Small Area Estimation workshop
Lecture 2 – Stat-JR, MCMC, eBooks and SAE modelling

Professor William Browne,
Centre for Multilevel Modelling, University of Bristol
(with Chris Charlton, Nikos Tzavidis and Timo Schmid)

What will we cover ?

- Stat-JR
- MCMC estimation
- eBooks
- Statistical Analysis Assistants
- MCMC algorithms for SAE.
- An SAA for Small Area Estimation.

Stat-JR

- A statistical package developed by the team at the centre for multilevel modelling with colleagues at Southampton.
- Contains it's own estimation engine.
- System based on the idea of a suite of templates where each template performs a specific operation.
- Also allows interoperability with other software packages, so for example might have a regression template that fits regressions using various software packages.
- The initial TREE interface runs in a web browser.
- There are also newer eBook and workflow interfaces.
- Several ESRC grants have enabled Stat-JR to be written.

Stat-JR - Jon Rasbash's big vision

- The multilevel modelling centre had developed MLwiN for many years and Jon the main programmer was thinking on where our software went next.
- The frequentist IGLS algorithm was hard to extend further.
- WinBUGS showed that MCMC as an algorithm could be extended easily but the difficulty in MLwiN was in extending the MCMC code I developed!
- The big vision was an all-singing all-dancing system where expert users could add functionality easily and which interoperates with other software..
- The ESRC LEMMA II, LEMMA III and eSTAT grants would enable this to be achieved

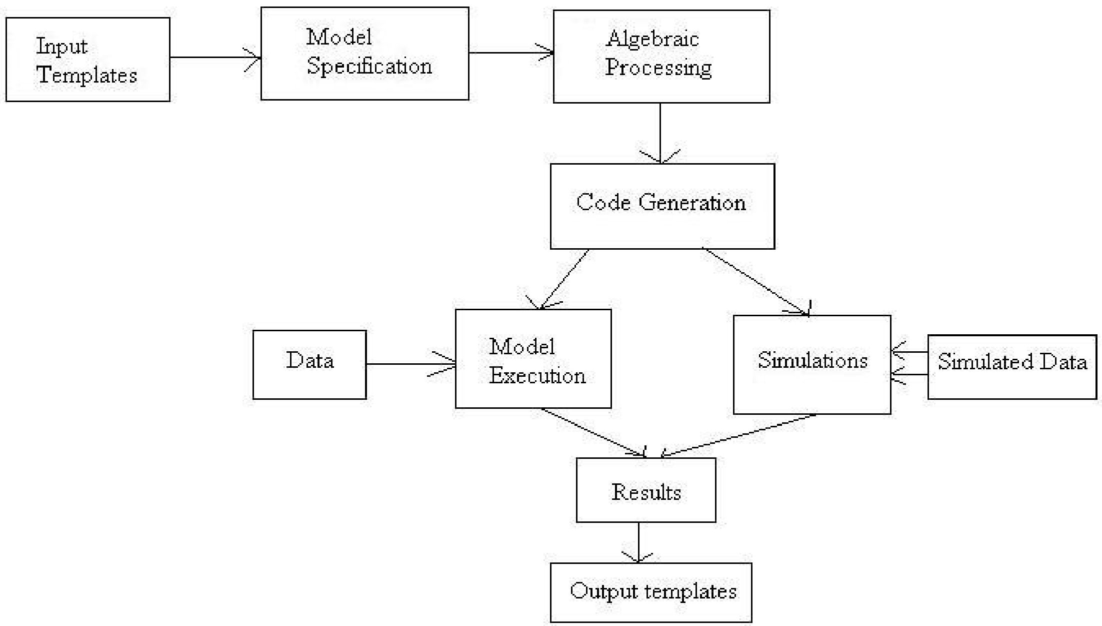
STAT-JR

Jon identified 3 groups of users:

- Novice practitioners who want to use statistical software that is user friendly and maybe tailored to their discipline
- Advanced practitioners who are the experts in their fields and also want to develop tools for the novice practitioners
- Algorithm Developers who want their algorithms used by practitioners.
- See http://www.cmm.bristol.ac.uk/research/NCESS-EStat/news.shtml for details of documentation for STAT-JR and several meeting presentations.

STAT-JR component based approach

Below is an early diagram of how we envisioned the system. Here you will see boxes representing components some of which are built into the STAT-JR system. The system is written in Python with a VB.net algebra processing system. A team of coders have worked together on the system.



Templates

Backbone of Stat-JR.

Consist of a set of code sections for advanced users to write. A bit like R packages.

For a model template it consists of at least:

- an *inputs* method which specifies inputs and types
- A *model* method that creates (BUGS like) model code for the algebra system
- An (optional) *latex* method can be used for outputting LaTeX code for the model.

Other optional functions required for more complex templates

Bayesian Statistics

- MCMC is often used with Bayesian statistical modelling.
- In Bayesian statistics we add prior distributions to a model so that all unknown parameters have a prior distribution which suggests what we know about the parameter BEFORE to data collection (often we know nothing and thus priors are often flat).
- This prior is combined with the data as it appears in a likelihood to produce a posterior distribution i.e. what we know about the parameter AFTER collecting the data.
- Posterior distributions are the key element of Bayesian statistics but often we look for summary measures from the posterior distribution e.g. it's mean or median.

MCMC estimation

- MCMC estimation works by calculating the conditional posterior distributions for each parameter i.e. the distribution if all other parameter values are known.
- Random (i.e. *Monte Carlo*) draws are taken in turn from each conditional distribution and the conditional distributions adapted to the new values.
- These draws then form a sample (*chain*) of estimates for each parameter. The chain is *Markov* (memory-less) as it only depends on the last value of the other parameters
- From the sample summary measures e.g. the posterior mean and SD can be used to describe the parameter but in addition the full sample allows easy calculation of interval estimates (from the quantiles of the chain of values).

Regression 1 Example

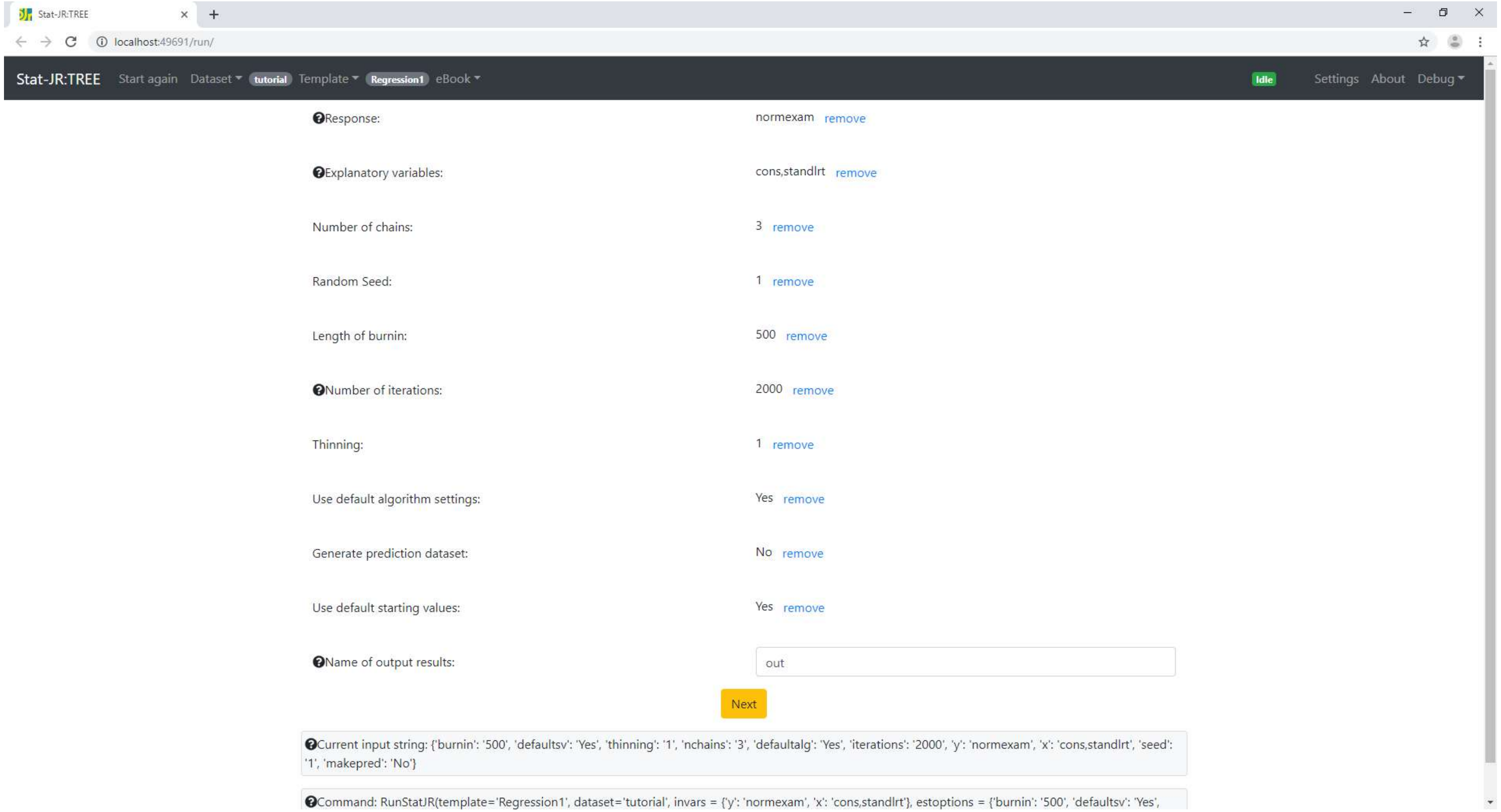
```
from EStat.Templating import *

class Regression1(Template):
   'A model template for fitting 1 level Normal multiple
        regression model in eStat only.‘
 tags = [ 'Model', '1-Level', 'eStat', 'Normal' ]
 engines = ['eStat']
 inputs = '''
y = DataVector('Response: ')
x = DataMatrix('Explanatory variables: ', allow_cat=True,
        help= 'predictor variables')
beta = ParamVector(parents=[x], as_scalar=True)
tau = ParamScalar()
sigma = ParamScalar(modelled = False)
sigma2 = ParamScalar(modelled = False)
deviance = ParamScalar(modelled = False)
'''
```

```
 model = '''
model{
    for (i in 1:length(${y})) {
        ${y}[i] ~ dnorm(mu[i], tau)
        mu[i] <- ${mmult(x, 'beta', 'i')}
    }

    # Priors
    % for i in range(0, x.ncols()):
    beta${i} ~ dflat()
    % endfor
    tau ~ dgamma(0.001000, 0.001000)
    sigma2 <- 1 / tau
    sigma <- 1 / sqrt(tau)
}
'''
       latex = r'''
\begin{aligned}
 \mbox{${y}}_i & \sim \mbox{N}(\mu_i, \sigma^2) \\
\mu_i & =
  ${mmulttex(x, r'\beta', 'i')} \\
%for i in range(0, len(x)):
\beta_${i} & \propto 1 \\
%endfor
\tau & \sim \Gamma (0.001,0.001) \\
\sigma^2 & = 1 / \tau
\end{aligned}
'''
```

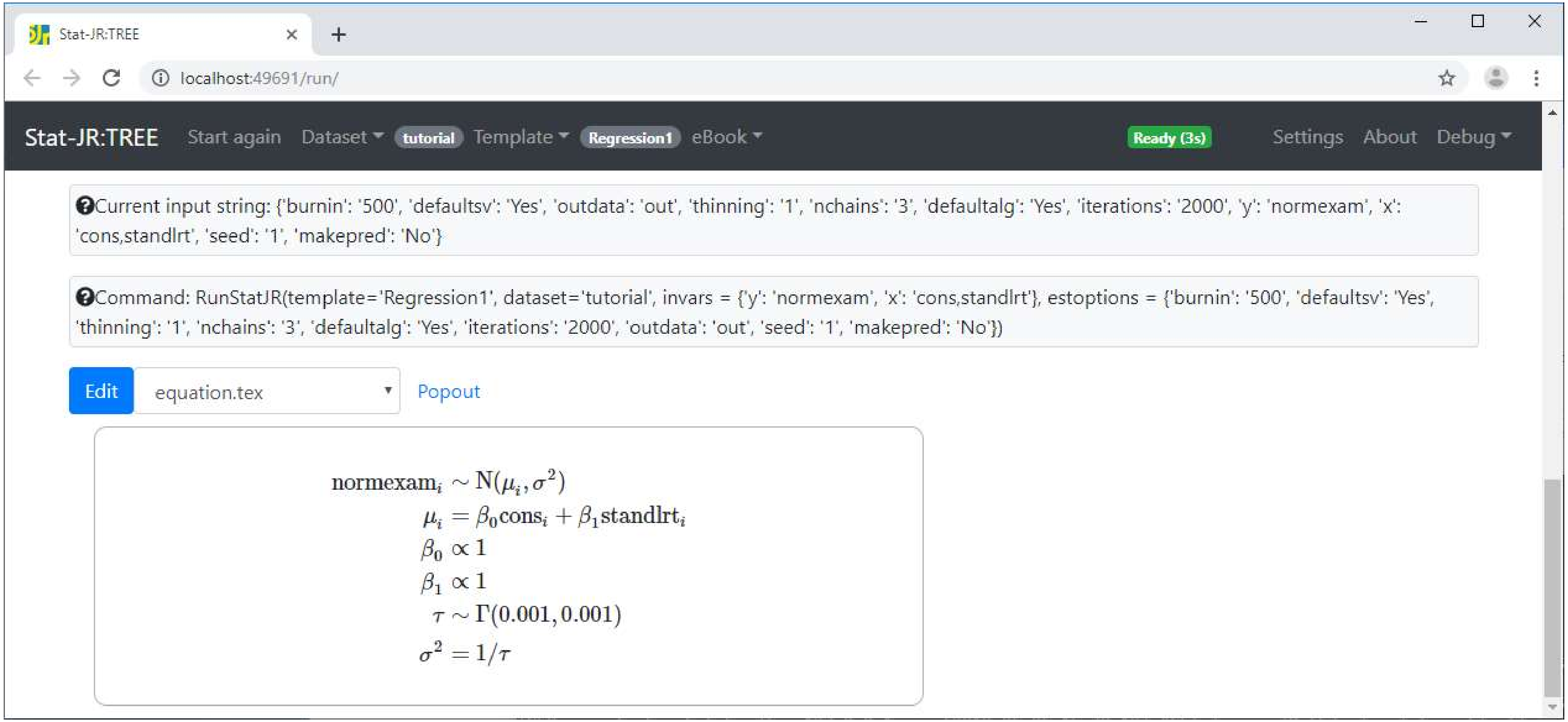
An example of STAT-JR – setting up a model



An example of STAT-JR – setting up a model

Response:	normexam remove
Explanatory variables:	cons,standlrt remove
Number of chains:	3 remove
Random Seed:	1 remove
Length of burnin:	500 remove
Number of iterations:	2000 remove
Thinning:	1 remove
Use default algorithm settings:	Yes remove
Generate prediction dataset:	Yes remove
Use default starting values:	Yes remove
Name of output results:	out

Equations for model



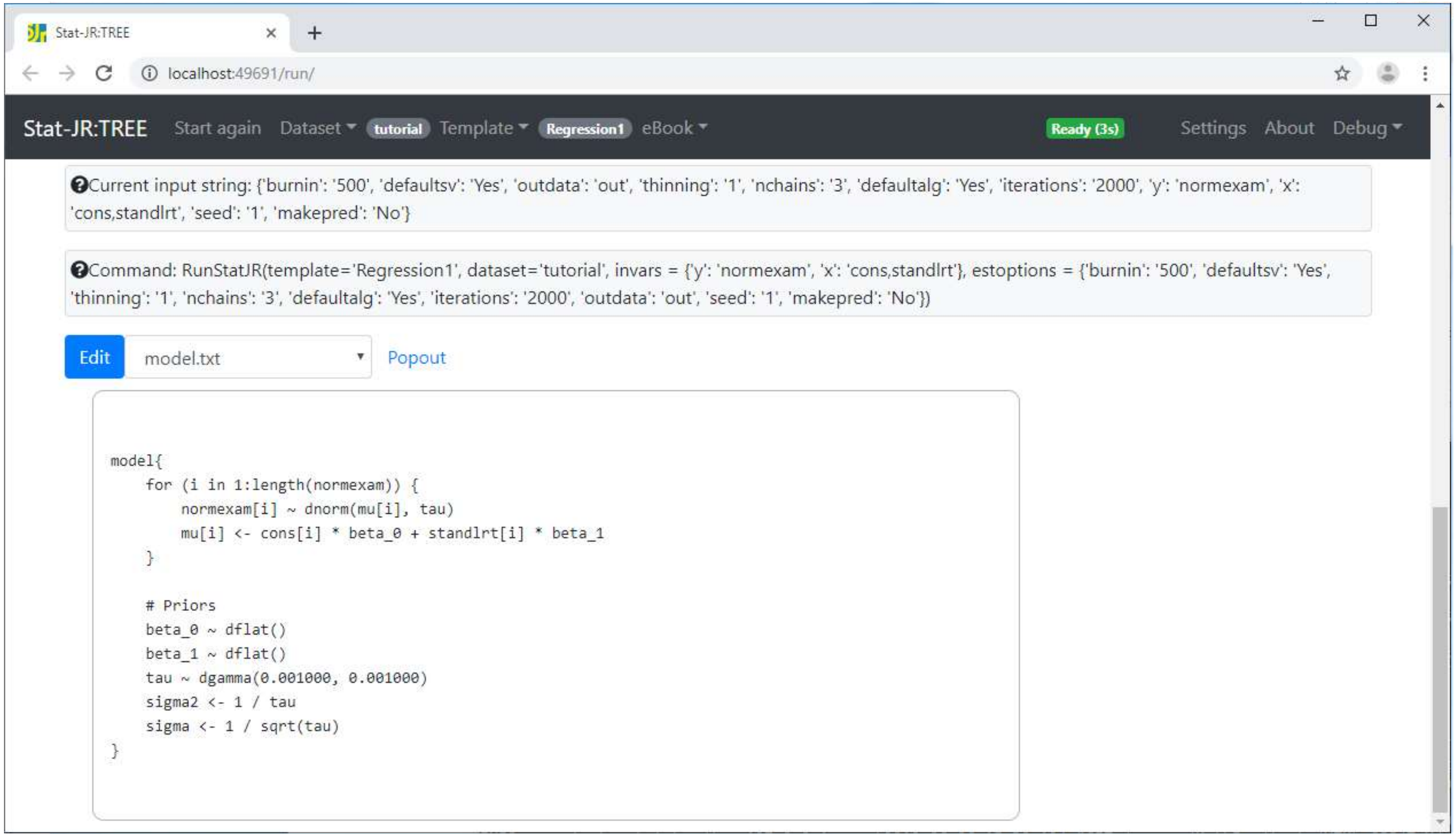
All objects created available from one pull down and can be popped out to separate tabs in browser.

Equations for model

$$
\begin{aligned}
\text{normexam}_i &\sim \mathrm{N}(\mu_i, \sigma^2) \\
\mu_i &= \beta_0 \text{cons}_i + \beta_1 \text{standlrt}_i \\
\beta_0 &\propto 1 \\
\beta_1 &\propto 1 \\
\tau &\sim \Gamma(0.001, 0.001) \\
\sigma^2 &= 1/\tau
\end{aligned}
$$

- Note: Equations use MathJax and so underlying LaTeX can be copied and paste. The model code is based around the WinBUGS language with some variation.

Model code



All objects created available from one pull down and can be popped out to separate tabs in browser.

Model code in detail

```
model{
   for (i in 1:length(normexam)) {
      normexam[i] ~ dnorm(mu[i], tau)
      mu[i] <- cons[i] * beta0 + standlrt[i] * beta1
 }
# Priors
   beta0 ~ dflat()
   beta1 ~ dflat()
   tau ~ dgamma(0.001000, 0.001000)
   sigma2 <- 1 / tau
   sigma <- 1/sqrt(tau)
}
```

For this template the code is, aside from the length function, standard WinBUGS model code.

