Developing a statistical analysis assistant using the Stat-JR software system

This guide was written by **William Browne***, **Chris Charlton*** and **Richard Parker*** with input from **Danius Michaelides****, **Harvey Goldstein***, **George Leckie***, **Kelvyn Jones*** and **Luc Moreau****

*Centre for Multilevel Modelling, University of Bristol, UK

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

September 2019

<u>Citing Stat-JR:</u>

Please cite Stat-JR as:

Charlton, C.M.J., Michaelides, D.T., Parker, R.M.A., Cameron, B., Szmaragd, C., Yang, H., Zhang, Z., Frazer, A.J., Goldstein, H., Jones, K., Leckie, G., Moreau, L. and Browne, W.J. (2018). *Stat-JR version 1.0.6.* Centre for Multilevel Modelling, University of Bristol & Electronics and Computer Science, University of Southampton, UK.

Developing a statistical analysis assistant using the Stat-JR software system (1.0.7)

ISBN: To be confirmed

Printed in the United Kingdom

Screenshots updated for 1.0.6 by Rhiannon Moore

Funding acknowledgement

We are grateful to the Economic and Social Research Council for funding the work upon which this guide has been developed (ESRC grant ES/K007246/1).

Contents

Chapter 1 - Introduction

The world around us has changed dramatically with the introduction of computers into more and more aspects of our everyday lives. In academia, both in teaching and research, computers have been in use for many years and gradually computer programs have taken on many roles that were traditionally done by hand. Our interest is in statistics and data analysis and here statistical software packages have been around for many decades. A software package is generally used to perform specific operations on a dataset to answer questions related to the data or more generally to answer questions for which the dataset offers evidence.

The traditional model here is that a specific function within a software package will, whether called by a command or selected via pull down menus, take user inputs and some data and produce a series of output objects. The user then has to interpret the objects produced to decide on what the data shows, although sometimes such interpretation is included to some degree in the objects produced.

It is possible to string together series of such operations in many statistics packages to produce a 'workflow' of operations that can be executed in sequence. It is also possible to create functions that are effectively such workflows and produce their objects via a series of operations rather than just one. We will consider combining both these approaches in this book where we look at how one might get a computer to perform a complete statistical analysis / study with limited user input.

We have several aims here that take us beyond the traditional approach. We would like the system that we create to be informative to the user and create objects with included contextual information to aid the user's understanding. Here we envisage that the system will not only create objects but will produce its own interpretation of the objects it produces or at least tell the user what to look for and explain what it believes the user might do next. Such information may also include greater details of how objects were created in order to increase learning of the statistical methods used.

The idea behind a statistical analysis assistant (SAA) is that by asking the user a series of questions about their problem and their dataset, the computer can then attempt an automated analysis of the data and produce an annotated report of its findings. In this book we will present several SAAs which will be used to consider statistical analyses of different levels of generality and different levels of complexity. We will aim to get to, by the end, a system that is capable of doing many different analyses but to get there we will start small and work up.

As we go we will assess how well our SAAs do in various scenarios and pinpoint how we can make improvements based on observing how the SAAs do on real data before moving on to the next level of detail. We will use our Stat-JR software system throughout this book and we will come across several interfaces to the Stat-JR package – TREE, DEEP and LEAF. We will not describe in detail how the software works as there are already four user guides to the various interfaces which can be used by readers who want to know more about features of the software.

We will begin describing the system with something simple – the automation of single operations and here the focus will be on how we can enhance what are essentially commonly used commands / options available in most statistical software packages to aid interpretation and give more contextual information. These single operations will then form the building blocks for the SAAs that we go on to describe in further chapters.

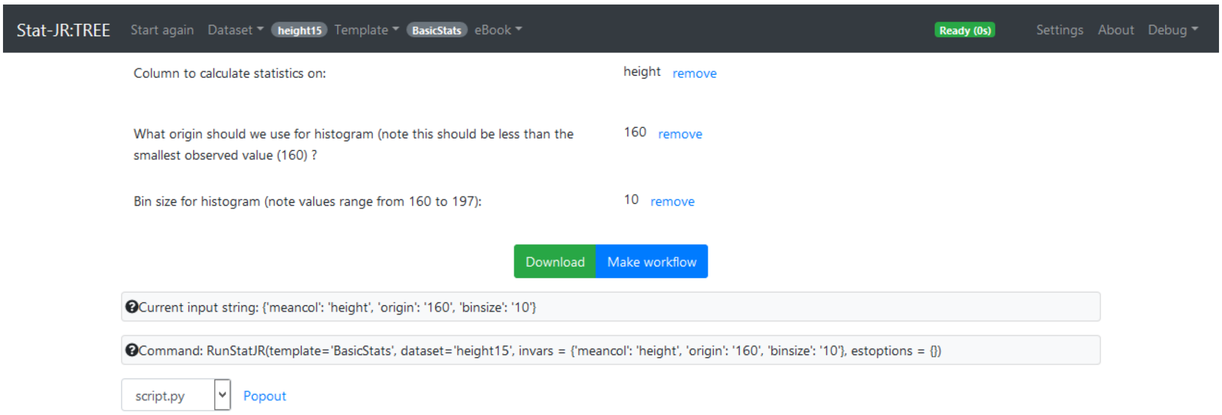
Chapter 2 – Automating single operations

2.1 Introducing the BasicStats template

In this chapter we will look at some basic statistical operations and statistical tests that one might consider using to initially get to know a dataset. Often when teaching statistics one of the first topics to cover is summary statistics and here we are interested in describing one or more variables in terms of measures of location (means, medians and modes) and measures of spread (standard deviation, range and intra-quartile range). Calculation of each of these measures will generally return a single number so we will consider here how we might add contextual information to explain how each of these measures is constructed. We will also consider one further operation which is a simple pictorial representation of the dataset – namely a histogram.

We will use the Stat-JR TREE software: so load up the software and choose from the Template list the template *BasicStats* and from the dataset list *height15*. If you are unsure on how to do this we suggest you read the Beginner's guide to Stat-JR first. This test dataset consists of 15 height measurements of adult males measured to the nearest cm.

The dataset only has one column, *height* and so we use this for the first input and then, in order to construct a histogram, we are asked for a starting point (origin) and a bin width (bin size). If we then press **Run** the screen will look as follows:



Column to calculate statistics on: height remove

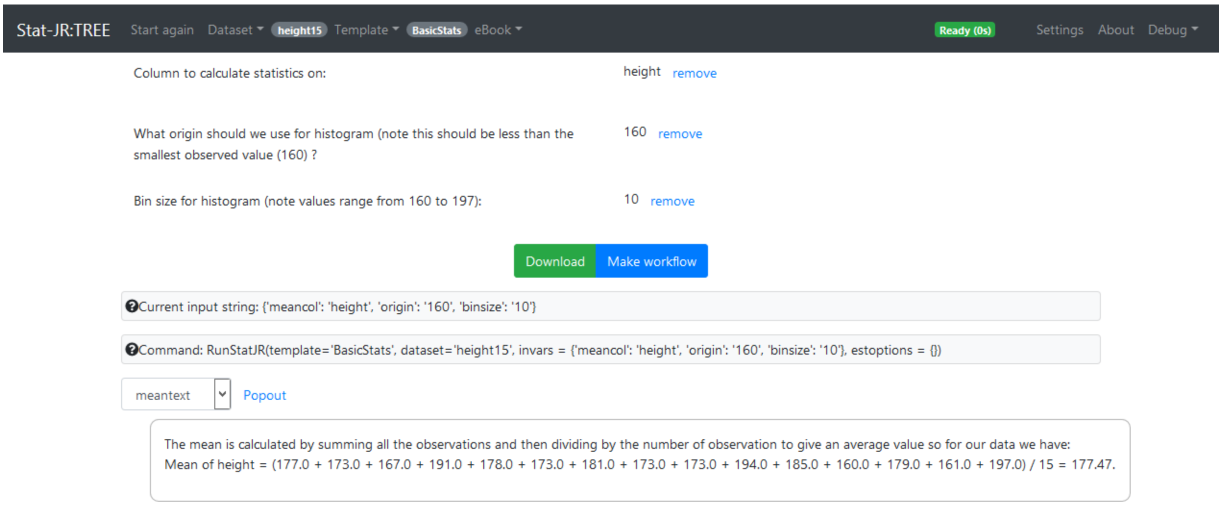
What origin should we use for histogram (note this should be less than the smallest observed value (160) ? 160 remove

Bin size for histogram (note values range from 160 to 197): 10 remove

Current input string: {'meancol': 'height', 'origin': '160', 'binsize': '10'}

Command: RunStatJR(template='BasicStats', dataset='height15', invars = {'meancol': 'height', 'origin': '160', 'binsize': '10'}, estoptions = {})

The pulldown object list then has several objects that we can access. If you have used Stat-JR before then you are probably used to seeing tables and graphs here but for this *BasicStats* template there are several html objects which consist of contextual text. So for example *meantext* shows how to calculate the mean and if we select it we see the following:



Column to calculate statistics on: height remove

What origin should we use for histogram (note this should be less than the smallest observed value (160) ? 160 remove

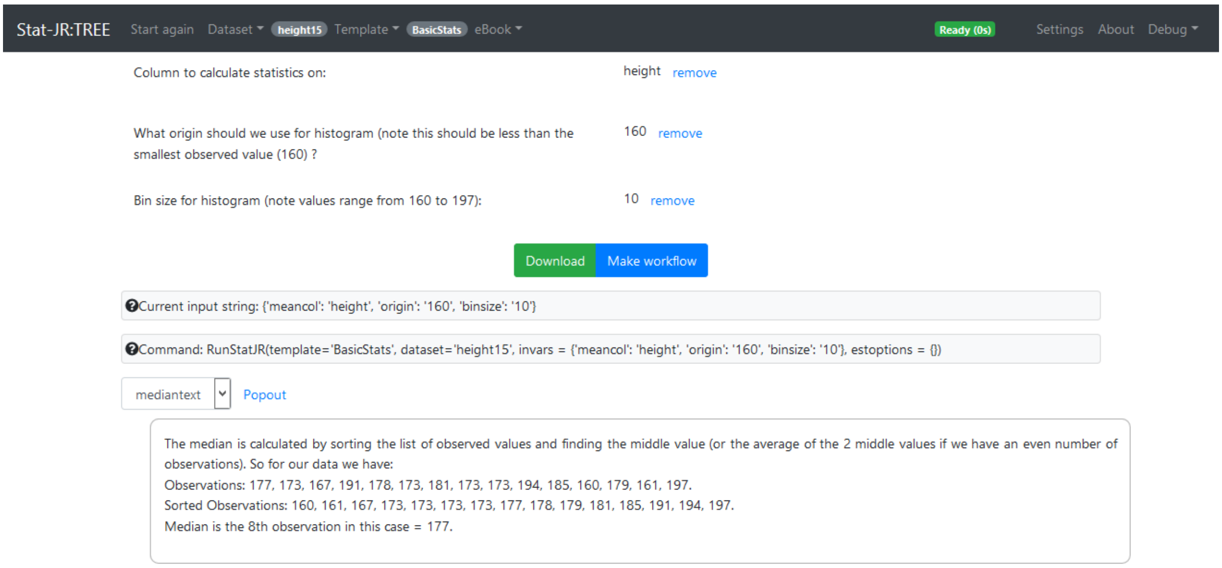
Bin size for histogram (note values range from 160 to 197): 10 remove

Current input string: {'meancol': 'height', 'origin': '160', 'binsize': '10'}

Command: RunStatJR(template='BasicStats', dataset='height15', invars = {'meancol': 'height', 'origin': '160', 'binsize': '10'}, estoptions = {})

The mean is calculated by summing all the observations and then dividing by the number of observation to give an average value so for our data we have:
Mean of height = (177.0 + 173.0 + 167.0 + 191.0 + 178.0 + 173.0 + 181.0 + 173.0 + 173.0 + 194.0 + 185.0 + 160.0 + 179.0 + 161.0 + 197.0) / 15 = 177.47.

Here the text not only shows the one number summary (177.47) but also how this was calculated. Similarly we see for the median, *mediantext* as follows:



Column to calculate statistics on: height remove

What origin should we use for histogram (note this should be less than the smallest observed value (160) ? 160 remove

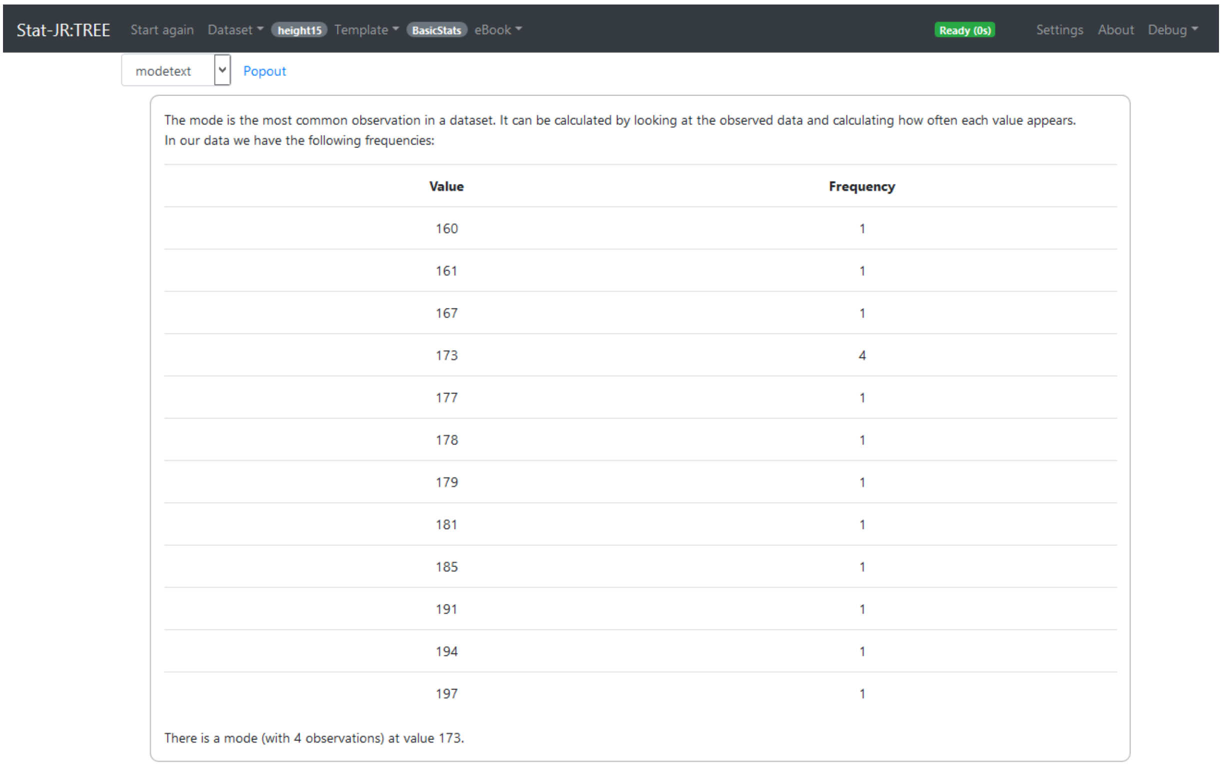
Bin size for histogram (note values range from 160 to 197): 10 remove

Current input string: {'meancol': 'height', 'origin': '160', 'binsize': '10'}

Command: RunStatJR(template='BasicStats', dataset='height15', invars = {'meancol': 'height', 'origin': '160', 'binsize': '10'}, estoptions = {})

The median is calculated by sorting the list of observed values and finding the middle value (or the average of the 2 middle values if we have an even number of observations). So for our data we have:
Observations: 177, 173, 167, 191, 178, 173, 181, 173, 173, 194, 185, 160, 179, 161, 197.
Sorted Observations: 160, 161, 167, 173, 173, 173, 173, 177, 178, 179, 181, 185, 191, 194, 197.
Median is the 8th observation in this case = 177.

Here the method is more involved and we can show more details to show the two operations of firstly sorting the list of variables and secondly picking the middle one from the sorted list. We can continue with the mode and *modetext* thus:



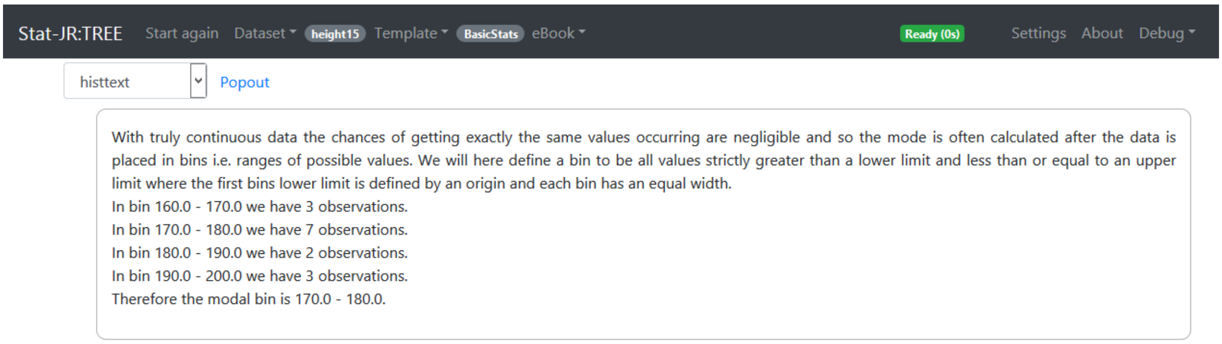
The mode is the most common observation in a dataset. It can be calculated by looking at the observed data and calculating how often each value appears.
In our data we have the following frequencies:

Value	Frequency
160	1
161	1
167	1
173	4
177	1
178	1
179	1
181	1
185	1
191	1
194	1
197	1

There is a mode (with 4 observations) at value 173.

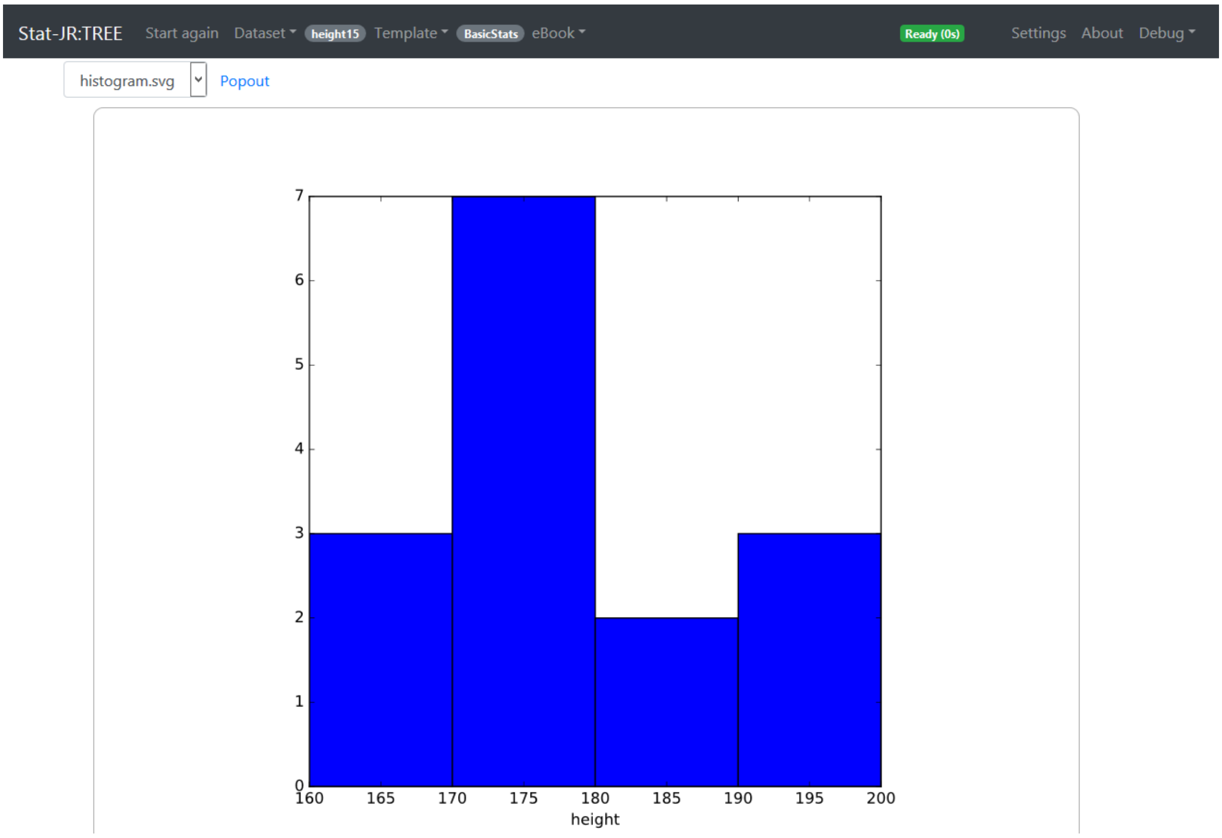
Here the operations are a basic tally for each unique value of how many instances we find (often called a frequency distribution) followed by picking the largest frequency. This begins to illustrate one possible issue with mode calculations in that all bar one value occurs only once! Clearly if the data had been continuous this would have been a worse issue as the chance of 2 truly continuous values being identical is very small.

Fortunately a solution is available with *histtext*, which is the text that explains how the numbers that are used to form the histogram are constructed. Here we tally counts within ranges rather than specific values and thus we get less counts but each count is larger and we can now construct the modal category rather than the modal value thus:



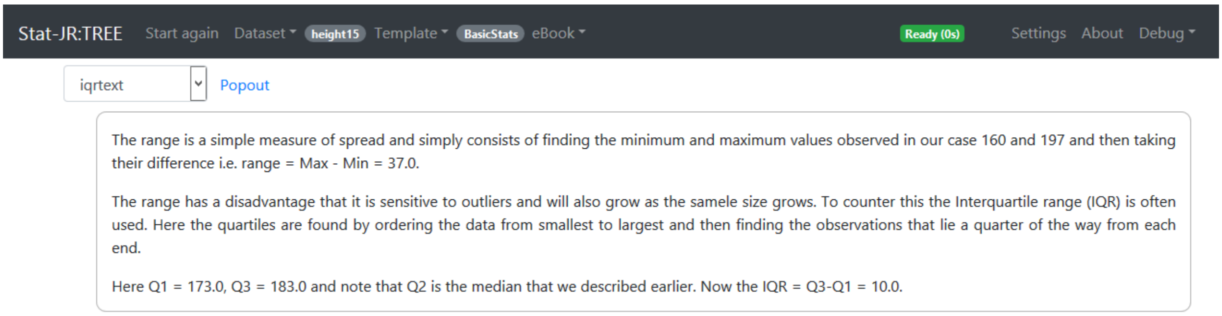
With truly continuous data the chances of getting exactly the same values occurring are negligible and so the mode is often calculated after the data is placed in bins i.e. ranges of possible values. We will here define a bin to be all values strictly greater than a lower limit and less than or equal to an upper limit where the first bins lower limit is defined by an origin and each bin has an equal width.
In bin 160.0 - 170.0 we have 3 observations.
In bin 170.0 - 180.0 we have 7 observations.
In bin 180.0 - 190.0 we have 2 observations.
In bin 190.0 - 200.0 we have 3 observations.
Therefore the modal bin is 170.0 - 180.0.

This directly translates to the histogram, *histogram.svg*



Here we have restricted the bins of the histogram to be of equal width (the *10* in the input questions) and we have put the tally numbers (frequencies) on the y axis. In practice it is the area of the bars that corresponds to the number of observations but often with equal bins the frequencies are placed on the y axis so that the histogram gives additional information on top of the shape of the distribution. Of course with equal widths the area is proportional to the heights so there is in practice no conflict here.

We can continue with the measures of spread and here we have two html output objects , *iqrtext* and *sdtext* shown below:

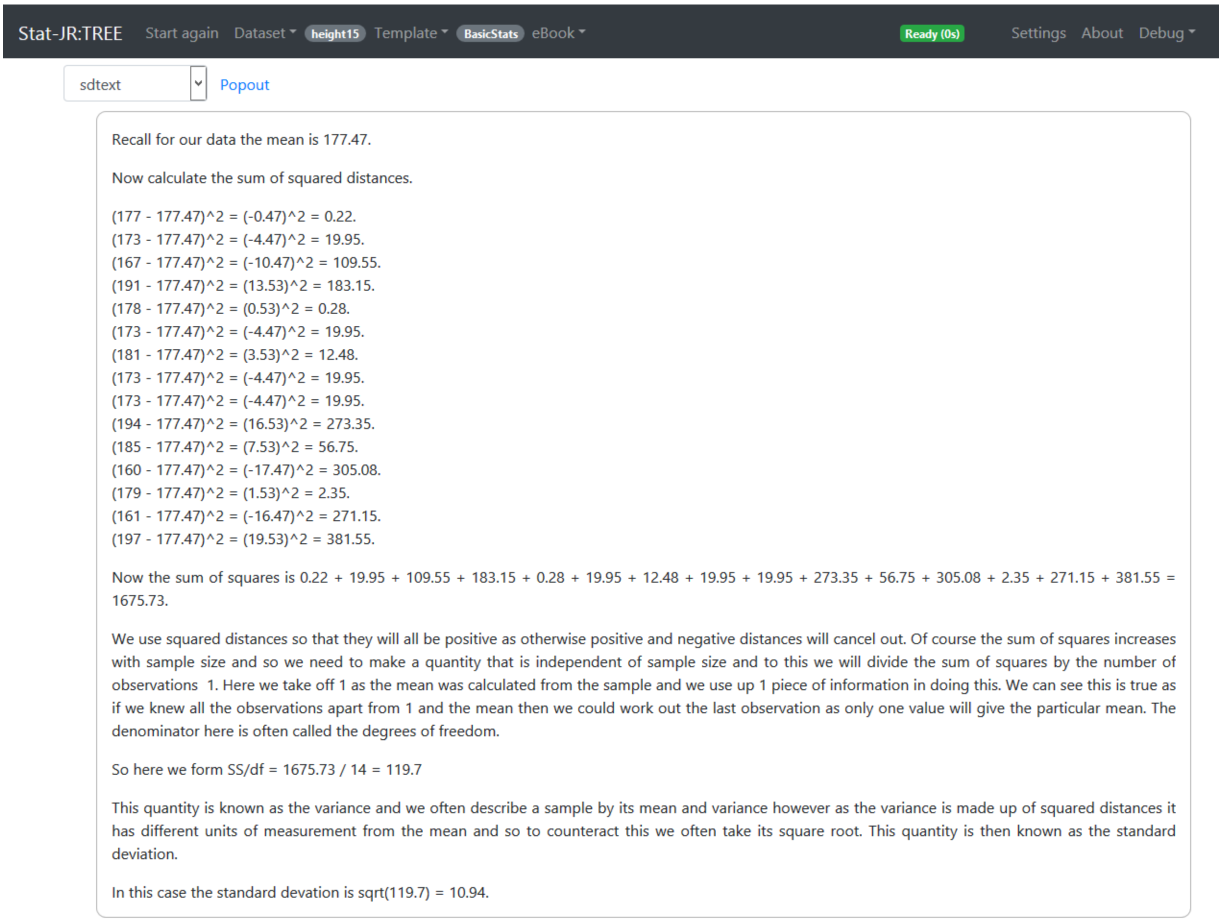


The range is a simple measure of spread and simply consists of finding the minimum and maximum values observed in our case 160 and 197 and then taking their difference i.e. range = Max - Min = 37.0.

The range has a disadvantage that it is sensitive to outliers and will also grow as the samele size grows. To counter this the Interquartile range (IQR) is often used. Here the quartiles are found by ordering the data from smallest to largest and then finding the observations that lie a quarter of the way from each end.

Here Q1 = 173.0, Q3 = 183.0 and note that Q2 is the median that we described earlier. Now the IQR = Q3-Q1 = 10.0.

and



Recall for our data the mean is 177.47.

Now calculate the sum of squared distances.

(177 - 177.47)^2 = (-0.47)^2 = 0.22.
(173 - 177.47)^2 = (-4.47)^2 = 19.95.
(167 - 177.47)^2 = (-10.47)^2 = 109.55.
(191 - 177.47)^2 = (13.53)^2 = 183.15.
(178 - 177.47)^2 = (0.53)^2 = 0.28.
(173 - 177.47)^2 = (-4.47)^2 = 19.95.
(181 - 177.47)^2 = (3.53)^2 = 12.48.
(173 - 177.47)^2 = (-4.47)^2 = 19.95.
(173 - 177.47)^2 = (-4.47)^2 = 19.95.
(194 - 177.47)^2 = (16.53)^2 = 273.35.
(185 - 177.47)^2 = (7.53)^2 = 56.75.
(160 - 177.47)^2 = (-17.47)^2 = 305.08.
(179 - 177.47)^2 = (1.53)^2 = 2.35.
(161 - 177.47)^2 = (-16.47)^2 = 271.15.
(197 - 177.47)^2 = (19.53)^2 = 381.55.

Now the sum of squares is 0.22 + 19.95 + 109.55 + 183.15 + 0.28 + 19.95 + 12.48 + 19.95 + 19.95 + 273.35 + 56.75 + 305.08 + 2.35 + 271.15 + 381.55 = 1675.73.

We use squared distances so that they will all be positive as otherwise positive and negative distances will cancel out. Of course the sum of squares increases with sample size and so we need to make a quantity that is independent of sample size and to this we will divide the sum of squares by the number of observations 1. Here we take off 1 as the mean was calculated from the sample and we use up 1 piece of information in doing this. We can see this is true as if we knew all the observations apart from 1 and the mean then we could work out the last observation as only one value will give the particular mean. The denominator here is often called the degrees of freedom.

