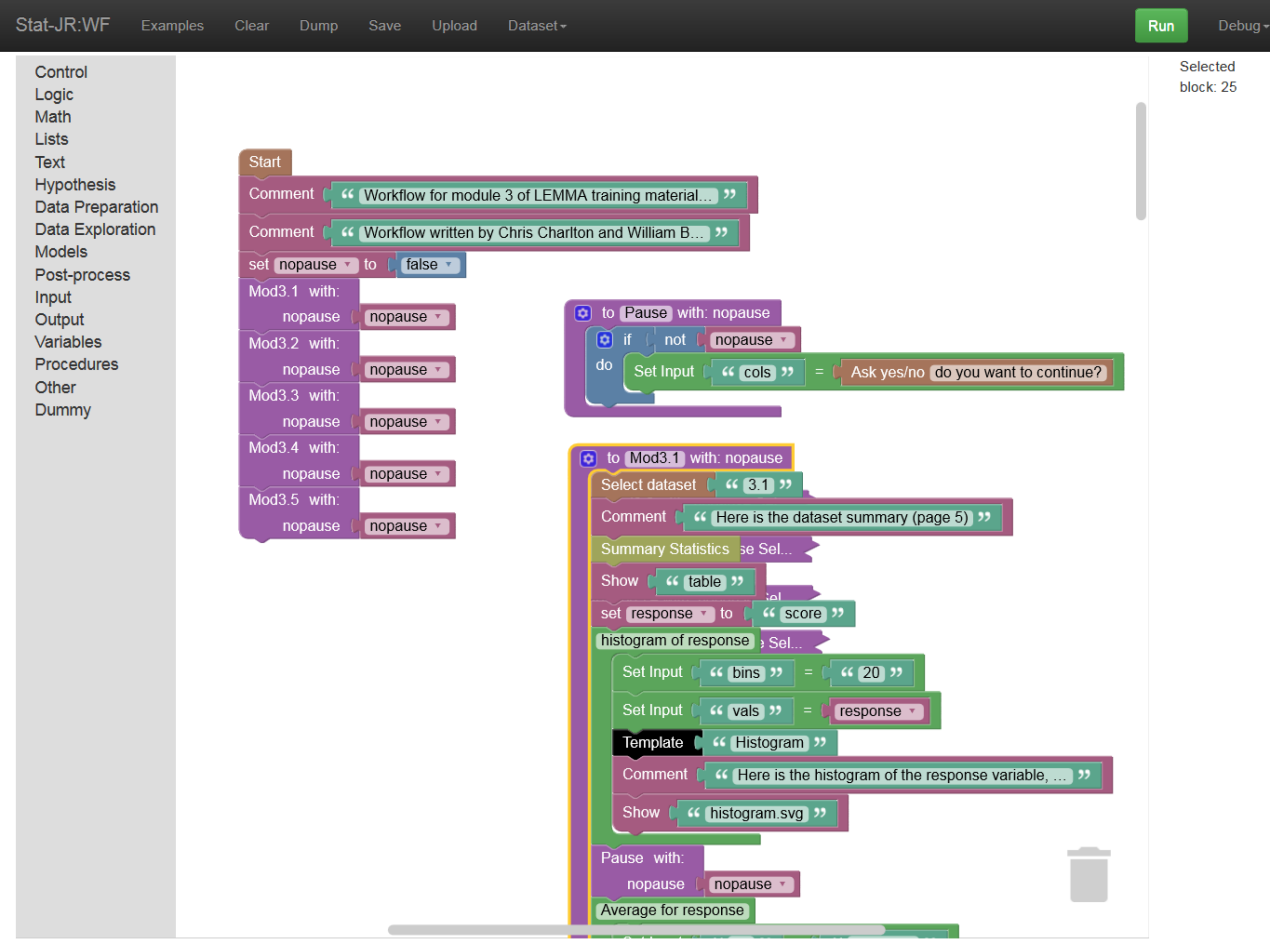
Because there are only five variables in the data file, all columns can be seen. When there are more variables, you can view any selection of columns, spreadsheet fashion, as follows:

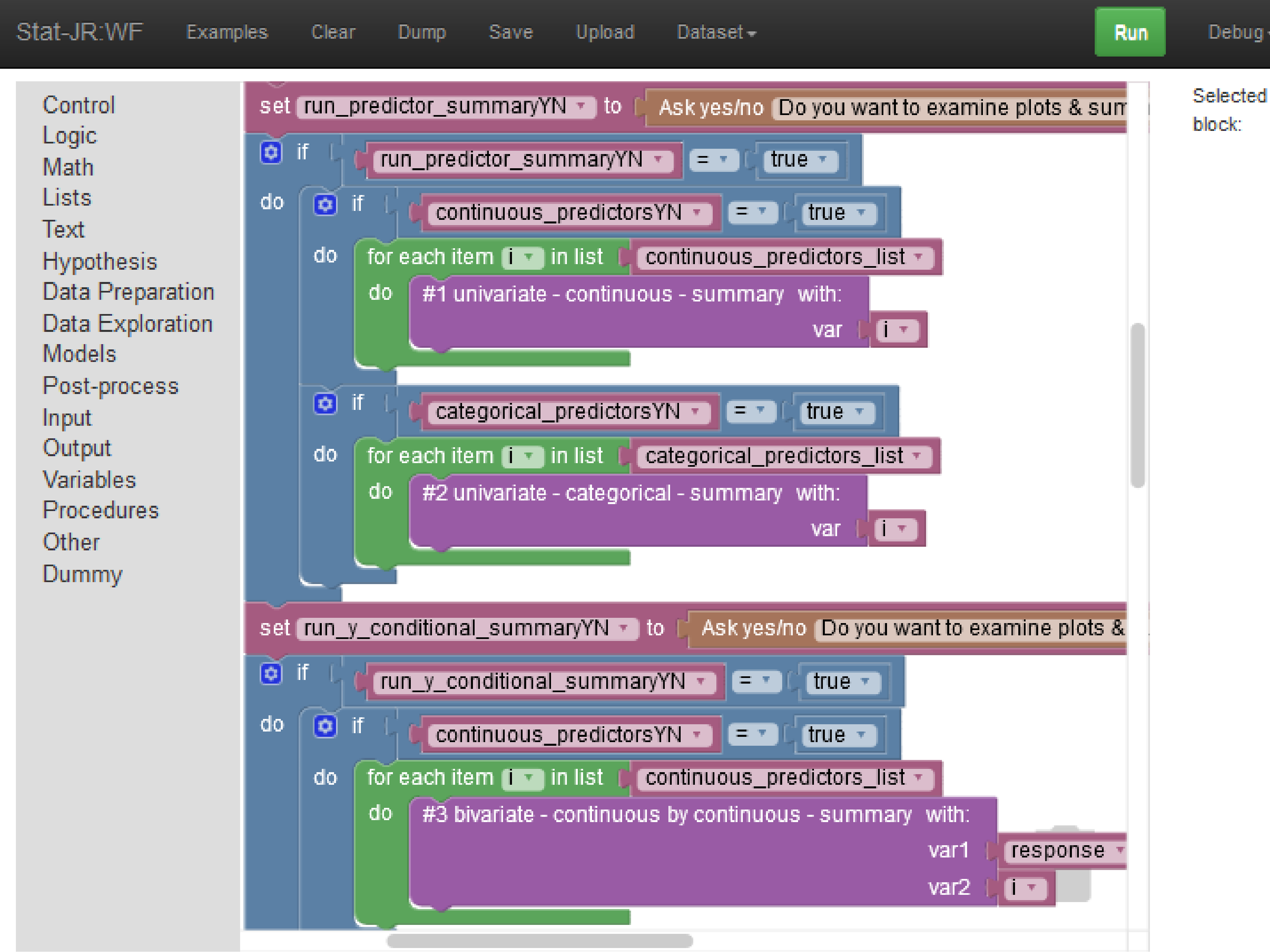
- Click the **View** button
- Select columns to view
- Click **OK**