Algebra system steps to calculate conditional posterior

Stat-JR:TREE | Stat-JR:TREE

localhost:49691/output/node_beta_0.xml

Stat-JR:TREE

Log posterior $\tau\left(\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i(\mathrm{normexam}_i - \mathrm{beta\_}_1\mathrm{standlrt}_i)\right)\mathrm{beta\_}_0 - \frac{\tau\left(\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2\right)\mathrm{beta\_}_0{}^2}{2}$

Distribution dnorm

Match $\mathrm{A} = \tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i(\mathrm{normexam}_i - \mathrm{beta\_}_1\mathrm{standlrt}_i)$

Match $\mathrm{B} = -\left(\frac{\tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2}{2}\right)$

Sampling parameter $\mu = \frac{\tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i(\mathrm{normexam}_i - \mathrm{beta\_}_1\mathrm{standlrt}_i)}{\tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2}$

Sampling parameter $\tau = \tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2$

Sampling distribution $\mathrm{beta\_}_0 \sim \mathrm{dnorm}\left(\frac{\tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i(\mathrm{normexam}_i - \mathrm{beta\_}_1\mathrm{standlrt}_i)}{\tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2}, \tau\sum_{i=1}^{\mathrm{length}\,(\mathrm{normexam})} \mathrm{cons}_i{}^2\right)$

Algebra system steps for all parameters

Stat-JR:TREE | Stat-JR:TREE | Stat-JR:TREE

localhost:49691/output/algorithm.tex

Stat-JR:TREE

LaTeX version of algorithm

Conditional posterior for tau for Gibbs sampling

$$\tau \sim \Gamma\left(0.001 + 0.5 \times \mathrm{length(normexam)}, 0.001000 + \frac{\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \left(\mathrm{normexam_i} - \mathrm{beta\_0} \times \mathrm{cons_i} - \mathrm{beta\_1} \times \mathrm{standlrt_i}\right)^2}{2}\right)$$

Deviance Function

$$\mathrm{deviance} = 2 \times \left(\frac{\tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \left(\mathrm{normexam_i} - \mathrm{beta\_0} \times \mathrm{cons_i} - \mathrm{beta\_1} \times \mathrm{standlrt_i}\right)^2\right)}{2} + 0.5 \times (\ln(\pi) - \ln(\tau)) \times \mathrm{length(normexam)} + 0.346573590279973 \times \mathrm{length(normexam)}\right)$$

Conditional posterior for beta_0 for Gibbs sampling

$$\mathrm{beta\_0} \sim \mathrm{N}\left(\frac{\tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{cons_i} \times \left(\mathrm{normexam_i} - \mathrm{beta\_1} \times \mathrm{standlrt_i}\right)\right)}{\tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{cons_i}^2\right)}, \tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{cons_i}^2\right)\right)$$

Conditional posterior for beta_1 for Gibbs sampling

$$\mathrm{beta\_1} \sim \mathrm{N}\left(\frac{\tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{standlrt_i} \times \left(\mathrm{normexam_i} - \mathrm{beta\_0} \times \mathrm{cons_i}\right)\right)}{\tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{standlrt_i}^2\right)}, \tau \times \left(\sum_{\mathrm{i}=1}^{\mathrm{length(normexam)}} \mathrm{standlrt_i}^2\right)\right)$$

Deterministic formula for parameter sigma

$$\sigma = \frac{1}{\mathrm{sqrt}(\tau)}$$

Deterministic formula for parameter sigma2

$$\sigma_2 = \frac{1}{\tau}$$

Algebra system steps

Use Gibbs sampling from conditional posterior for beta1:

$$\beta_1 \sim \mathrm{N}\left(\frac{\tau\times\left(\sum_{i=1}^{\mathrm{length(normexam)}} \mathrm{standlrt}_i\times\left(\mathrm{normexam}_i-\beta_0\times\mathrm{cons}_i\right)\right)}{\tau\times\left(\sum_{i=1}^{\mathrm{length(normexam)}} \mathrm{standlrt}_i^{\,2}\right)}, \tau\times\left(\sum_{i=1}^{\mathrm{length(normexam)}} \mathrm{standlrt}_i^{\,2}\right)\right)$$

$$\beta_1 \sim \mathrm{N}(0.000249799765395\times(2382.12631198+\beta_0\times(-7.34783096611)), 4003.20632175\times\tau)$$

Here the first line is what is returned by the algebra system – which works solely on the model code.
The second line is what can be calculated when values are added for constants and data etc.
System then constructs C code and fits model

Output of generated C++ code

Stat-JR:TREE

localhost:49691/output/modelcode.cpp

Stat-JR:TREE

```
    }

        // Update beta_0
// This code was generated by the Stat-JR package (copyright 2012 University of Bristol and University of Southampton).

    {

            double sum0=0;
            double csum0=0;
            double sum1=0;
            double csum1=0;
            for(int i=0; i<4059; i++) {
                    double ysum0 = (cons[i]*(normexam[i]-(beta_1*standlrt[i]))) - csum0;
                    double tsum0 = sum0 + ysum0;
                    double ysum1 = pow(cons[i],2.0) - csum1;
                    double tsum1 = sum1 + ysum1;
                    csum0 = (tsum0 - sum0) - ysum0;
                    sum0 = tsum0;
                    csum1 = (tsum1 - sum1) - ysum1;
                    sum1 = tsum1;
            }


        beta_0 = dnorm(((tau*sum0)/(tau*sum1)),(tau*sum1));


    }

        // Update beta_1
// This code was generated by the Stat-JR package (copyright 2012 University of Bristol and University of Southampton).

    {

            double sum0=0;
            double csum0=0;
            double sum1=0;
            double csum1=0;
            for(int i=0; i<4059; i++) {
                    double ysum0 = (standlrt[i]*(normexam[i]-(beta_0*cons[i]))) - csum0;
                    double tsum0 = sum0 + ysum0;
                    double ysum1 = pow(standlrt[i],2.0) - csum1;
```

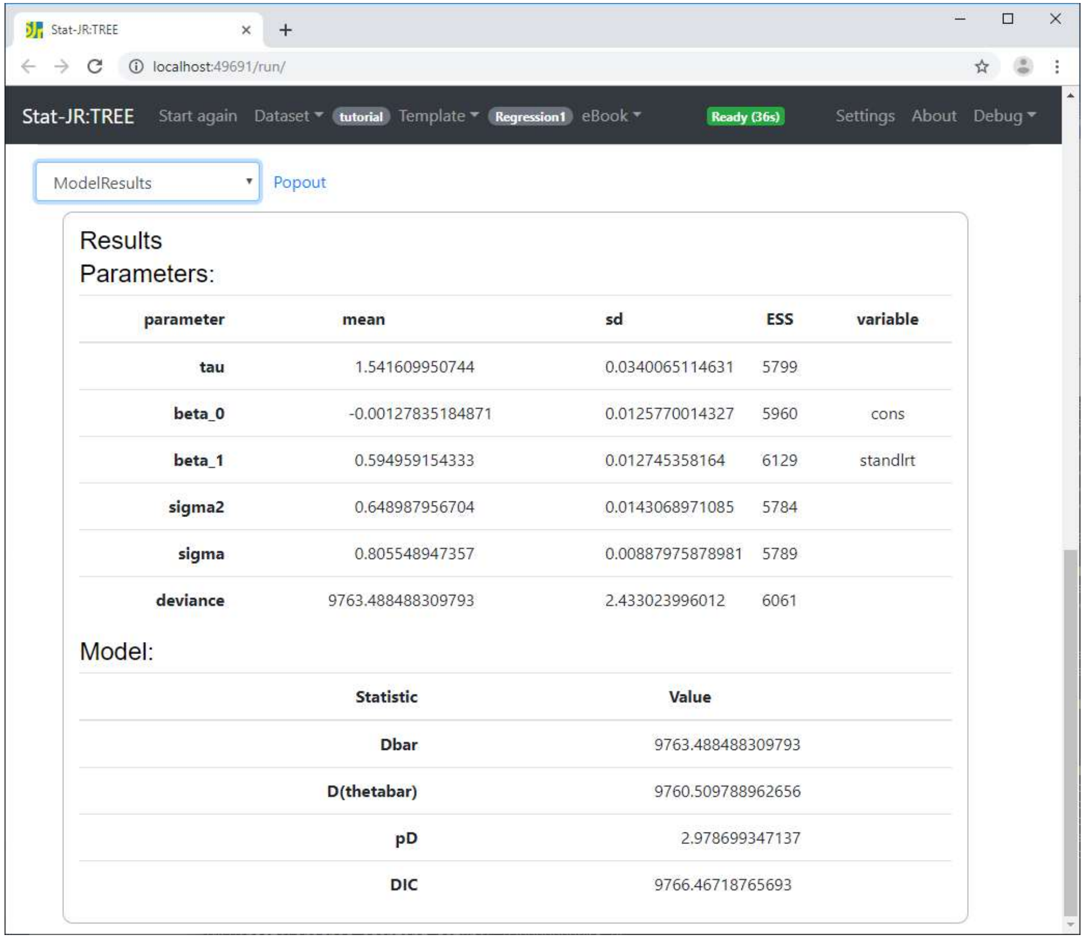
The package can output C++ code that can then be taken away by software developers and modified.

Output of generated C++ code

```
// Update beta1
{
beta1 = dnorm((0.000249799765395*(2382.12631198+(beta0*(-
    7.34783096611)))),(4003.20632175*tau));
}
// Update beta0
{
beta0 = dnorm((((-0.462375992909)+((-
    7.34783096611)*beta1))*0.000246366100025),(tau*4059.0));
}
```

- Note now that the code includes the actual data in place of constants and so looks less like the familiar algebraic expressions

Output from the eStat engine



Results

Parameters:

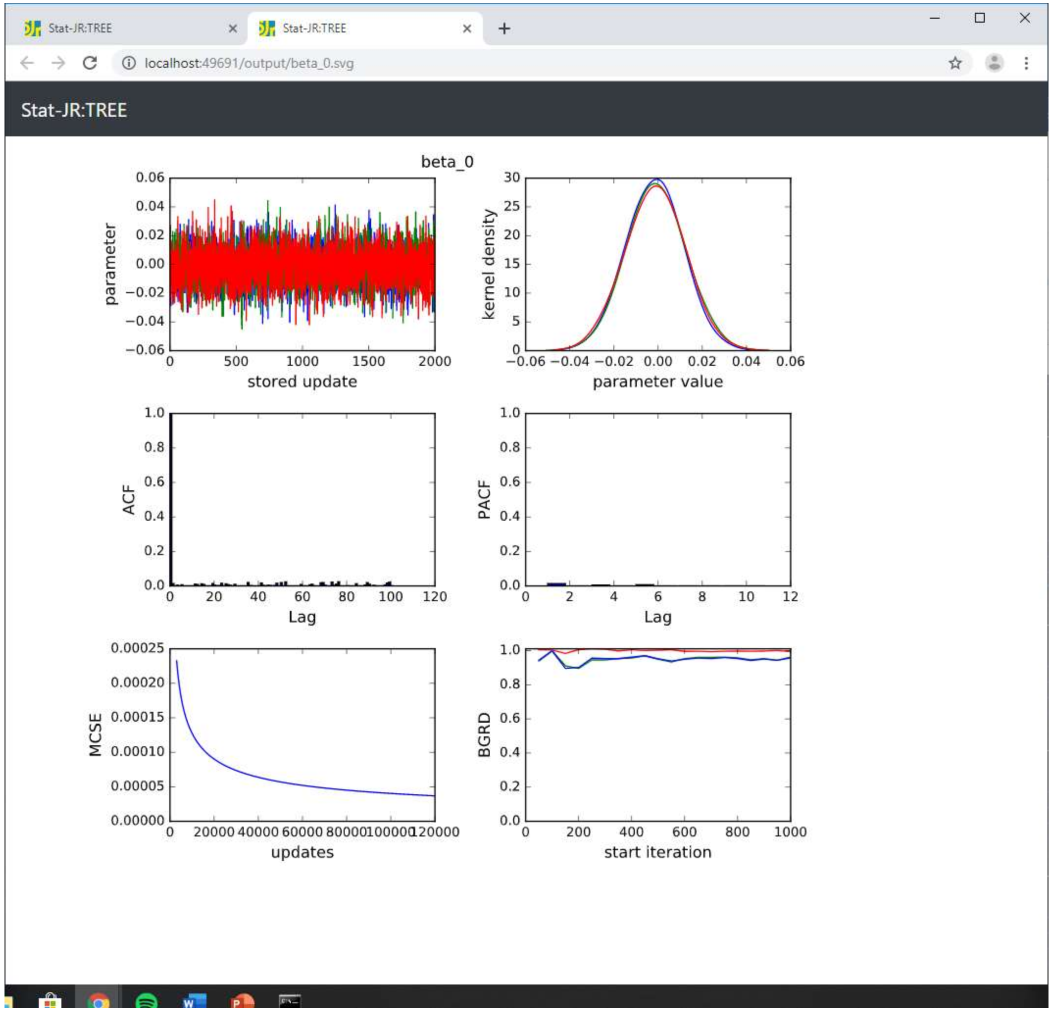
parameter	mean	sd	ESS	variable
tau	1.541609950744	0.0340065114631	5799	
beta_0	-0.00127835184871	0.0125770014327	5960	cons
beta_1	0.594959154333	0.012745358164	6129	standlrt
sigma2	0.648987956704	0.0143068971085	5784	
sigma	0.805548947357	0.00887975878981	5789	
deviance	9763.488488309793	2.433023996012	6061	

Model:

Statistic	Value
Dbar	9763.488488309793
D(thetabar)	9760.509788962656
pD	2.978699347137
DIC	9766.46718765693

Estimates and the DIC diagnostic can be viewed for the model fitted.

Output from the eStat engine



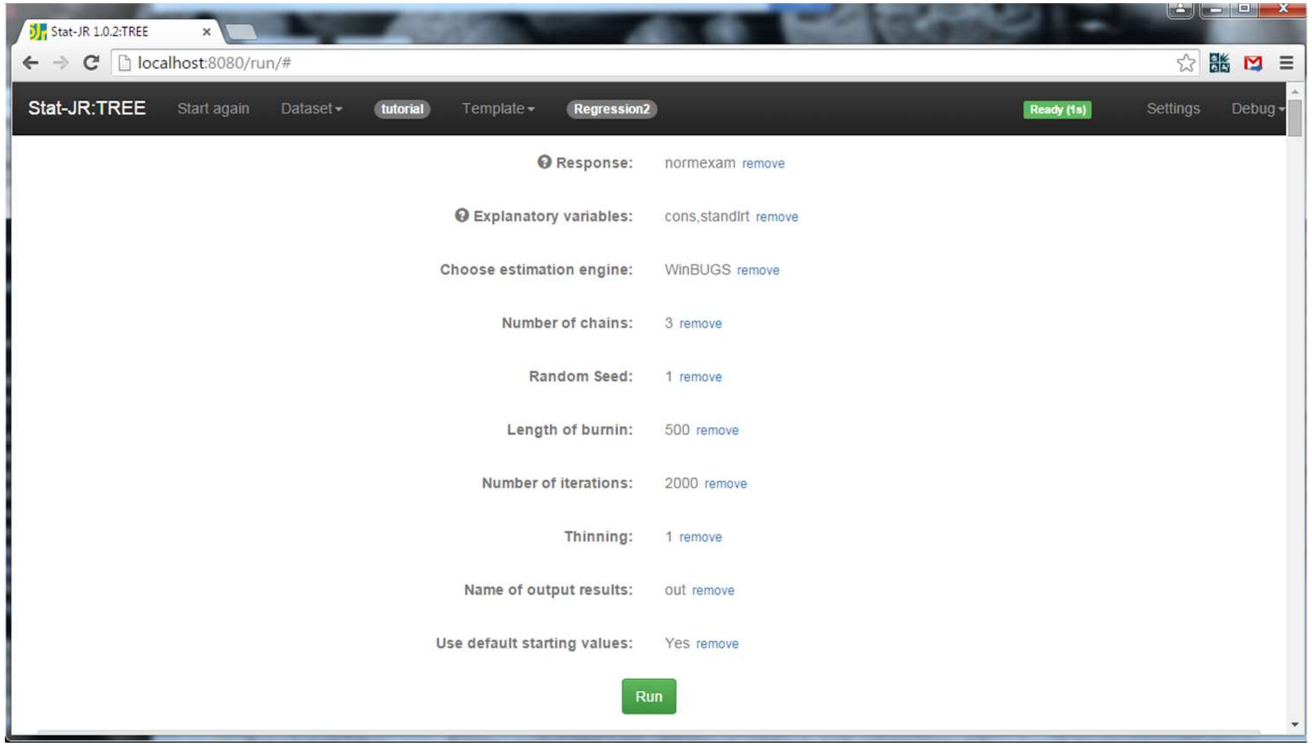
eStat offers multiple chains so that we can use multiple chain diagnostics to aid convergence checking.

Otherwise the graphs are borrowed from the 6-way plotting in the MLwiN package.

Graphics are in svg format so scale nicely.

INTEROPERABILITY

Interoperability with WinBUGS (Regression 2)

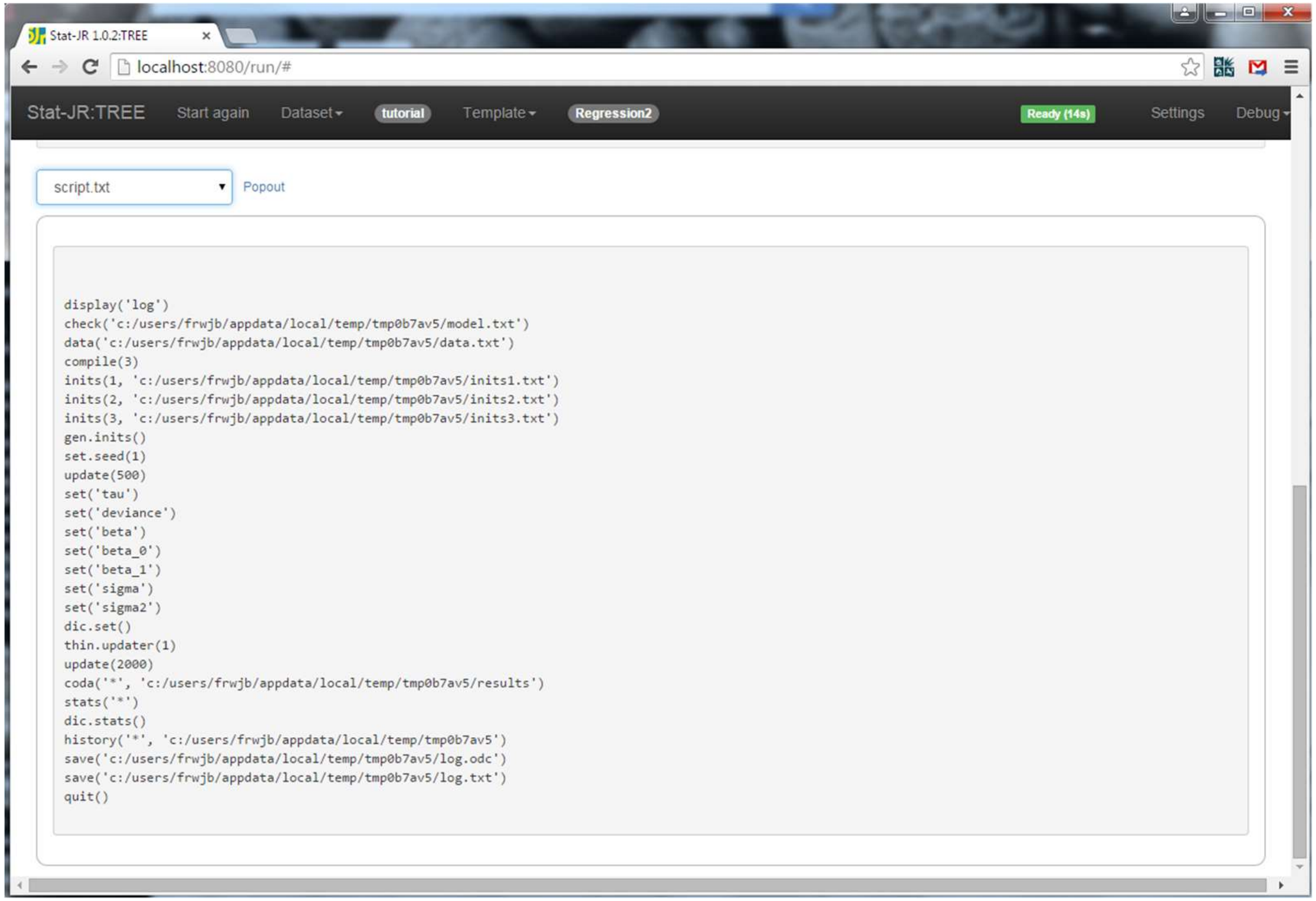


This template offers the choice of many software packages for fitting a regression model.

STAT-JR checks what is installed on the machine and only offers packages that are installed.
Here we choose WinBUGS.

Interoperability in the user interface is obtained via a few extra inputs. In fact in the template code user written functions are required for all packages apart from WinBUGS, OpenBUGS and JAGS. The transfer of data between packages is however generic.

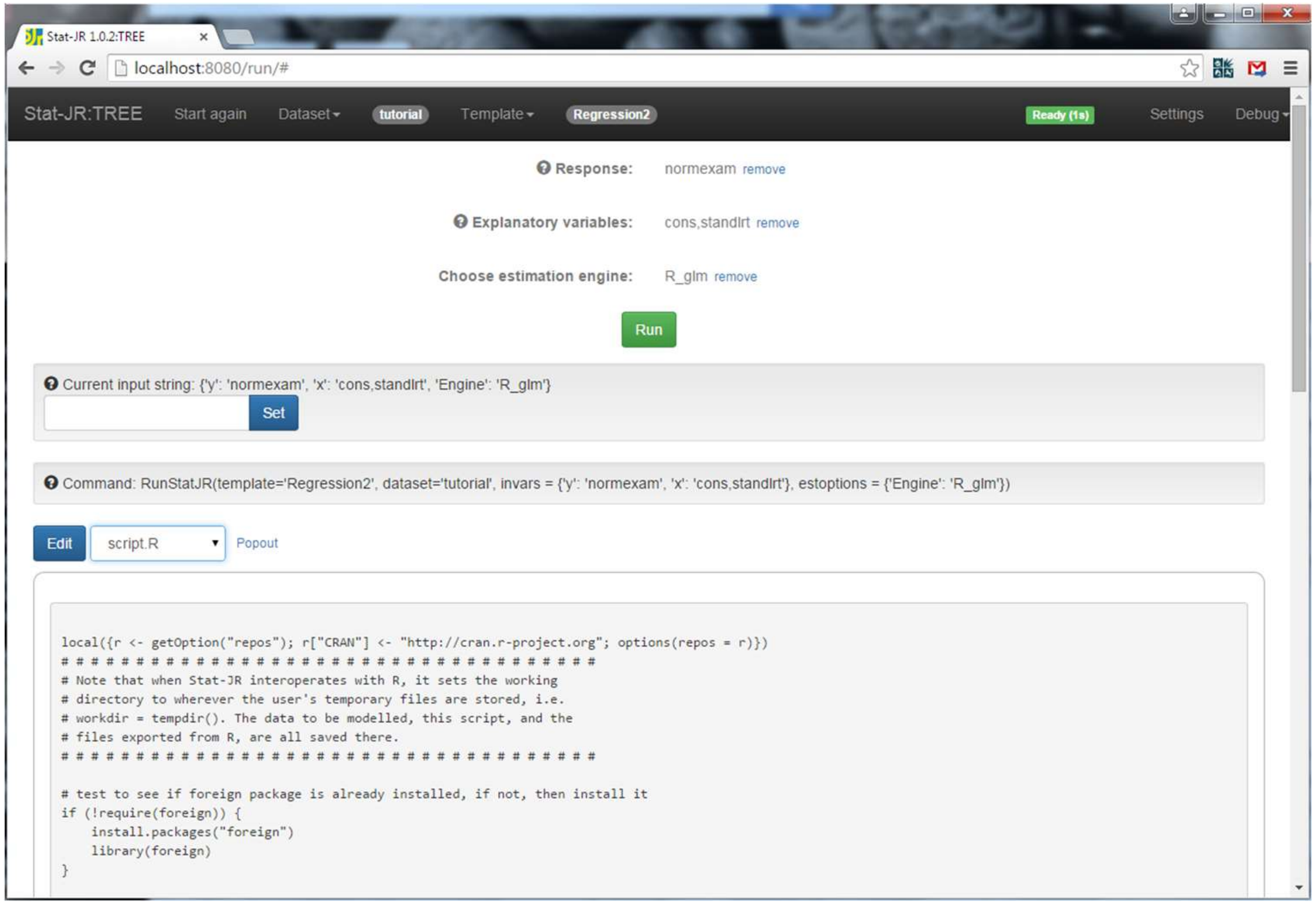
Interoperability with WinBUGS (Regression 2)



Here we can view the files required to run WinBUGS in the pane (script file shown but model, inits and data also available)

The model can be run by press of a button.