So here we form SS/df = 1675.73 / 14 = 119.7

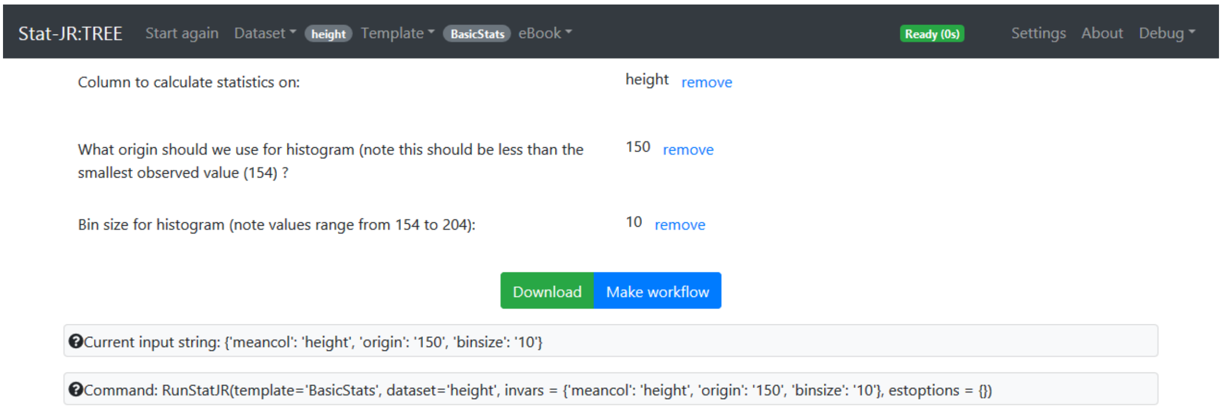
This quantity is known as the variance and we often describe a sample by its mean and variance however as the variance is made up of squared distances it has different units of measurement from the mean and so to counteract this we often take its square root. This quantity is then known as the standard deviation.

In this case the standard devation is sqrt(119.7) = 10.94.

So here we see for the three measures of spread quite detailed explanations of how the quantities are calculated.

2.2 What about bigger datasets?

Clearly in practice with real datasets, a dataset of only 15 observations would be unusual and so how does our *BasicStats* template cope with bigger datasets. In fact in this template the answer is that how it copes varies with the particular quantity of interest. So if we choose a slightly bigger dataset called *height* (Rasbash, Steele, Browne, & Goldstein, 2012) which contains 100 heights (as opposed to 15) then we can choose the following:



Column to calculate statistics on: height remove

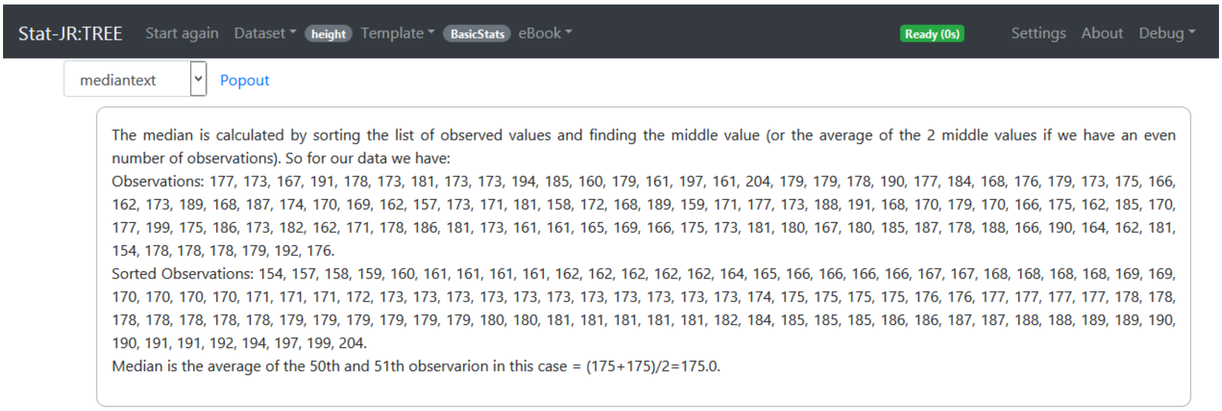
What origin should we use for histogram (note this should be less than the smallest observed value (154) ? 150 remove

Bin size for histogram (note values range from 154 to 204): 10 remove

Current input string: {'meancol': 'height', 'origin': '150', 'binsize': '10'}

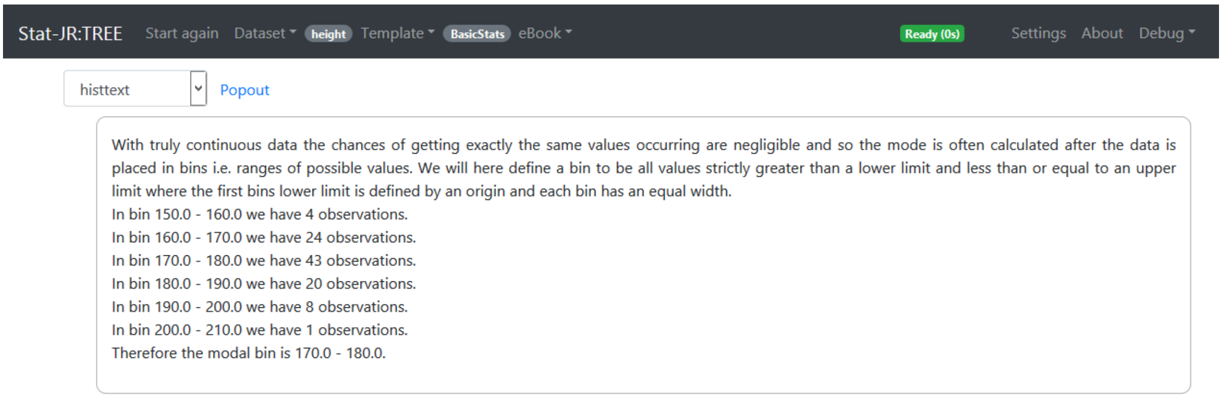
Command: RunStatJR(template='BasicStats', dataset='height', invars = {'meancol': 'height', 'origin': '150', 'binsize': '10'}, estoptions = {})

Here we have decreased the first origin to 150 as the smallest height in this bigger dataset is below 160. Now if we look at say *mediantext* we see a calculation that doesn't scale nicely and the two lists are now very long:



The median is calculated by sorting the list of observed values and finding the middle value (or the average of the 2 middle values if we have an even number of observations). So for our data we have:
Observations: 177, 173, 167, 191, 178, 173, 181, 173, 173, 194, 185, 160, 179, 161, 197, 161, 204, 179, 179, 178, 190, 177, 184, 168, 176, 179, 173, 175, 166, 162, 173, 189, 168, 187, 174, 170, 169, 162, 157, 173, 171, 181, 158, 172, 168, 189, 159, 171, 177, 173, 188, 191, 168, 170, 179, 170, 166, 175, 162, 185, 170, 177, 199, 175, 186, 173, 182, 162, 171, 178, 186, 181, 173, 161, 161, 165, 169, 166, 175, 173, 181, 180, 167, 180, 185, 187, 178, 188, 166, 190, 164, 162, 181, 154, 178, 178, 178, 179, 192, 176.
Sorted Observations: 154, 157, 158, 159, 160, 161, 161, 161, 161, 162, 162, 162, 162, 162, 164, 165, 166, 166, 166, 166, 167, 167, 168, 168, 168, 168, 169, 169, 170, 170, 170, 170, 171, 171, 171, 172, 173, 173, 173, 173, 173, 173, 173, 173, 173, 173, 173, 174, 175, 175, 175, 175, 176, 176, 177, 177, 177, 177, 178, 178, 178, 178, 178, 178, 178, 179, 179, 179, 179, 179, 179, 180, 180, 181, 181, 181, 181, 181, 182, 184, 185, 185, 185, 186, 186, 187, 187, 188, 188, 189, 189, 190, 190, 191, 191, 192, 194, 197, 199, 204.
Median is the average of the 50th and 51th observarion in this case = (175+175)/2=175.0.

By contrast the *histtext* object scales nicely and is not affected by increasing the sample size to be summarised.



With truly continuous data the chances of getting exactly the same values occurring are negligible and so the mode is often calculated after the data is placed in bins i.e. ranges of possible values. We will here define a bin to be all values strictly greater than a lower limit and less than or equal to an upper limit where the first bins lower limit is defined by an origin and each bin has an equal width.
In bin 150.0 - 160.0 we have 4 observations.
In bin 160.0 - 170.0 we have 24 observations.
In bin 170.0 - 180.0 we have 43 observations.
In bin 180.0 - 190.0 we have 20 observations.
In bin 190.0 - 200.0 we have 8 observations.
In bin 200.0 - 210.0 we have 1 observations.
Therefore the modal bin is 170.0 - 180.0.

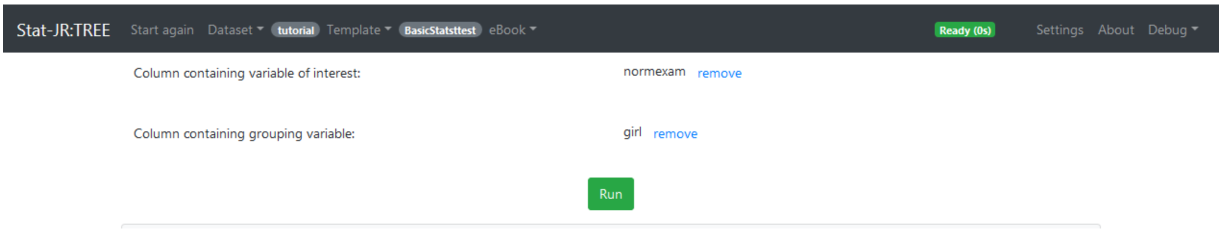
In practice outputs that require the whole list of observations (or the majority of them) to be part of the text do not scale so well so a solution might be to use '…' when the number of observations is bigger than some value for example something like the following for *mediantext*

Observations: 177,173,…,192,176
Sorted Observations: 154,157,…,175,175,…,199,204
Median is the average of the 50th and 51th observation in this case = (175+175)/2=175.0

With regard the basic statistics template we have really just added some calculation explanation to what would be done generally by a simple basic statistics function. We next look at another simple statistics building block which is the hypothesis test and illustrate true contextual text.

2.3 Hypothesis testing

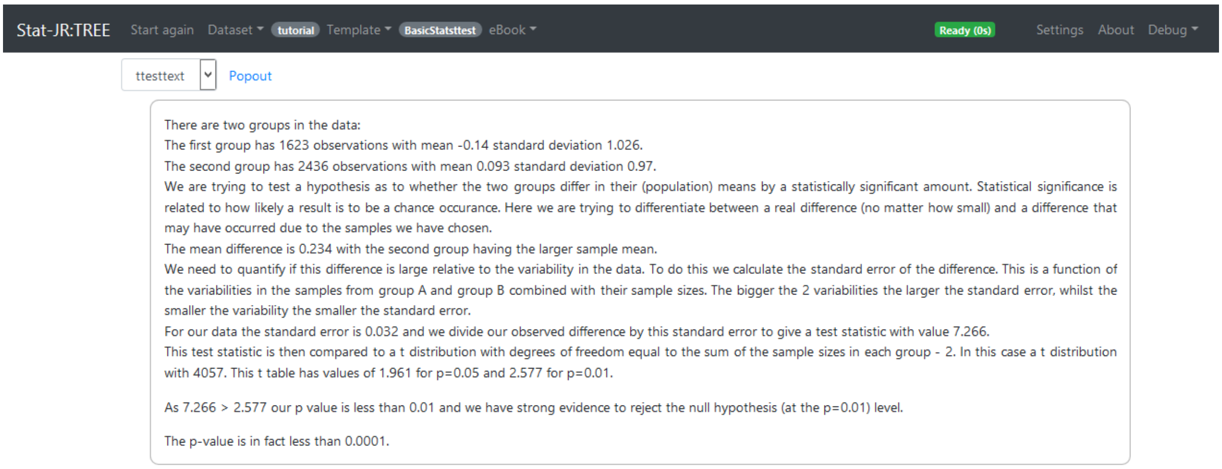
In statistics we are often interested in trying to test whether particular hypotheses hold by using our collected data to test the hypothesis in question. There are many different hypothesis tests which are used for different types of variable and different experimental designs. We will firstly consider the case of a continuous variable (assumed normally distributed) and one dichotomous (binary) predictor variable and test whether the distribution (or at least the mean) of the continuous variable is the same for each category of the binary variable. We will test this using a larger dataset, the *tutorial* (Goldstein, et al., 1993) dataset, in which our variable of interest is exam scores at age 16, *normexam* and our predictor is the gender of the student, *girl*. The template is called *BasicStatsttest* as the test to be performed is the (2-sample) t test. The inputs are as follows:



Column containing variable of interest: normexam

Column containing grouping variable: girl

Running the template gives the output *ttesttext*:



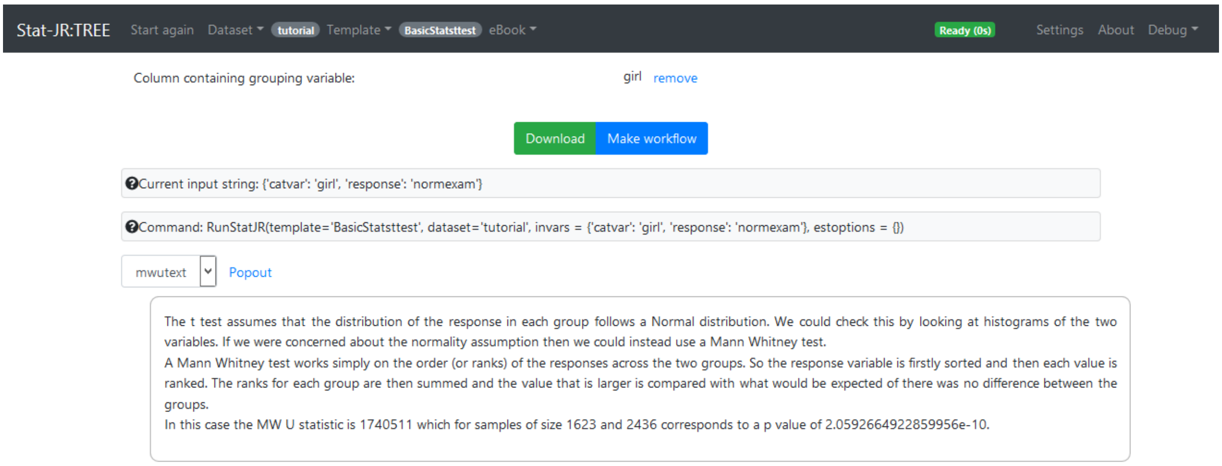
There are two groups in the data:
The first group has 1623 observations with mean -0.14 standard deviation 1.026.
The second group has 2436 observations with mean 0.093 standard deviation 0.97.
We are trying to test a hypothesis as to whether the two groups differ in their (population) means by a statistically significant amount. Statistical significance is related to how likely a result is to be a chance occurance. Here we are trying to differentiate between a real difference (no matter how small) and a difference that may have occurred due to the samples we have chosen.
The mean difference is 0.234 with the second group having the larger sample mean.
We need to quantify if this difference is large relative to the variability in the data. To do this we calculate the standard error of the difference. This is a function of the variabilities in the samples from group A and group B combined with their sample sizes. The bigger the 2 variabilities the larger the standard error, whilst the smaller the variability the smaller the standard error.
For our data the standard error is 0.032 and we divide our observed difference by this standard error to give a test statistic with value 7.266.
This test statistic is then compared to a t distribution with degrees of freedom equal to the sum of the sample sizes in each group - 2. In this case a t distribution with 4057. This t table has values of 1.961 for p=0.05 and 2.577 for p=0.01.

As 7.266 > 2.577 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.

Here we see that to perform a t test consists of several operations: calculating summary statistics (means and sds); working out other statistics from these summary statistics (mean difference and standard error) and from these the test statistic is formed. The test statistic is then compared with critical values (based on the number of observations) and from this comparison the null hypothesis is either rejected or not. Finally the p value of the test is given. Unlike many software packages we give different interpretations to this p value and effectively have contextual text for various scenarios for p>0.05, 0.05 >p> 0.01 and p<0.01 where we fail to reject, reject and have strong evidence to reject the null hypothesis respectively. The template is therefore written in such a way that different contextual text appears depending on the p value that results from the dataset and variables provided.

The t test assumes a normal distribution for the variable of interest but this template does not test whether this is appropriate. Instead it simply also gives the non-parametric alternative, the Mann Whitney test and this is shown in the output *mwutext* thus:



Column containing grouping variable: girl

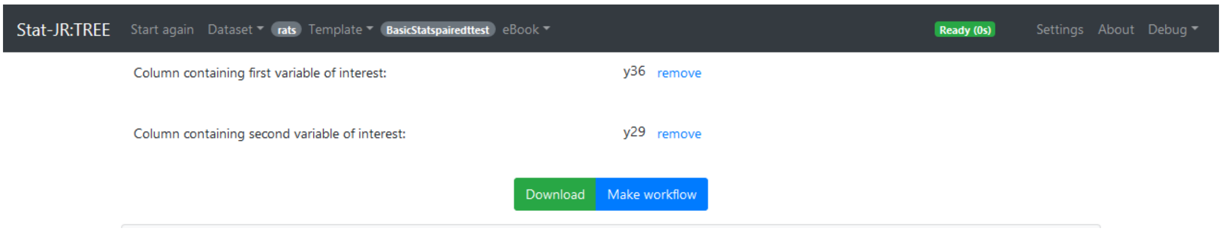
❓Current input string: {'catvar': 'girl', 'response': 'normexam'}

❓Command: RunStatJR(template='BasicStatsttest', dataset='tutorial', invars = {'catvar': 'girl', 'response': 'normexam'}, estoptions = {})

The t test assumes that the distribution of the response in each group follows a Normal distribution. We could check this by looking at histograms of the two variables. If we were concerned about the normality assumption then we could instead use a Mann Whitney test.
A Mann Whitney test works simply on the order (or ranks) of the responses across the two groups. So the response variable is firstly sorted and then each value is ranked. The ranks for each group are then summed and the value that is larger is compared with what would be expected of there was no difference between the groups.
In this case the MW U statistic is 1740511 which for samples of size 1623 and 2436 corresponds to a p value of 2.0592664922859956e-10.

Here we describe the method rather than showing the details and we get a rather small P value and so here we see that for this large dataset, both the parametric and non-parametric tests result in a significant difference between the 2 groups.

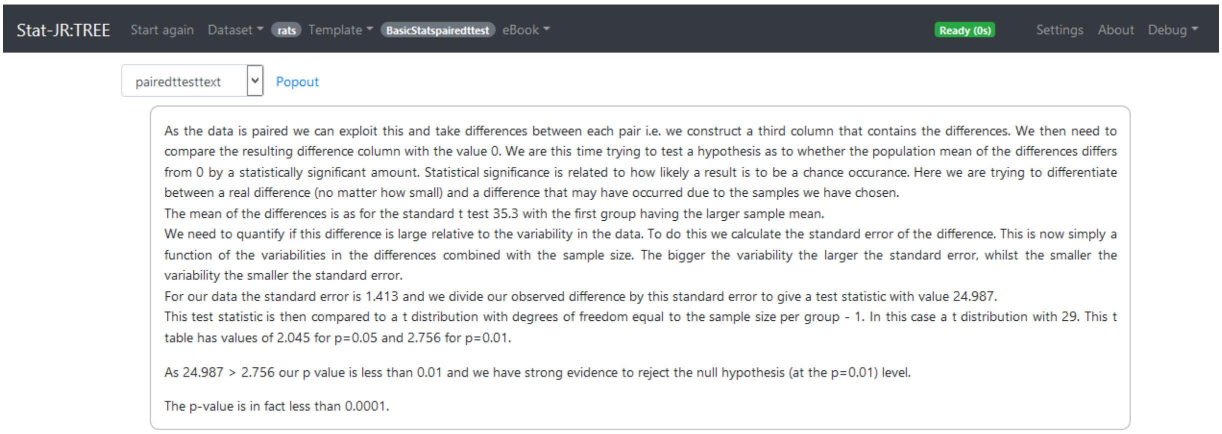
There is another form of t test which is used when the data have a particular data structure (often by design) where for the two groups (or variables) to be compared there is a pairing structure. An example would be an experiment where a variable is measured before and after an intervention, and so for each individual there are two measures and the hypothesis is that there is a difference in the means of the variable before and after the intervention. The paired t test respects the data structure and as a result can be more efficient than a 2-sample t test when data is paired as the between individual variability can be removed and focus can instead be on the differences between the two groups within individuals. We can demonstrate this here using the template *BasicStatspairedttest* with a rather artificial example looking at the weights of the rats in the *rats* (Gelfand, Hills, Racine-Poon, & Smith, 1990) dataset at 29 (*y29*) and 36 (*y36*) days old respectively. One should note that for the paired t test that the data is in 2 columns (rather than 1 column plus an indicator column) and the data for a pair occur on the same row. The inputs look as follows:



Column containing first variable of interest: y36

Column containing second variable of interest: y29

Then the details on the method and results can be seen below:



As the data is paired we can exploit this and take differences between each pair i.e. we construct a third column that contains the differences. We then need to compare the resulting difference column with the value 0. We are this time trying to test a hypothesis as to whether the population mean of the differences differs from 0 by a statistically significant amount. Statistical significance is related to how likely a result is to be a chance occurance. Here we are trying to differentiate between a real difference (no matter how small) and a difference that may have occurred due to the samples we have chosen.
The mean of the differences is as for the standard t test 35.3 with the first group having the larger sample mean.
We need to quantify if this difference is large relative to the variability in the data. To do this we calculate the standard error of the difference. This is now simply a function of the variabilities in the differences combined with the sample size. The bigger the variability the larger the standard error, whilst the smaller the variability the smaller the standard error.
For our data the standard error is 1.413 and we divide our observed difference by this standard error to give a test statistic with value 24.987.
This test statistic is then compared to a t distribution with degrees of freedom equal to the sample size per group - 1. In this case a t distribution with 29. This t table has values of 2.045 for p=0.05 and 2.756 for p=0.01.

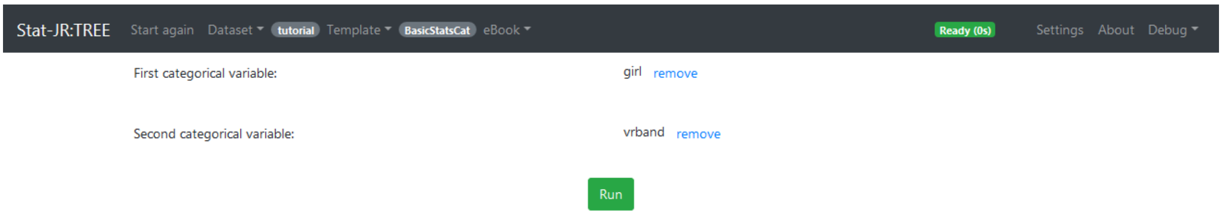
As 24.987 > 2.756 our p value is less than 0.01 and we have strong evidence to reject the null hypothesis (at the p=0.01) level.

The p-value is in fact less than 0.0001.

Here we see very strong evidence for a difference in means. This template will also show the results of a standard 2-sample t test where in this case the p value is still very small. Again in this template we are using contextual text based on the p value that comes from the specific dataset.

Moving along in our possible hypothesis tests lets consider next the case where we have two categorical variables. Here we often want to test whether they are related and this is done via a chi-squared test. In Stat-JR this can be done using the *BasicStatsCat* template So for example we could consider the tutorial dataset again and see if there is any relation between gender (*girl*) and the verbal reasoning test banding (*vrband*) each child fits in i.e. are proportionally more girls than boys in higher bands.

To do this we first need to tell Stat-JR the columns that containing the two variables:



First categorical variable: girl

Second categorical variable: vrband

We then look at the distribution of the two variables in isolation (under *table*):

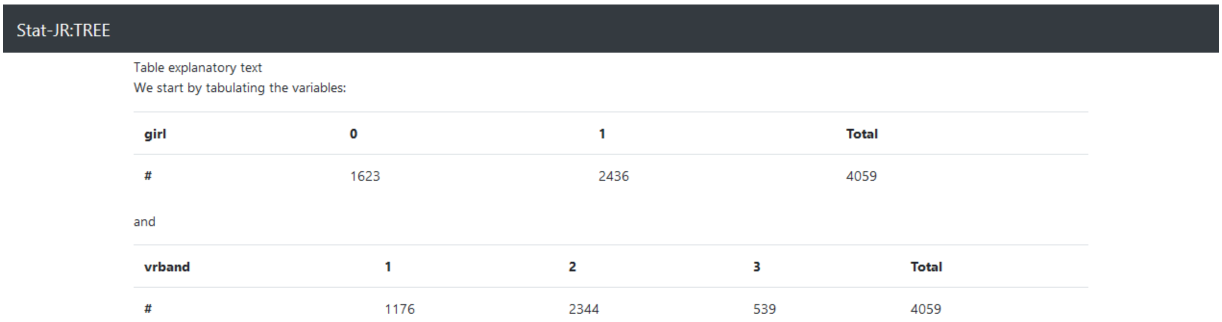


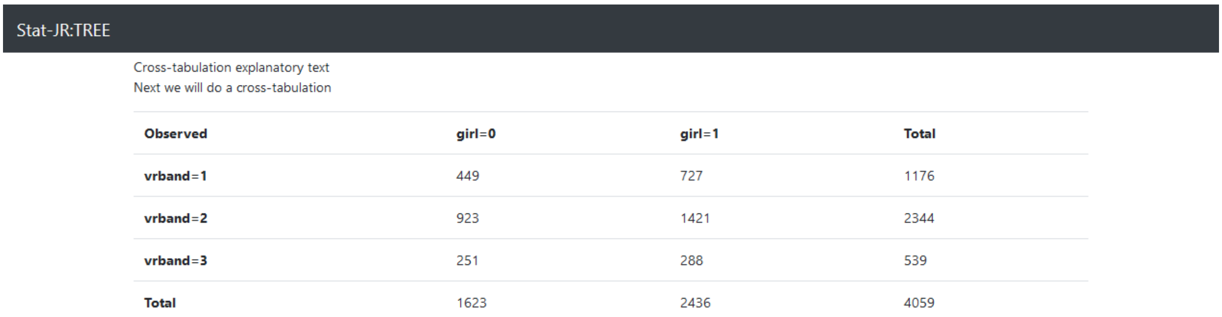
Table explanatory text
We start by tabulating the variables:

girl	0	1	Total
#	1623	2436	4059

and

vrband	1	2	3	Total
#	1176	2344	539	4059

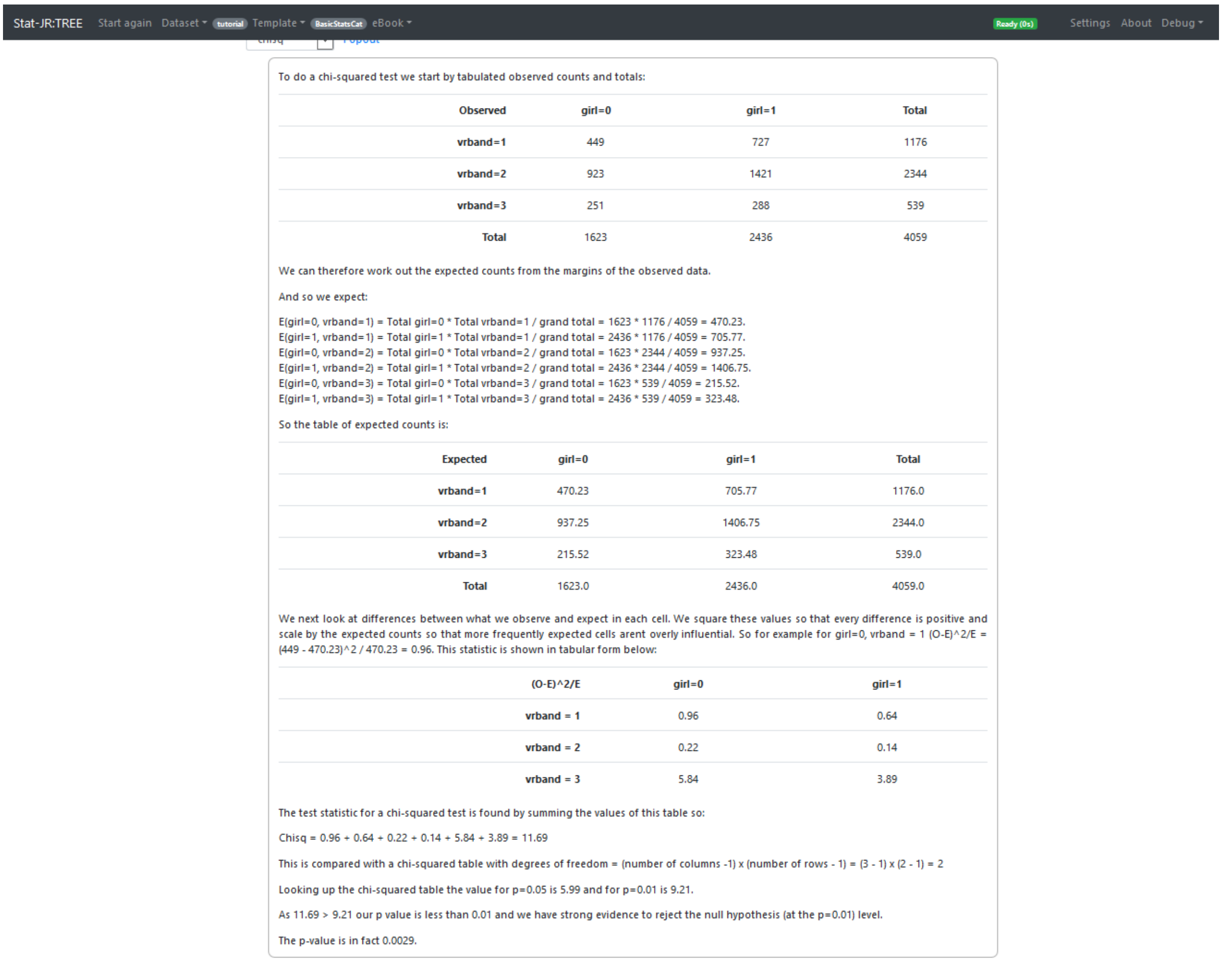
Here we see that there are more girls than boys and the majority of children are in band 2 with only just over 500 in band 3. We next perform a cross-tabulation (under *crosstable*) to look at how the two observed variables are related:



Cross-tabulation explanatory text
Next we will do a cross-tabulation

Observed	girl=0	girl=1	Total
vrband=1	449	727	1176
vrband=2	923	1421	2344
vrband=3	251	288	539
Total	1623	2436	4059

Here we are tabulating the counts of each combination so for example we have observed 449 rows of the data where girl is 0 i.e. the observation is a boy and VR band is 1 etc. The chi-squared test then looks to see if these observed counts are what one might expect to observe if there was no relationship between the two variables. To do this it calculates how many observations (on average) we would expect to see in each cell of the table and then compares these expected counts to the observed counts. How this is done is shown in the output (under *chisq*) below:



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

Observed	girl=0	girl=1	Total
vrband=1	449	727	1176
vrband=2	923	1421	2344
vrband=3	251	288	539
Total	1623	2436	4059

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

And so we expect:

E(girl=0, vrband=1) = Total girl=0 * Total vrband=1 / grand total = 1623 * 1176 / 4059 = 470.23.
E(girl=1, vrband=1) = Total girl=1 * Total vrband=1 / grand total = 2436 * 1176 / 4059 = 705.77.
E(girl=0, vrband=2) = Total girl=0 * Total vrband=2 / grand total = 1623 * 2344 / 4059 = 937.25.
E(girl=1, vrband=2) = Total girl=1 * Total vrband=2 / grand total = 2436 * 2344 / 4059 = 1406.75.
E(girl=0, vrband=3) = Total girl=0 * Total vrband=3 / grand total = 1623 * 539 / 4059 = 215.52.
E(girl=1, vrband=3) = Total girl=1 * Total vrband=3 / grand total = 2436 * 539 / 4059 = 323.48.