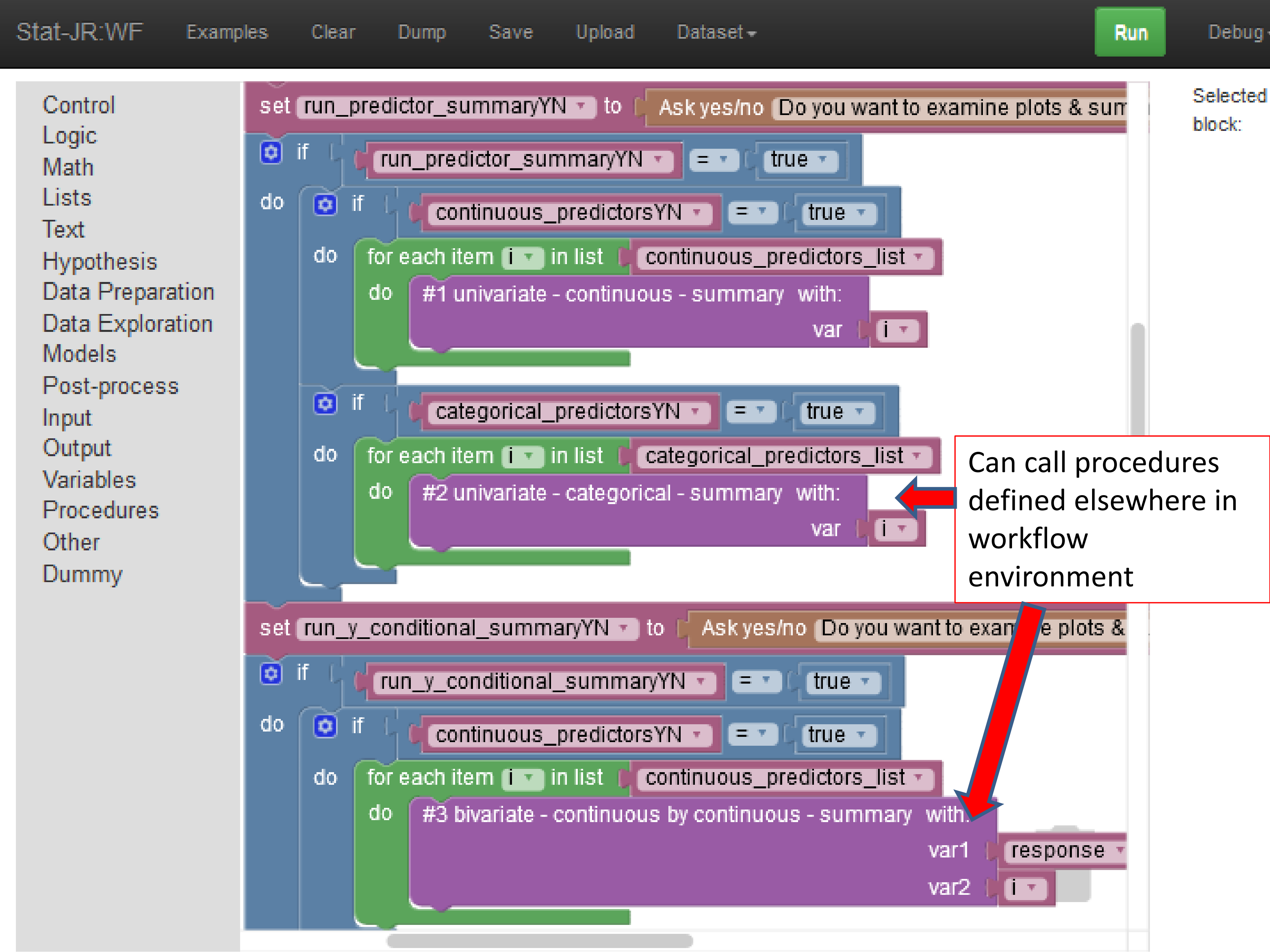
You can select a block of adjacent columns either by pointing and dragging or by selecting the column at one end of the block and holding down 'Shift' while you select the column at the other end. You can add to an existing selection by holding down 'Ctrl' while you select new columns or blocks. Use the scroll bars of the **Data** window to move horizontally and vertically through the data, and move or resize the window if you wish. You can go straight to line 1000, for example, by typing 1000 in the **goto line** box, and you can highlight a particular cell by pointing and clicking. This provides a means to edit data.

Having viewed the data we will examine SCORE and COHORT90, the variables to be considered in our first regression analysis.


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Selected block: 1306
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Start
Comment " Workflow for module 3 of LEMMA training material... "
Comment " Workflow written by Chris Charlton and William B... "
set nopause to false
Mod3.1 with: nopause nopause
Mod3.2 with: nopause nopause
Mod3.3 with: nopause nopause
Mod3.4 with: nopause nopause
Mod3.5 with: nopause nopause
to Pause with: nopause
if not nopause
do Set Input " cols " = Ask yes/no do you want to continue?
to Mod3.1 with: nopause Sel...
to Mod3.2 with: nopause Sel...
to Mod3.3 with: nopause Sel...
to Mod3.4 with: nopause Sel...
to Mod3.5 with: nopause Sel...


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Selected block: 25
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Start
Comment “Workflow for module 3 of LEMMA training material…”
Comment “Workflow written by Chris Charlton and William B…”
set nopause to false
Mod3.1 with: nopause nopause
Mod3.2 with: nopause nopause
Mod3.3 with: nopause nopause
Mod3.4 with: nopause nopause
Mod3.5 with: nopause nopause
to Pause with: nopause
if not nopause
do Set Input “cols” = Ask yes/no do you want to continue?
to Mod3.1 with: nopause
Select dataset “3.1”
Comment “Here is the dataset summary (page 5)”
Summary Statistics
Show “table”
set response to “score”
histogram of response
Set Input “bins” = “20”
Set Input “vals” = response
Template “Histogram”
Comment “Here is the histogram of the response variable, …”
Show “histogram.svg”
Pause with: nopause nopause
Average for response


Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
set run_y_conditional_summaryYN to Ask yes/no Do you want to examine plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i
Selected block:



Stat-JR:WF
Examples
Clear
Dump
Save
Upload
Dataset
Run
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Dummy
Selected block:
set run_predictor_summaryYN to Ask yes/no Do you want to examine plots & sum
if run_predictor_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #1 univariate - continuous - summary with:
var i
if categorical_predictorsYN = true
do for each item i in list categorical_predictors_list
do #2 univariate - categorical - summary with:
var i
Can call procedures defined elsewhere in workflow environment
set run_y_conditional_summaryYN to Ask yes/no Do you want to exam e plots &
if run_y_conditional_summaryYN = true
do if continuous_predictorsYN = true
do for each item i in list continuous_predictors_list
do #3 bivariate - continuous by continuous - summary with:
var1 response
var2 i

# Practical 3…