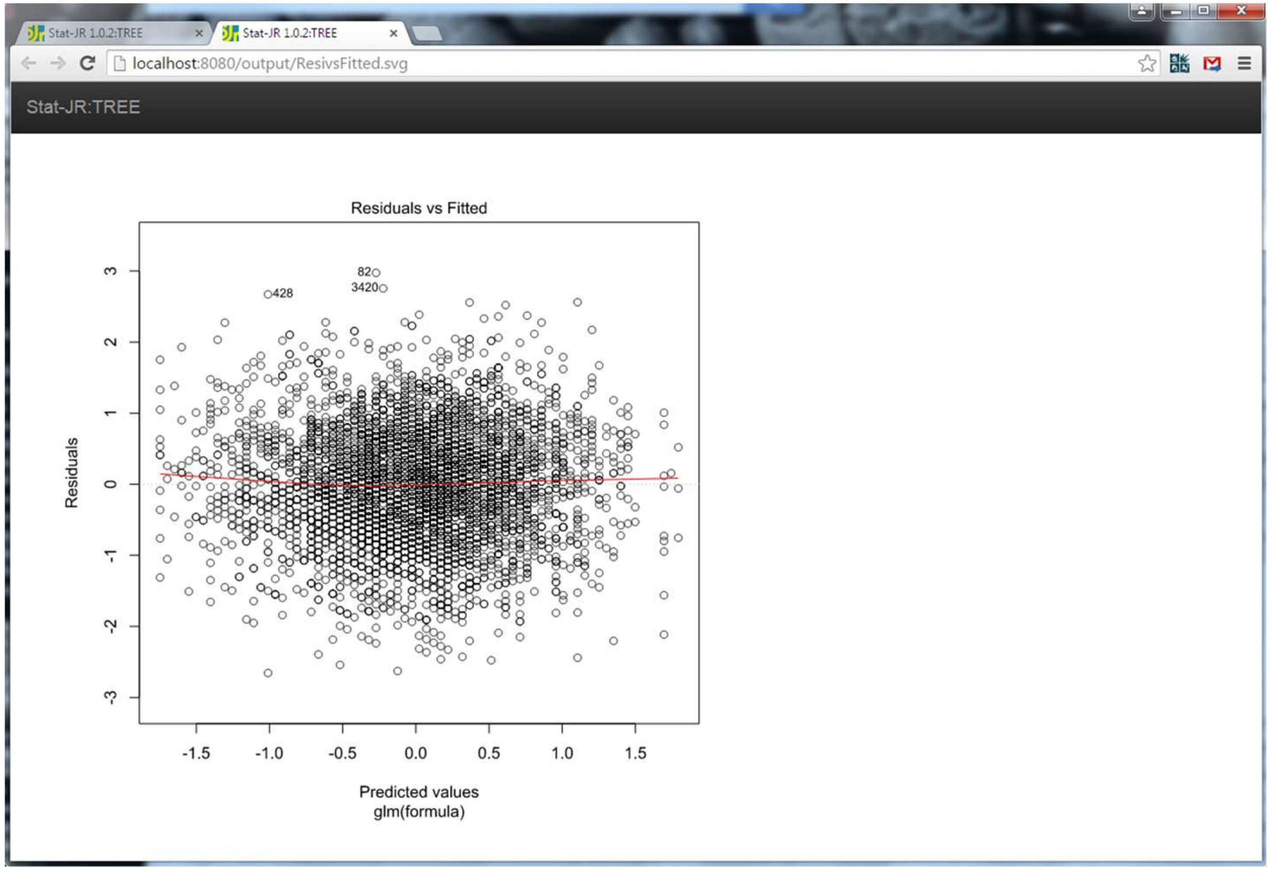
Interoperability with R



R can be chosen as another alternative. In fact here we have 2 choices – glm or MCMCglmm.

You will see in the pane the script file ready for input to R. There will also be the data file that R requires.

Interoperability with R



If written in to the code in the template – graphics from other software can be extracted.

Here for example is a residual plot associated with the R fit of the model.

eBooks

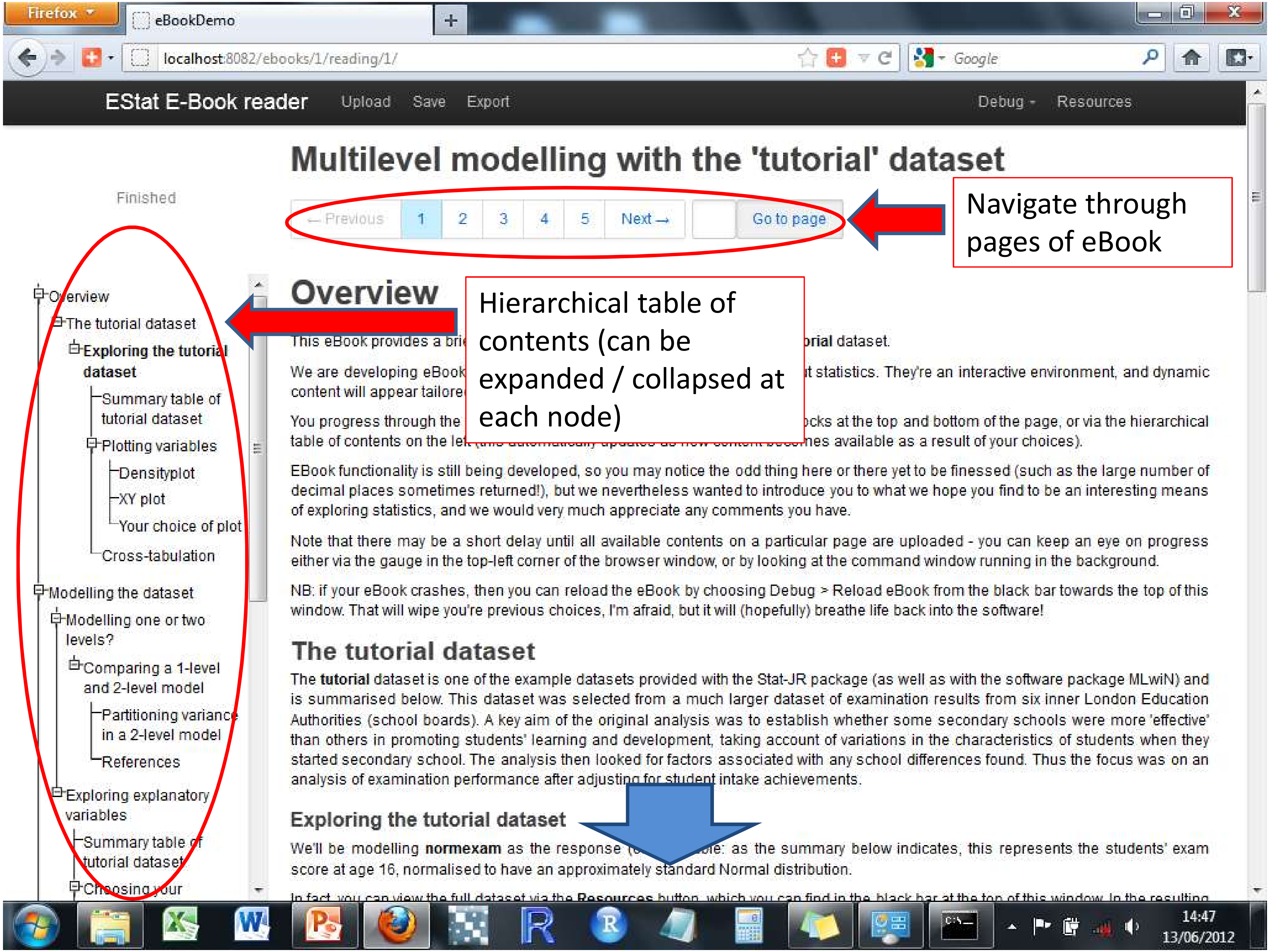
+

=

An electronic book is a book-publication in digital form.
In the US more books are published online than distributed in hard copy in book shops.

Statistical (and Mathematical) eBooks

- The idea is can we incorporate statistical content into an eBook? Of course a statistical textbook is no different on paper to any other document when it comes to creating a pdf file (aside from maybe more equations!)
- The difference is in what 'enhancements' we can add and so the idea here is combining the text book with the statistics package i.e. interactive examples, allowing the user to include their own dataset etc.


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
Hierarchical table of contents (can be expanded / collapsed at each node)
Overview
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
nes 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

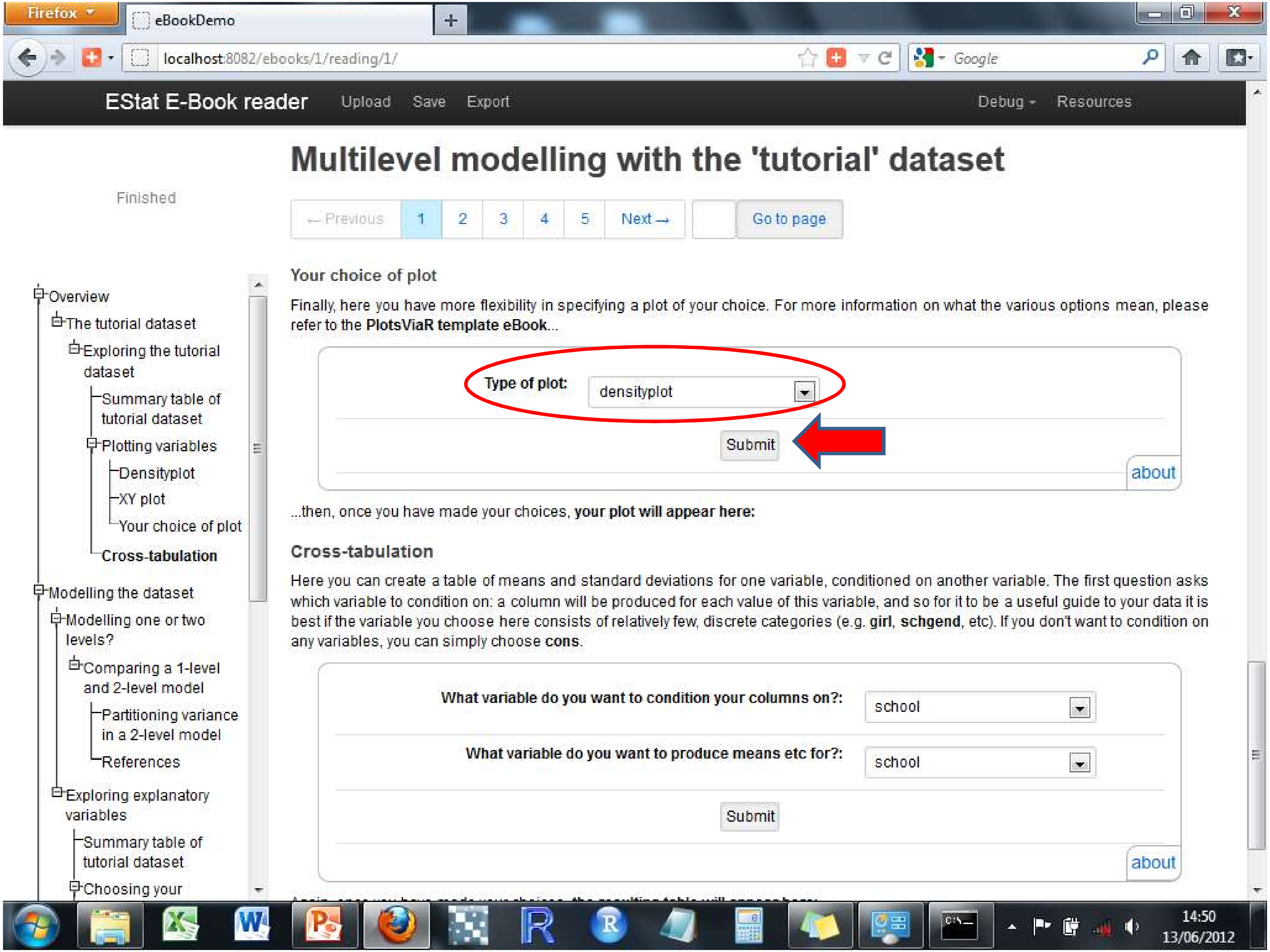
Column name	n	Missing	Min	Max	Description
school	4059	0	1	65	Numeric school identifier
student	4059	0	1	198	Numeric student identifier
normexam	4059	0	-3.67	3.67	Students' exam score at age 16, normalised to have approximately a standard Normal distribution.
cons	4059	0	1	1	A column of ones. If included 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%

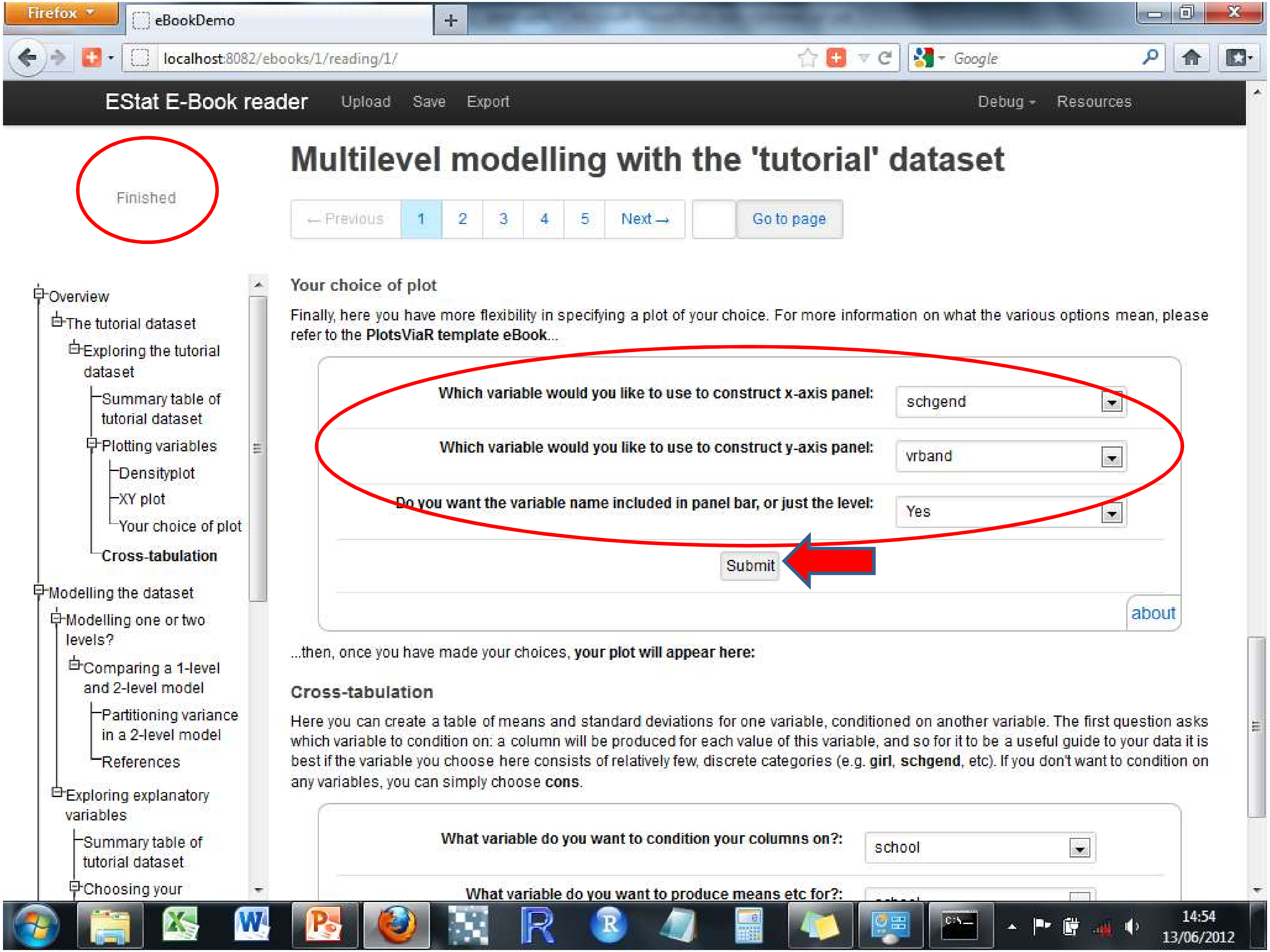
Plotting variables

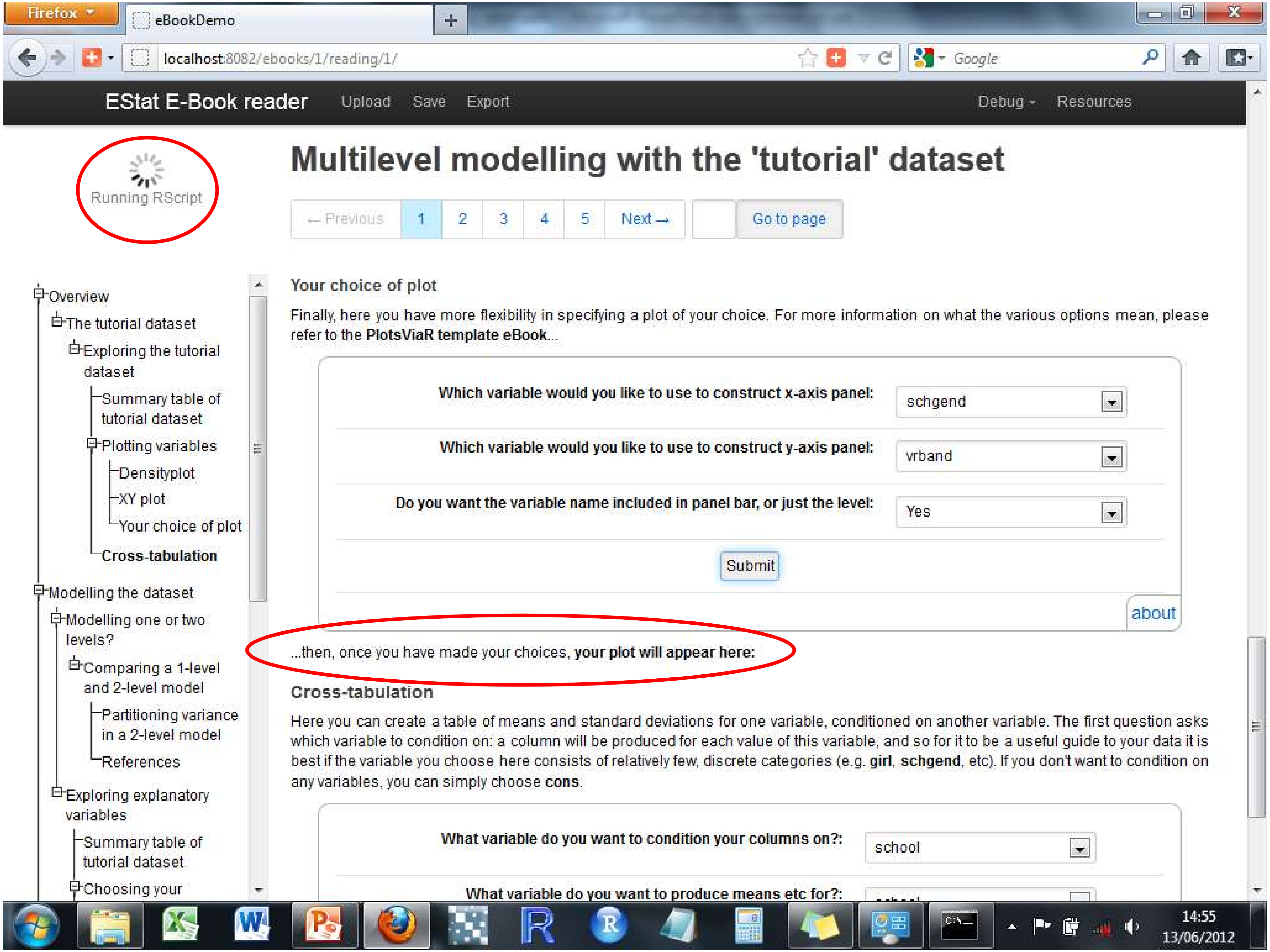
Here you can graphically-explore the **tutorial** dataset.

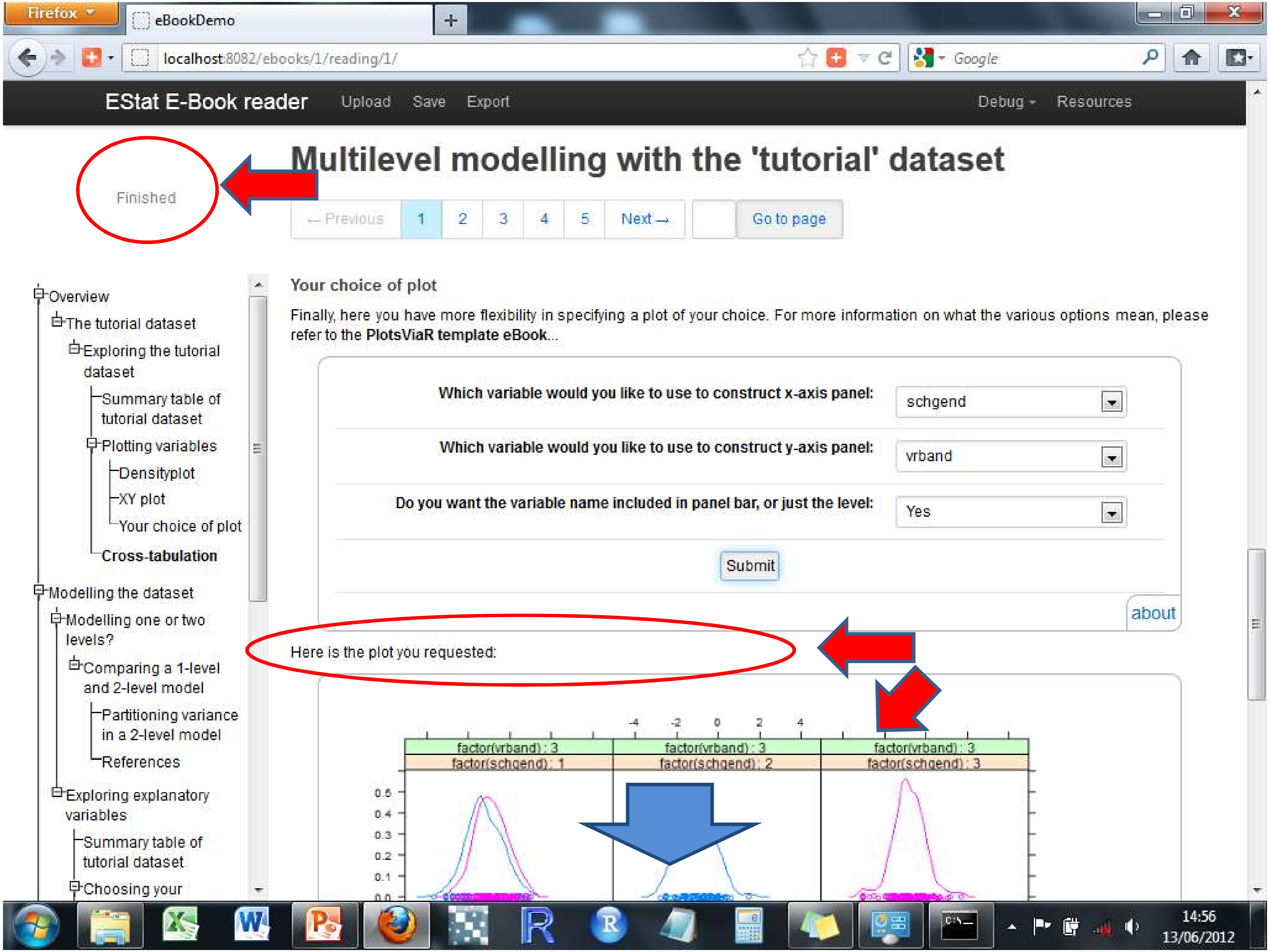
In the first two sections, below, you can produce a densityplot and XY plot, respectively; here you can re-specify your choice of variables