So the table of expected counts is:

Expected	girl=0	girl=1	Total
vrband=1	470.23	705.77	1176.0
vrband=2	937.25	1406.75	2344.0
vrband=3	215.52	323.48	539.0
Total	1623.0	2436.0	4059.0

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 girl=0, vrband = 1 (O-E)^2/E = (449 - 470.23)^2 / 470.23 = 0.96. This statistic is shown in tabular form below:

(O-E)^2/E	girl=0	girl=1
vrband = 1	0.96	0.64
vrband = 2	0.22	0.14
vrband = 3	5.84	3.89

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

Chisq = 0.96 + 0.64 + 0.22 + 0.14 + 5.84 + 3.89 = 11.69

This is compared with a chi-squared table with degrees of freedom = (number of columns -1) x (number of rows - 1) = (3 - 1) x (2 - 1) = 2

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

As 11.69 > 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.

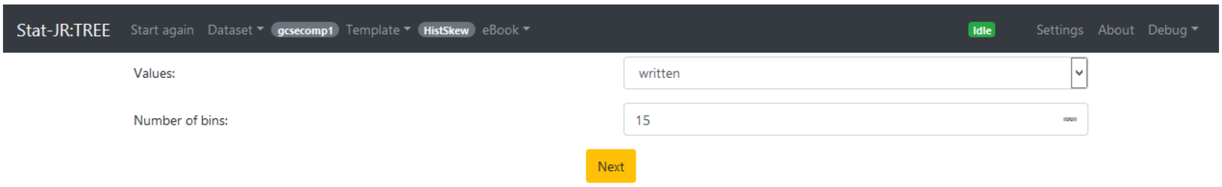
The p-value is in fact 0.0029.

Here we see that we have more girls than expected in VR bands 1 and 2 and less in VR band 3 and that with this size of dataset this is statistically significantly different to what one would expect by chance. It should be noted that the Chi-squared test treats the VR band variable as a nominal categorical variable i.e. it doesn’t use the fact that VR band 1 is closer to VR band 2 than band 3 in the test. There are other tests that do account for ordered categorical variables but we are not considering them here.

2.4 Interpreting figures

In practice it is fairly easy for a computer to describe the steps involved in single operations and in performing hypothesis tests as they involve algorithms that break down into a series of steps that calculate and compare numerical values. It can also interpret a p value and explain whether the hypothesis is rejected or not but it will struggle to explain what this means in the context of a real world problem i.e. it can spot that more girls than expected are in VR band category 3 and that this is significant but can’t say why or what to do about it! We will now move onto another challenging task for the computer and that is interpreting figures. Basically if a human looks at a graph it can describe its shape by simply observing it whereas although a computer can plot a graph internally it is stored as a set of numbers and so the task of describing shape is more complex for it. To attempt to describe the shape the computer needs to construct a statistic that correlates with different shapes and we will illustrate this with plotting histograms and establishing whether they are symmetric or skew.

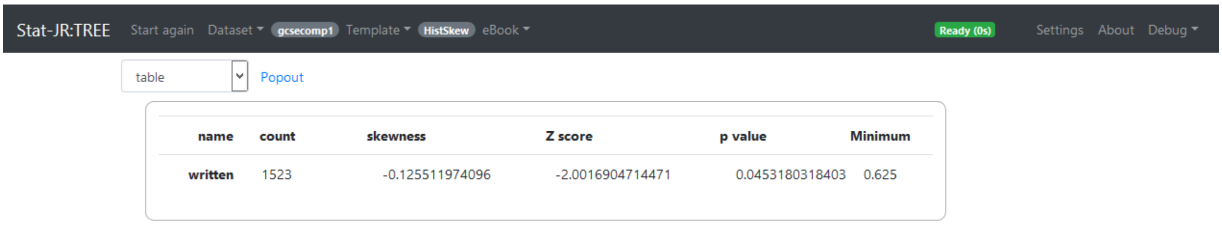
Here we will use the template *HistSkew*. We begin by selecting a variable (*written* – which here is the mark on a written test for a set of school children) to plot and a number of bins for the histogram as shown below:



Values: written

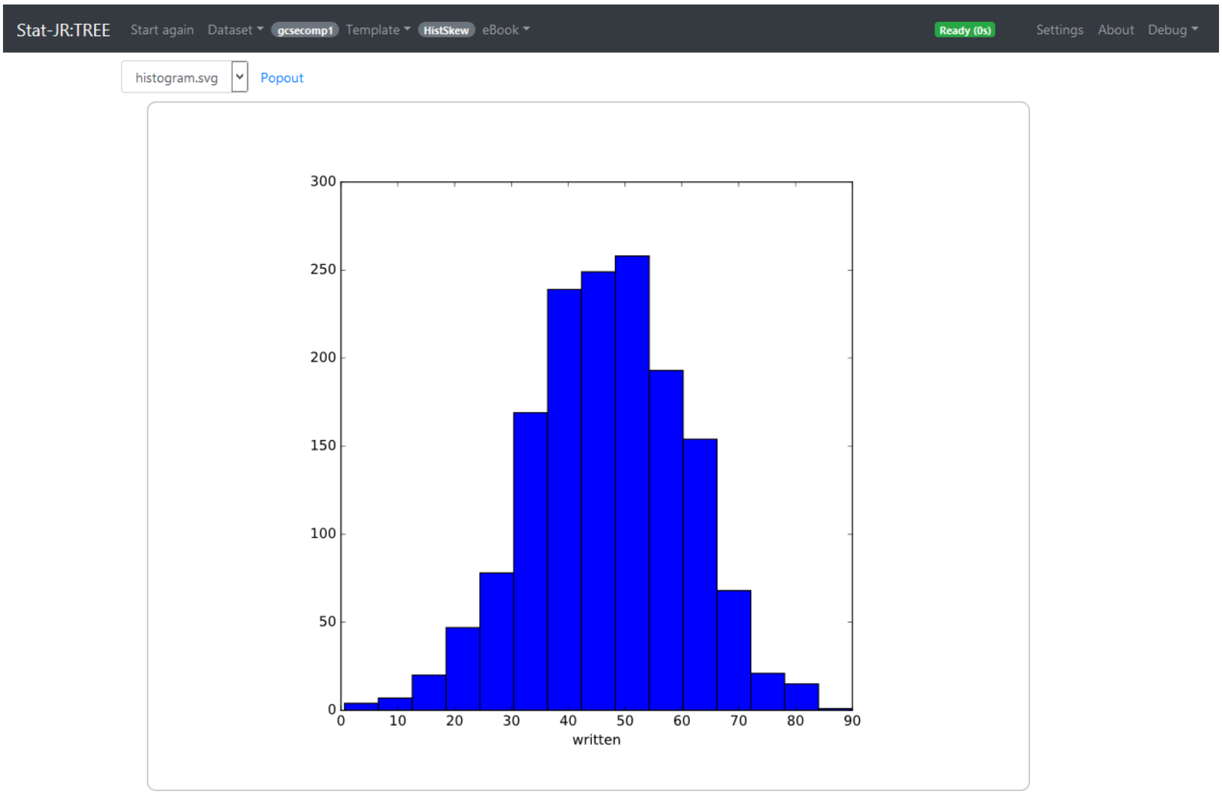
Number of bins: 15

We are fortunate that there is a statistic for a distribution known as skewness and we can also test whether the value of skewness obtained is significantly different from 0 (which represents symmetry). Here below we see that the skewness value (-0.1255) is slightly negative but is significantly different from 0 (p = 0.045) which may be in part due to the large dataset size of 1523 observations that we are using.

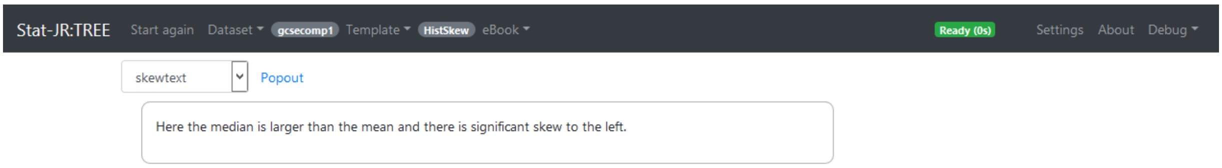


name	count	skewness	Z score	p value	Minimum
written	1523	-0.125511974096	-2.0016904714471	0.0453180318403	0.625

We can then plot the histogram as shown below:

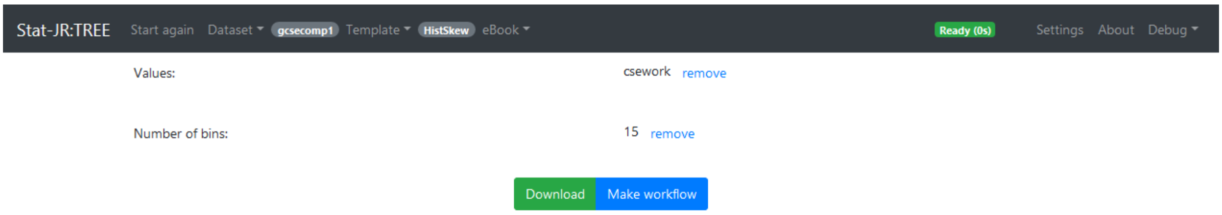


Here in fact to the eye the histogram looks fairly symmetrical with perhaps a slight skew to the left. This is backed up by the slight negative value and so the template sends back the following text (*skewtext*):



Here the median is larger than the mean and there is significant skew to the left.

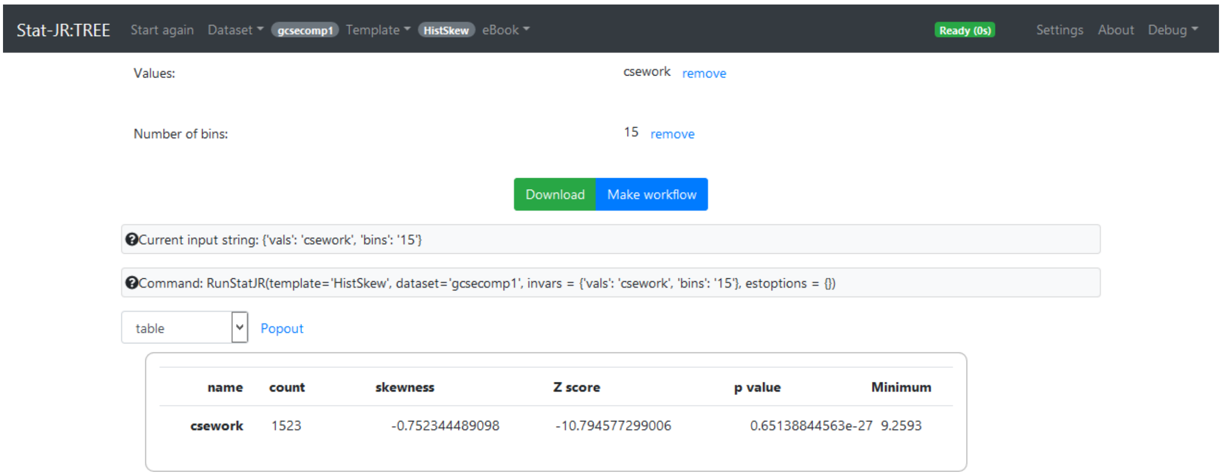
If we also repeat this analysis with a second variable (*csework*) as follows:



Values: csework

Number of bins: 15

This time we see a larger negative skewness and a really significant result:



Values: csework

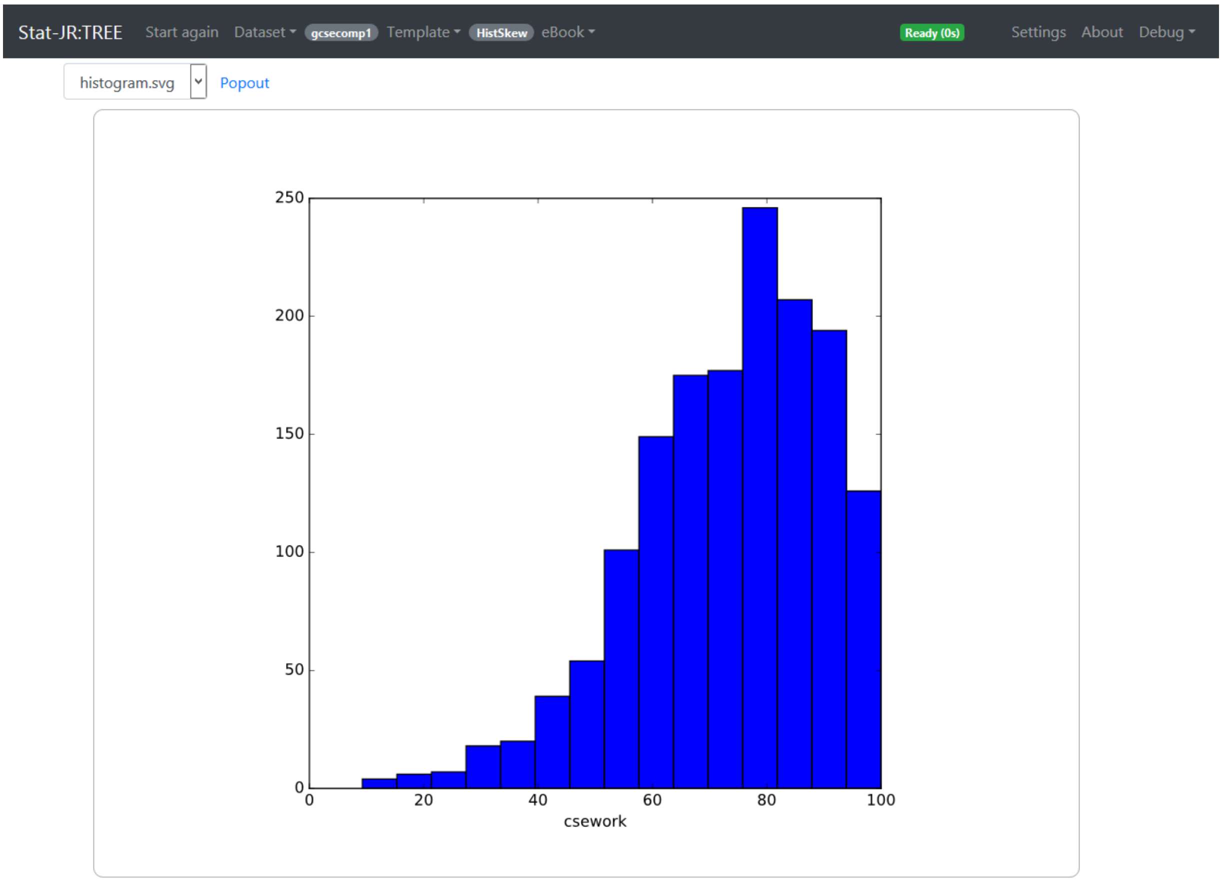
Number of bins: 15

```
Current input string: {'vals': 'csework', 'bins': '15'}
```

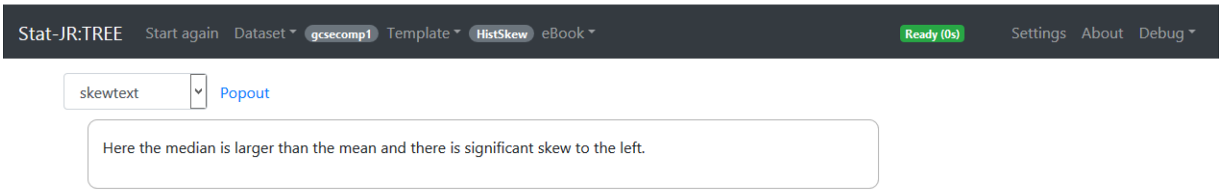
```
Command: RunStatJR(template='HistSkew', dataset='gcsecomp1', invars = {'vals': 'csework', 'bins': '15'}, estoptions = {})
```

name	count	skewness	Z score	p value	Minimum
csework	1523	-0.752344489098	-10.794577299006	0.65138844563e-27	9.2593

This is backed up by the histogram which by eye is clearly skewed to the left



The text produced is however the same which shows the somewhat limited nature of what we can do here:



Here the median is larger than the mean and there is significant skew to the left.

It would be possible to vary the text to give some indicator of the amount of skew based on differing ranges of p value but a real challenge is that for large sample sizes most data will exhibit some significant skew and for small highly skewed data the skew might not be statistically significant. This variable has other interesting properties, for example it is clearly constrained to lie in a range (0-100) which we can see from the histogram, and we could accompany the graph with text describing the range. The histogram might also show multiple modes (peaks in the data) or in fact outlying values and it is a greater challenge still to help the user interpret this automatically.

2.5 Bringing it all together

We have so far shown in isolation how we might construct Stat-JR templates that perform some of the building blocks of a statistical analysis. We have covered in this chapter summary statistics and hypothesis testing and rather briefly plotting of data. Of course which of these elements are required for a particular dataset will depend on the type of data collected and what the researcher is requiring from their data. As a statistician one often gets involved in collaboration and/or consulting. By enquiring of the (applied) researcher what they are hoping to achieve with their data (or indeed if you are lucky to be involved at the start of the project what data they intend to collect) the statistician can suggest appropriate techniques and building blocks for their problem. Interestingly not all statisticians will use or suggest exactly the same approach to a problem based on their interaction with the researcher and so the dream of a single automated statistical analysis assistant (SAA) is just that – a dream. There is however still some merit in attempting to create SAAs and when we consider for example the hypothesis testing we have talked about in this chapter one might consider an SAA based around the flowchart idea of hypothesis tests that one often finds in textbooks i.e. if one wishes to test a hypothesis there are usually a few questions like: is normality of the response variable an appropriate assumption? Are there two or more groups that we wish to compare? Are the data structured in pairs? By following a few such questions we can pinpoint the specific test required and so one could imagine an SAA that asks these questions and then pointed the researcher to the appropriate template to perform their test or indeed a super template that combines all the standard tests.

The one thing we have seen in the work so far is that performing a test in an automatic fashion is a far easier task than interpreting a figure constructed from the data. This will be a challenge in the SAAs that we introduce in the rest of this book and we will often (skewness of histograms aside) simply duck the issue and instead instruct the user what to look for. At present also none of the decisions made in the SAA are based on interpreting figures but are based on either user inputs or model fits. We will now leave our section on single operations and consider a first statistical modelling adventure into the world of the linear regression.

Chapter 3 – The stages of a statistical analysis and a first simple analysis assistant to perform all the steps of a linear regression

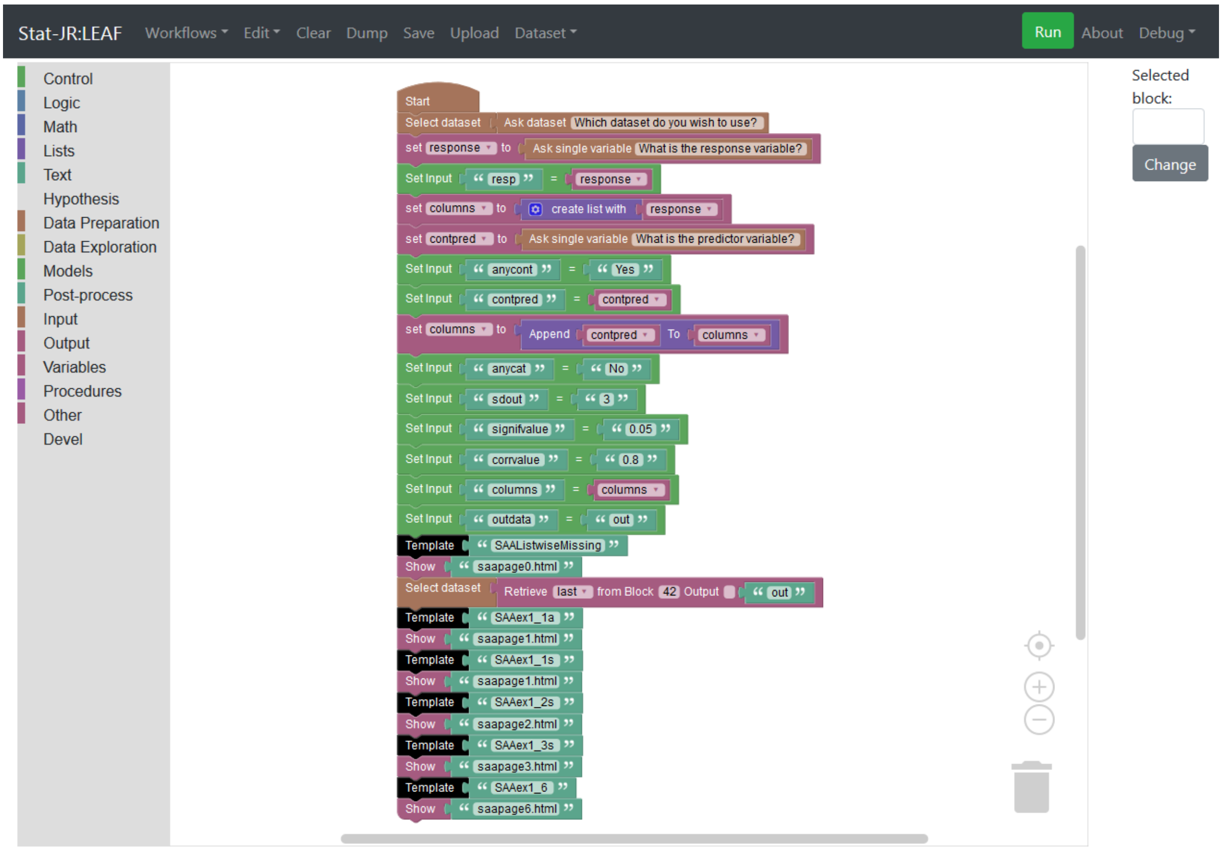
3.1 - Linear Regression

In the last chapter we finished by briefly discussing the idea of a flowchart that would allow questions about the hypothesis to be answered and the variables to be considered by the user to identify a simple statistical test that is required. In this chapter we move things on by considering the steps that together might make a specific statistical analysis. We will keep things simple by considering a very specific analysis with one continuous (and normally distributed) response variable and one continuous predictor variable, namely a simple linear regression. We should note that hear and onwards in this book the SAA workflows that we describe use as their underlying estimation software the MLwiN (Charlton, Rasbash, Browne, Healy, & Cameron, 2017) package and so you will need to ensure you have MLwiN installed and pointed at correctly by Stat-JR to run the models.

We will then piece together a series of steps that one might do in this situation. Before we start we will run a template that ensures that only observations with no missing data are used (we will talk about other things to do with missing data in later SAAs). Then one might next do some exploratory data analysis (exploration of the two variables independently) before looking at them together and whether there is any correlation between them. Although we intend simply to fit a linear regression we might also consider whether in practice we should fit polynomial terms (as we will do this in later SAAs). We will then fit our chosen linear model and display a plot of the predicted model fit before showing some residual plots to look at if the model fit is good or if there are identifiable outliers (values that the model does not fit well).

We can firstly look at a workflow for performing these steps in the LEAF interface. The LEAF system is designed for users to set up their own workflows using Stat-JR templates and there is also the possibility of automatically saving a series of operations performed using TREE as a workflow to be used in LEAF (see the LEAF workflow guide (Browne, Parker, Charlton, Michaelides, & Moreau, 2016) for more details.)

For now we'll begin by loading up LEAF and selecting from the *Workflows/Stat_Assistant* menu the workflow *linreg* from the list and then the screen should look as follows:



```
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"
```

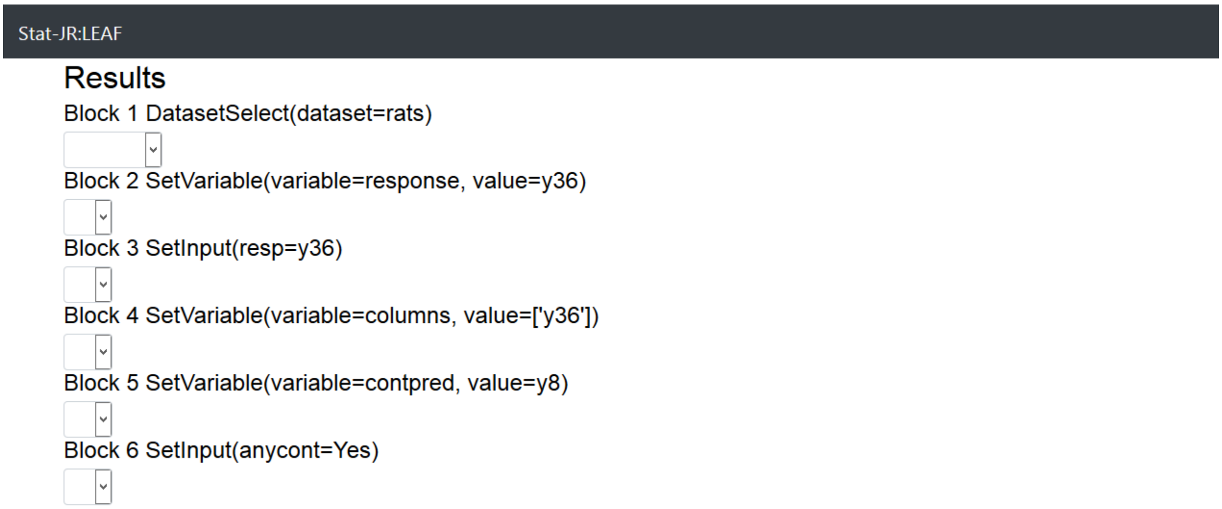
Basically the workflow consists of a few inputs from the user at the outset i.e. what is the dataset, response variable and predictor variable. The workflow then sets a few other inputs required by the various templates given below before running a set of templates and displaying their outputs.

We see that first a template *SAAListwiseMissing* is run and this template will ensure that only observations that are not missing for the variables used in the modelling are included in the later operations. This template outputs: a html object *saapage0.html* which simply describes how many observations are used, and a dataset, *out* containing only the observations required. The dataset is then retrieved in the next block so that it is used in the blocks that follow before the series of templates that make up the stages of the regression are run in turn.

The *SAAex1_1a* template will give some summary statistics and a plot of the response variable whilst the *SAAex1_1s* template will do the same for the predictor variable. The *SAAex1_2s* template will produce correlations between the predictor and response and assess what order of polynomial fits the relationship best. The *SAAex1_3s* template then fits a linear regression model and shows the estimates along with a predicted plot. Finally the *SAAex1_6* template gives residual plots for the model fitted (note this template actually has to refit the model to get the residuals as nothing is passed from template to template here).

It should be noted that the numbering is such because originally an SAA for more general linear models was written with templates running from SAAex1_1 to SAAex1_7 and this has been adapted for the special case of the linear regression by replacing some templates with others and we have used the addition of an s to some template names to indicate a single predictor.