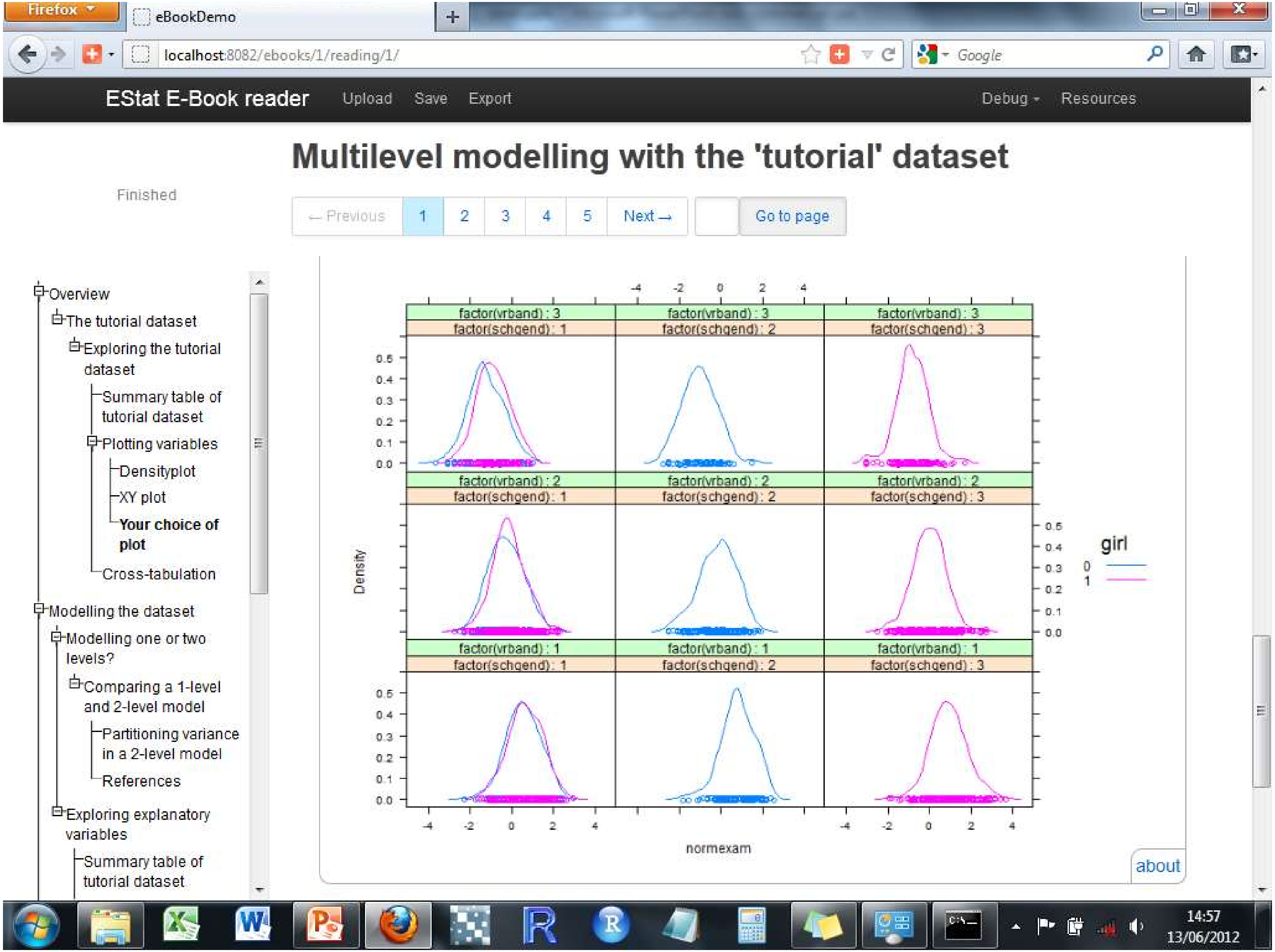
14:48
13/06/2012


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
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
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


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
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

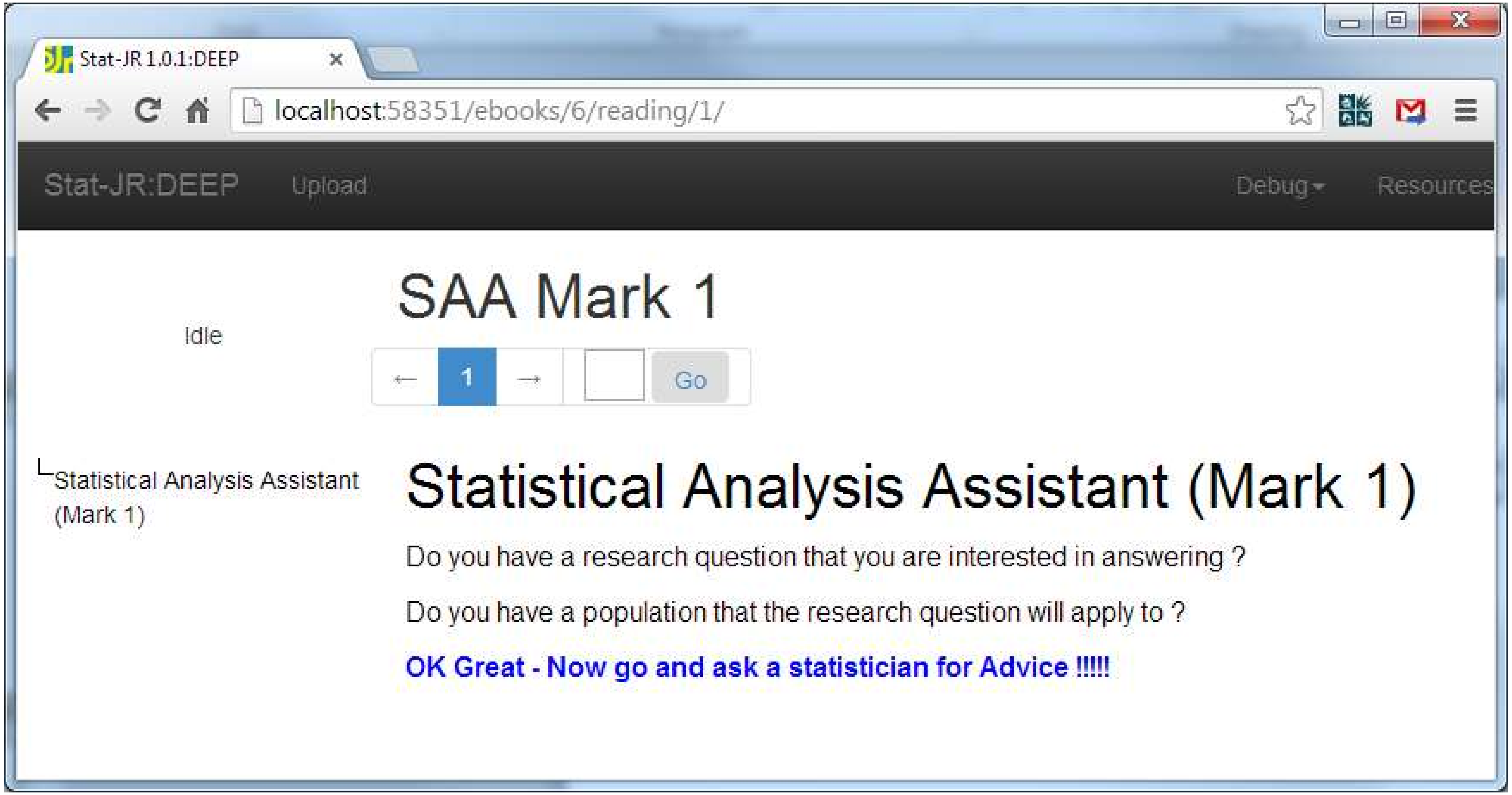


Firefox
eBookDemo
localhost:8082/ebooks/1/reading/1/
Google
EStat E-Book reader
Upload
Save
Export
Debug
Resources
Multilevel modelling with the 'tutorial' dataset
Finished
← Previous
1
2
3
4
5
Next →
Go to page
Overview
The tutorial dataset
Exploring the tutorial dataset
Summary table of tutorial dataset
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(schgend) : 1
factor(schgend) : 2
factor(schgend) : 3
factor(vrband) : 2
factor(vrband) : 1
Density
normexam
girl
0
1
about
14:57
13/06/2012

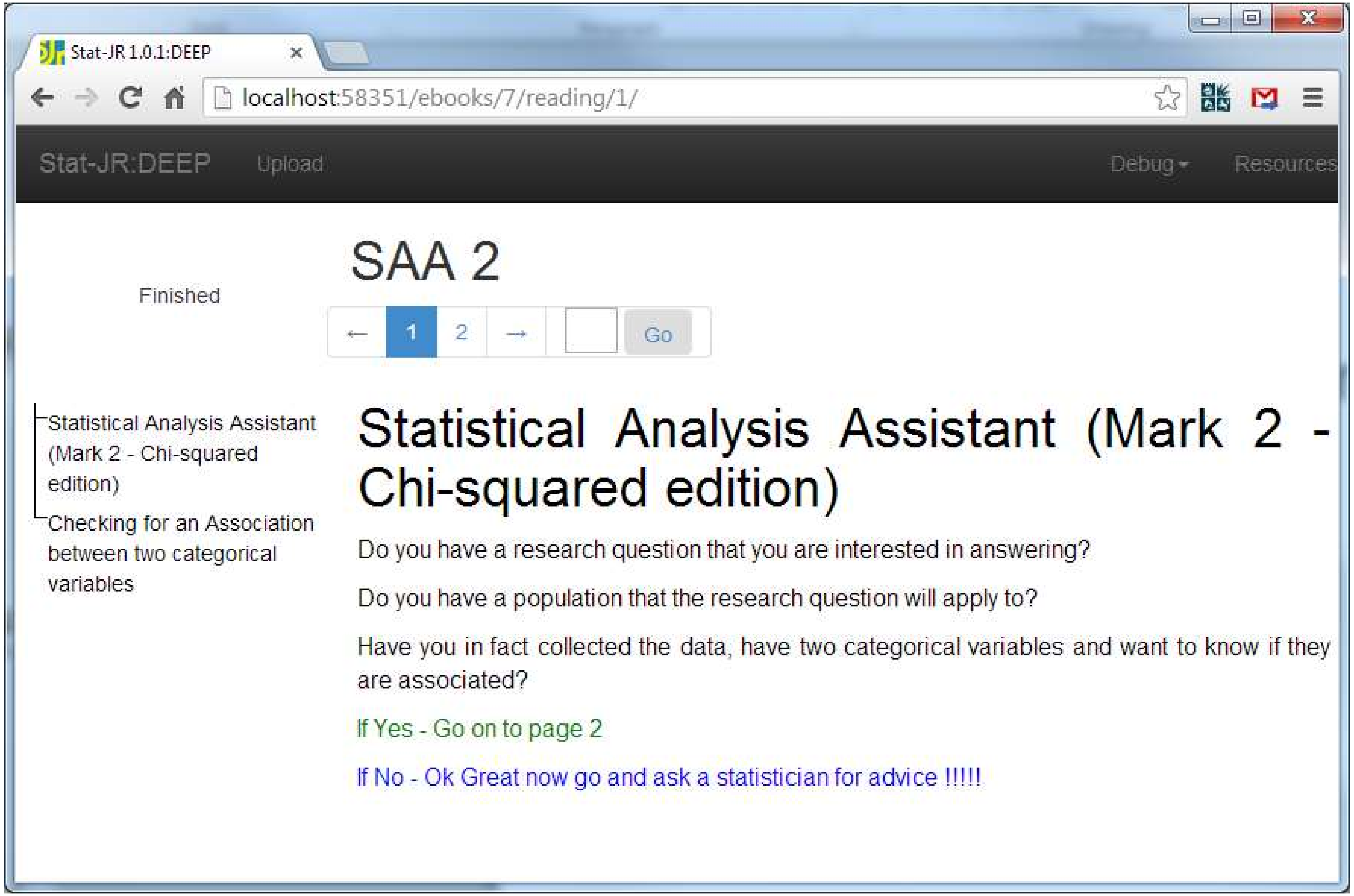
Statistical Analysis Assistants

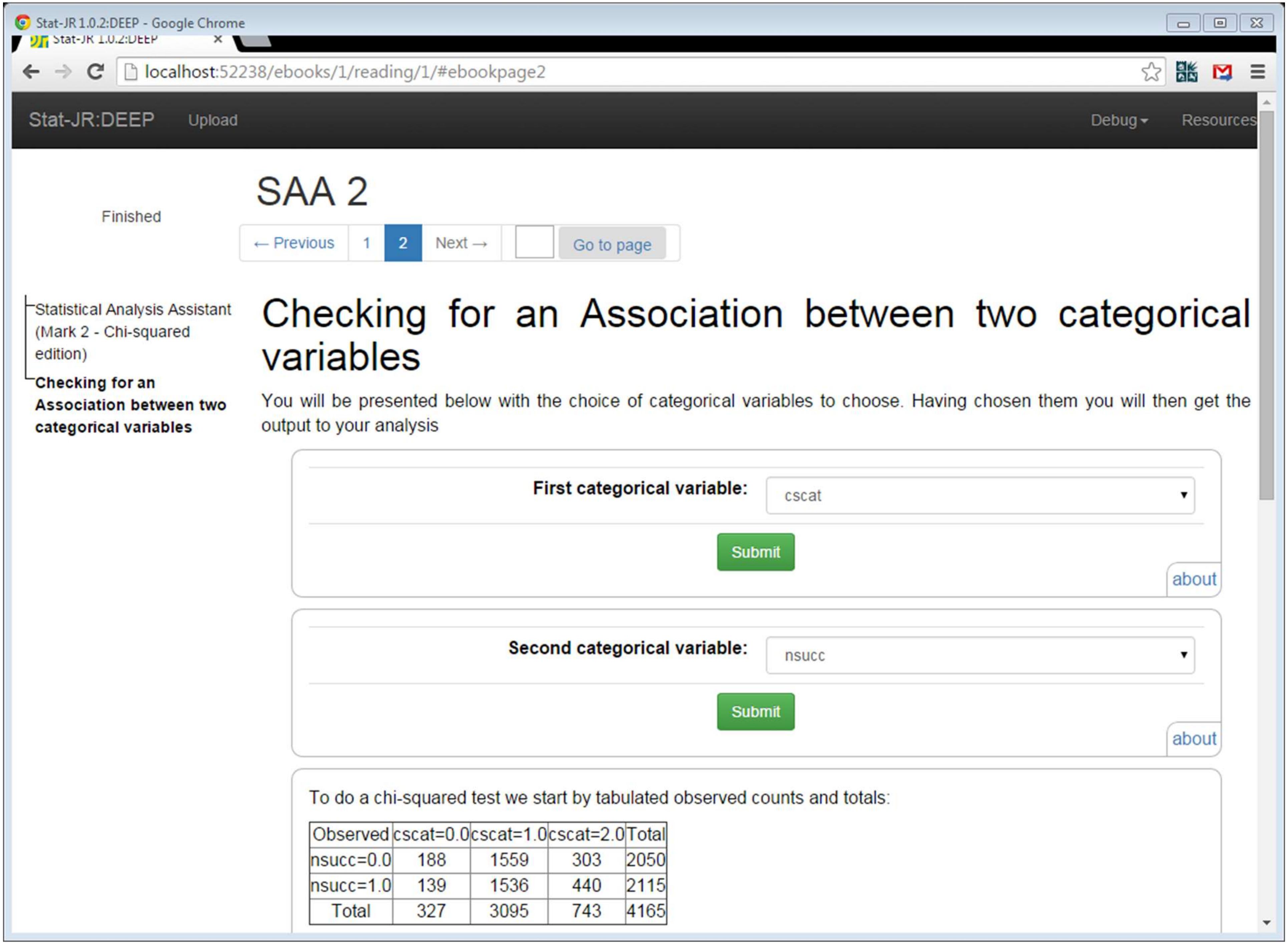
- We adapt our eBook system to allow workflows that will be constructed to describe how the steps in a statistical analysis fit together.
- There may be many SAAs adapted to different researcher's approaches – e.g. one might want to answer a research question/analyse a dataset as a specific expert might do it.
- Opinion is divided on how far one can take the idea – from nowhere to complete automation i.e. pour in the dataset at the top and let the computer sort it out.
- Probable end point will be somewhere in between or in fact a series of SAAs that lie on this continuum.
- Easiest to start with automating single operations.

A statistical analysis assistant we are all happy with!



One Step further




Stat-JR 1.0.2:DEEP - Google Chrome
Stat-JR 1.0.2:DEEP
localhost:52238/ebooks/1/reading/1/#ebookpage2
Stat-JR:DEEP
Upload
Debug
Resources
SAA 2
Finished
← Previous
1
2
Next →
Go to page
Statistical Analysis Assistant (Mark 2 - Chi-squared edition)
Checking for an Association between two categorical variables
Checking for an Association between two categorical variables
You will be presented below with the choice of categorical variables to choose. Having chosen them you will then get the output to your analysis
First categorical variable:
cscat
Submit
about
Second categorical variable:
nsucc
Submit
about
To do a chi-squared test we start by tabulated observed counts and totals:

Observed	cscat=0.0	cscat=1.0	cscat=2.0	Total
nsucc=0.0	188	1559	303	2050
nsucc=1.0	139	1536	440	2115
Total	327	3095	743	4165

Stat-JR:DEEP

localhost:59271/ebooks/4/reading/1/#ebookpage2

Apps MyBristol: Welcome LibrariesWest The Bloggess Common Cooking Mi {homemade} Healthy The Atlantic — News BBC News - Home

Stat-JR:DEEP Upload Resources About Debug

SAA 2

Finished

← Previous 1 2 Next → Go to page

Statistical Analysis Assistant (Mark 2 - Chi-squared edition)

Checking for an Association between two categorical variables

To do a chi-squared test we start by tabulated observed counts and totals:

Observed	cscat=0.0	cscat=1.0	cscat=2.0	Total
nsucc=0.0	188	1559	303	2050
nsucc=1.0	139	1536	440	2115
Total	327	3095	743	4165

We can therefore work out the expected counts from the margins of the observed data

And so we expect

E(cscat=0.0,nsucc=0.0)= Total cscat=0.0* Total nsucc=0.0/grand total = 327*2050/4165=160.95
E(cscat=1.0,nsucc=0.0)= Total cscat=1.0* Total nsucc=0.0/grand total = 3095*2050/4165=1523.35
E(cscat=2.0,nsucc=0.0)= Total cscat=2.0* Total nsucc=0.0/grand total = 743*2050/4165=365.7
E(cscat=0.0,nsucc=1.0)= Total cscat=0.0* Total nsucc=1.0/grand total = 327*2115/4165=166.05
E(cscat=1.0,nsucc=1.0)= Total cscat=1.0* Total nsucc=1.0/grand total = 3095*2115/4165=1571.65
E(cscat=2.0,nsucc=1.0)= Total cscat=2.0* Total nsucc=1.0/grand total = 743*2115/4165=377.3

So the table of expected counts is

Expected	cscat=0.0	cscat=1.0	cscat=2.0	Total
nsucc=0.0	160.95	1523.35	365.7	2050.0
nsucc=1.0	166.05	1571.65	377.3	2115.0
Total	327.0	3095.0	743.0	4165.0

We next look at differences between what we observe and expect in each cell. We square these values so that every difference is positive and scale by the expected counts so that more frequently expected cells arent overly influential. So for example for cscat=0.0, nsucc=0.0 (O-E)^2/E = (188-160.95)^2/160.95=4.55. This statistic is shown in tabular form below

Stat-JR:DEEP Upload Resources About Debug

SAA 2

Finished

← Previous 1 2 Next → Go to page

Statistical Analysis Assistant (Mark 2 - Chi-squared edition)

Checking for an Association between two categorical variables

So the table of expected counts is

Expected	cscat=0.0	cscat=1.0	cscat=2.0	Total
nsucc=0.0	160.95	1523.35	365.7	2050.0
nsucc=1.0	166.05	1571.65	377.3	2115.0
Total	327.0	3095.0	743.0	4165.0

We next look at differences between what we observe and expect in each cell. We square these values so that every difference is positive and scale by the expected counts so that more frequently expected cells arent overly influential. So for example for cscat=0.0, nsucc=0.0 (O-E)^2/E = (188-160.95)^2/160.95=4.55. This statistic is shown in tabular form below

(O-E)^2/E	cscat=0.0	cscat=1.0	cscat=2.0
nsucc=0.0	4.55	0.83	10.75
nsucc=1.0	4.41	0.81	10.42

The test statistic for a chi-squared test is found by summing the values of this table so

Chisq=4.55+0.83+10.75+4.41+0.81+10.42=31.77

This is compared with a chi-squared table with degrees of freedom = (number of columns -1)x(number of rows - 1) =

(2-1)x(3-1)=2

Looking up the chi-squared table the value for P=0.05 is 5.99 and for P=0.01 = 9.21

as 31.77 > 9.21 our P value is less than 0.01 and we have strong evidence to reject the null hypothesis (at the P=0.01) level

The p-value is in fact less than 0.0001

about

Adding contextual text to a single operation

- As we have seen with the Chi-squared example it is easy to enhance a single statistical operation like a statistical test.
- We can easily expose the steps required for the test in this case –

1. The tabulation of the observed counts
2. The calculation of the corresponding expected counts
3. The calculation of the test statistic and degree of freedom
4. The interpretation of the test, the P value and what it means in words.

What is harder is to then put what the result means into context.

Statistical tests and tables are fairly easy to enhance with intelligent textual information whilst graphs and figures are harder to enhance. Generally one has to calculate a statistic related to the figure and work with that e.g. skewness and histograms as shown later.

‘The Warlock of Firetop Mountain’ approach

- The first of a genre of interactive books published in 1982 and lapped up by 10 year old boys like myself!
- A combination of book and flowchart
- Worked something like:

‘The goblin advances towards you, shouting words that you can’t understand, do you try to make conversation (turn to page 231), run past the goblin (turn to page 176) or draw your sword and fight (turn to page 134)’

- Basically underpinning the book was effectively a flowchart disguised by random page movements with a variety of endings (99% of them involved you dying), possible loops etc.

The use of Flowcharts in Statistics

- The equivalent exists in (at least) basic statistical analysis and a variety of books have flowcharts to guide the uninitiated to the appropriate test.
- The branching rules are usually things like – how many variables do you have?, what type are they?, is a normality assumption appropriate?
- The example flowcharts usually then say you need a t test / Mann Whitney test / ANOVA etc.
- One could expand this idea to include branches where we haven't written material – i.e. the equivalent of ending up dead would be the default 'go and ask a statistician' end point – possibly taking your answers to the flow chart with you.

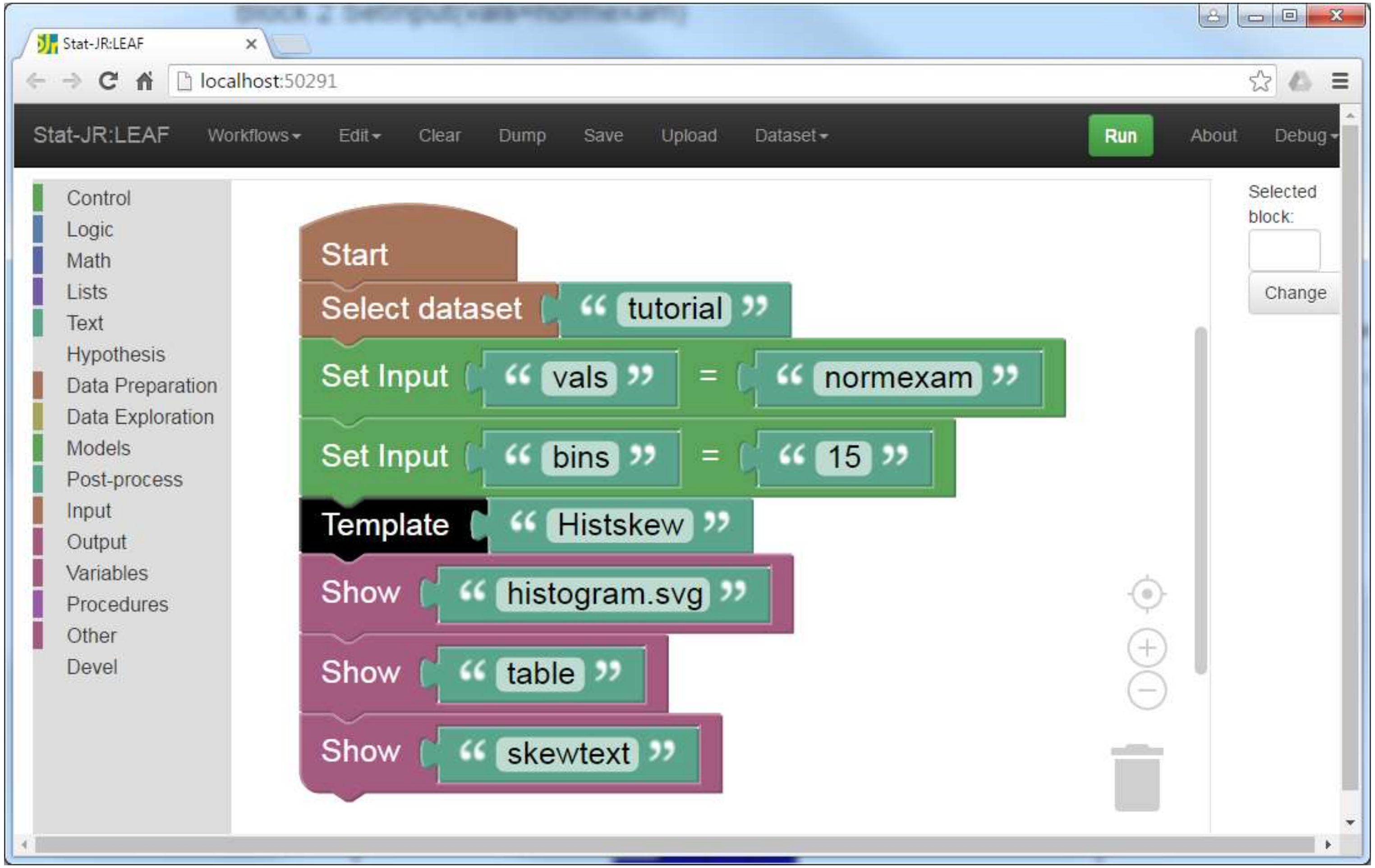
Where might this go?

- The flow chart idea is appealing as it may to some degree mimic a statistical consultation.
- If the system is flexible enough then each statistician can tune the SAA to their own approach to analysis and to how much they feel can be comfortably automated.
- Where there is uncertainty / options in what one should do this could be incorporated
- E-books can contain hyperlinks so that further background on proposed statistical methods or examples can be easily found

Workflows and StatJR LEAF

- Workflows allow the sequencing of a series of operations to perform an analysis.
- StatJR LEAF is based around a new front end written using the Blockly system.
- It allows the user to link up templates themselves in a user-friendly visual way.
- Work flows can then be included in eBooks.
- We will use this system in the SAAs.

Skewness / Histogram workflow



Here is a logfile style workflow. Basically we select a dataset then fit a histogram to a variable and display several objects.

Skewness / Histogram workflow

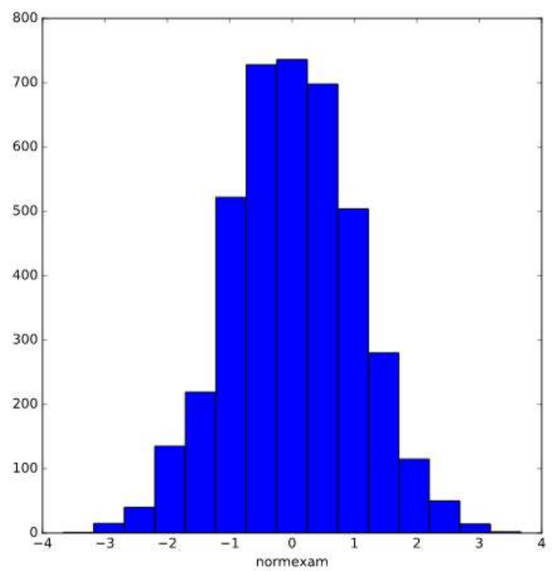
Stat-JR:LEAF

Stat-JR:LEAF

localhost:50291/run/

Stat-JR:LEAF

Block 5 OutputObject(histogram.svg)



Block 7 OutputObject(skewtext)

Here the distribution is reasonably symmetric.

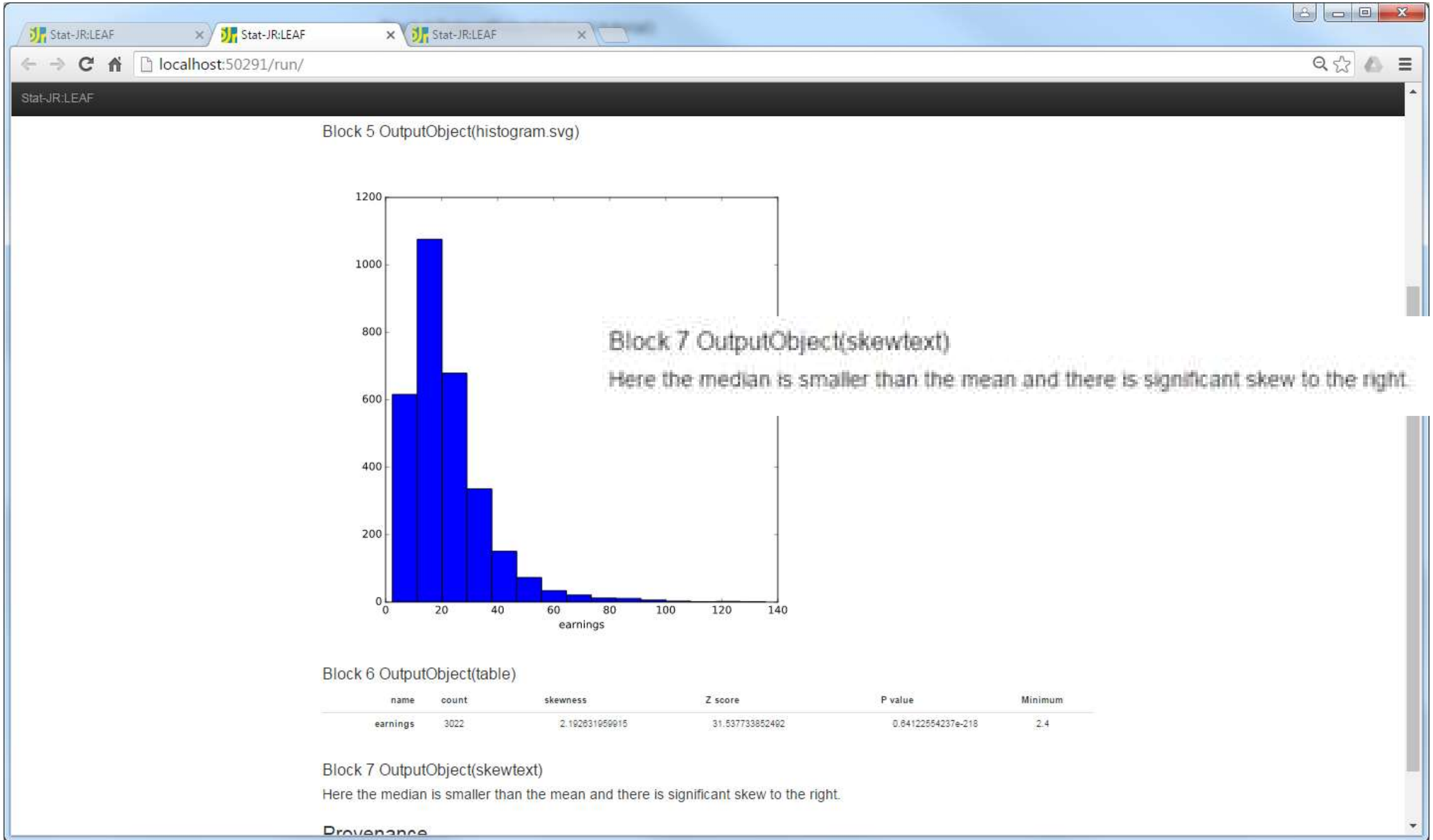
Block 6 OutputObject(table)

name	count	skewness	Z score	P value	Minimum
normexam	4059	0.0037421034649	0.0975093737986	0.922321890491	-3.66607

Block 7 OutputObject(skewtext)

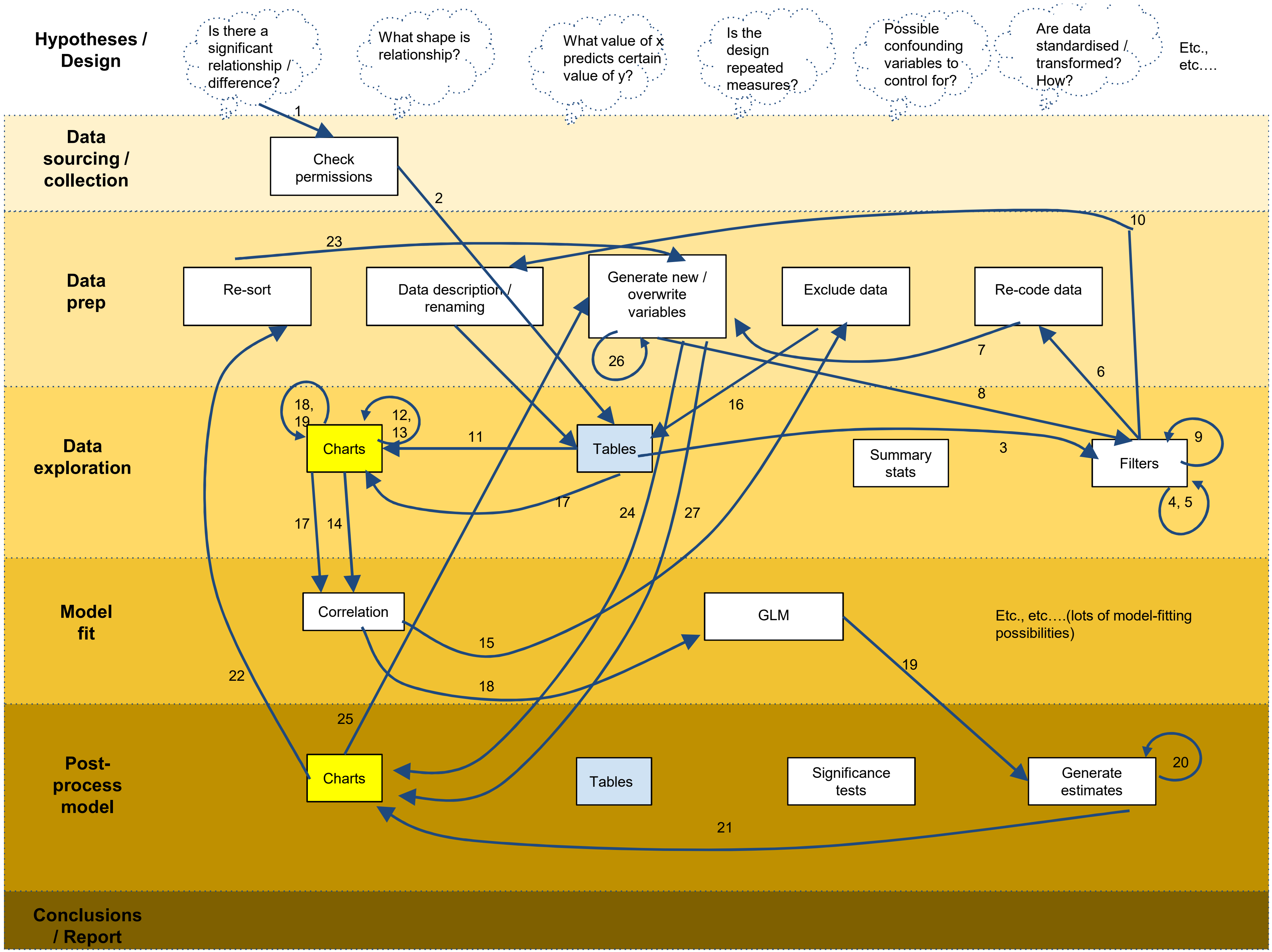
Here the distribution is reasonably symmetric.

Skewness / Histogram workflow

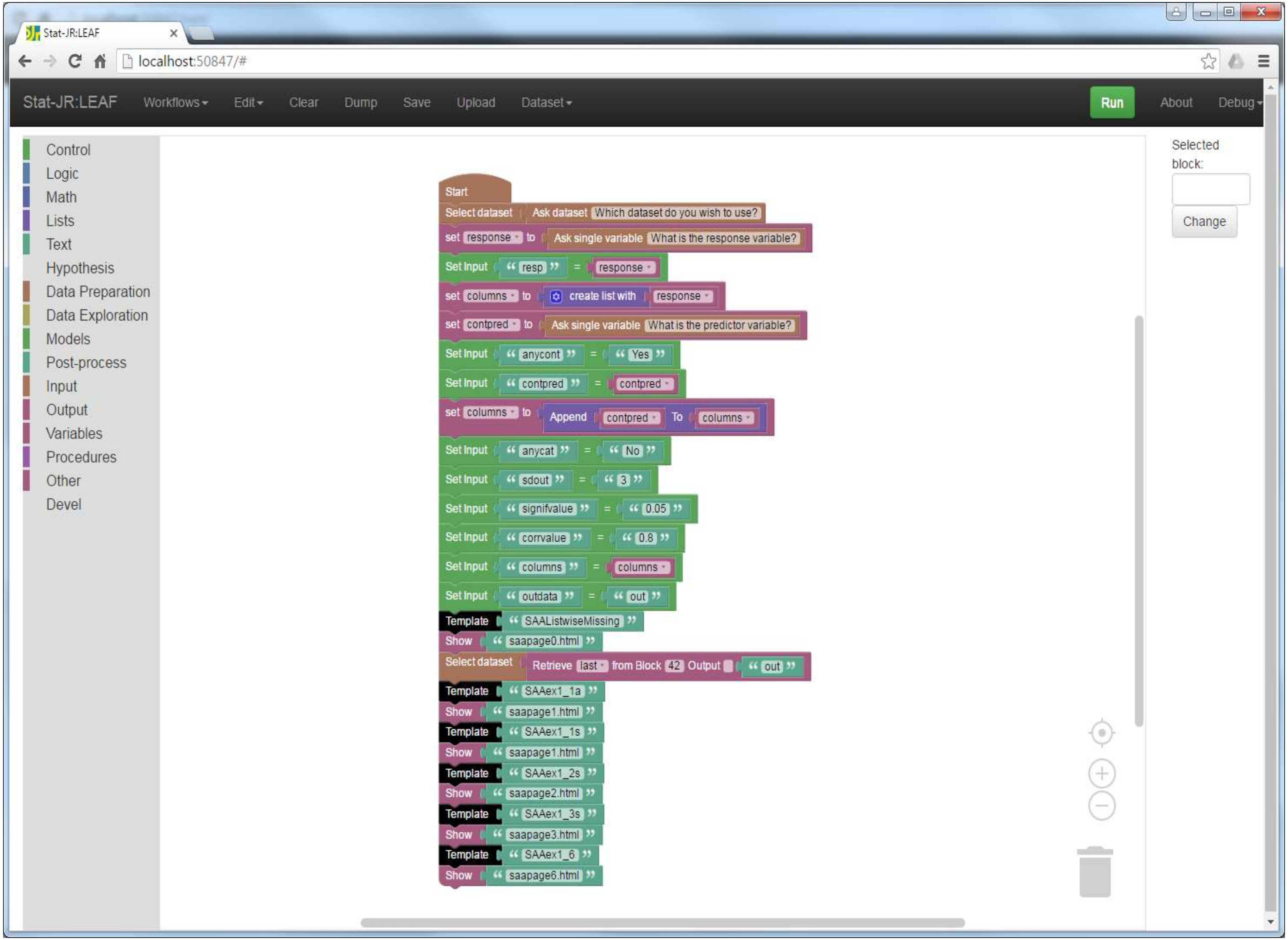


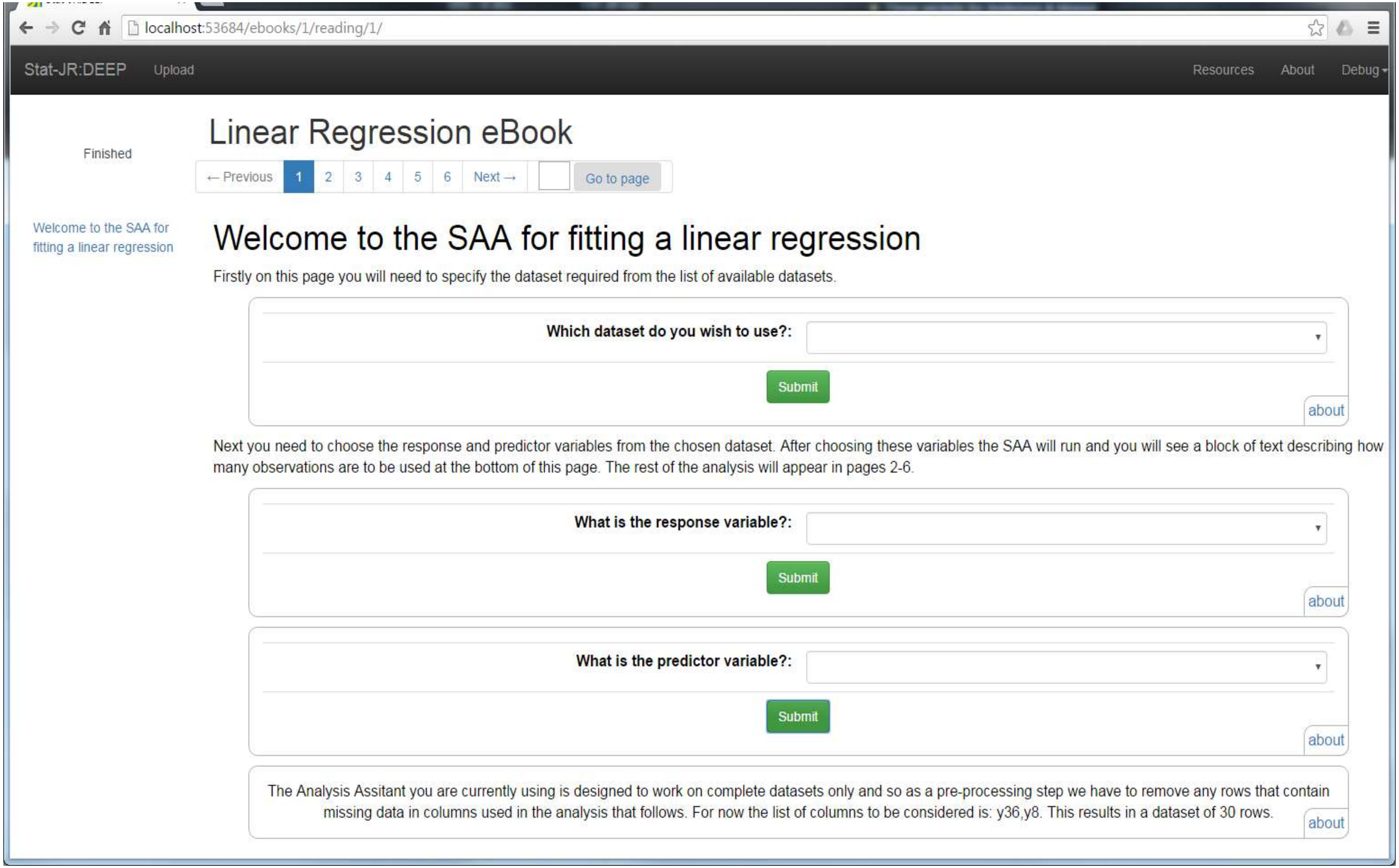
More complex operations – linear regression

- When we looked at the chi-squared test earlier we already broke the test down into a series of steps which formed the test.
- For a regression analysis we might have additional steps to translate from simply a test to an analysis.
- We might do some initial exploratory data analysis and possible transform variables.
- We will clearly do the model fit itself but we will probably then also do some post-processing steps – for example analysis of the residuals and plotting the model predictions
- We will demonstrate an SAA for a linear regression but first show an example of a flow-chart for a real analysis.