If we run the workflow in LEAF and use dataset – *rats*, response – *y36* and predictor *y8* we will get pages of output:



Results

Block 1 DatasetSelect(dataset=rats)

Block 2 SetVariable(variable=response, value=y36)

Block 3 SetInput(resp=y36)

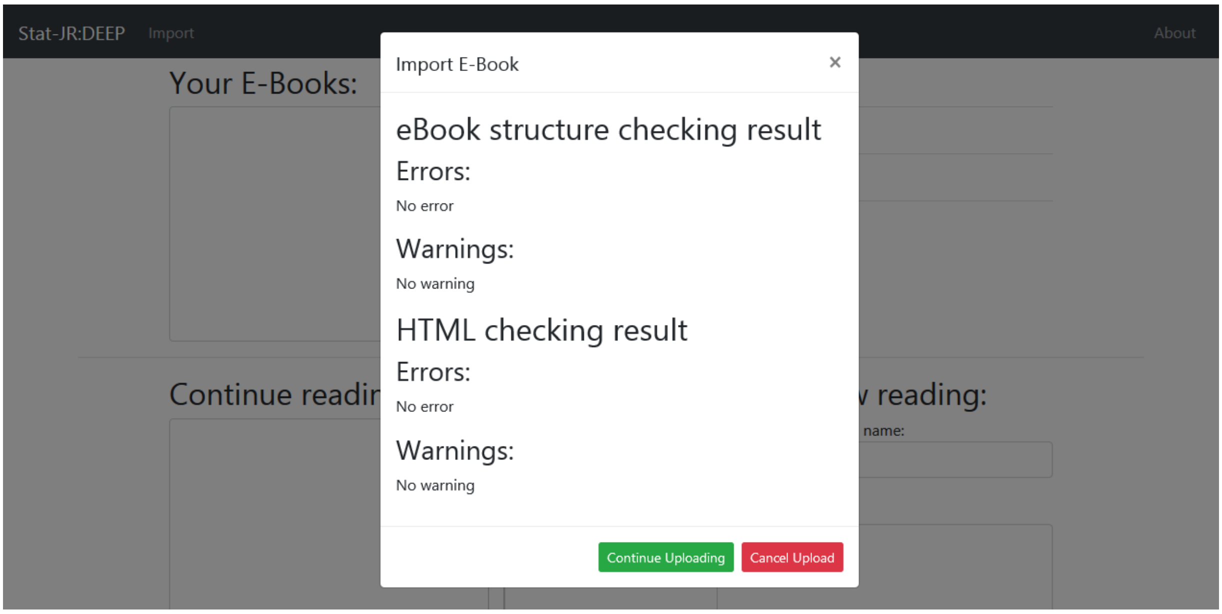
Block 4 SetVariable(variable=columns, value=['y36'])

Block 5 SetVariable(variable=contpred, value=y8)

Block 6 SetInput(anycont=Yes)

This output will show the various output objects embedded into a list of blocks if you scroll down through it.

We can also run the E-book version of the SAA so to do this we need to load up the DEEP interface to Stat-JR. We then click on **Import** and from the list of eBooks found in the eBooks subdirectory under the Stat-JR install you will need to select *linreg.zip*. Click on **Open** to see:



eBook structure checking result

Errors:

No error

Warnings:

No warning

HTML checking result

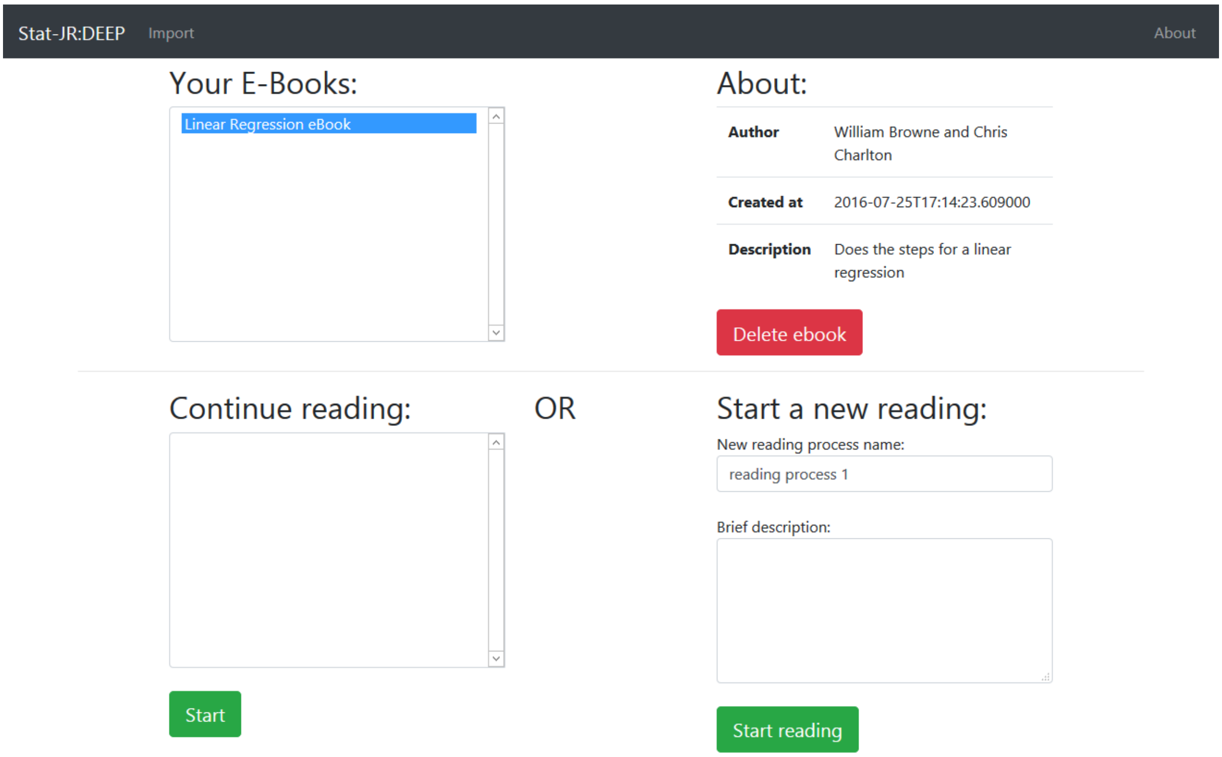
Errors:

No error

Warnings:

No warning

And then click on *Continue Uploading* and select the eBook that appears by clicking on it in the list under *Your E-Books* and then typing a name in the *New Reading process name* box:



Your E-Books:

Linear Regression eBook

About:

Author	William Browne and Chris Charlton
Created at	2016-07-25T17:14:23.609000
Description	Does the steps for a linear regression

Continue reading:

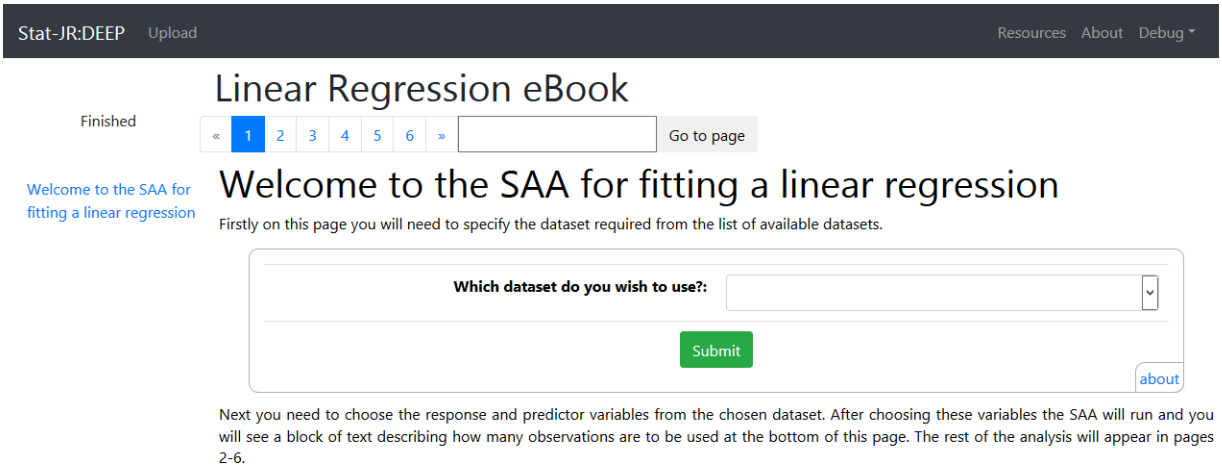
OR

Start a new reading:

New reading process name: reading process 1

Brief description:

Clicking on *Start Reading* and you get the following:



Linear Regression eBook

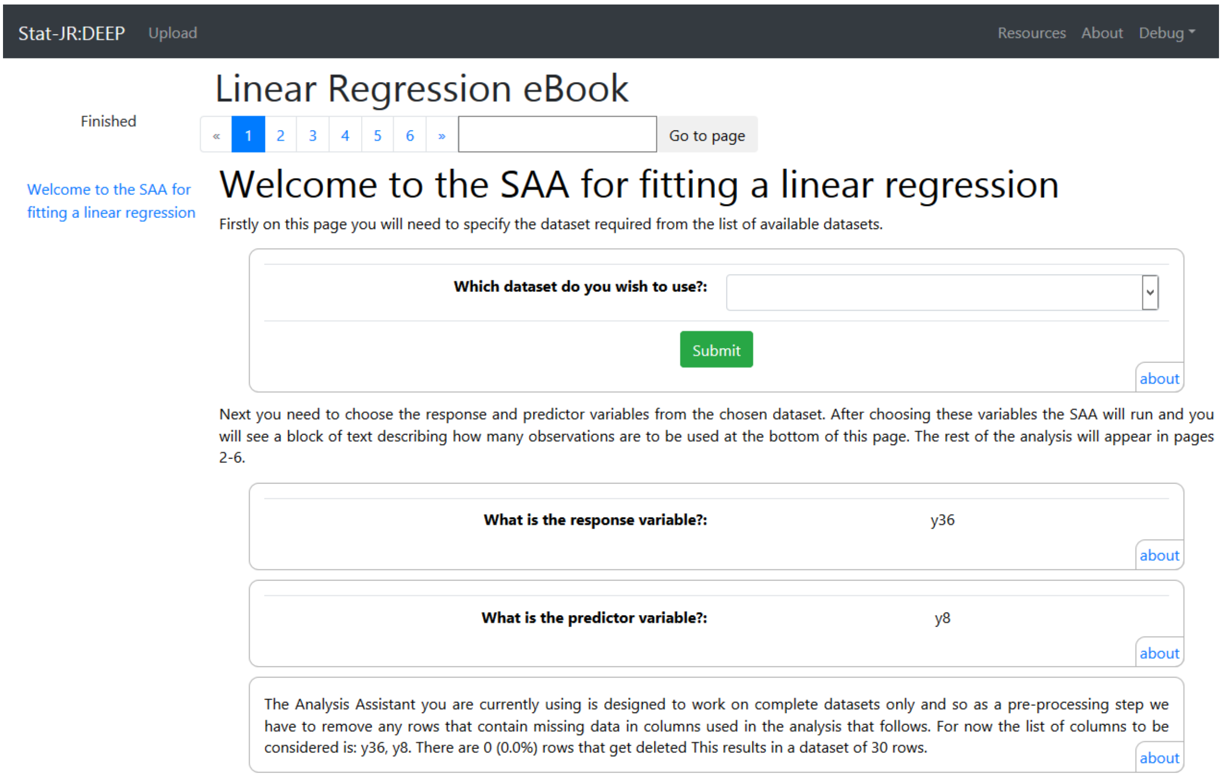
Welcome to the SAA for fitting a linear regression

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?:

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.

The eBook has the workflow embedded in it but also has additional text written around the input boxes to make it more user-friendly. Now we will select again for the dataset, *rats*, for the response variable, *y36,* and for the predictor variable, *y8* to get:



Linear Regression eBook

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?:

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.

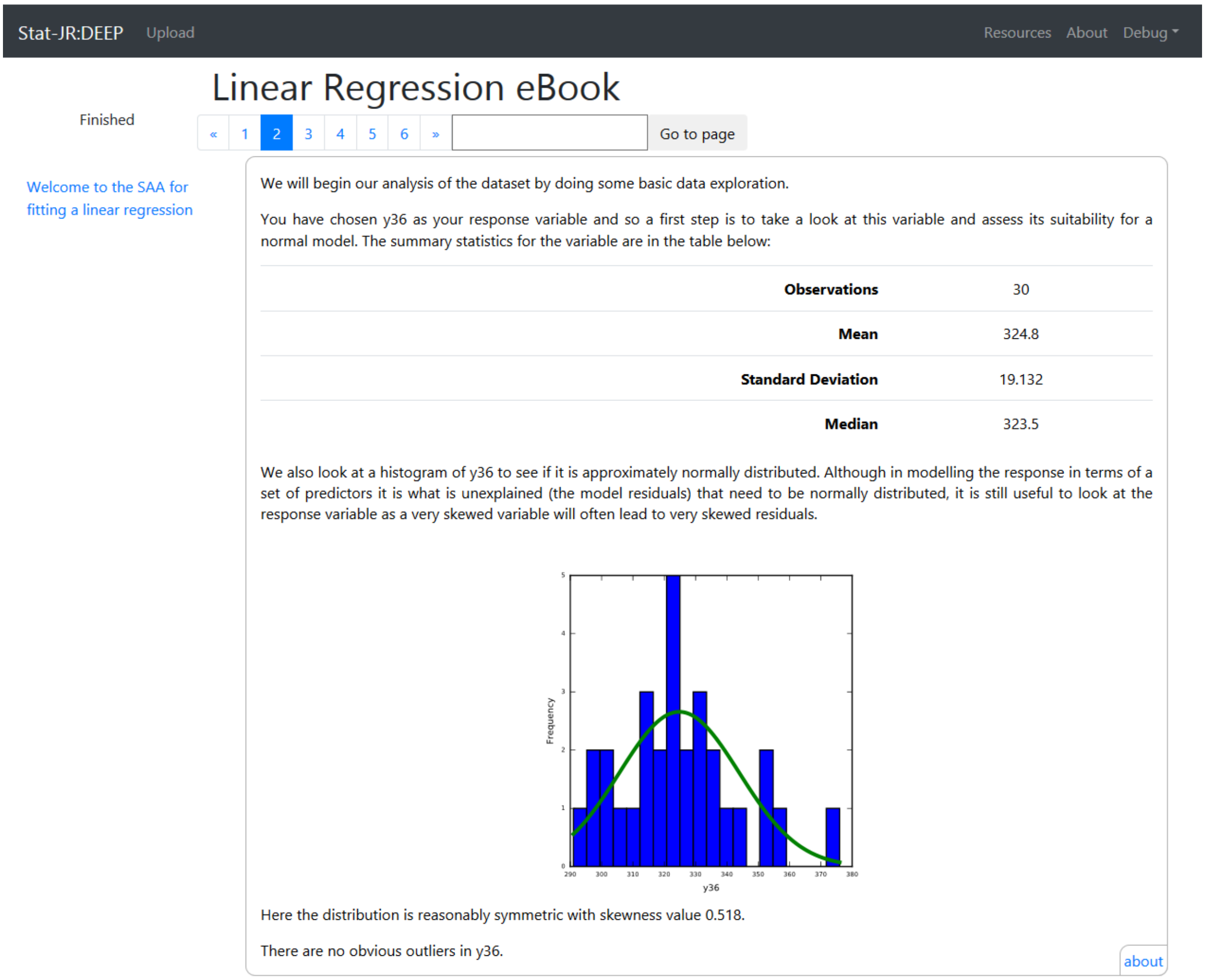
What is the response variable?: y36

What is the predictor variable?: y8

The Analysis Assistant 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. There are 0 (0.0%) rows that get deleted This results in a dataset of 30 rows.

Note that the next input to be selected appears as a blank list so that here the *datasets* box is blank as having made all your selections the eBook has run and is ready to start over. You will also see that a box at the bottom that has indicated that missing data has been removed.

Page 2 will show you some background on the response. This includes some basic summary statistics along with a histogram of the response with a description of how symmetric it is. These should be familiar now as they are similar to the basic statistics we described in the last chapter.



Linear Regression eBook

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:

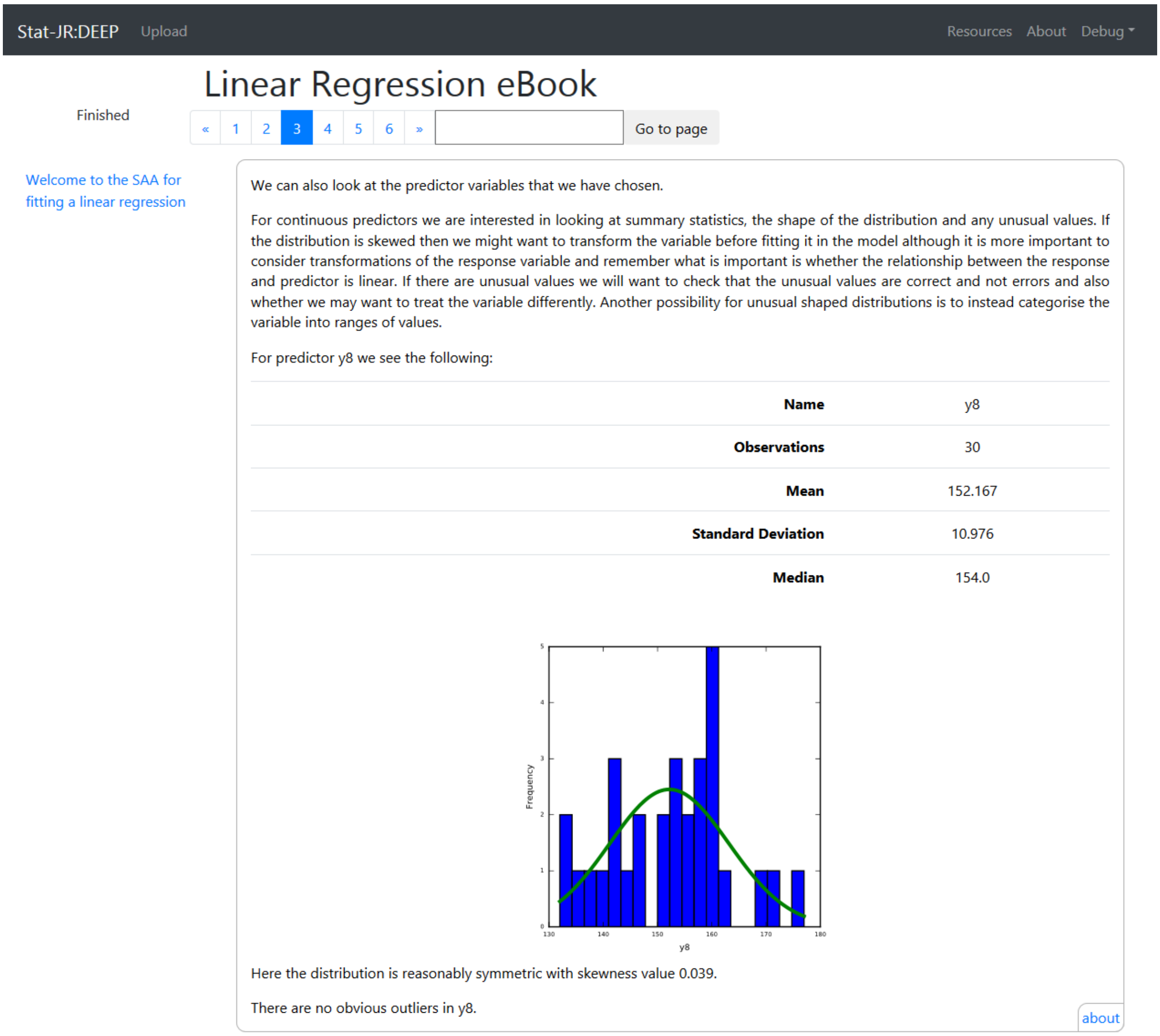
Observations	30
Mean	324.8
Standard Deviation	19.132
Median	323.5

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.

Here we see despite the small sample size a fairly symmetric distribution (with normal curve superimposed) with no outliers reported. Moving to page 3 this repeats the same summary statistics along with the histogram and skewness description for the chosen continuous predictor as shown below.



Linear Regression eBook

We can also look at the predictor variables that we have chosen.

For continuous predictors we are interested in looking at summary statistics, the shape of the distribution and any unusual values. If the distribution is skewed then we might want to transform the variable before fitting it in the model although it is more important to consider transformations of the response variable and remember what is important is whether the relationship between the response and predictor is linear. If there are unusual values we will want to check that the unusual values are correct and not errors and also whether we may want to treat the variable differently. Another possibility for unusual shaped distributions is to instead categorise the variable into ranges of values.

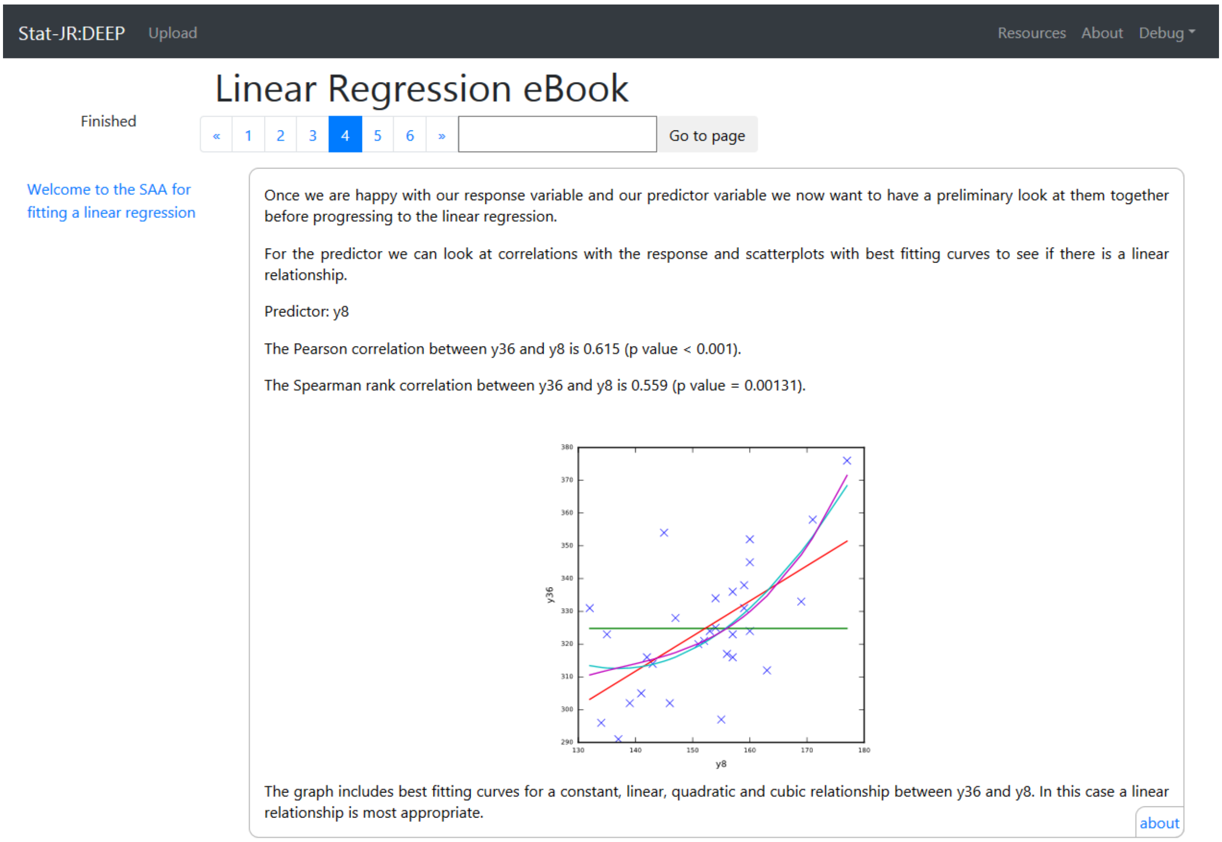
For predictor y8 we see the following:

Name	y8
Observations	30
Mean	152.167
Standard Deviation	10.976
Median	154.0

Here the distribution is reasonably symmetric with skewness value 0.039.

There are no obvious outliers in y8.

Page 4 will next look at the correlation (quoting both the Pearson and Spearman coefficients) between the response and the predictor. It will also fit a series of models to the response for different types of effect of the predictor – constant or no effect, linear effect, quadratic effect and cubic effect and will suggest which is the most appropriate (based on model fit) while superimposing the four fitted relationships over the data points so that the user can look by eye at what the relationships look like. This modelling is actually more complex than the simple linear regression that this SAA is designed to do but we have borrowed this template from the workflow in chapter 4 for more general linear models and so the user may consider the output here to suggest whether they need to consider a more complex SAA. In fact for our example the SAA suggests that the linear relationship is the best fit.



Linear Regression eBook

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.

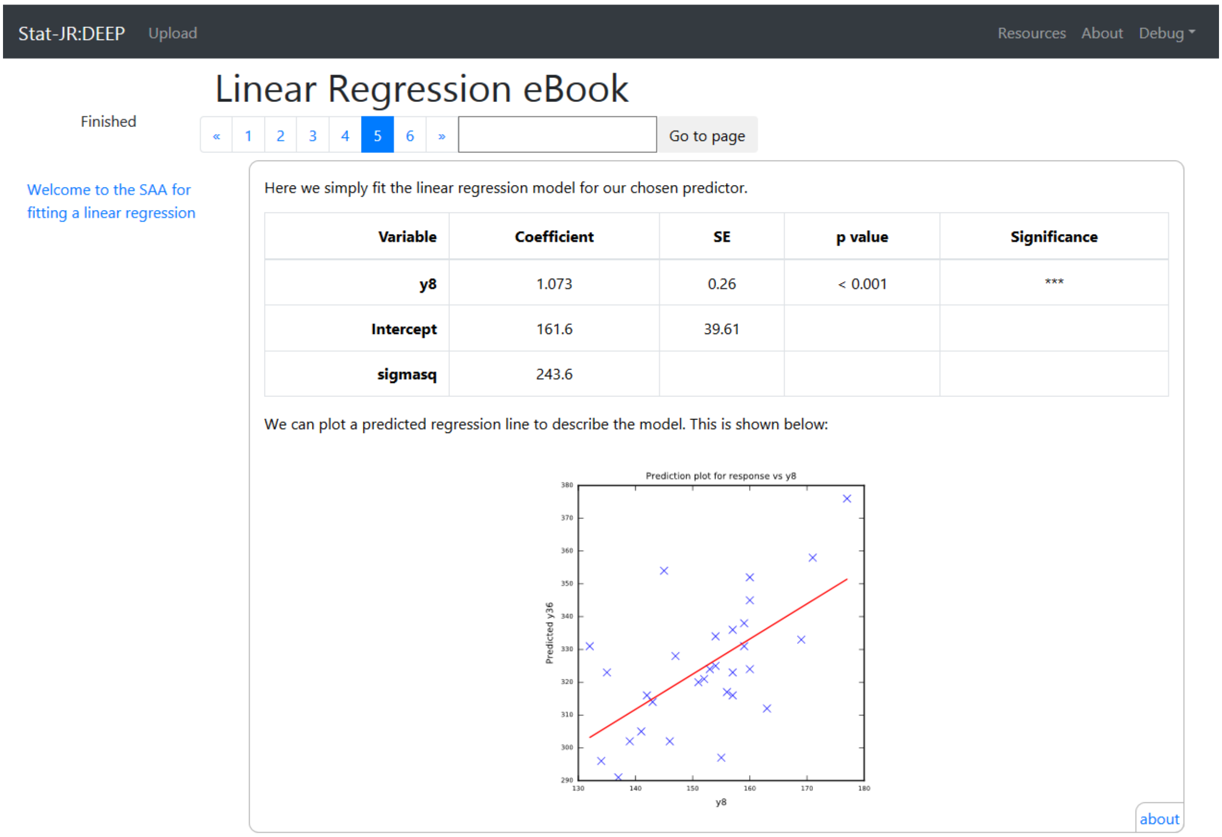
Predictor: y8

The Pearson correlation between y36 and y8 is 0.615 (p value < 0.001).

The Spearman rank correlation between y36 and y8 is 0.559 (p value = 0.00131).

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.