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
17
18
18, 19
19
20
21
22
23
24
25
26
27


Stat-JR:LEAF
localhost:50847/#
Stat-JR:LEAF
Workflows
Edit
Clear
Dump
Save
Upload
Dataset
Run
About
Debug
Control
Logic
Math
Lists
Text
Hypothesis
Data Preparation
Data Exploration
Models
Post-process
Input
Output
Variables
Procedures
Other
Devel
Selected block:
Change
Start
Select dataset Ask dataset Which dataset do you wish to use?
set response to Ask single variable What is the response variable?
Set Input " resp " = response
set columns to create list with response
set contpred to Ask single variable What is the predictor variable?
Set Input " anycont " = " Yes "
Set Input " contpred " = contpred
set columns to Append contpred To columns
Set Input " anycat " = " No "
Set Input " sdout " = " 3 "
Set Input " signifvalue " = " 0.05 "
Set Input " corrvalue " = " 0.8 "
Set Input " columns " = columns
Set Input " outdata " = " out "
Template " SAAListwiseMissing "
Show " saapage0.html "
Select dataset Retrieve last from Block 42 Output " out "
Template " SAAex1_1a "
Show " saapage1.html "
Template " SAAex1_1s "
Show " saapage1.html "
Template " SAAex1_2s "
Show " saapage2.html "
Template " SAAex1_3s "
Show " saapage3.html "
Template " SAAex1_6 "
Show " saapage6.html "


localhost:53684/ebooks/1/reading/1/
Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
Linear Regression eBook
← Previous
1
2
3
4
5
6
Next →
Go to page
Welcome to the SAA for fitting a linear regression
Welcome to the SAA for fitting a linear regression
Firstly on this page you will need to specify the dataset required from the list of available datasets.
Which dataset do you wish to use?:
Submit
about
Next you need to choose the response and predictor variables from the chosen dataset. After choosing these variables the SAA will run and you will see a block of text describing how many observations are to be used at the bottom of this page. The rest of the analysis will appear in pages 2-6.
What is the response variable?:
Submit
about
What is the predictor variable?:
Submit
about
The Analysis Assitant you are currently using is designed to work on complete datasets only and so as a pre-processing step we have to remove any rows that contain missing data in columns used in the analysis that follows. For now the list of columns to be considered is: y36,y8. This results in a dataset of 30 rows.
about

Stat-JR:DEEP

localhost:53684/ebooks/1/reading/1/#page2

Stat-JR:DEEP Upload Resources About Debug

Finished

Welcome to the SAA for fitting a linear regression

Linear Regression eBook

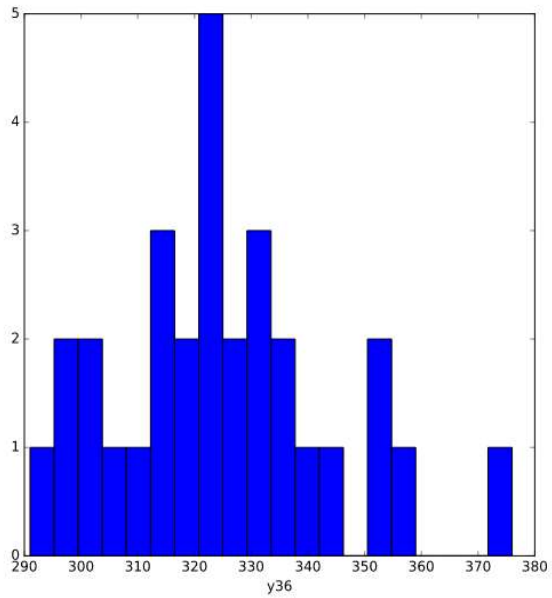
← Previous 1 2 3 4 5 6 Next → Go to page

We will begin our analysis of the dataset by doing some basic data exploration.

You have chosen y36 as your response variable and so a first step is to take a look at this variable and assess its suitability for a normal model. The summary statistics for the variable are in the table below

Observations	30
Mean	324.8
Standard Deviation	19.132
Median	323.5

We also look at a histogram of y36 to see if it is approximately normally distributed. Although in modelling the response in terms of a set of predictors it is what is unexplained (the model residuals) that need to be normally distributed, it is still useful to look at the response variable as a very skewed variable will often lead to very skewed residuals.

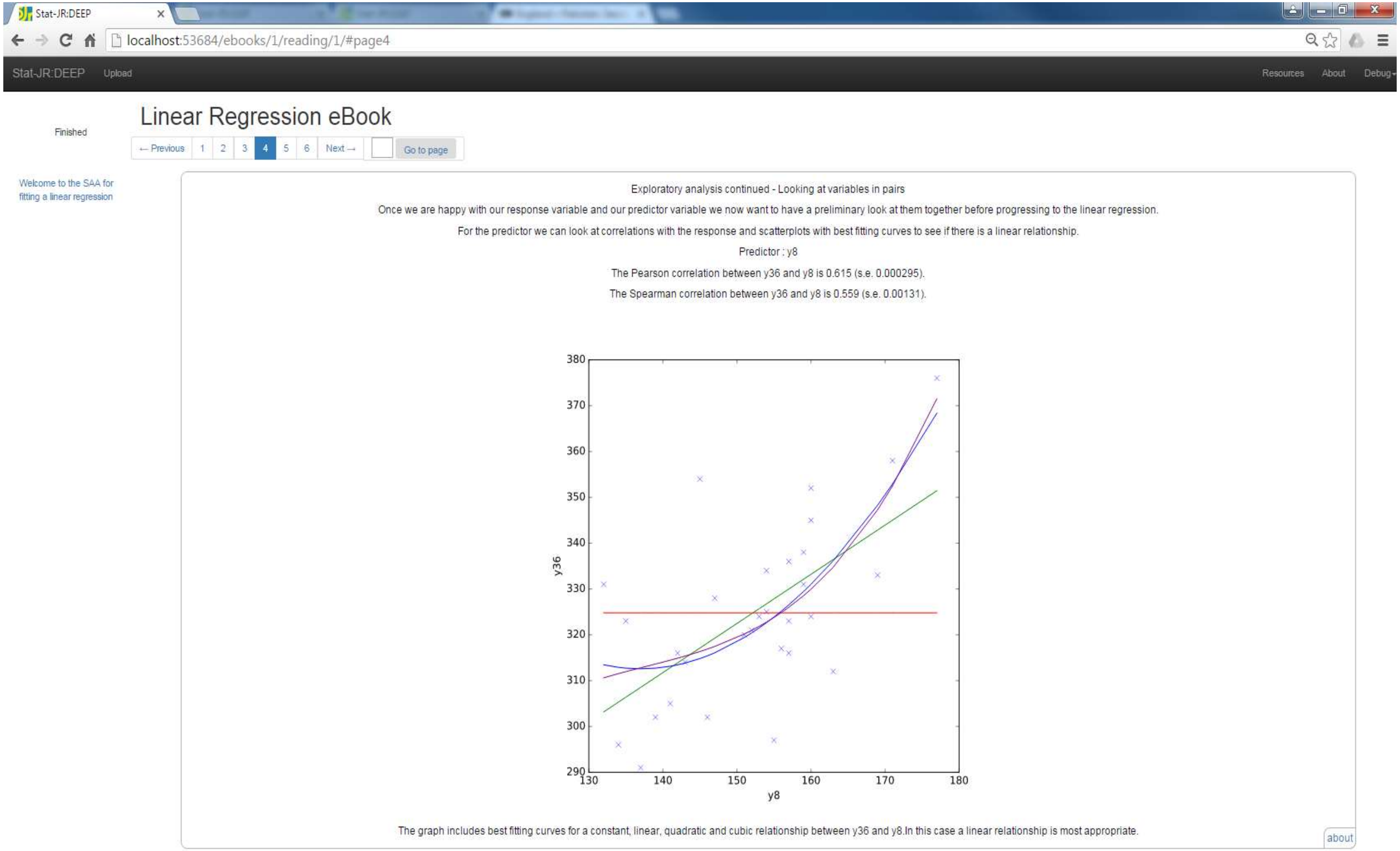


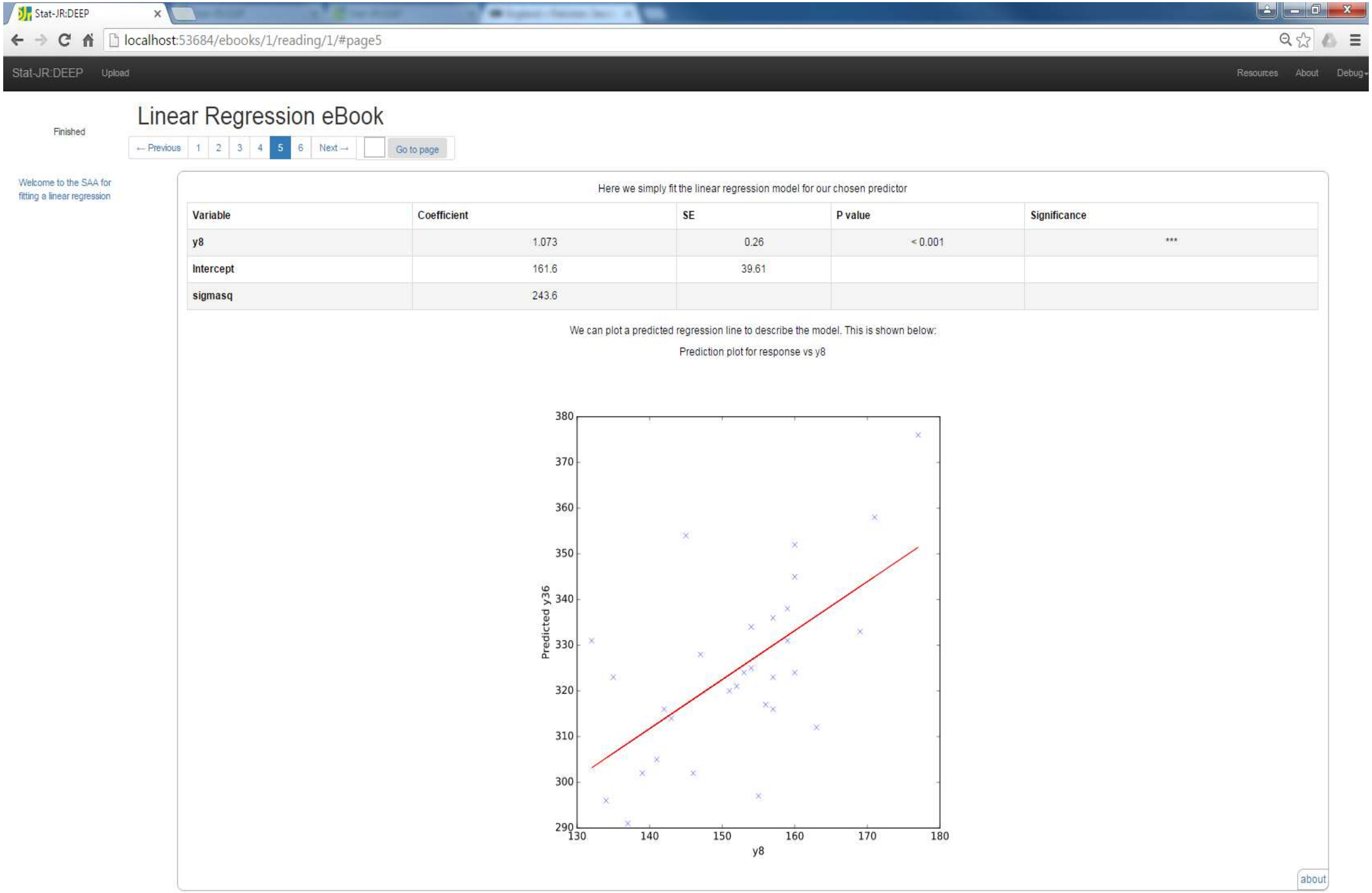
Here the distribution is reasonably symmetric with skewness value 0.518.

There are no obvious outliers in y36.

about

University of BRISTOL Centre for Multilevel Modelling

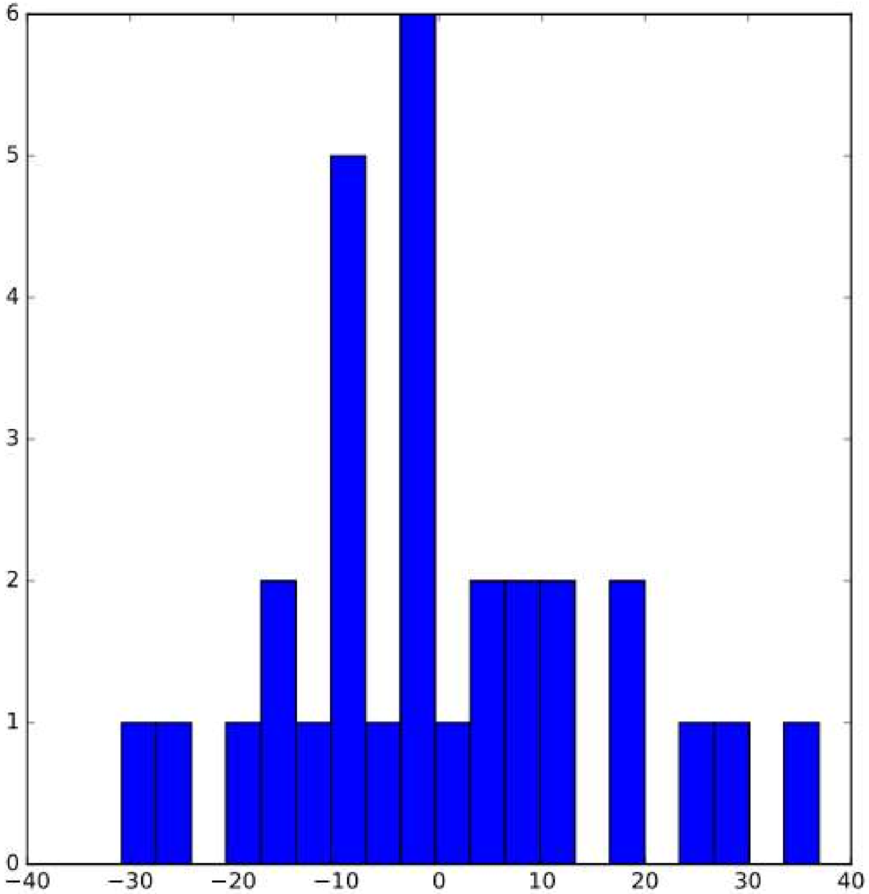

Stat-JR:DEEP
localhost:53684/ebooks/1/reading/1/#page4
Stat-JR:DEEP
Upload
Resources
About
Debug
Linear Regression eBook
Finished
← Previous 1 2 3 4 5 6 Next →
Go to page
Welcome to the SAA for fitting a linear regression
Exploratory analysis continued - Looking at variables in pairs
Once we are happy with our response variable and our predictor variable we now want to have a preliminary look at them together before progressing to the linear regression.
For the predictor we can look at correlations with the response and scatterplots with best fitting curves to see if there is a linear relationship.
Predictor : y8
The Pearson correlation between y36 and y8 is 0.615 (s.e. 0.000295).
The Spearman correlation between y36 and y8 is 0.559 (s.e. 0.00131).
y36
y8
The graph includes best fitting curves for a constant, linear, quadratic and cubic relationship between y36 and y8.In this case a linear relationship is most appropriate.
about



Variable	Coefficient	SE	P value	Significance
y8	1.073	0.26	< 0.001	***
Intercept	161.6	39.61		
sigmasq	243.6			

Here we look at the residuals from the model and plot them in various ways.

Histogram of residuals

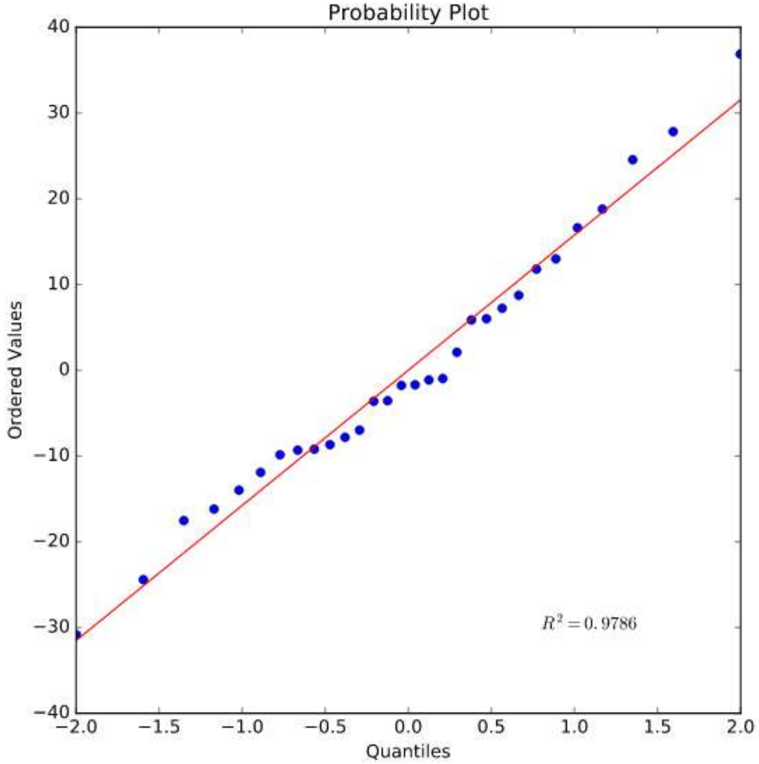


Here the distribution is reasonably symmetric with skewness value 0.399.

There are no obvious outliers in the residuals.

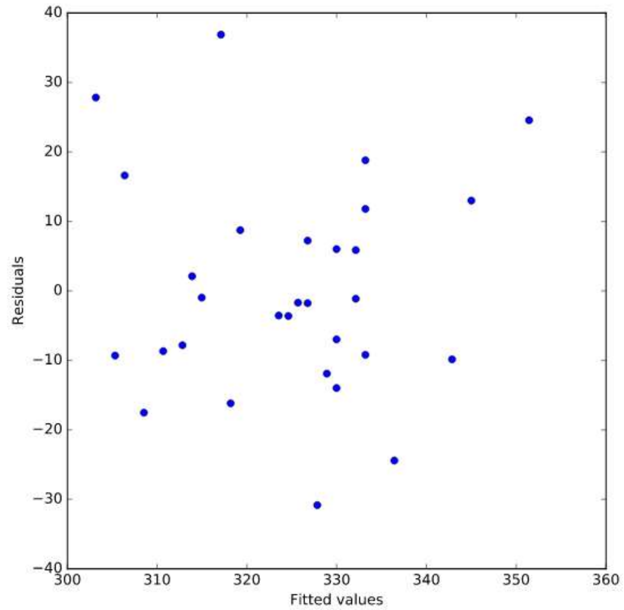
See http://www.bristol.ac.uk/cmm/media/software/statjr/downloads/manuals/1-06/manual-saa.pdf for more details and multilevel SAAs

Quantile-Quantile Plot of residuals



If the residuals are fairly normally distributed then the points in this graph should be close to the red line.

Residuals vs fitted values



Here you should consider whether there are any patterns in this plot. Ideally we would like to see similar variability of the residuals across the range of fit

Small Area Estimation (SAE) – Recap

- In SAE the focus on estimating quantities of interest for each of a series of 'small areas' – for example voting constituencies, administrative regions or schools.
- Each such 'small area' contains a large number of units – voters, people, children on which we can measure a variable e.g. voting intention, annual pay, exam results and we then wish to construct summary estimates (mean, variability, maximum etc.) of this variable for each small area.
- There are two main approaches for SAE which depend on what data is available, area level models and unit level models and here we will concern ourselves with unit level models.

Unit Level Models

- Unit level models are so called because we have some information on the individual units.
- Unit level models require 2 datasets:

The first **census** (population) dataset contains a list of variables X for all units in the population.

The second **survey** (sample) dataset contains both Y, our variable of interest and the same X variables as in the **census** dataset.

- The basic idea is to fit a model to the **survey** dataset relating Y to X and typically this is a multilevel model.
- Then we use the estimates from this model to predict Y for all units in the **census** dataset.
- Finally we can construct summary small area estimates by constructing summary statistics from the predictions on the census dataset.

Classical approach using emdi

- The emdi package (Kreutzmann et al. 2017) in R will fit SAE models using a classical approach (Tzavidis project).
- It offers direct estimation and model-based estimation using the empirical Bayes prediction approach of Molina and Rao (2010).
- As well as calculating SAE estimates for the mean, standard deviation and quantiles it also calculates several poverty based indicators such as the Gini coefficient, poverty gap, headcount ratio and quintile share ratio.
- It only fits normal responses but allows transformations (log and Box-Cox)
- Other classical approaches include the World Bank method (e.g. Elbers et al., 2002)

MCMC approach in StatJR

- In StatJR we have implemented the World Bank method which essentially contains 3 steps.
- Firstly a 2-level (units within small areas) model is fitted to the sample dataset to produce chains of parameter estimates.
- Secondly these estimate chains and the data for each unit in the population are used to then produce chains of values for each unit.
- Finally the estimated values for the units in each small area at each iteration are then used to construct the aggregate small area estimates e.g. small area mean, sd, median etc.
- As there is a chain of values we can therefore measure the uncertainty in these estimates.

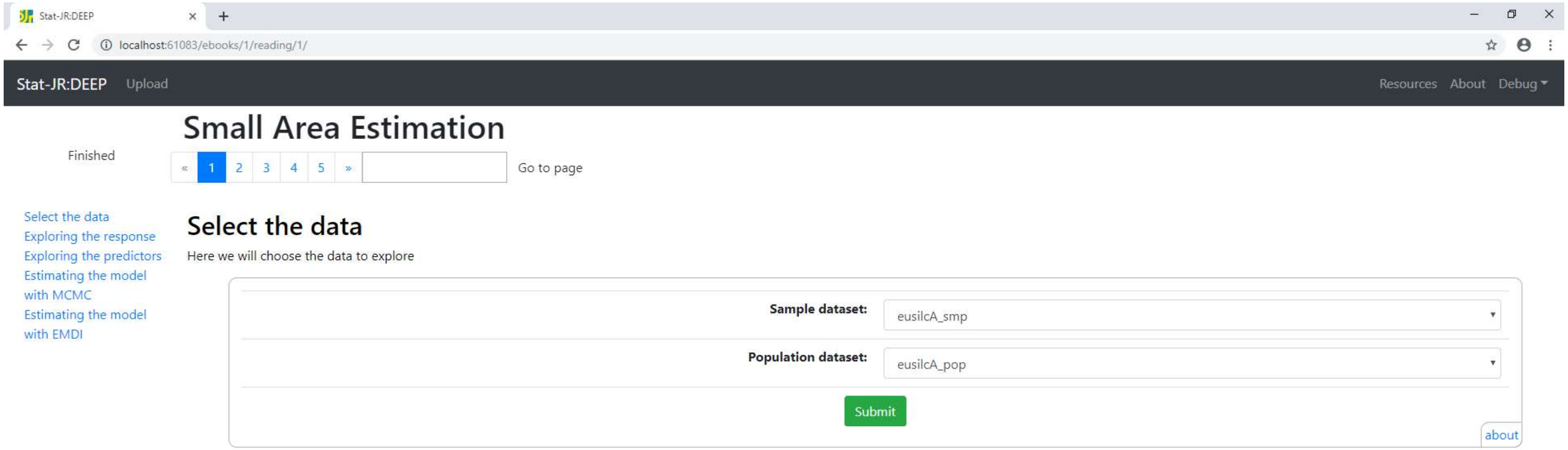
Parallel processing speedups

- Generally the sample dataset is significantly smaller than the population dataset and so the computations in stage 2 on the last slide are more computationally intensive than in stage 1.
- However calculating predictions for each unit in each small area can be done independently conditional on stage 1 and so thus we farm out the process to different processors.
- We have investigated several approaches including OpenMP along with the standard C++ library functions for parallel processing.
- In the end we found that the standard approach worked better and was a large speed up compared to the non-parallel approach and this is implemented in StatJR.