On page 5 the simple linear regression has then been fitted on its own, and the estimates are shown along with an indication (via a p value and stars with one for less than 0.05, two for less than 0.01 and three for less than 0.001) of the significance of the predictor. This page also draws a plot of the fitted line for the linear regression as shown below:



Linear Regression eBook

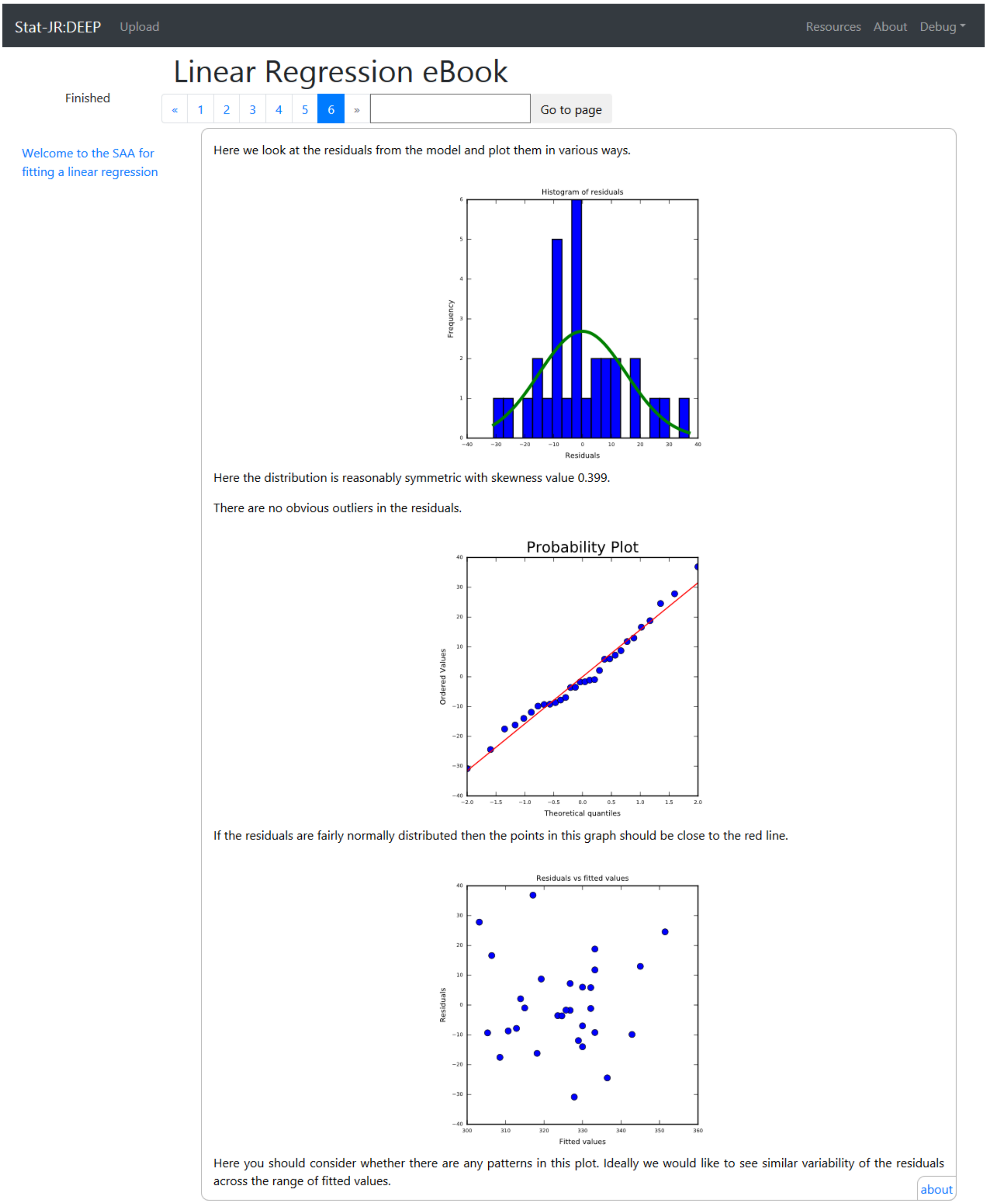
Here we simply fit the linear regression model for our chosen predictor.

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

We can plot a predicted regression line to describe the model. This is shown below:

Finally page 6 looks at post model fit checking using the residuals. Here three plots are displayed. The first is a simple histogram of the raw residuals which is accompanied by a description of the skewness of their distribution (in a similar way to that done for the response earlier). Here for our example pleasingly these are symmetric with no outliers.

We next plot a Q-Q plot of the residuals against normal distributed quantiles. We do not include a formal test here but suggest the user compares the plotted residuals to the red straight line as any divergence would indicate some degree of non-normality. Finally there is a plot of the residuals against the fitted values. Again we currently don't give any interpreted guidance based on the plot but simply suggest that the variability of the residuals should be roughly constant across the range of fitted values if the model fits well.



Linear Regression eBook

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

Here the distribution is reasonably symmetric with skewness value 0.399.

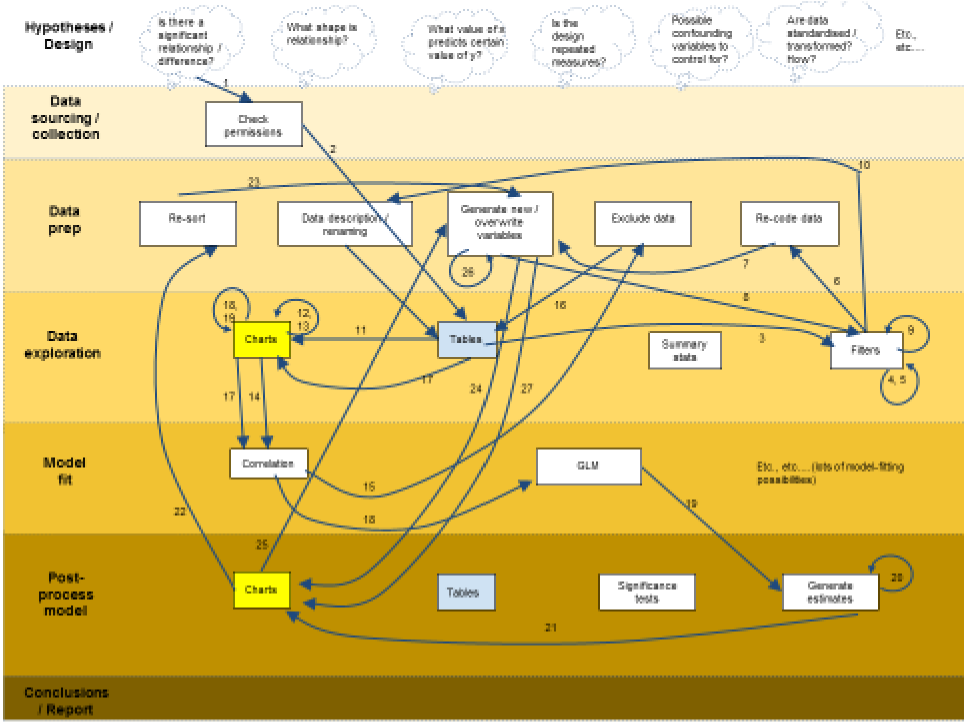
There are no obvious outliers in the residuals.

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

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 fitted values.

Note that now if you wish to restart this eBook and try a different combination of response and/or predictor then you can do this on page 1.

Note also that this is just one set of operations that might be considered in doing an analysis involving one predictor and a continuous response variable. When we started on the project that has developed Stat-JR's LEAF interface and our SAAs we set out a task for a reading group of statisticians to demonstrate how they approached similar problems and then one of us (Richard Parker) sat down and tried to create flow charts illustrating how this works in practice. Below is one example of such an analysis for a regression type analysis.

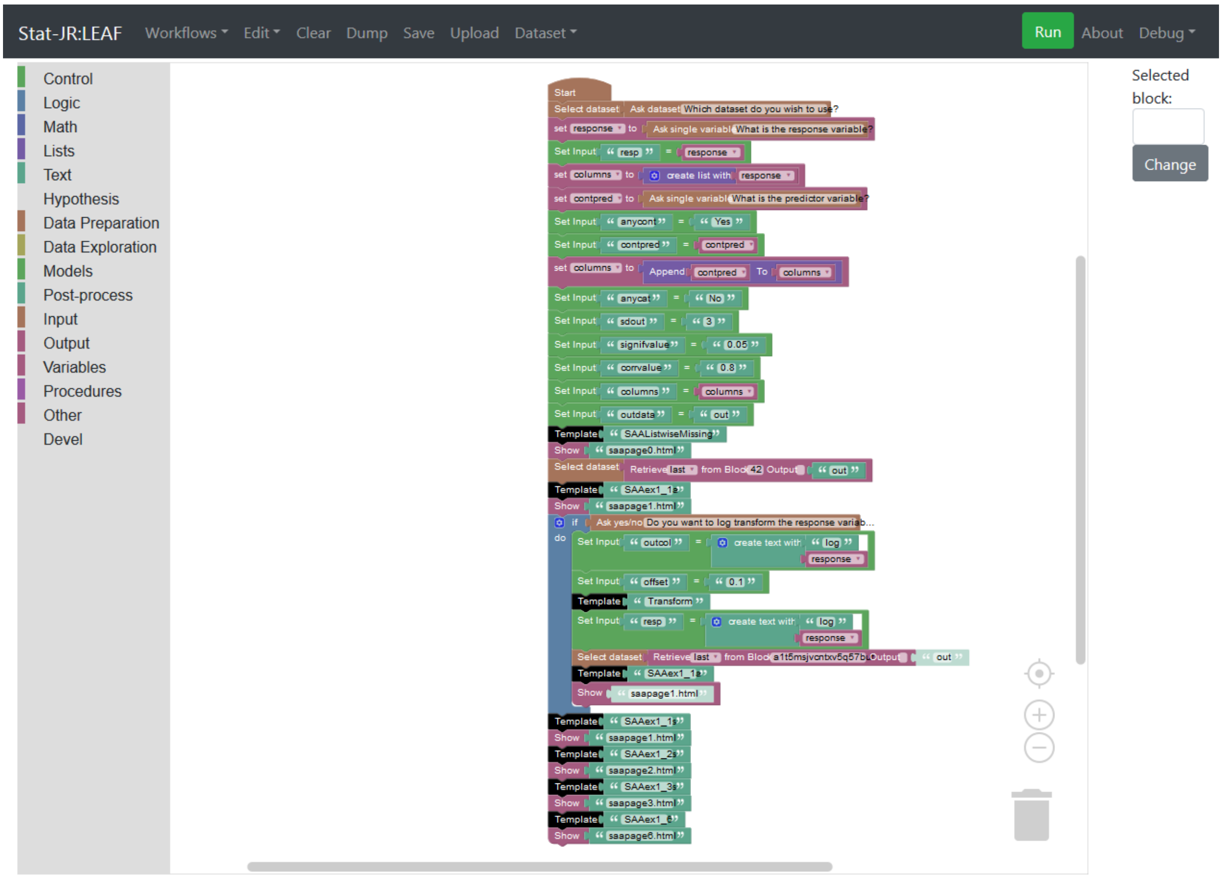

Hypotheses / Design
Is there a significant relationship / difference?
What shape is relationship?
What value of x predicts certain value of y?
Is the design repeated measures?
Possible confounding variables to control for?
Are data standardised / transformed? How?
Etc., etc....
Data sourcing / collection
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
18, 19
19
20
21
22
23
24
25
26
27

What one can immediately notice is that the whole process has many steps including several data preparation / processing steps that are done based on preliminary analysis of the data. It is of course possible to use the linear regression SAA to do a preliminary analysis and on the basis of some of the outputs go away and do some further data processing before reusing the SAA. With this in mind there is consider merit in using the SAAs we have created to quickly get to know your data and maybe, in particular from the EDA they produce, spot things that need altering (including errors!) within the data prior to doing a more thorough analysis.

3.2 Transformations

One way we might extend this workflow and eBook is to allow the user some flexibility into whether they fit a model to the response or, having inspected the shape of the response, to a transformed response. To illustrate this here we will add a question to the workflow after displaying the histogram to allow the user the option to instead use a log transformed variable.

The workflow is called *linreglog* and the additional blocks can be seen in the window below:



```
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 "
if  Ask yes/no Do you want to log transform the response variab...
do  Set Input " outcol " = create text with " log "
                                             response
    Set Input " offset " = " 0.1 "
    Template " Transform "
    Set Input " resp " = create text with " log "
                                           response
    Select dataset  Retrieve last from Block a1t5msjvcntxv5q57bu 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 "
```

In the workflow we have added a question block and then if it is answered *yes* the workflow constructs a new column which has the same name as the old column but with log added to the start. The *Transform* template then constructs a logged version of the response and adds it to the dataset *out* which is then used. The initial summary statistics section is then repeated for this logged response so that we can observe the shape of the logged response. Note that the Transform template has an *offset* parameter which is used to ensure that we don't log a negative number as if any values are zero then the minimum value plus the offset is added before logging.

We do not show the output screens here but there is also an eBook entitled *linreglog.zip.* One possible choice is to look at the *mmmec* dataset with response *obs* and predictor *uvbi*. Here you can contrast the impact of logging (or not) the response variable. In reality it is better to fit Poisson models to this dataset (as we illustrate in chapter 10), and include the (logged) expected counts as an offset but it is still interesting to look at a simple regression. One other interesting observation with this dataset is that the SAA suggests that a linear relationship is not the most appropriate and we will revisit this in section 4.2. Before that we will move from simple linear regressions to linear models.

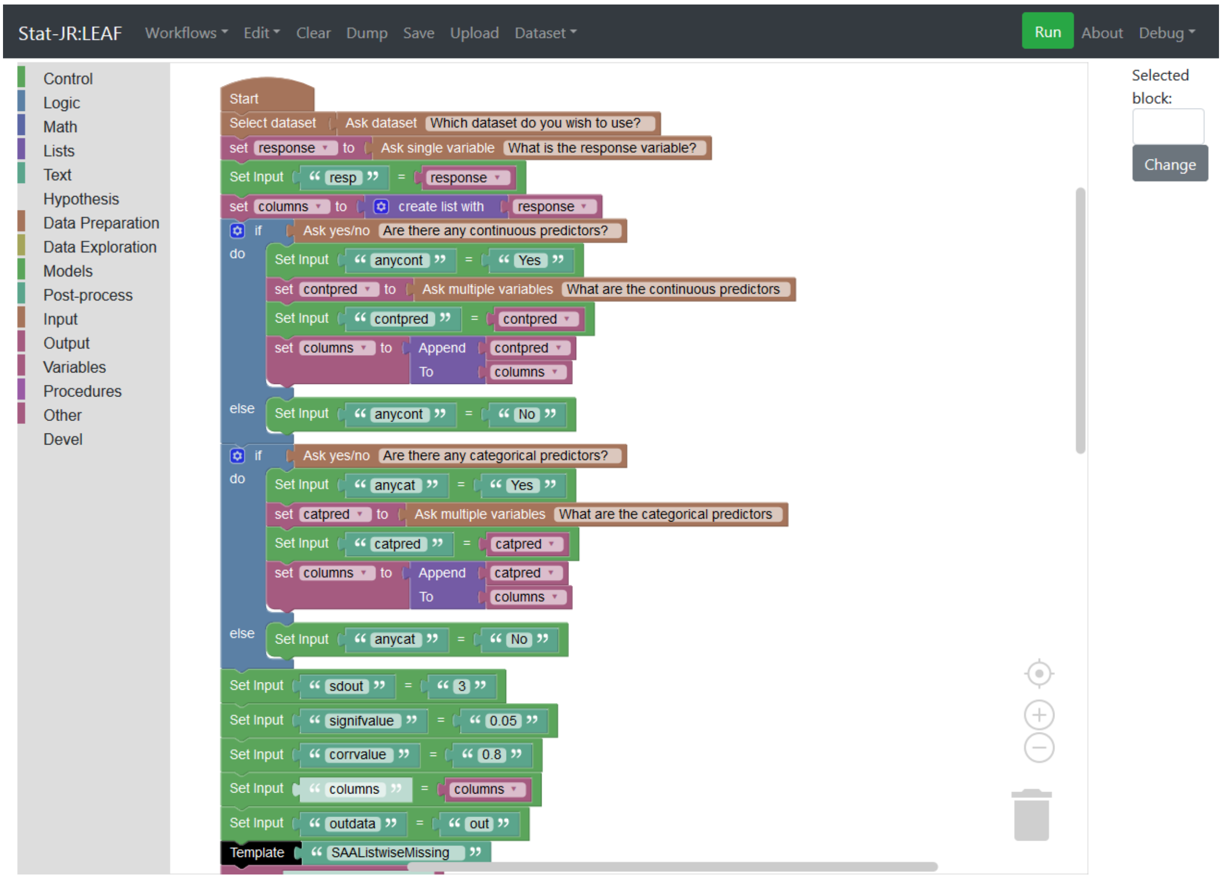
Chapter 4 – Linear Modelling

4.1 - Extending the analysis to allow for more than one predictor variable

In this section we will include several new features at once. Fitting a regression with one continuous predictor variable and restricting the relationship to be linear is rather restrictive. Often we will have several predictors to test and these will be of different types (both continuous and categorical). We might also like to build up our model fitting to include several predictor variables at once and in some way find a ‘best’ fitting model.

Fitting more general linear models (as models which assume a normally distributed response) is fairly straightforward and we would anticipate doing similar exploratory data analysis and model fitting steps but we may also look at relationships between predictor variables to check for issues such as colinearity. We will also do slightly different exploratory analysis for categorical predictors.

We will start by showing the workflow in the LEAF interface so load up *LEAF* and from the *Workflows/Stat_Assistant* menu select *lmodel*. The top of the workflow looks as follows:

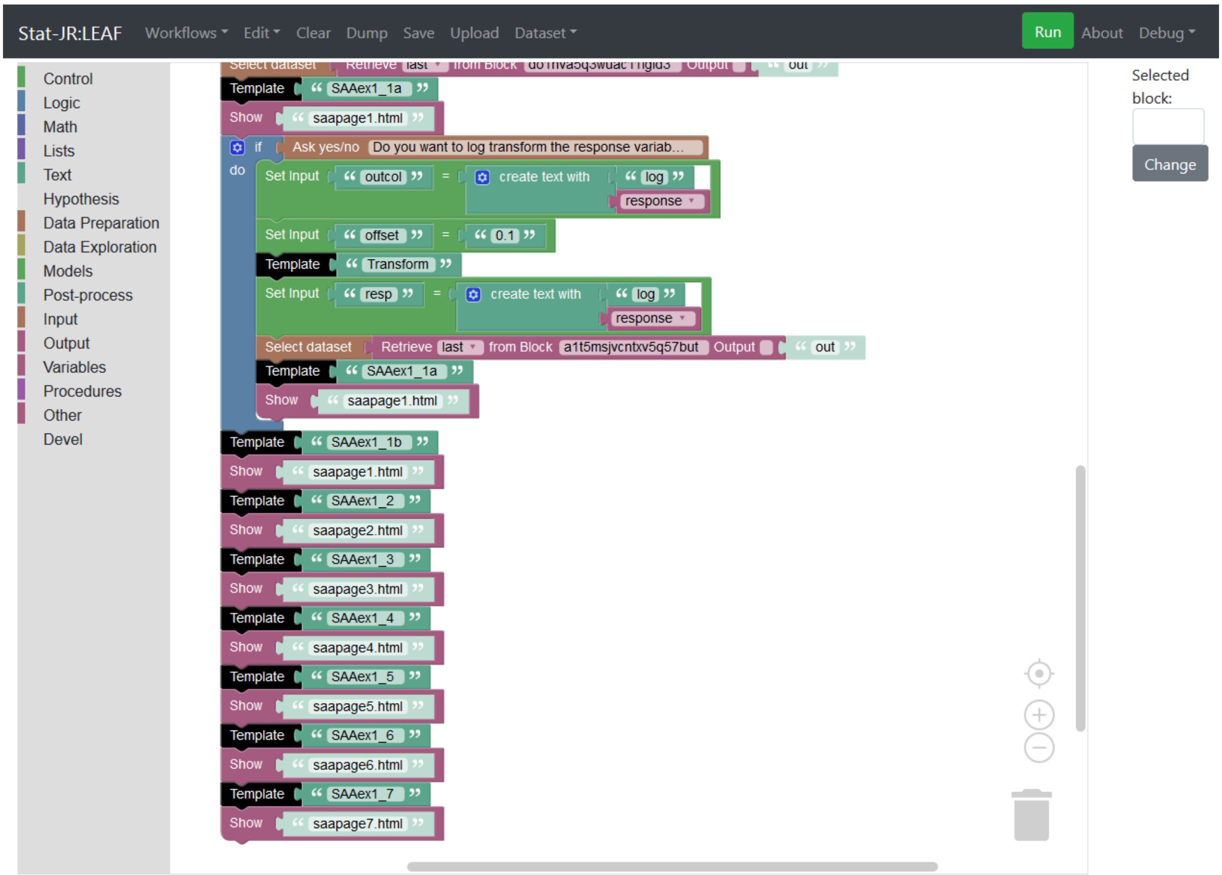


```
Start
Select dataset  Ask dataset  Which dataset do you wish to use?
set response to  Ask single variable  What is the response variable?
Set Input  " resp "  =  response
set columns to  create list with  response
if  Ask yes/no  Are there any continuous predictors?
do  Set Input  " anycont "  =  " Yes "
    set contpred to  Ask multiple variables  What are the continuous predictors
    Set Input  " contpred "  =  contpred
    set columns to  Append  contpred
                    To  columns
else  Set Input  " anycont "  =  " No "
if  Ask yes/no  Are there any categorical predictors?
do  Set Input  " anycat "  =  " Yes "
    set catpred to  Ask multiple variables  What are the categorical predictors
    Set Input  " catpred "  =  catpred
    set columns to  Append  catpred
                    To  columns
else  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 "
```

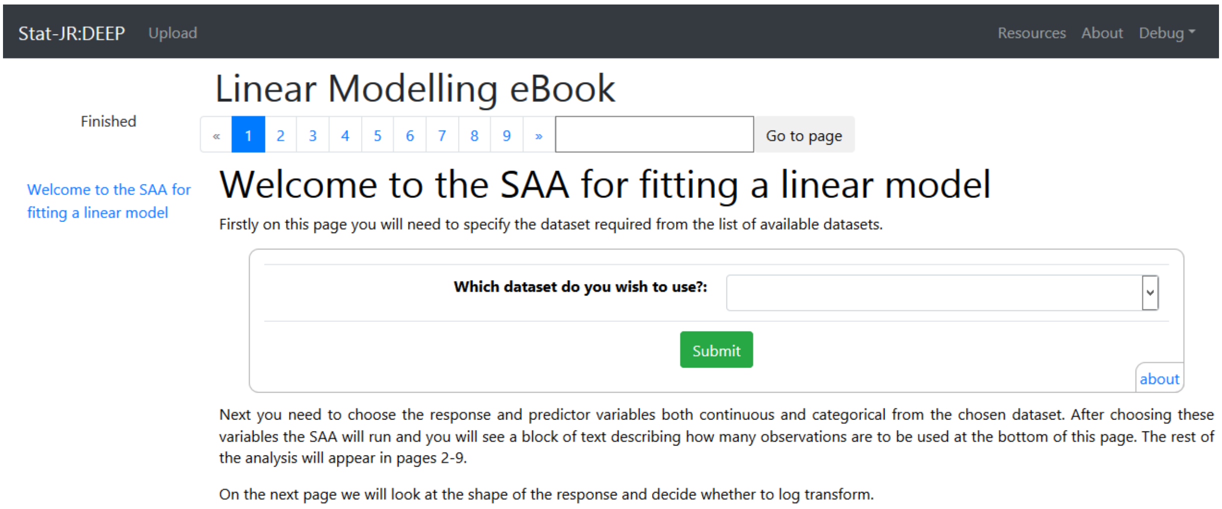
Here compared with the linear regression model there are more input questions and some of these inputs require conditional operations i.e. there is an input asking whether there are any continuous predictors and if (and only if) there are then the additional question of what columns are the continuous predictors is asked. We differentiate between continuous and categorical predictors here as they are treated differently in the modelling and also have different forms of exploratory data analysis associated with them. Aside from setting the inputs the workflow also creates a *columns* list and this is used to store all the columns that are used in the modelling so that when the list-wise sweeping out of missing data is carried out the appropriate rows are removed. Basically we remove any data rows that contain missing data in any of the variables involved in our modelling.

Moving down the workflow we have adapted the linear regression workflow that allowed the possibility of a log transform to the response variable and thus there is another condition block that asks this and if the transform is required then this is done.

This is followed by the progression of black box templates that each create a page of the resulting output. This is possibly not the most attractive way of creating an SAA as it makes things less transparent to the user. The main reason for this approach is that for this example and more generally the number of output objects – graphs, tables etc. is often dependent on the inputs for example there may be a histogram for each continuous predictor. The possibility of such an unknown number of objects to display is problematic for the DEEP interface and so instead of putting more of the workings in the workflow they are occurring in the (super) templates and each template then creates one (or more) pages of output as html objects. These pages can have varying numbers of graphs etc. but the DEEP system only needs to know that one page will appear.



We will not run this workflow here however you can if you wish. Instead we will move on to the equivalent eBook which is called linmodel (*linmod.zip*). We will assume you are able to load up DEEP and select this eBook as we demonstrated in the last chapter and so when you have done this you should be presented by page 1 of the eBook thus:



Linear Modelling eBook

Welcome to the SAA for fitting a linear model

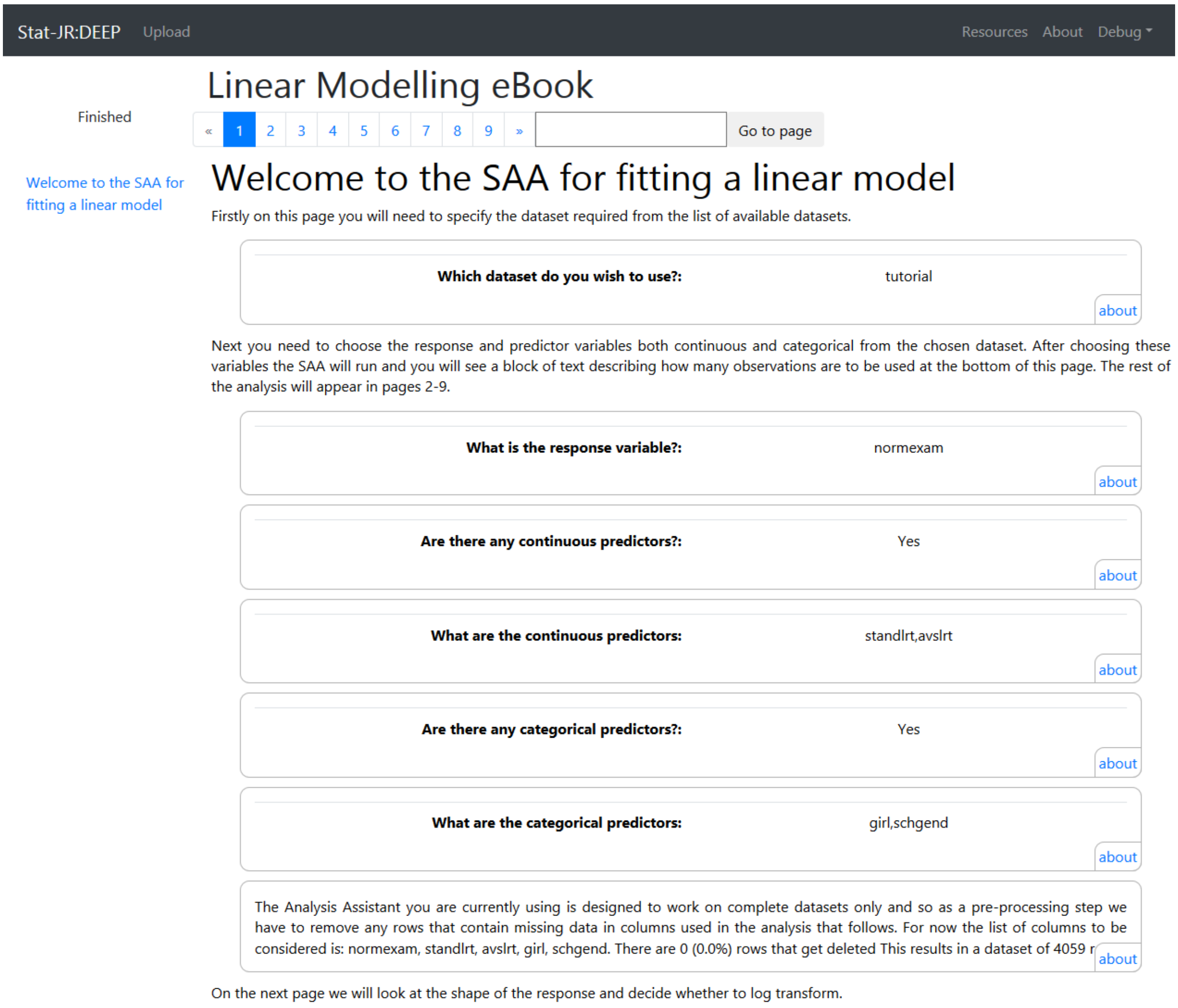
Firstly on this page you will need to specify the dataset required from the list of available datasets.

Which dataset do you wish to use?:

Next you need to choose the response and predictor variables both continuous and categorical 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 2-9.

On the next page we will look at the shape of the response and decide whether to log transform.

We now have to choose the initial set of inputs and so we will select the *tutorial* dataset along with response, *normexam.* We will have both continuous (*standlrt* and *avslrt*) and categorical (*girl* and *schgend*) predictors and having filled in these six questions the eBook will start working and page 1 will look as follows:



Linear Modelling eBook

Welcome to the SAA for fitting a linear model

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?: tutorial

Next you need to choose the response and predictor variables both continuous and categorical 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 2-9.