Example: EU-SILC data from Austria 2006

- This is the example data used with *emdi* and consists of a simulated dataset that mimics the European Union Statistics on Income and Living Conditions (EU-SILC) dataset for Austria in 2006.
- The response of interest is equivalized household income (income / household size) and there are several explanatory variables.
- The population (census) dataset is of size 25,000 individuals from 94 districts whilst the sample (survey) dataset is of size 1,945 individuals with some districts having no observations.
- As the response is income we can look at some of the interesting poverty related summary measures.

On page 1 of the eBook we simply input the datasets that contain the sample and population data.



On page 2 we do some exploratory data analysis of the response variable including looking at possible transformations to allow normality.

Stat-JR:DEEP Upload Resources About Debug

Small Area Estimation

Finished

« 1 2 3 4 5 » Go to page

Select the data
Exploring the response
Exploring the predictors
Estimating the model with MCMC
Estimating the model with EMDI

In this SAA we will particularly be concerning ourselves with unit level models. For a unit level model we require 2 datasets – a sample dataset and a population dataset. The sample dataset contains a sample of individual from some (but not necessarily all) of the groups in the population and for each individual the variable of interest which we will call Y (e.g. salary, voting intention) is collected along with a lot of other variables which we will call X that might be thought to predict the variable of interest (e.g. gender, benefits, family size).

The second population dataset contains records for the WHOLE population i.e. everybody in all groups. This dataset contains the same predictor variables X but here the variable of interest Y is absent. The rationale for the unit level model is therefore to fit a (multilevel) regression model to the Y in the sample dataset to investigate the relationship between Y and X. We then use this model to predict the values of Y for the WHOLE population using the population dataset and then use the estimated Y produced to estimate small area quantities (e.g. means, proportions and percentiles) for each small area.

We will first take a look at the response variable that we wish to estimate at our small areas. We will on this page look at some summary information about this variable and also consider whether the variable needs transforming. We often transform variables so that we can fit a Normal response model and assume normality for the residuals. So firstly we ask for the name of the response variable and a value for the parameter lambda used in the Box Cox transformation later.

Response variable: eqIncome

Common ID variable: district

Lambda parameter for response tranformation: 0.6

Submit

about

Here we see a summary of how big the sample is and then a table explaining the relative numbers in the sample from each small area

Stat-JR:DEEP Upload Resources About Debug

Small Area Estimation

Finished

« 1 2 3 4 5 » Go to page

- Select the data
- **Exploring the response**
- Exploring the predictors
- Estimating the model with MCMC
- Estimating the model with EMDI

In your dataset the small areas are represented by the variable name district and there are 25000 individuals in the population that come from 94 areas in total. The sample dataset has 1945 individuals in total (7.78% of the population) with individuals in the sample coming from 70 areas.

about

In the table below we will look at how representative the sample is of the population in each small area. The larger percentage of the population that is in the sample the more confidence we will have in our small area estimates and the less we will have to use the response – predictor variable relationships across all areas to estimate those small area estimates.

Code	Name	Nsamp	Npop	%
1	Eisenstadt-Umgebung	0	115	0.00
2	Eisenstadt (Stadt)	0	37	0.00
3	Güssing	0	74	0.00
4	Jennersdorf	0	49	0.00
5	Mattersburg	0	109	0.00
6	Neusiedl am See	16	155	10.32
7	Oberpullendorf	0	105	0.00
8	Oberwart	15	150	10.00

Next we look at the VPC to see how important the small areas are in the modelling. This is followed by a histogram of the untransformed variable.

Stat-JR:DEEP Upload Resources About Debug

Small Area Estimation

Finished

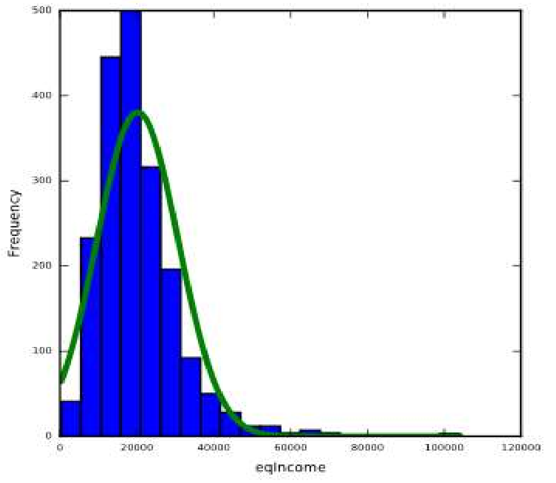
« 1 2 3 4 5 » Go to page

Select the data
Exploring the response
Exploring the predictors
Estimating the model with MCMC
Estimating the model with EMDI

When we look at eqIncome in the sample dataset there is variability in the average response across the small areas that we are estimating values for. It is important to estimate how much variation in the response is between small areas and how much is within small areas and this is done with a statistic called the VPC. Here the VPC is 0.4. This means that 40% of the variation in eqIncome is between small areas and therefore due to differences across areas. These differences may be explained by the predictor variables in our later modelling.

about

We next look at the shape of the response variable to see whether it needs transforming. First we look at the response itself as shown in the histogram below.



about

We can look at transformed variables, firstly log and finally the Box-Cox transform of the response and here for the value 0.6 for Lambda we still see a slight right hand skew

Stat-JR:DEEP Upload Resources About Debug

Small Area Estimation

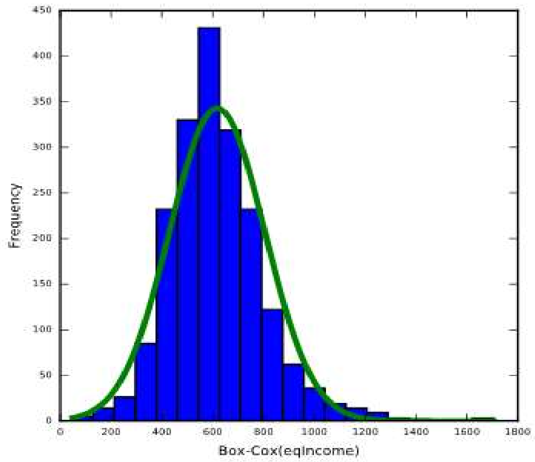
Finished

« 1 2 3 4 5 » Go to page

Select the data
Exploring the response
Exploring the predictors
Estimating the model with MCMC
Estimating the model with EMDI

A more general transformation is the Box-Cox transformation which transforms the original response y to the function $y_i^{(\lambda)} = \begin{cases} \frac{y_i^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\ ln y_i, & \lambda = 0 \end{cases}$

where lambda is a parameter that needs inputting (as you did at the top of the page). Again the Box-Cox requires a shift prior to transforming. For the current value of λ we get the histogram shown below

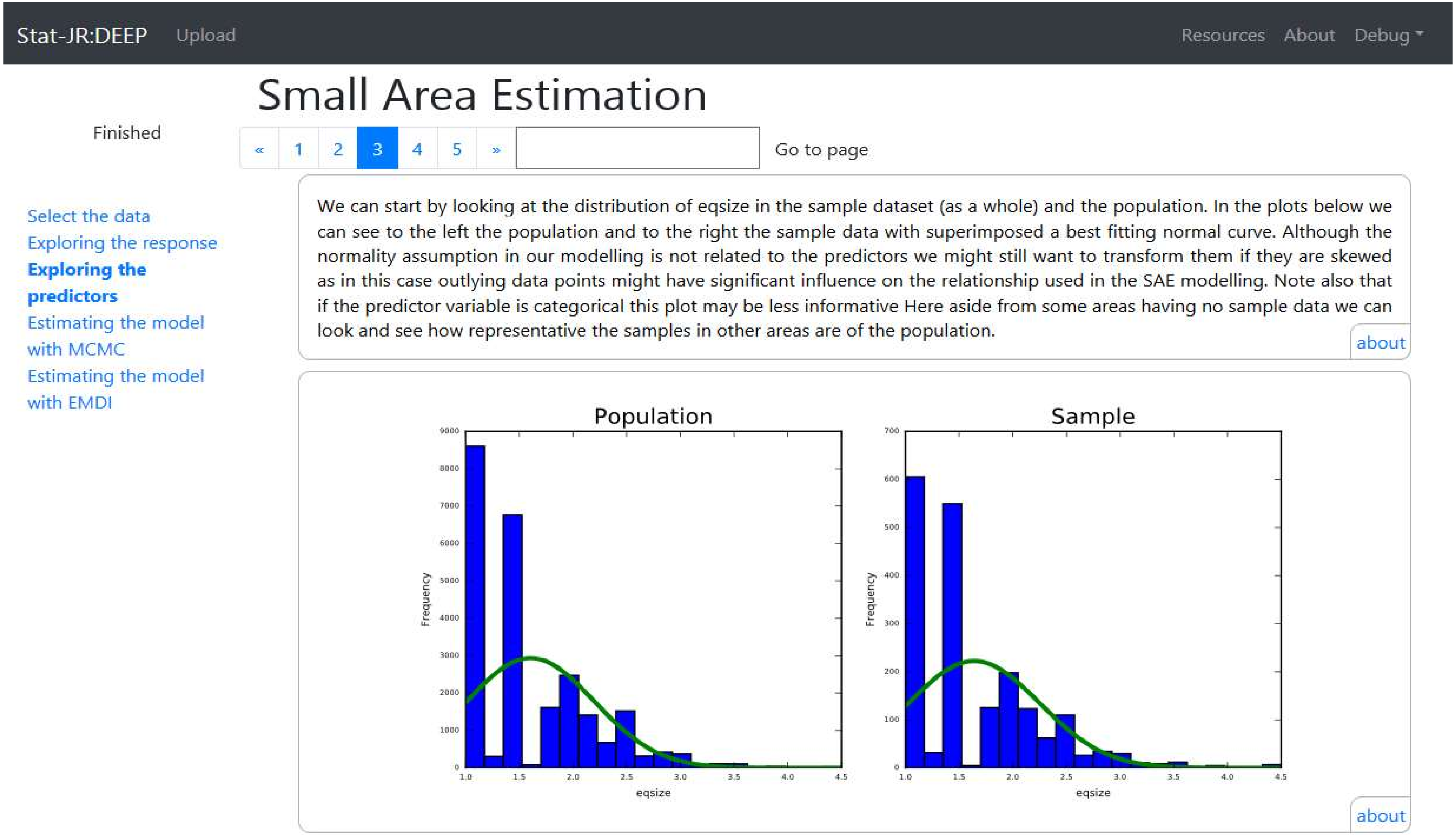


about

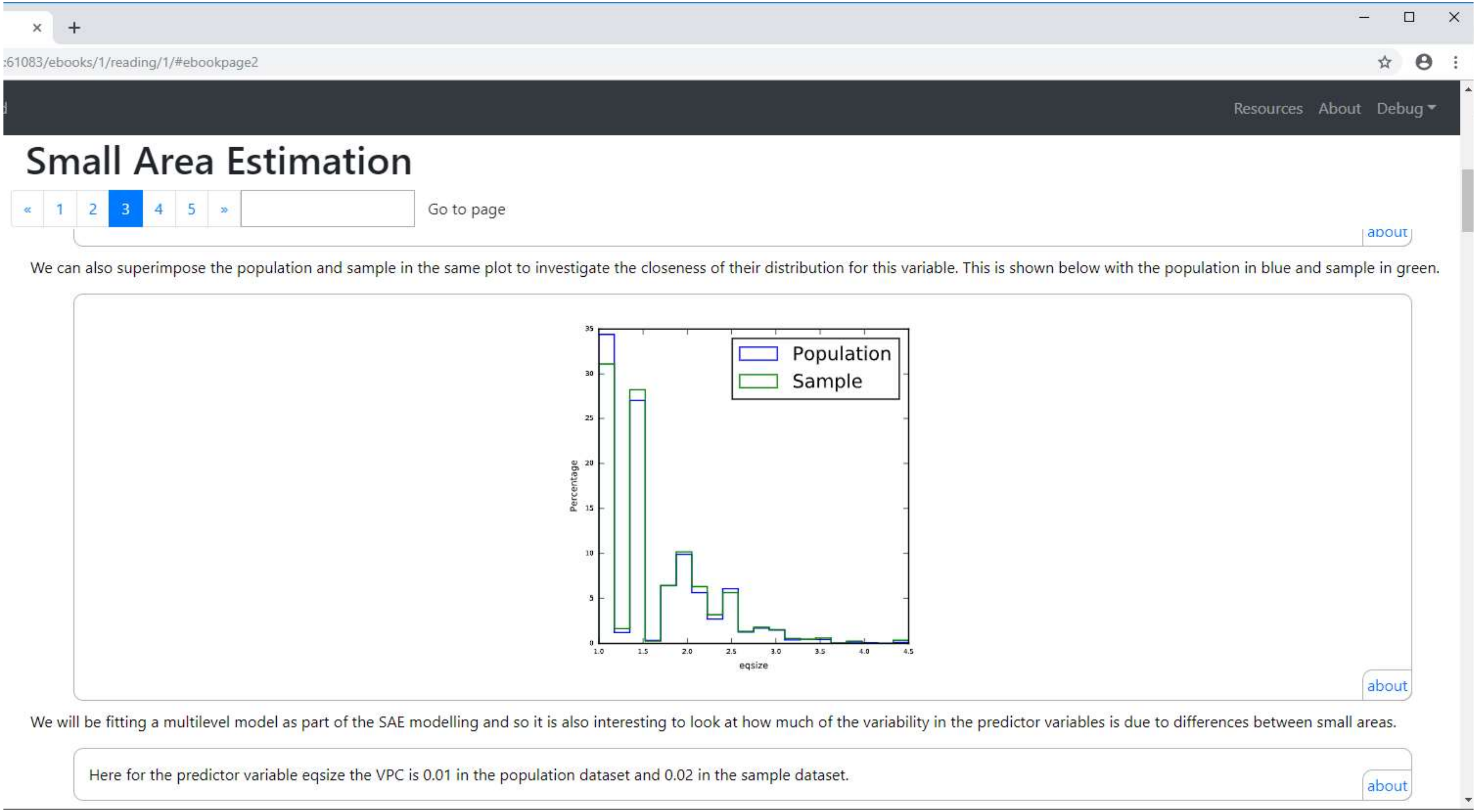
Here the median is smaller than the mean and there is significant skew to the right. The skewness value is 0.904. Here the statistical significance may be to some degree due to the large sample size as from a practical perspective values of skew less than 2 are not considered too big a skew.

about

Page 3 allows us to look at any of the predictor variables before we then fit them in the model. Here we have chosen eq-size and we can see that the distribution of the variable is similar in the sample as in the population



An alternative approach is to superimpose the two distributions for the population and sample on the same plot to look for differences. Here we also see the VPC calculated and the eqsize variable seems to exhibit very little variation across small areas



We can also look at differences for the sample and population for each small area

t:61083/ebooks/1/reading/1/#ebookpage2

Resources About Debug

Small Area Estimation

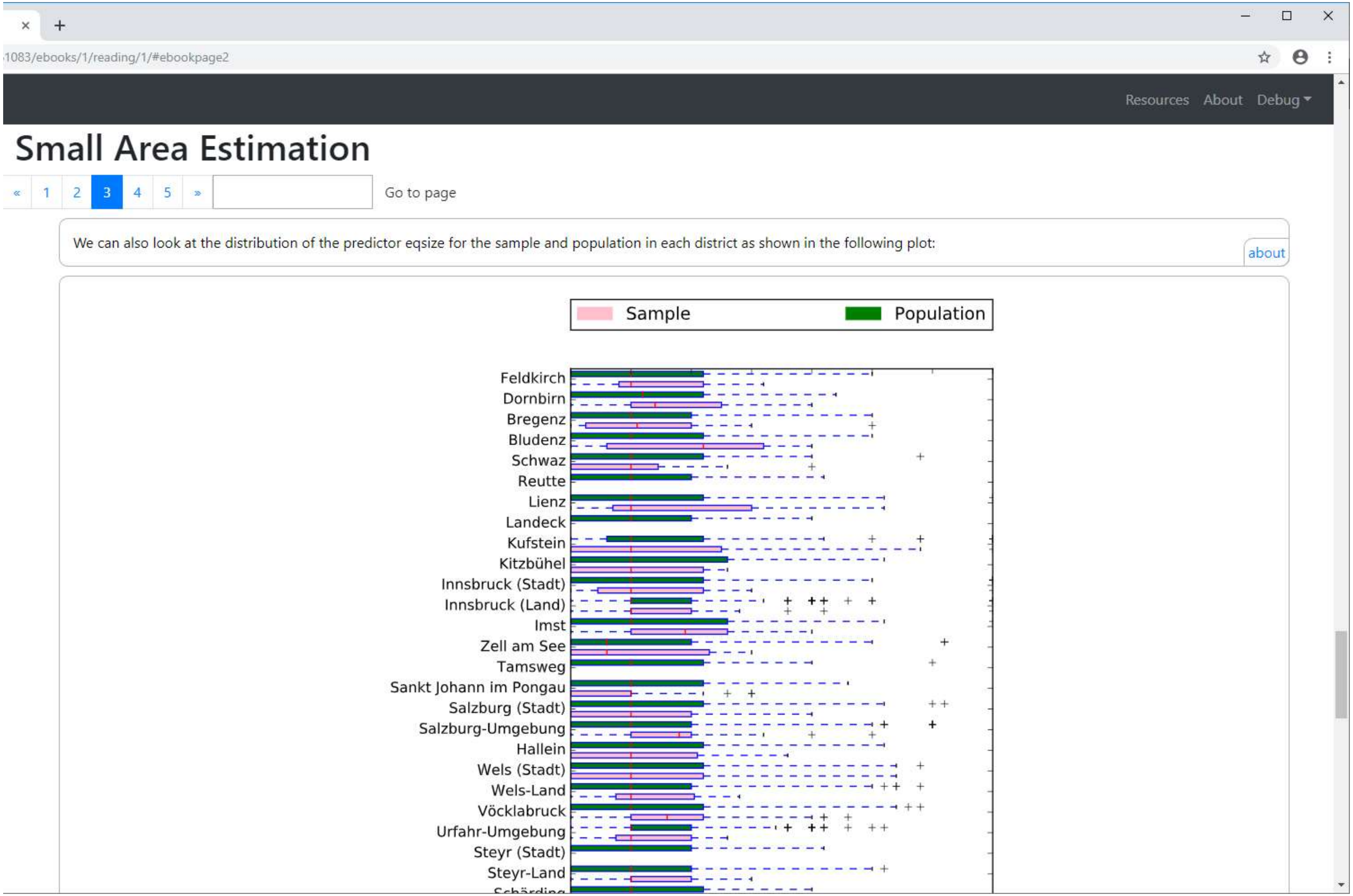
« 1 2 3 4 5 » Go to page

We will next look at the data in the individual small areas, and in particular the mean and standard deviation of the variable eqsize for both the sample and the population dataset. Here aside from some areas having no sample data we can look and see how representative the samples in other areas are of the population.

about

Code	Name	Sample mean	Sample sd	Population mean	Population sd
1	Eisenstadt-Umgebung	--	--	1.75	0.52
2	Eisenstadt (Stadt)	--	--	1.65	0.24
3	Güssing	--	--	1.59	0.59
4	Jennersdorf	--	--	1.58	0.61
5	Mattersburg	--	--	1.69	0.56
6	Neusiedl am See	1.62	0.41	1.64	0.55
7	Oberpullendorf	--	--	1.73	0.64
8	Oberwart	1.61	0.60	1.60	0.57
9	Rust (Stadt)	--	--	1.32	0.44
10	Amstetten	1.61	0.69	1.58	0.62
11	Baden	1.79	0.50	1.72	0.60
12	Bruck an der Leitha	1.83	0.73	1.72	0.58
13	Gänserndorf	1.91	0.64	1.71	0.60
14	Gmünd	--	--	1.52	0.60

This can also be done graphically via box plots as shown below:



On page 4 we finally get to fit small area estimation models and here we try a model with all the possible predictors

Stat-JR:DEEP Upload Resources About Debug

Running Python_script

Small Area Estimation

« 1 2 3 4 5 » Go to page

Select the data
Exploring the response
Exploring the predictors
Estimating the model with MCMC
Estimating the model with EMDI

Estimating the model with MCMC

Now that we have looked at the response and predictor variables we will next fit a small area estimation model. Here we fit a multilevel model to the sample dataset and then use the same model to predict the response in the population dataset and thus have predictions for all individuals in the population. From these predictions we can form small area statistics by using the predicted values for individuals in each small area.

In order to fit the model we here have to reinput the response variable and all of the predictors we wish to use in the estimation. We are also given the choice of whether to fit a model to the original response or to use a logged or Box-Cox transformation. To start the model running make these selections from the box below. Note that we are using MCMC estimation which will not only give us small area estimates but also Bayesian credible intervals. It is however a computationally intensive procedure and so this page will take some time to run.

Response variable:	eqIncome change
Specify distribution:	Normal change
Transformation:	Box-Cox change
Lambda:	Modelled change
Common ID variable:	district change
Common predictor variables	gender,eqsize,cash,self_empl,unempl_ben,age_ben,surv_ben,sick_ben,dis_ben,rent,fam_allow,house_allow,cap change
Do you want to calculate poverty related estimates, e.g. Head count ratio?:	Yes change
Threshold value:	Automatic change

The template gives equations for the model being fitted to the sample data and some estimates for the model fit as shown here. Note the MCMC algorithm treats lambda as a parameter to be estimated in the modelling.

Stat-JR:DEEP Upload Resources About Debug

Small Area Estimation

Finished

« 1 2 3 4 5 » Go to page

Select the data
Exploring the response
Exploring the predictors
Estimating the model with MCMC
Estimating the model with EMDI

beta_4	self_empl	0.4647642	0.0178588	8459.
beta_5	unempl_ben	0.3920562	0.05746135	44366.
beta_6	age_ben	0.5503508	0.01642028	9745.
beta_7	surv_ben	0.584027	0.114855	50782.
beta_8	sick_ben	0.4924603	0.1503358	56522.
beta_9	dis_ben	0.6082253	0.03702702	51163.
beta_10	rent	0.3773787	0.02605691	285.
beta_11	fam_allow	0.0305386	0.04151562	54229.
beta_12	house_allow	0.8691035	0.3172981	55855.
beta_13	cap_inv	0.4038095	0.03507246	12629.
beta_14	tax_adj	-0.2604875	0.06284644	49197.
sigma2	Level-1 variance	21707770.	855081.6	17257.
sigma_u	Level-2 variance	5880221.	1244259.	26391.
lamb	Box-Cox Lambda	0.5792535	0.07403507	47

about

Here we see that in the sample the predictors eqsize, cash, self_empl, unempl_ben, age_ben, surv_ben, sick_ben, dis_ben, rent, house_allow, cap_inv, and tax_adj have significant effects on the response variable.

about

Here we see that the model has been used to produce mean and SD estimates for each of the small areas. These are compared with the mean and SD of the data in the sample (where present).