What is the response variable?: normexam

Are there any continuous predictors?: Yes

What are the continuous predictors: standlrt,avslrt

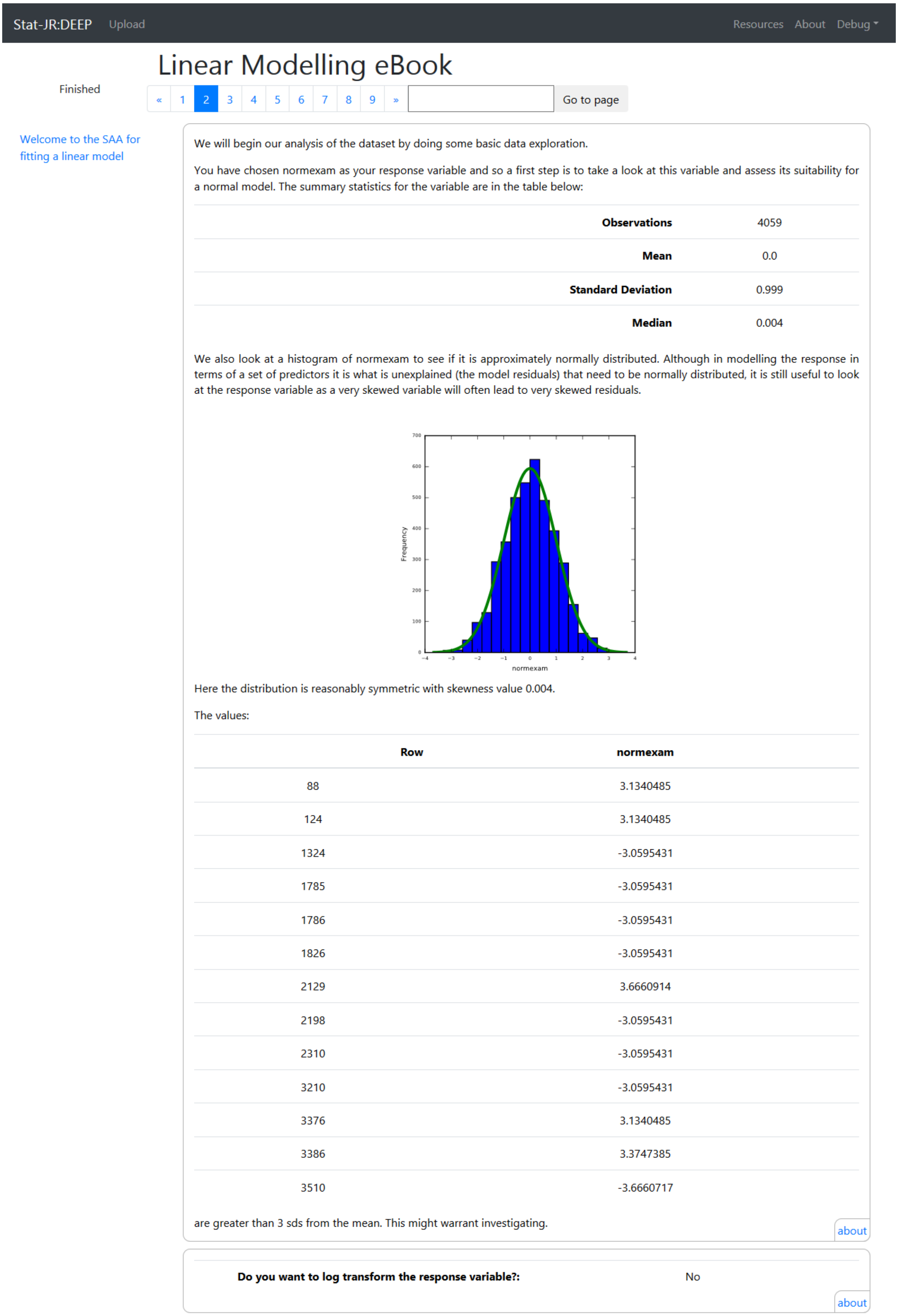
Are there any categorical predictors?: Yes

What are the categorical predictors: girl,schgend

The Analysis Assistant 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: normexam, standlrt, avslrt, girl, schgend. There are 0 (0.0%) rows that get deleted This results in a dataset of 4059 r

On the next page we will look at the shape of the response and decide whether to log transform.

Here we can see the last few inputs along with the generated text describing the removal of missing data and the resulting 4059 observations. The eBook has started running the underlying workflow by this point and this generated the box you see along with some further output on page 2 but it has then stopped as there is another question for the user on page 2. We will therefore move to page 2 to see this.



Linear Modelling eBook

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

You have chosen normexam 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	4059
Mean	0.0
Standard Deviation	0.999
Median	0.004

We also look at a histogram of normexam 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 0.004.

The values:

Row	normexam
88	3.1340485
124	3.1340485
1324	-3.0595431
1785	-3.0595431
1786	-3.0595431
1826	-3.0595431
2129	3.6660914
2198	-3.0595431
2310	-3.0595431
3210	-3.0595431
3376	3.1340485
3386	3.3747385
3510	-3.6660717

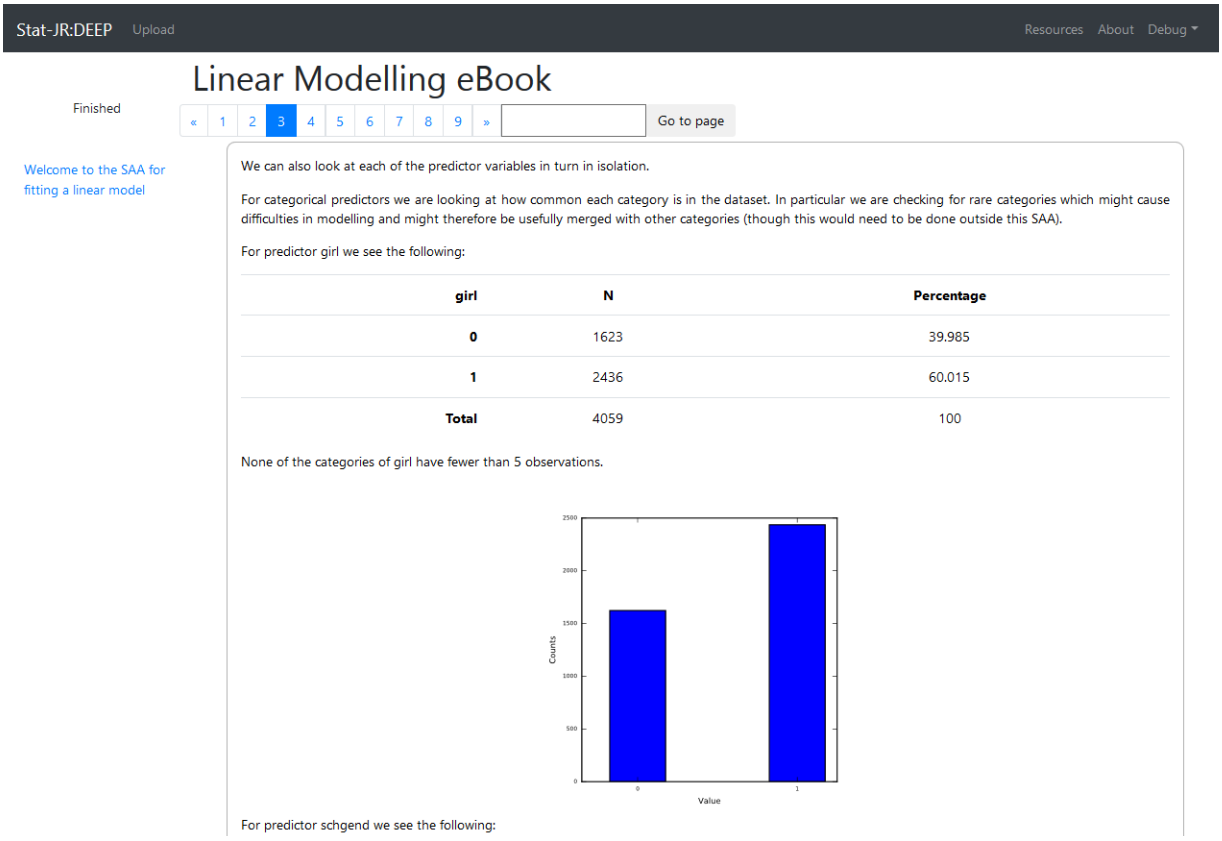
are greater than 3 sds from the mean. This might warrant investigating.

Do you want to log transform the response variable?:	No

Here as with the linear regression template the first stage is to show some descriptive statistics and a histogram with normal curve for the response (*normexam* in this case)

Then lower down the page is an additional input asking whether to log transform the variable. Here we will select *No* as the histogram gives us no reason to want to log transform and upon clicking on *Submit* the rest of the SAA eBook will now be run and generated.

It will be informative to look at aspects of each of the pages that follow in turn. On page 3 each of the predictors is looked at in isolation. As with the earlier regression SAA, for each of the continuous predictors some summary statistics and a histogram are produced. For categorical predictors we give a bar chart and percentages as illustrated for *girl* below:



Linear Modelling eBook

We can also look at each of the predictor variables in turn in isolation.

For categorical predictors we are looking at how common each category is in the dataset. In particular we are checking for rare categories which might cause difficulties in modelling and might therefore be usefully merged with other categories (though this would need to be done outside this SAA).

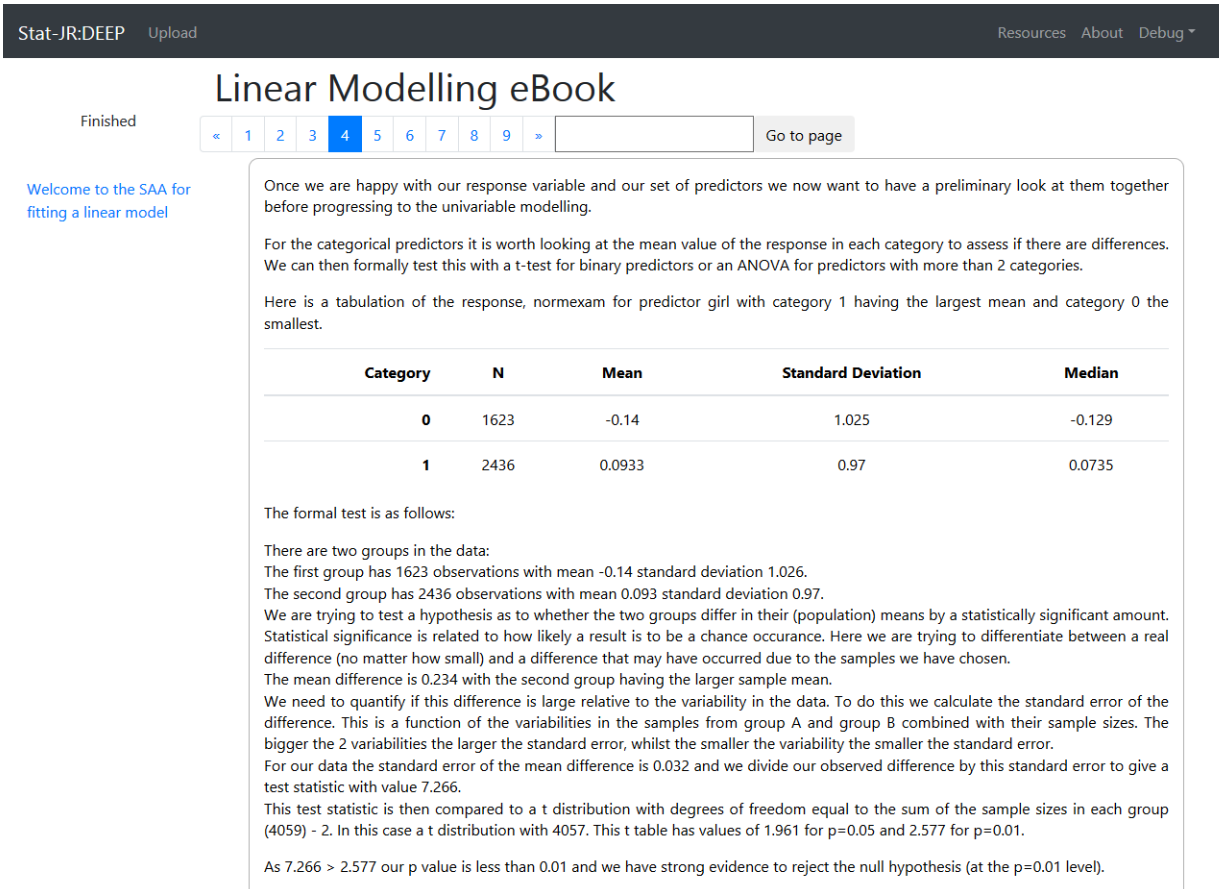
For predictor girl we see the following:

girl	N	Percentage
0	1623	39.985
1	2436	60.015
Total	4059	100

None of the categories of girl have fewer than 5 observations.

For predictor schgend we see the following:

Next on page 4 we look at how each predictor links to the response in isolation. For binary categorical predictors e.g. *girl* this includes showing summary statistics for the response for each category and then performing both a t test and a Mann Whitney test to look at significance as shown below:



Linear Modelling eBook

Once we are happy with our response variable and our set of predictors we now want to have a preliminary look at them together before progressing to the univariable modelling.

For the categorical predictors it is worth looking at the mean value of the response in each category to assess if there are differences. We can then formally test this with a t-test for binary predictors or an ANOVA for predictors with more than 2 categories.

Here is a tabulation of the response, normexam for predictor girl with category 1 having the largest mean and category 0 the smallest.

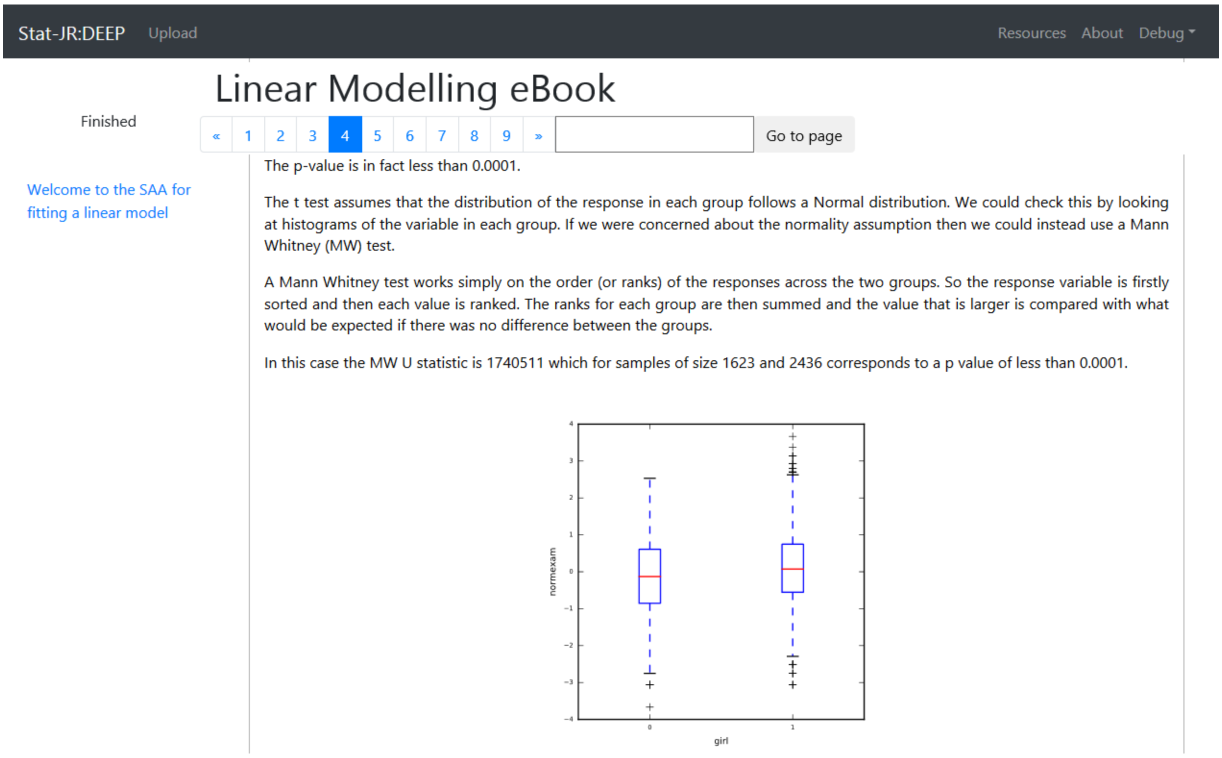
Category	N	Mean	Standard Deviation	Median
0	1623	-0.14	1.025	-0.129
1	2436	0.0933	0.97	0.0735

The formal test is as follows:

There are two groups in the data:
The first group has 1623 observations with mean -0.14 standard deviation 1.026.
The second group has 2436 observations with mean 0.093 standard deviation 0.97.
We are trying to test a hypothesis as to whether the two groups differ in their (population) means by a statistically significant amount. Statistical significance is related to how likely a result is to be a chance occurance. Here we are trying to differentiate between a real difference (no matter how small) and a difference that may have occurred due to the samples we have chosen.
The mean difference is 0.234 with the second group having the larger sample mean.
We need to quantify if this difference is large relative to the variability in the data. To do this we calculate the standard error of the difference. This is a function of the variabilities in the samples from group A and group B combined with their sample sizes. The bigger the 2 variabilities the larger the standard error, whilst the smaller the variability the smaller the standard error.
For our data the standard error of the mean difference is 0.032 and we divide our observed difference by this standard error to give a test statistic with value 7.266.
This test statistic is then compared to a t distribution with degrees of freedom equal to the sum of the sample sizes in each group (4059) - 2. In this case a t distribution with 4057. This t table has values of 1.961 for p=0.05 and 2.577 for p=0.01.

As 7.266 > 2.577 our p value is less than 0.01 and we have strong evidence to reject the null hypothesis (at the p=0.01 level).

This is followed up by a box plot to illustrate the distribution of the response in each category:



Linear Modelling eBook

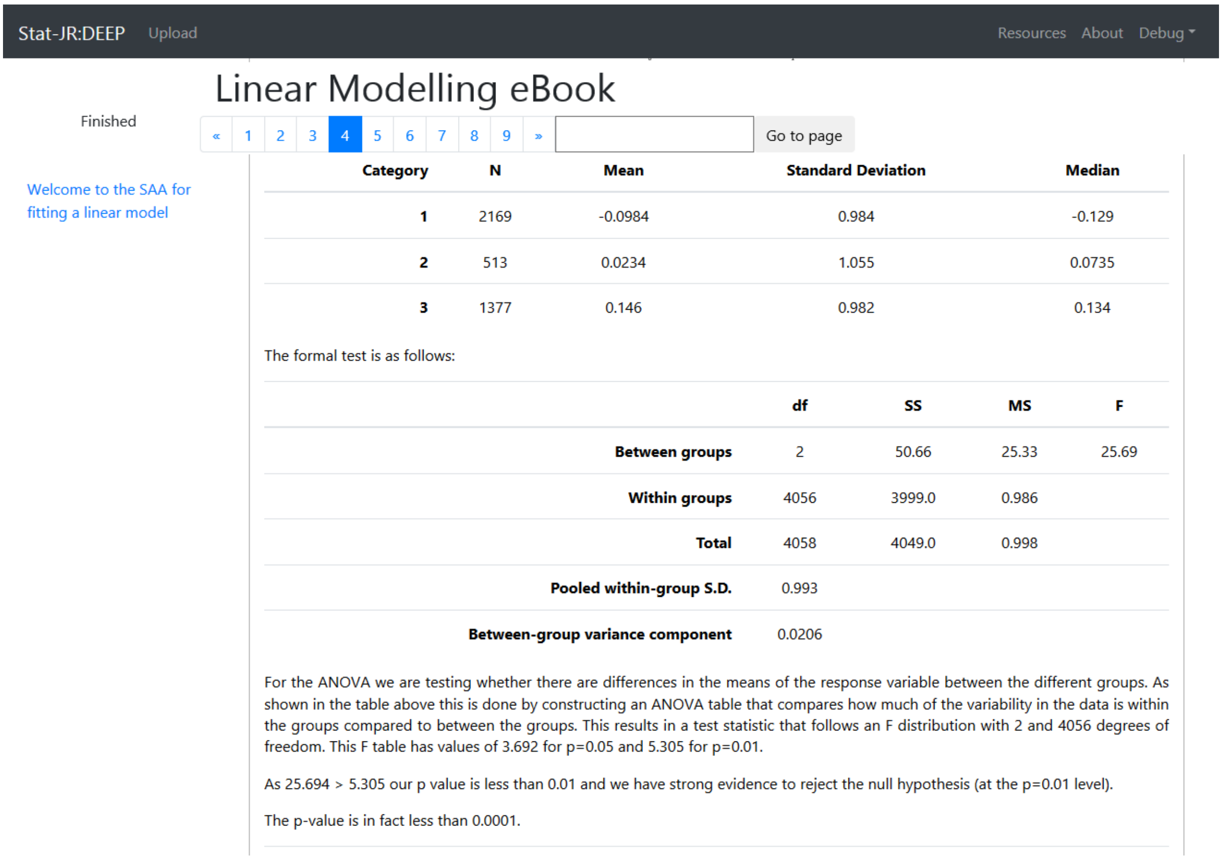
The p-value is in fact less than 0.0001.

The t test assumes that the distribution of the response in each group follows a Normal distribution. We could check this by looking at histograms of the variable in each group. If we were concerned about the normality assumption then we could instead use a Mann Whitney (MW) test.

A Mann Whitney test works simply on the order (or ranks) of the responses across the two groups. So the response variable is firstly sorted and then each value is ranked. The ranks for each group are then summed and the value that is larger is compared with what would be expected if there was no difference between the groups.

In this case the MW U statistic is 1740511 which for samples of size 1623 and 2436 corresponds to a p value of less than 0.0001.

For categorical predictors with more categories then the t test is replaced by an Analysis of Variance (ANOVA) as illustrated for *schgend*:



Linear Modelling eBook

Category	N	Mean	Standard Deviation	Median
1	2169	-0.0984	0.984	-0.129
2	513	0.0234	1.055	0.0735
3	1377	0.146	0.982	0.134

The formal test is as follows:

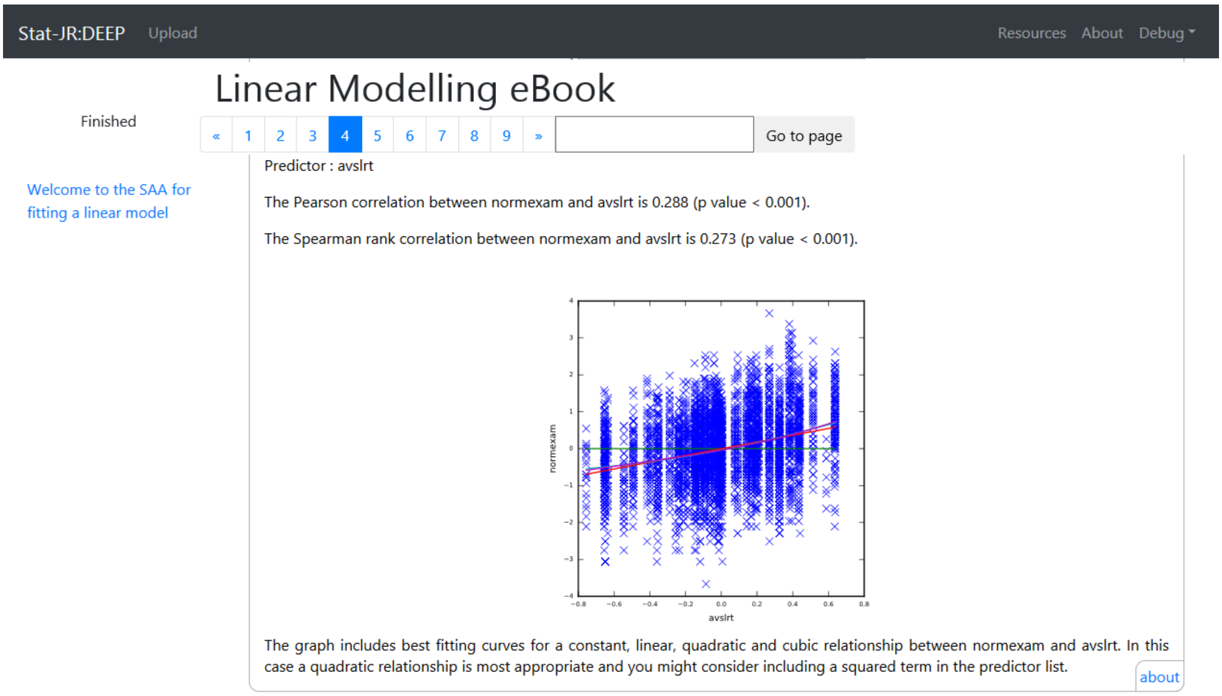
	df	SS	MS	F
Between groups	2	50.66	25.33	25.69
Within groups	4056	3999.0	0.986	
Total	4058	4049.0	0.998	
Pooled within-group S.D.	0.993			
Between-group variance component	0.0206			

For the ANOVA we are testing whether there are differences in the means of the response variable between the different groups. As shown in the table above this is done by constructing an ANOVA table that compares how much of the variability in the data is within the groups compared to between the groups. This results in a test statistic that follows an F distribution with 2 and 4056 degrees of freedom. This F table has values of 3.692 for p=0.05 and 5.305 for p=0.01.

As 25.694 > 5.305 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.

For continuous predictors as in the linear regression eBook we look at correlations with the response and fitting various polynomial curves as illustrated for *avslrt* below:



Linear Modelling eBook

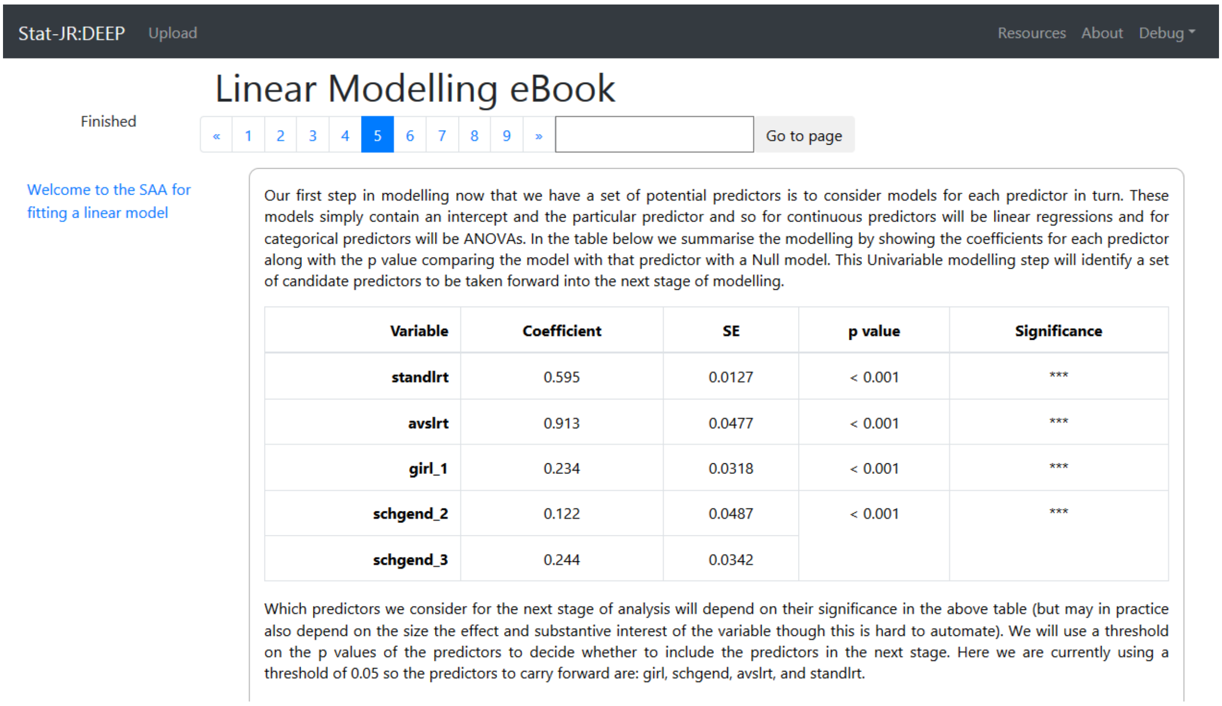
Predictor : avslrt

The Pearson correlation between normexam and avslrt is 0.288 (p value < 0.001).

The Spearman rank correlation between normexam and avslrt is 0.273 (p value < 0.001).

The graph includes best fitting curves for a constant, linear, quadratic and cubic relationship between normexam and avslrt. In this case a quadratic relationship is most appropriate and you might consider including a squared term in the predictor list.

This completes the analysis on page 4 and next on page 5 we perform the actual univariable linear modelling. This means we fit each predictor variable on its own – which is effectively like doing the linear regression (for the continuous predictors) in the last eBook but for each predictor in turn. On page 5 we summarise this with a table showing the estimates for each predictor (each category apart from the base category for categorical predictors) and their significance as shown below:



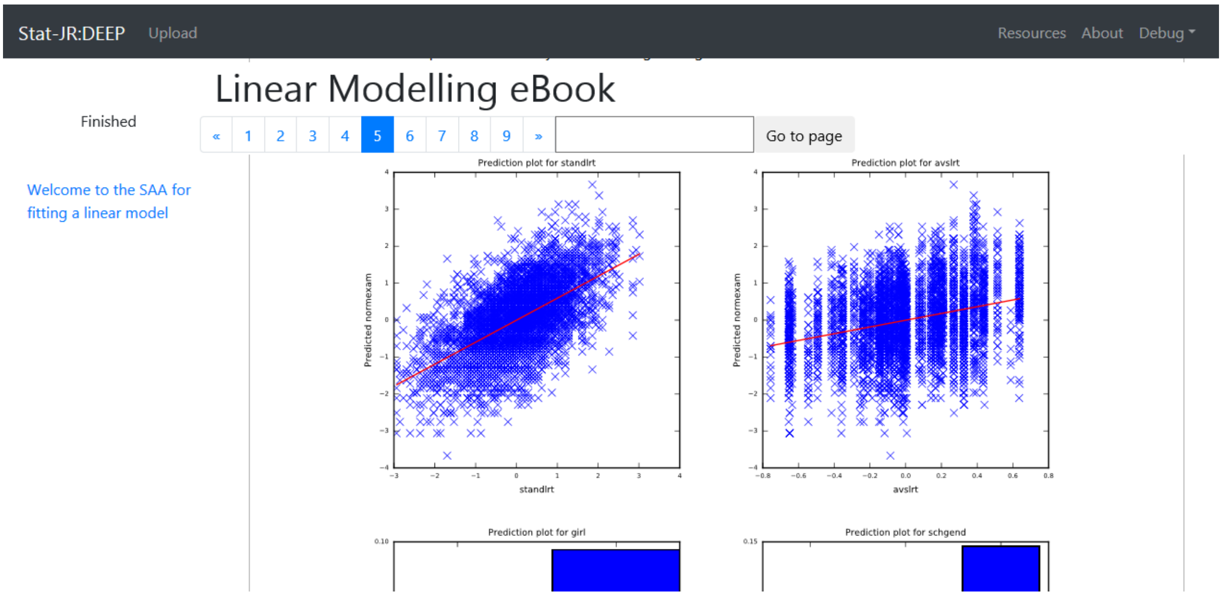
Linear Modelling eBook

Our first step in modelling now that we have a set of potential predictors is to consider models for each predictor in turn. These models simply contain an intercept and the particular predictor and so for continuous predictors will be linear regressions and for categorical predictors will be ANOVAs. In the table below we summarise the modelling by showing the coefficients for each predictor along with the p value comparing the model with that predictor with a Null model. This Univariable modelling step will identify a set of candidate predictors to be taken forward into the next stage of modelling.

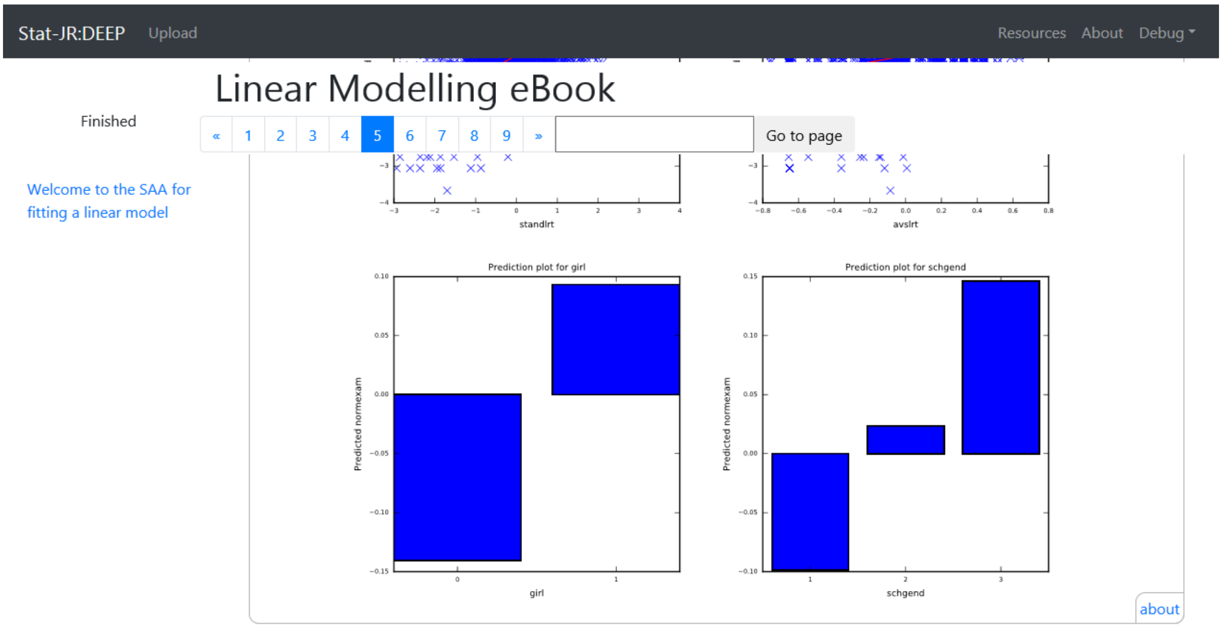
Variable	Coefficient	SE	p value	Significance
standlrt	0.595	0.0127	< 0.001	***
avslrt	0.913	0.0477	< 0.001	***
girl_1	0.234	0.0318	< 0.001	***
schgend_2	0.122	0.0487	< 0.001	***
schgend_3	0.244	0.0342		

Which predictors we consider for the next stage of analysis will depend on their significance in the above table (but may in practice also depend on the size the effect and substantive interest of the variable though this is hard to automate). We will use a threshold on the p values of the predictors to decide whether to include the predictors in the next stage. Here we are currently using a threshold of 0.05 so the predictors to carry forward are: girl, schgend, avslrt, and standlrt.

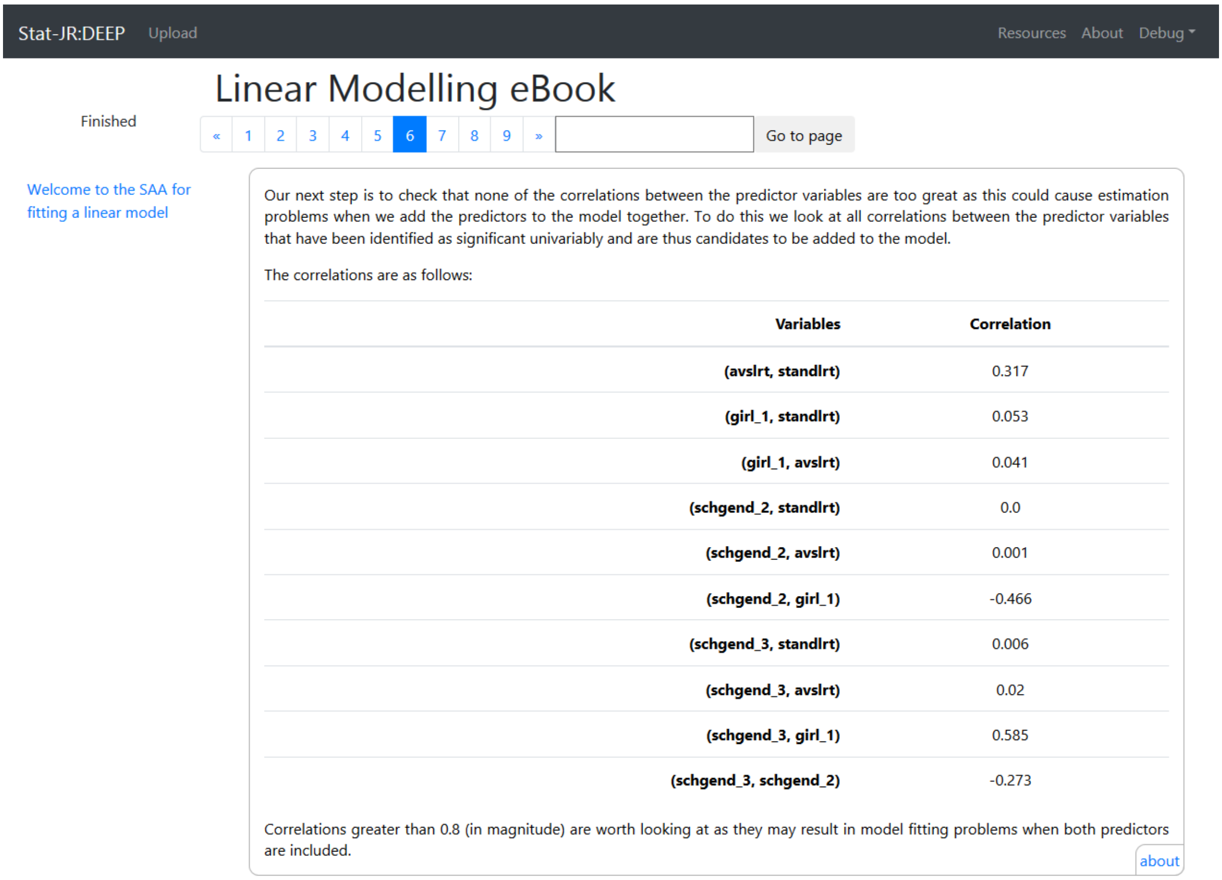
We then show the predictions that go with these univariable models – for continuous predictors as line graphs as illustrated for *standlrt* and *avslrt*:



For categorical predictors the predictions are shown as bar graphs as illustrated for *girl* and *schgend* below:



We then move on to page 6 where we test the relationships between the various predictors via correlations to try and identify if there are any problems due to collinearity etc.



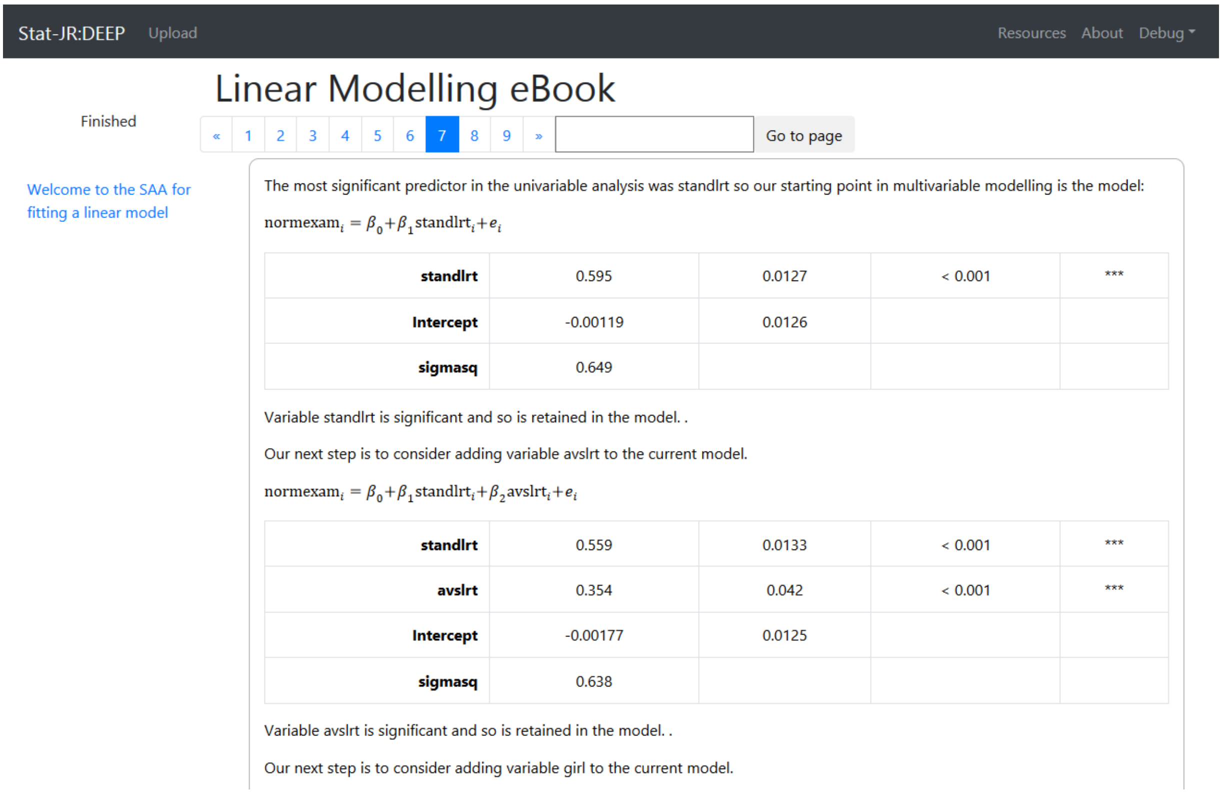
Our next step is to check that none of the correlations between the predictor variables are too great as this could cause estimation problems when we add the predictors to the model together. To do this we look at all correlations between the predictor variables that have been identified as significant univariably and are thus candidates to be added to the model.

The correlations are as follows:

Variables	Correlation
(avslrt, standlrt)	0.317
(girl_1, standlrt)	0.053
(girl_1, avslrt)	0.041
(schgend_2, standlrt)	0.0
(schgend_2, avslrt)	0.001
(schgend_2, girl_1)	-0.466
(schgend_3, standlrt)	0.006
(schgend_3, avslrt)	0.02
(schgend_3, girl_1)	0.585
(schgend_3, schgend_2)	-0.273

Correlations greater than 0.8 (in magnitude) are worth looking at as they may result in model fitting problems when both predictors are included.

Here we see no huge correlations and so things should be OK so we can move on to page 7 where we consider model building and creating a ‘best’ model. Here we build on the univariable models that we looked at on page 5 and use a form of automated forward selection (forward pass) to add variables in turn based on their significance in univariable fitting:



Linear Modelling eBook

The most significant predictor in the univariable analysis was standlrt so our starting point in multivariable modelling is the model:

$$\text{normexam}_i = \beta_0 + \beta_1 \text{standlrt}_i + e_i$$

standlrt	0.595	0.0127	< 0.001	***
Intercept	-0.00119	0.0126		
sigmasq	0.649			

Variable standlrt is significant and so is retained in the model. .

Our next step is to consider adding variable avslrt to the current model.

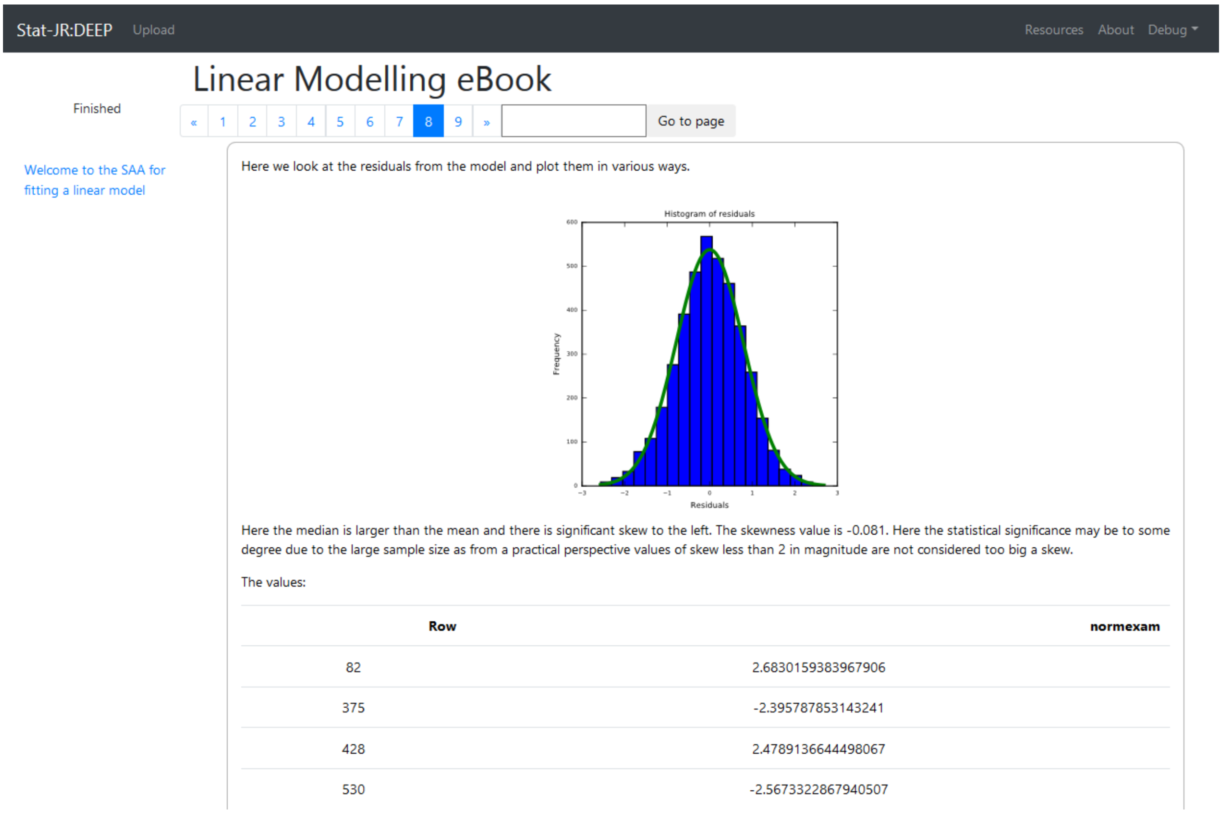
$$\text{normexam}_i = \beta_0 + \beta_1 \text{standlrt}_i + \beta_2 \text{avslrt}_i + e_i$$

standlrt	0.559	0.0133	< 0.001	***
avslrt	0.354	0.042	< 0.001	***
Intercept	-0.00177	0.0125		
sigmasq	0.638			

Variable avslrt is significant and so is retained in the model. .

Our next step is to consider adding variable girl to the current model.

Each variable is added in turn and kept if it is significant (at the 0.05 level). In our case all four predictors are significant and thus appear in the final model. We then do some post-processing from this final model so as with the linear regression eBook we next have some residual plots on page 8:



Linear Modelling eBook

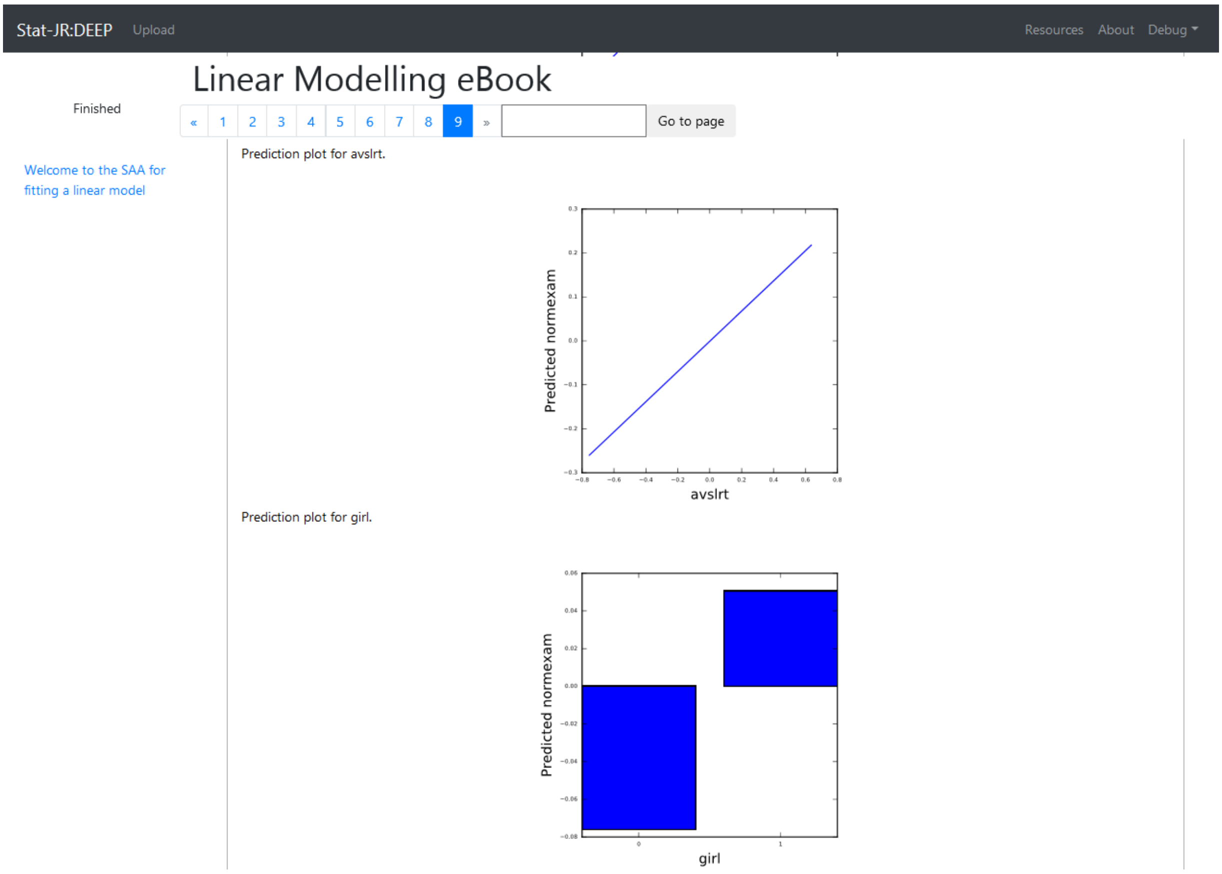
Here we look at the residuals from the model and plot them in various ways.

Here the median is larger than the mean and there is significant skew to the left. The skewness value is -0.081. 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 in magnitude are not considered too big a skew.

The values:

Row	normexam
82	2.6830159383967906
375	-2.395787853143241
428	2.4789136644498067
530	-2.5673322867940507

We finish our eBook with some prediction plots based on this final model. Predictions are more challenging when one has lots of variables. For now we have simply plotted the predicted values against each predictor in isolation assuming that all other predictors are held at their mean value. So below you will see the predicted line for *avslrt* and bar chart for *girl*:



Linear Modelling eBook

Prediction plot for avslrt.

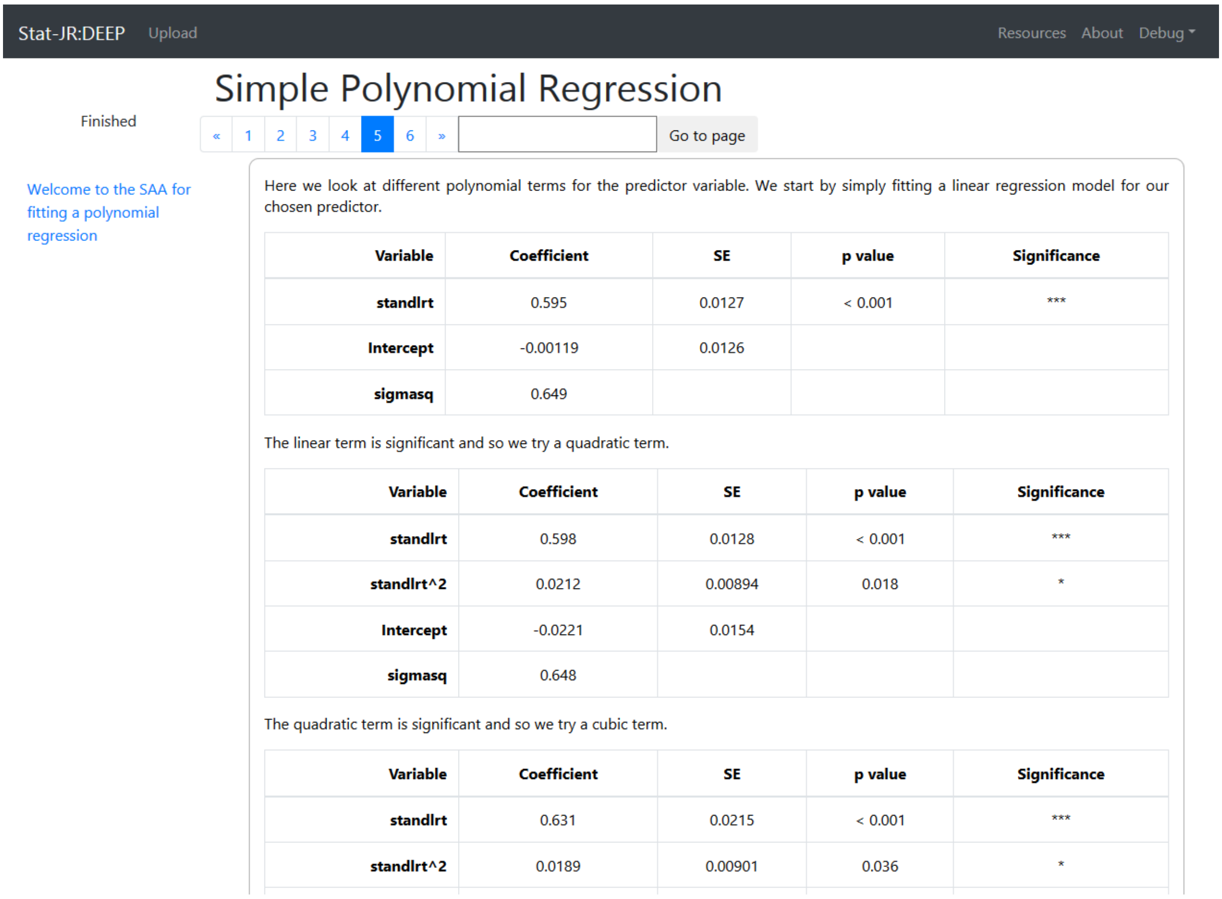
Prediction plot for girl.

4.2 – Other features of linear models - non-linear (polynomial) effects of predictors and interactions

It is possible to already use the eBooks we have thus far developed to include non-linear effects and/or interactions simply by manually including such terms in the list of predictors to be consider. This is however somewhat suboptimal as the SAA does not know that such terms are related and so will for example plot separate lines against a predictor and its squared term if both are included in the predictor list. To rectify this we look at including the possibility of adapting SAAs to include such terms.

We will firstly look at updating the simple linear regression SAA to included polynomial terms. This eBook we have called *linregpol.zip*

This eBook asks the same questions as the linear regression eBook but now in addition looks at model fitting of various polynomial orders (up to cubic) of the predictor. Here we see the output for the *tutorial* dataset looking at the effect of *standlrt* on *normexam* with the model building occurring on page 5.



Simple Polynomial Regression

Here we look at different polynomial terms for the predictor variable. We start by simply fitting a linear regression model for our chosen predictor.

Variable	Coefficient	SE	p value	Significance
standlrt	0.595	0.0127	< 0.001	***
Intercept	-0.00119	0.0126		
sigmasq	0.649			

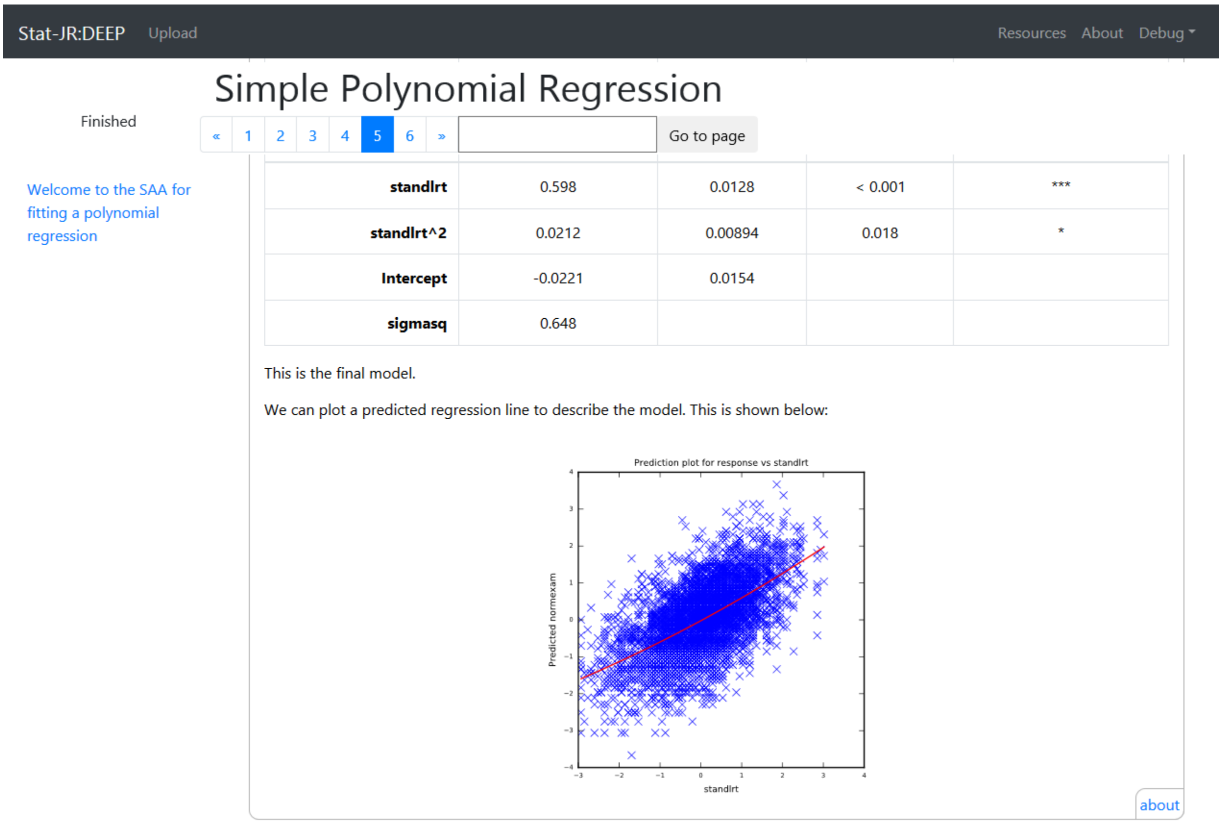
The linear term is significant and so we try a quadratic term.

Variable	Coefficient	SE	p value	Significance
standlrt	0.598	0.0128	< 0.001	***
standlrt^2	0.0212	0.00894	0.018	*
Intercept	-0.0221	0.0154		
sigmasq	0.648			

The quadratic term is significant and so we try a cubic term.

Variable	Coefficient	SE	p value	Significance
standlrt	0.631	0.0215	< 0.001	***
standlrt^2	0.0189	0.00901	0.036	*

Here we see (if you scroll further down) that the model is built up to a cubic but that the cubic term is not significant and so we revert to a quadratic relationship. We then plot this prediction as shown below:



Simple Polynomial Regression

standlrt	0.598	0.0128	< 0.001	***
standlrt^2	0.0212	0.00894	0.018	*
Intercept	-0.0221	0.0154		
sigmasq	0.648			

This is the final model.

We can plot a predicted regression line to describe the model. This is shown below:

Here this quadratic is not that far from linear but is a significant effect.