Resources About Debug

Small Area Estimation

« 1 2 3 4 5 » Go to page

Although the estimates are interesting our primary interest is in the small area estimates that this model produces. The table below gives estimates for the mean and SD for each of the 94 small areas in the dataset. These estimates can be found in the columns headed population whilst for comparison in the columns headed sample are the means and SDS for eqIncome just using the sample data.

about

Code	Name	Sample mean	Sample sd	Population mean	Population sd
1	Eisenstadt-Umgebung	-	--	27160.54	11731.92
2	Eisenstadt (Stadt)	-	--	55112.39	31483.61
3	Güssing	-	--	17195.27	6272.34
4	Jennersdorf	-	--	13657.69	5370.03
5	Mattersburg	-	--	21375.96	8467.11
6	Neusiedl am See	19692.53	5462.45	19176.11	6354.53
7	Oberpullendorf	-	--	17739.70	6752.91
8	Oberwart	13833.36	6830.92	13715.97	5109.59
9	Rust (Stadt)	-	--	15788.02	5581.03
10	Amstetten	15201.15	6458.63	14409.22	5431.82
11	Baden	22921.69	6293.34	22436.13	7512.82
12	Bruck an der Leitha	23753.31	6160.26	24150.02	8215.10
13	Gänserndorf	20279.97	5833.86	20574.64	7022.35
14	Gmünd	-	--	14525.73	5621.84

We can also plot the means along with credible intervals for each district graphically as shown below. Here we see the wide variety of mean estimates.

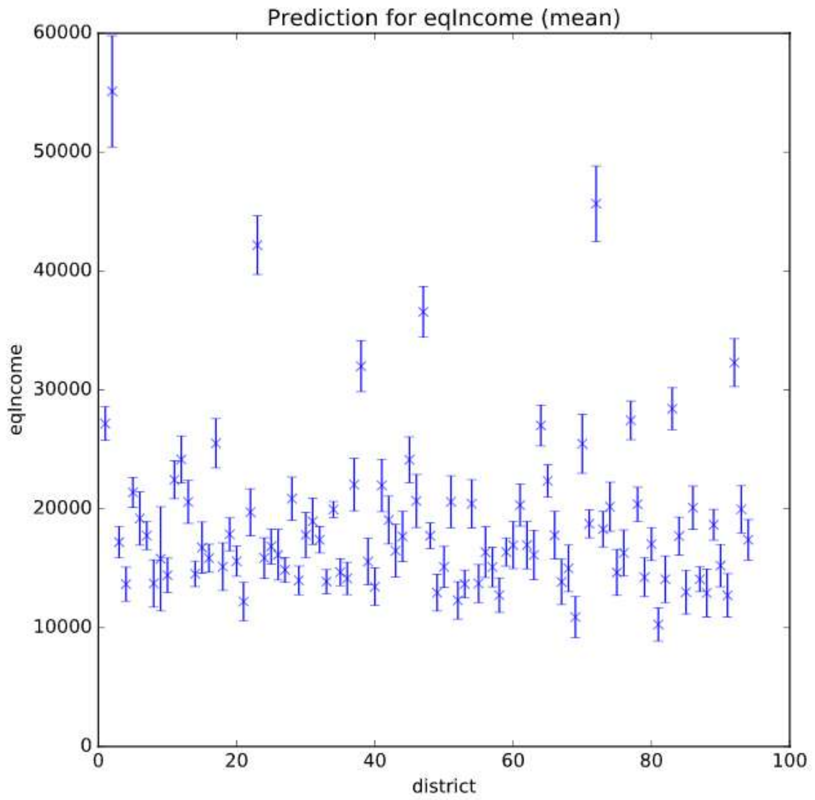
Resources About Debug

Small Area Estimation

« 1 2 3 4 5 » Go to page

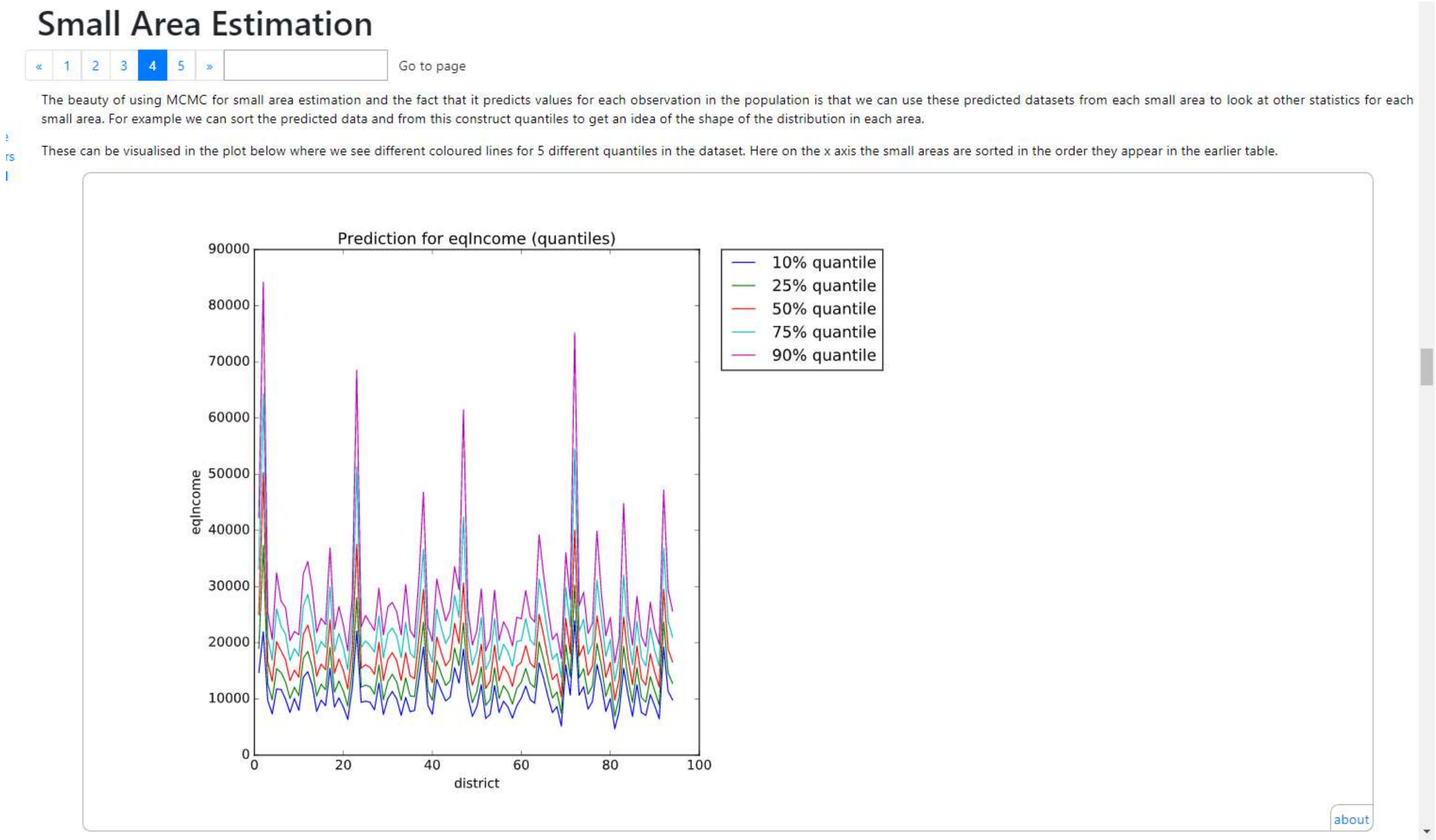
about

We can also visualise these data graphically and so in the graph below you will see the means plotted with 95% credible intervals to illustrate differences across the small areas. The small areas are listed in the order they appear in the table.



about

We can construct quantiles for each district from the predicted values for each individual and plot them as shown below.



These are also made available in tabular format so that the individual names of the districts can be established and also the precise estimates.

Small Area Estimation

« 1 2 3 4 5 » Go to page

about

To see these values in detail the table below gives the values for the same 7 quantiles in tabular form

Code	Name	Q10	Q25	Q50	Q75	Q90
1	Eisenstadt-Umgebung	14683.67	18984.76	24979.34	33073.70	42278.99
2	Eisenstadt (Stadt)	21938.74	37312.36	50276.98	64285.83	84163.51
3	Güssing	9745.10	12739.69	16604.29	21016.89	25394.93
4	Jennersdorf	7333.80	9844.37	13133.98	16889.33	20589.91
5	Mattersburg	11771.08	15387.15	20198.09	26043.00	32424.91
6	Neusiedl am See	11685.79	14705.87	18536.51	22903.12	27388.90
7	Oberpullendorf	9957.46	13030.06	16996.84	21561.79	26239.20
8	Oberwart	7586.47	10085.51	13251.47	16834.19	20416.89
9	Rust (Stadt)	10083.56	12059.86	15119.88	18962.46	22004.10
10	Amstetten	7963.34	10583.63	13883.92	17618.69	21432.41
11	Baden	13844.94	17209.24	21474.63	26521.20	32224.66
12	Bruck an der Leitha	14850.48	18473.87	23133.04	28598.97	34467.23
13	Gänserndorf	12464.32	15679.48	19775.65	24506.52	29513.60
14	Gmünd	7777.01	10524.03	14035.05	17966.27	21836.76
15	Hollabrunn	9784.07	12624.26	16202.00	20255.36	24363.19

The output includes information on several poverty/inequality type indicators including the Gini index shown here. The Gini has a range of 0 to 1.

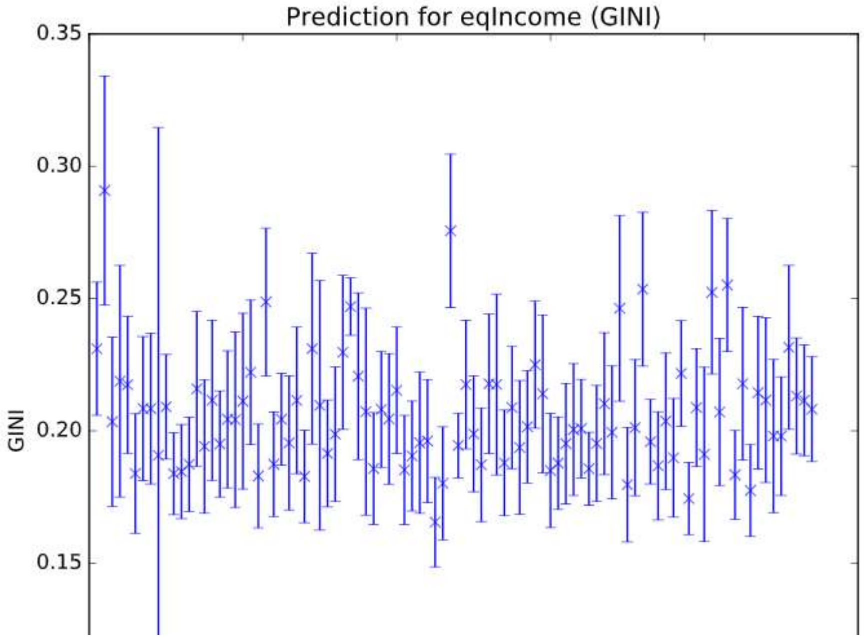
Small Area Estimation

« 1 2 3 4 5 » Go to page

The Gini index (or coefficient) is perhaps the most commonly used measure of inequality (particularly by economists). It attempts to measures dispersion in a frequency distribution with 0 meaning all individuals in an area have the same value of the response (often income) with a value of 1 then representing the extreme case of all the values for the response (all the income) in an area being concentrated on one individual with all other individuals having response 0. The Gini coefficient can therefore be used to measure relative inequality across a series of small areas.

The formula for the Gini index can be written $\widehat{Gini}_i = \left[\frac{2\sum_{j=1}^{n_i} j y_{ij}}{n_i \sum_{j=1}^{n_i} y_{ij}} - \frac{n_i+1}{n_i}\right]$

Below is a graph of the Gini index (with credible intervals) for the 94 small areas. about



After graphs of each indicator (Gini, Quintile share ratio, head count ratio and poverty gap index their values are summarised along with ranks for the small areas in the table shown below.

Small Area Estimation

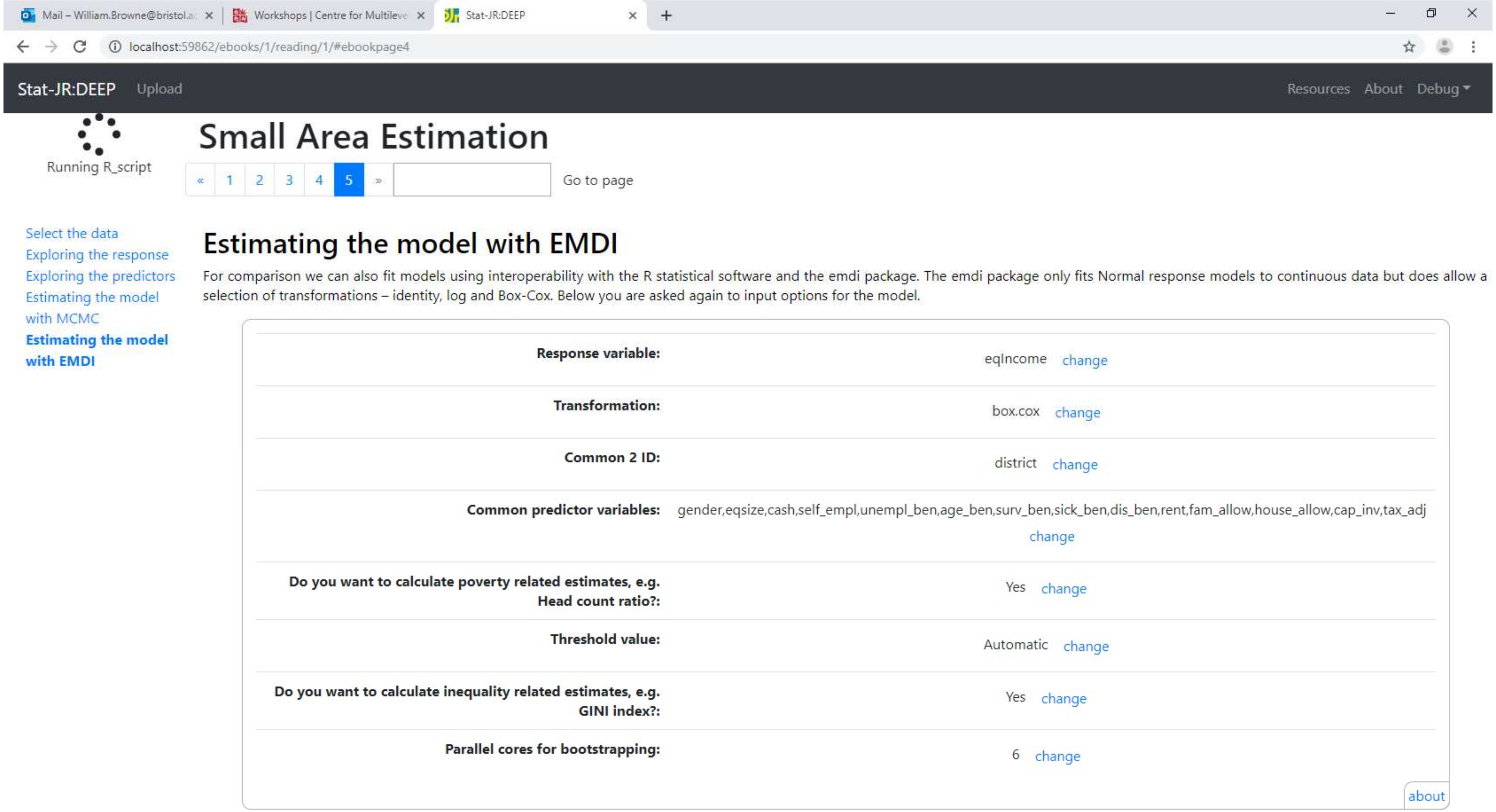
« 1 2 3 4 5 » Go to page

about

Finally we can summarise all of these indices in tabular form so that it is easier to see values for individual small areas. We do this in the table below.

Code	Area	Gini	Gini rank	QSR	QSR rank	HCR	HCR rank	PGI	PGI rank
1	Eisenstadt-Umgebung	0.23	84.00	3.25	82.00	0.03	14.00	0.01	15.00
2	Eisenstadt (Stadt)	0.29	94.00	4.70	93.00	0.03	13.00	0.01	14.00
3	Güssing	0.20	46.00	2.89	52.00	0.16	50.00	0.03	52.00
4	Jennersdorf	0.22	78.00	3.17	78.00	0.33	84.00	0.09	83.00
5	Mattersburg	0.22	73.00	3.07	72.00	0.08	29.00	0.02	31.00
6	Neusiedl am See	0.18	10.00	2.58	17.00	0.08	30.00	0.01	29.00
7	Oberpullendorf	0.21	56.00	2.96	62.00	0.14	45.00	0.03	46.00
8	Oberwart	0.21	57.00	3.00	65.00	0.32	80.00	0.08	80.00
9	Rust (Stadt)	0.19	24.00	inf	94.00	0.24	68.00	0.06	68.00
10	Amstetten	0.21	60.00	2.93	58.00	0.28	71.00	0.07	71.00
11	Baden	0.18	9.00	2.55	11.00	0.03	15.00	0.01	13.00
12	Bruck an der Leitha	0.18	11.00	2.56	13.00	0.02	12.00	0.00	12.00
13	Gänserndorf	0.19	18.00	2.61	20.00	0.06	23.00	0.01	23.00
14	Gmünd	0.22	72.00	3.11	74.00	0.28	73.00	0.07	73.00
15	Hollabrunn	0.19	28.00	2.74	33.00	0.16	51.00	0.03	50.00

The software also allows fitting the same models using the EMDI package in R with StatJR acting as an interface to R and the same eBook used.

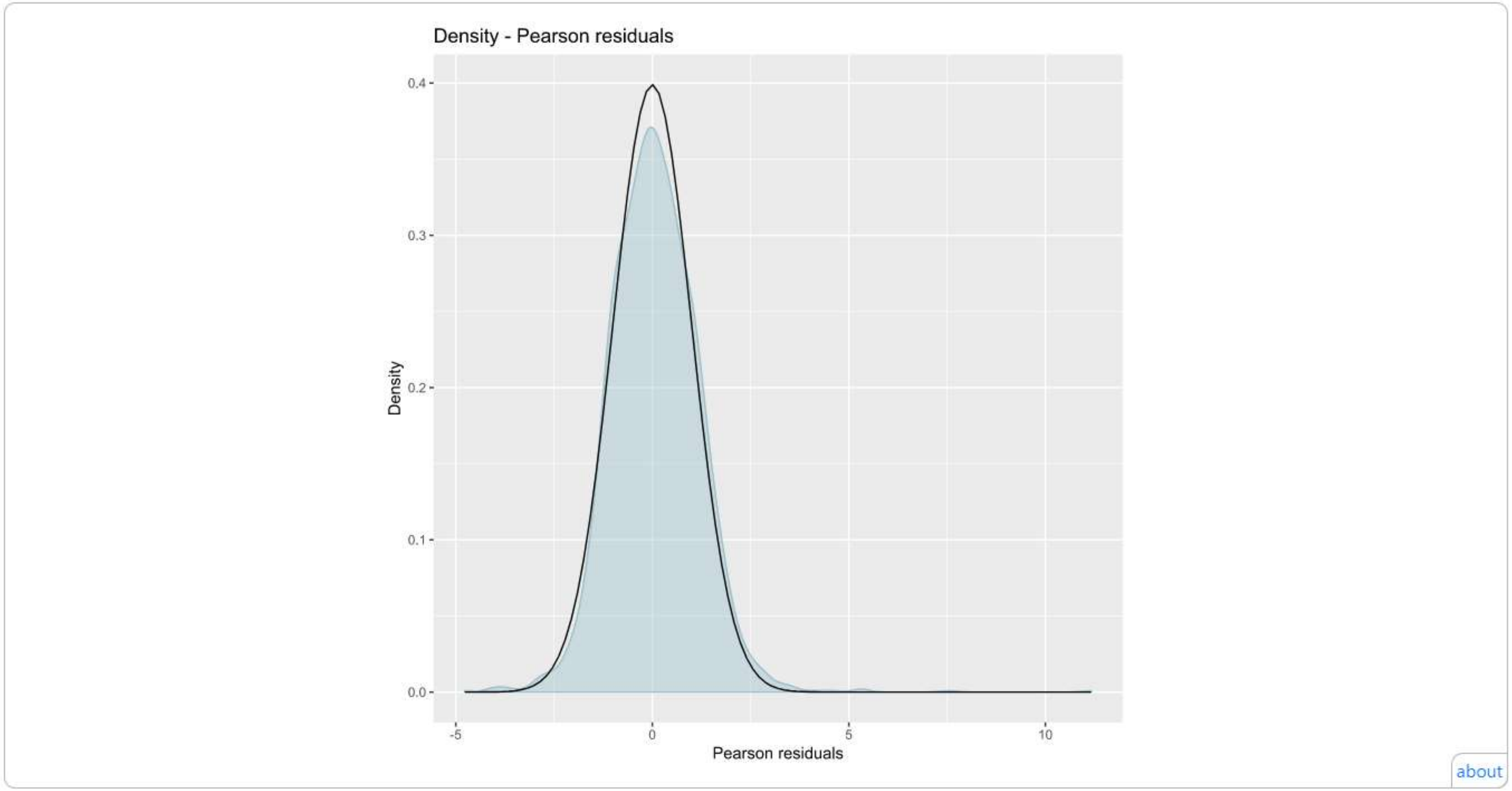


Many of the same outputs are available for EMDI along with some additional model fit checks as shown below.

Small Area Estimation

« 1 2 3 4 5 » Go to page

Emdi produces plots of the fit of the normal response model to the data. Below we can first see a plot of the individual level (Pearson) residuals from the sample plotted as a density plot in blue with superimposed a best fitting normal distribution. The amount the blue density appears outside the black normal curve represents how good/poor a fit the normal assumption is.



We can look at a similar plot for the level (random) effects at level 2 of the model fitted and again plot against a best fitting normal curve as shown below:

Extensions to current work – Binary responses

- Given the current Brexit crisis we might be interesting in where in the UK wants to leave the EU and where wants to stay.
- Small Area Estimation can be used along with data polling a random sample from the population.
- The only difference is that a (multilevel) logistic regression model would be used for the remain / leave response and possible predictors like age, gender, ethnicity and voting constituency would be included.
- Such an approach is often used with exit polls in parliamentary elections to predict the results in each constituency in the election and thus the overall result.

Practical

- In the practical that follows you will try out the SAE functionality in StatJR
- Here we are provided a stripped down version of StatJR on a memory stick with only the templates required for SAE.
- The practical has 3 examples – an educational tutorial example, the EU-SILC example in the slides and used this morning and a binary voting example
- This practical will become the LEMMA training materials practical for Small Area Estimation in StatJR
- We are not expecting you to finish it all this afternoon but you are free to take the stick with you and finish at home.