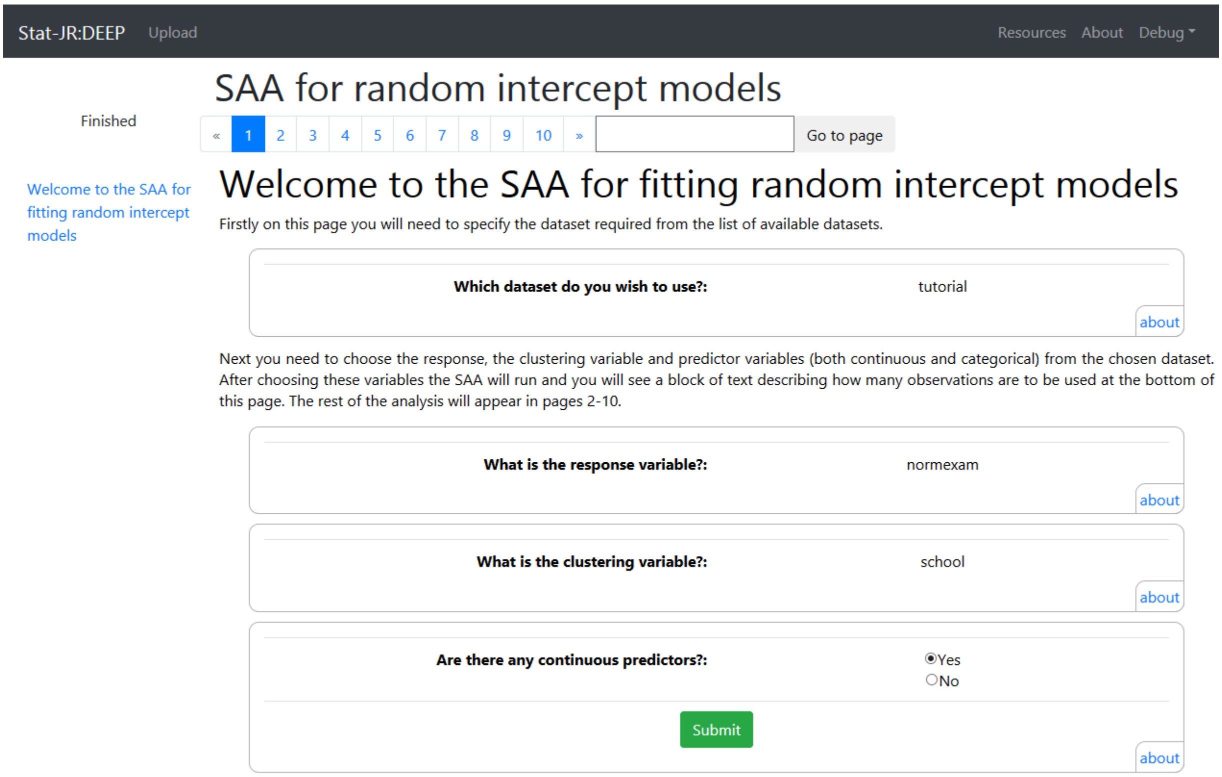
Another extension that we can apply within the linear modelling family is to include interaction terms. These are terms that are constructed from a pair (or a group) of predictor variables so that the effect of one predictor can vary with values of the other predictor. In other words the two predictor variables interact with each other in how they influence the response variable. We will not discuss interactions here but will introduce them in the later sections when we look at multilevel models so as not to repeat ourselves as they impact in a similar way in multilevel models as in linear models. We next move onto multilevel models and begin by looking at the simplest form – the two-level random intercept model.

Chapter 5 – Random intercept models

5.1 Extending linear models to incorporate random effects

For linear modelling there is an implicit assumption that the data come from a simple random sample or similar structure and that there are no explicit clustering variables. In practice much data is clustered and taking account of this clustering is important. Models that take account of this clustering are called multilevel models and we will begin in this chapter by first looking at an SAA that fits models that account for clustering in the response only by fitting random intercept models. This SAA eBook is in the file *randint.zip.*

Here we use the *tutorial* dataset with response being *normexam* and clustering variable *school*:



SAA for random intercept models

Welcome to the SAA for fitting random intercept models

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?: tutorial

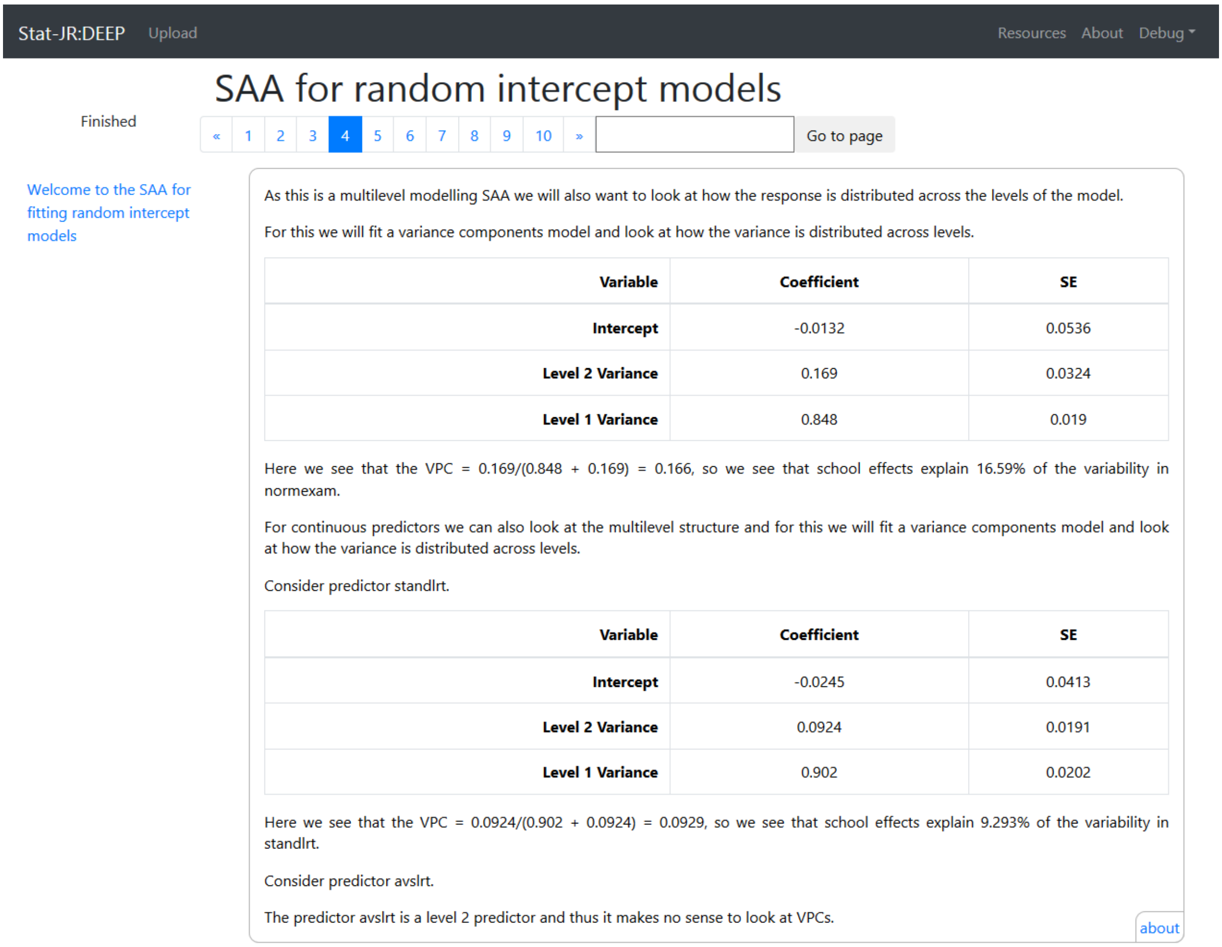
Next you need to choose the response, the clustering variable and predictor variables (both continuous and categorical) from the chosen dataset. After choosing these variables the SAA will run and you will see a block of text describing how many observations are to be used at the bottom of this page. The rest of the analysis will appear in pages 2-10.

What is the response variable?: normexam

What is the clustering variable?: school

Are there any continuous predictors?: ●Yes ○No

The SAA will aim to build up a best model for the data based on a set of candidate predictor variables and so we will try 5 predictor variables – 2 continuous (*standlrt*, *avslrt*) and 3 categorical predictors (*girl*, *schgend, vrband*). Once again we need on page 2 to say that we don’t want to log transform the response and now wait rather a while for the whole eBook to run. Much of the exploratory data analysis is borrowed from the linear modelling SAA but as we have clustering an additional exploratory step (on page 4 of the eBook) is to look at how the variation in the response is partitioned:



SAA for random intercept models

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

For this we will fit a variance components model and look at how the variance is distributed across levels.

Variable	Coefficient	SE
Intercept	-0.0132	0.0536
Level 2 Variance	0.169	0.0324
Level 1 Variance	0.848	0.019

Here we see that the VPC = 0.169/(0.848 + 0.169) = 0.166, so we see that school effects explain 16.59% of the variability in normexam.

For continuous predictors we can also look at the multilevel structure and for this we will fit a variance components model and look at how the variance is distributed across levels.

Consider predictor standlrt.

Variable	Coefficient	SE
Intercept	-0.0245	0.0413
Level 2 Variance	0.0924	0.0191
Level 1 Variance	0.902	0.0202

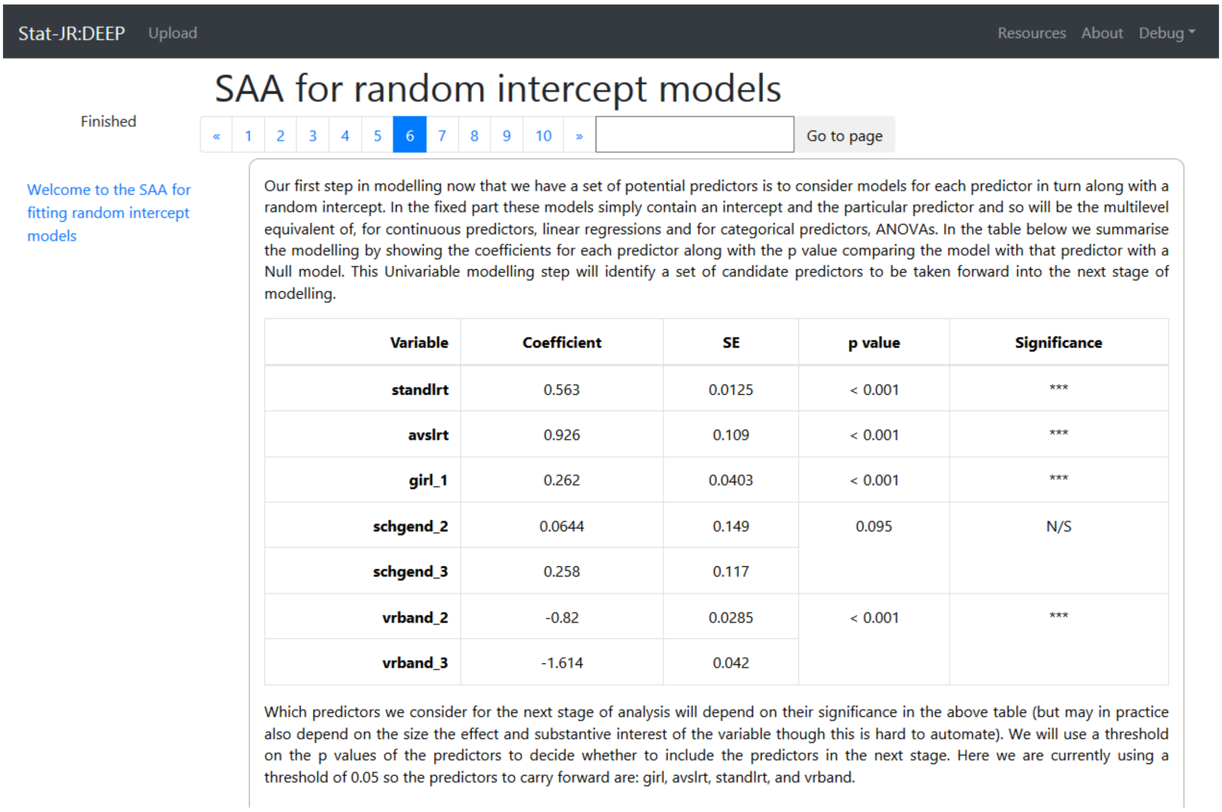
Here we see that the VPC = 0.0924/(0.902 + 0.0924) = 0.0929, so we see that school effects explain 9.293% of the variability in standlrt.

Consider predictor avslrt.

The predictor avslrt is a level 2 predictor and thus it makes no sense to look at VPCs.

Here we see that roughly 16.5% of the response variability is between schools with the remainder being within schools. We can also (as done here) repeat this calculation for the predictor variables and we see that 9% of the variability in the *standlrt* variable is between schools whilst *avslrt* is the school average scores of LRT and so partitioning doesn't make sense given the predictor is constant for the school and so the SAA has correctly identified this as a level 2 predictor.

The next change for the multilevel models is on page 6 where we now fit univariable models as we did for the linear model, but in this case these are random intercept models and include school random effects as well as one predictor at a time thus:



SAA for random intercept models

Finished

Welcome to the SAA for fitting random intercept models

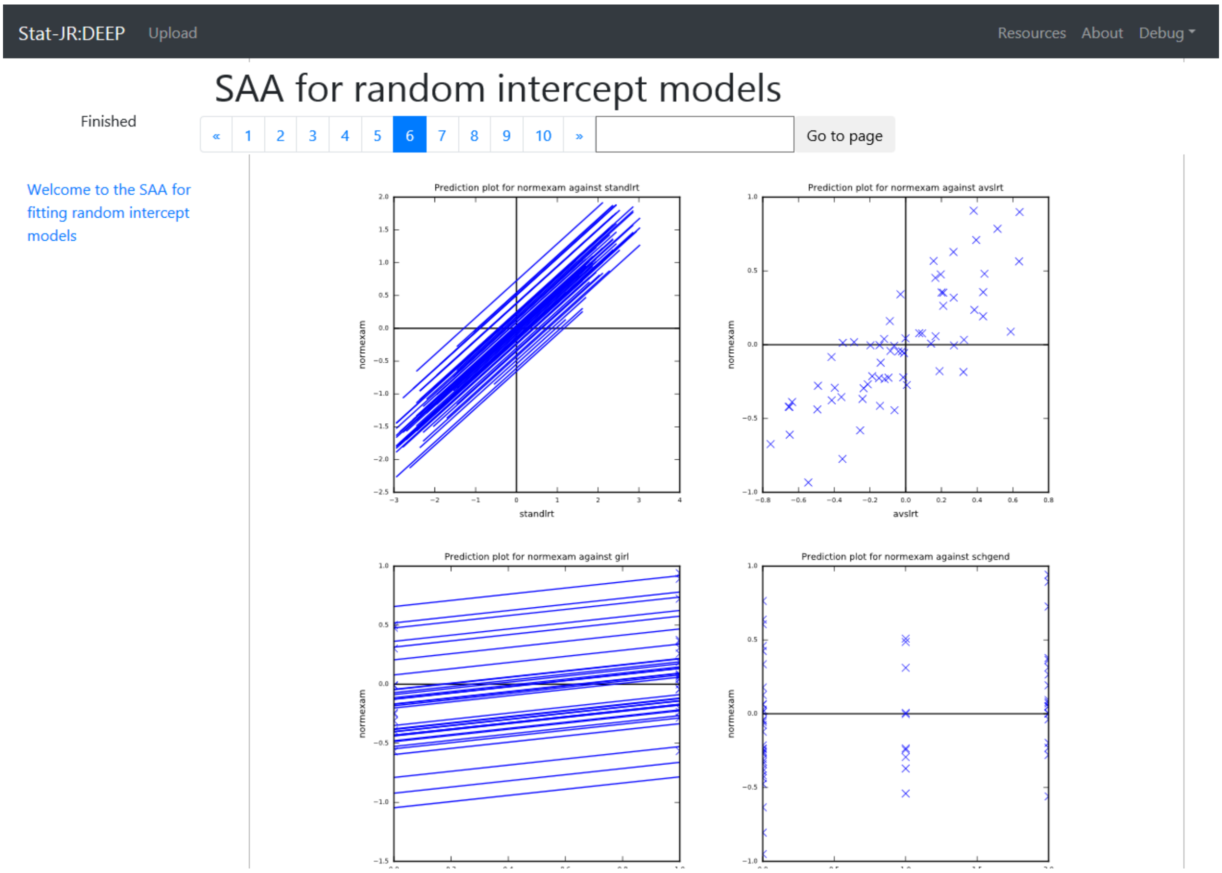
Our first step in modelling now that we have a set of potential predictors is to consider models for each predictor in turn along with a random intercept. In the fixed part these models simply contain an intercept and the particular predictor and so will be the multilevel equivalent of, for continuous predictors, linear regressions and for categorical predictors, ANOVAs. In the table below we summarise the modelling by showing the coefficients for each predictor along with the p value comparing the model with that predictor with a Null model. This Univariable modelling step will identify a set of candidate predictors to be taken forward into the next stage of modelling.

Variable	Coefficient	SE	p value	Significance
standlrt	0.563	0.0125	< 0.001	***
avslrt	0.926	0.109	< 0.001	***
girl_1	0.262	0.0403	< 0.001	***
schgend_2	0.0644	0.149	0.095	N/S
schgend_3	0.258	0.117		
vrband_2	-0.82	0.0285	< 0.001	***
vrband_3	-1.614	0.042		

Which predictors we consider for the next stage of analysis will depend on their significance in the above table (but may in practice also depend on the size the effect and substantive interest of the variable though this is hard to automate). We will use a threshold on the p values of the predictors to decide whether to include the predictors in the next stage. Here we are currently using a threshold of 0.05 so the predictors to carry forward are: girl, avslrt, standlrt, and vrband.

Here we see that school gender does not have a significant effect when fitted in this multilevel model – which is different to what we would find if we didn't include school random effects.

Page 6 also includes prediction plots for the various predictors as illustrated below for some of the predictors.


Stat-JR:DEEP
Upload
Resources
About
Debug
« 1 2 3 4 5 6 7 8 9 10 »
Go to page
Prediction plot for normexam against standlrt
normexam
standlrt
Prediction plot for normexam against avslrt
normexam
avslrt
Prediction plot for normexam against girl
normexam
Prediction plot for normexam against schgend
normexam

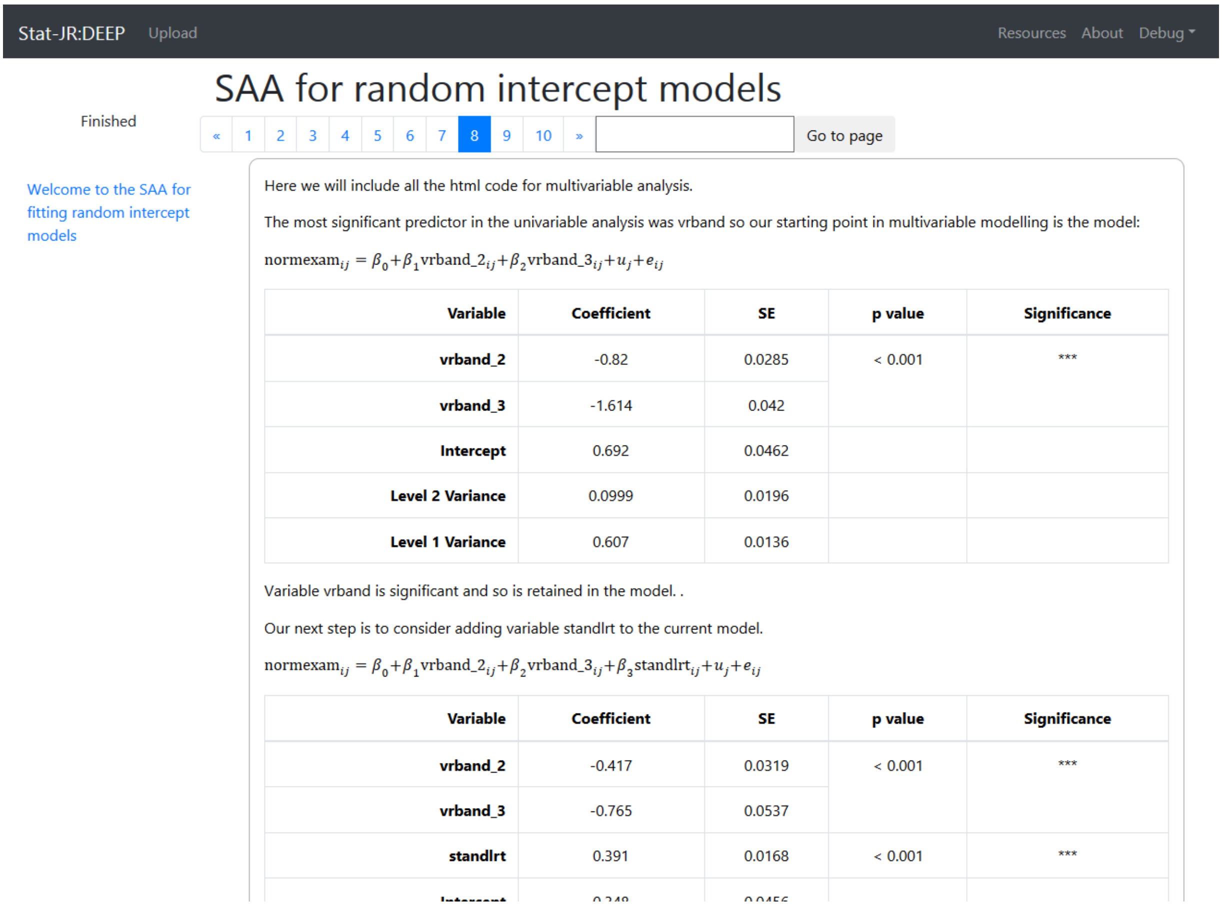
SAA for random intercept models

Finished

Welcome to the SAA for fitting random intercept models

Here we first see parallel lines for *standlrt*, one for each of the 65 schools in the dataset. There are then similar plots for the other predictors although school level predictors will only have points rather than lines.

On page 8 we perform the equivalent of the forward pass method for the random intercept models:



SAA for random intercept models

Here we will include all the html code for multivariable analysis.

The most significant predictor in the univariable analysis was vrband so our starting point in multivariable modelling is the model:

$\text{normexam}_{ij} = \beta_0 + \beta_1 \text{vrband\_2}_{ij} + \beta_2 \text{vrband\_3}_{ij} + u_j + e_{ij}$

Variable	Coefficient	SE	p value	Significance
vrband_2	-0.82	0.0285	< 0.001	***
vrband_3	-1.614	0.042		
Intercept	0.692	0.0462		
Level 2 Variance	0.0999	0.0196		
Level 1 Variance	0.607	0.0136		

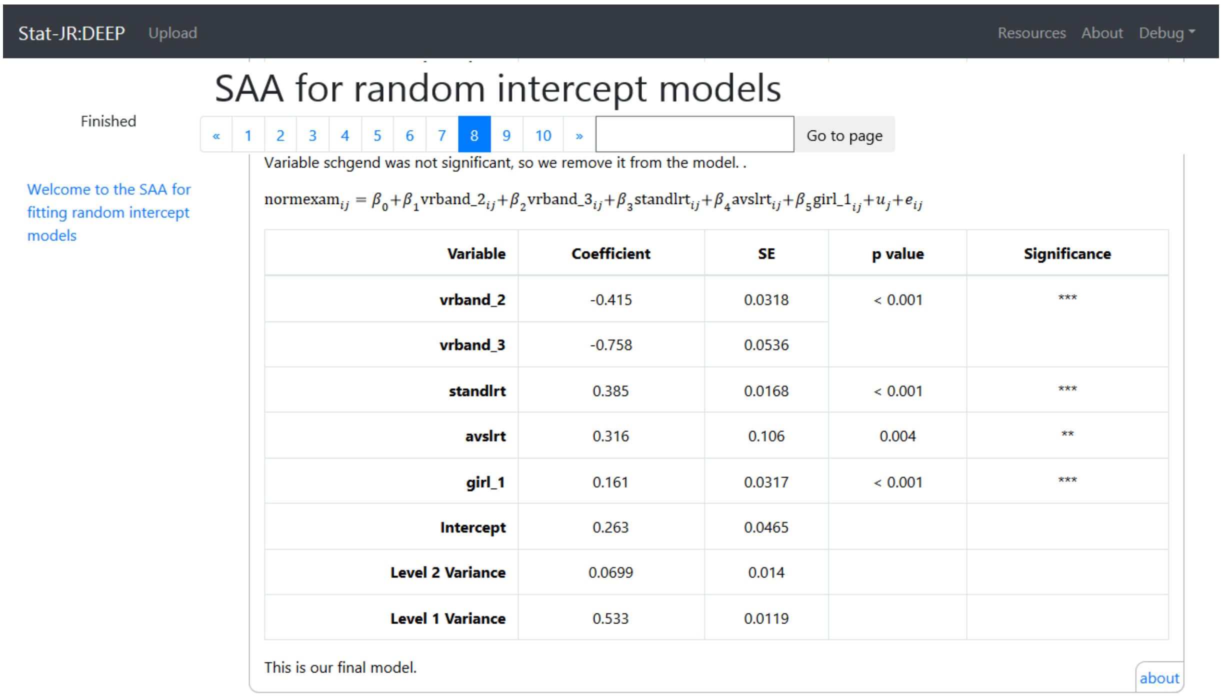
Variable vrband is significant and so is retained in the model. .

Our next step is to consider adding variable standlrt to the current model.

$\text{normexam}_{ij} = \beta_0 + \beta_1 \text{vrband\_2}_{ij} + \beta_2 \text{vrband\_3}_{ij} + \beta_3 \text{standlrt}_{ij} + u_j + e_{ij}$

Variable	Coefficient	SE	p value	Significance
vrband_2	-0.417	0.0319	< 0.001	***
vrband_3	-0.765	0.0537		
standlrt	0.391	0.0168	< 0.001	***
Intercept	0.248	0.0456		

This continues until we end up with a best model as shown below:



SAA for random intercept models

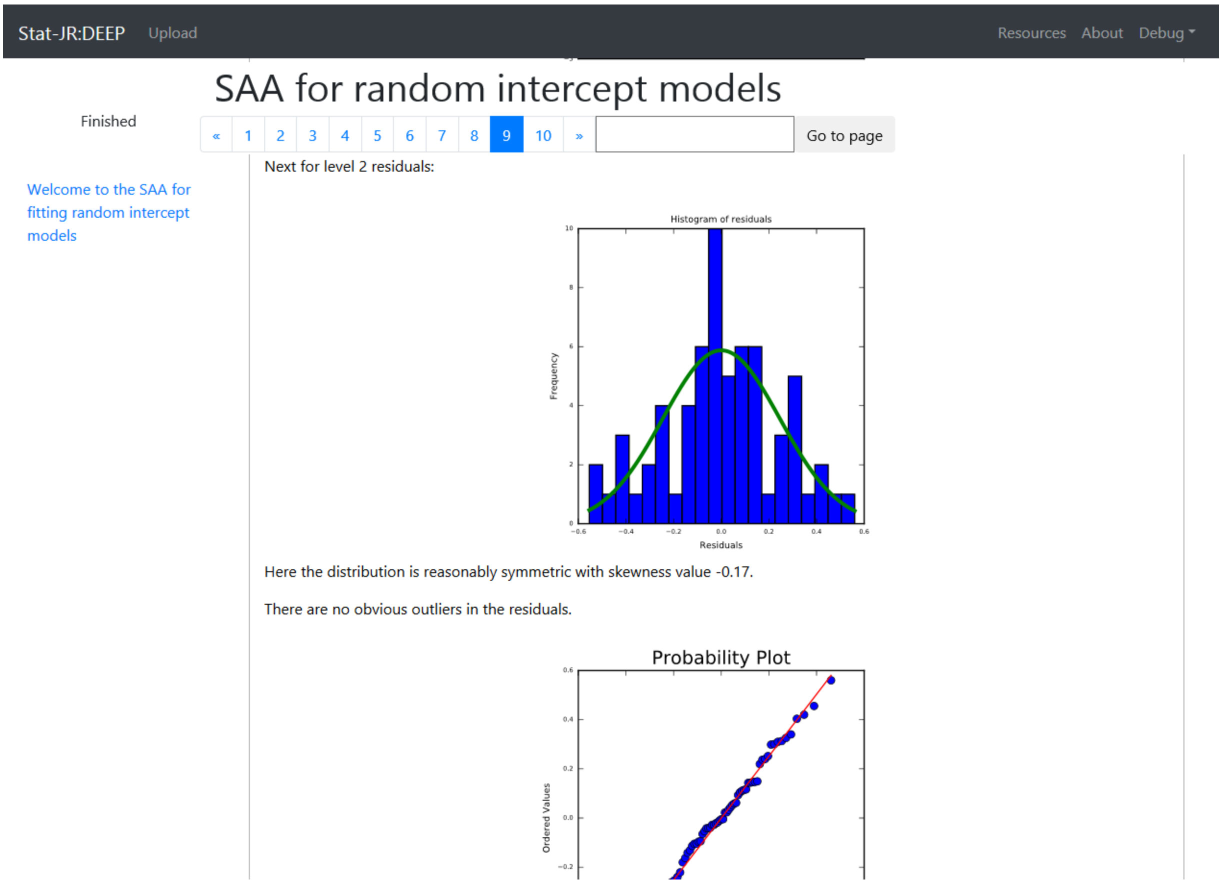
Variable schgend was not significant, so we remove it from the model. .

$\text{normexam}_{ij} = \beta_0 + \beta_1 \text{vrband\_2}_{ij} + \beta_2 \text{vrband\_3}_{ij} + \beta_3 \text{standlrt}_{ij} + \beta_4 \text{avslrt}_{ij} + \beta_5 \text{girl\_1}_{ij} + u_j + e_{ij}$

Variable	Coefficient	SE	p value	Significance
vrband_2	-0.415	0.0318	< 0.001	***
vrband_3	-0.758	0.0536		
standlrt	0.385	0.0168	< 0.001	***
avslrt	0.316	0.106	0.004	**
girl_1	0.161	0.0317	< 0.001	***
Intercept	0.263	0.0465		
Level 2 Variance	0.0699	0.014		
Level 1 Variance	0.533	0.0119		

This is our final model.

The best model contains three of the four predictors with school gender not included. On page 9 we look at residuals for this final model and here we have the addition of residuals at level 2 as well as at level 1 as shown below:



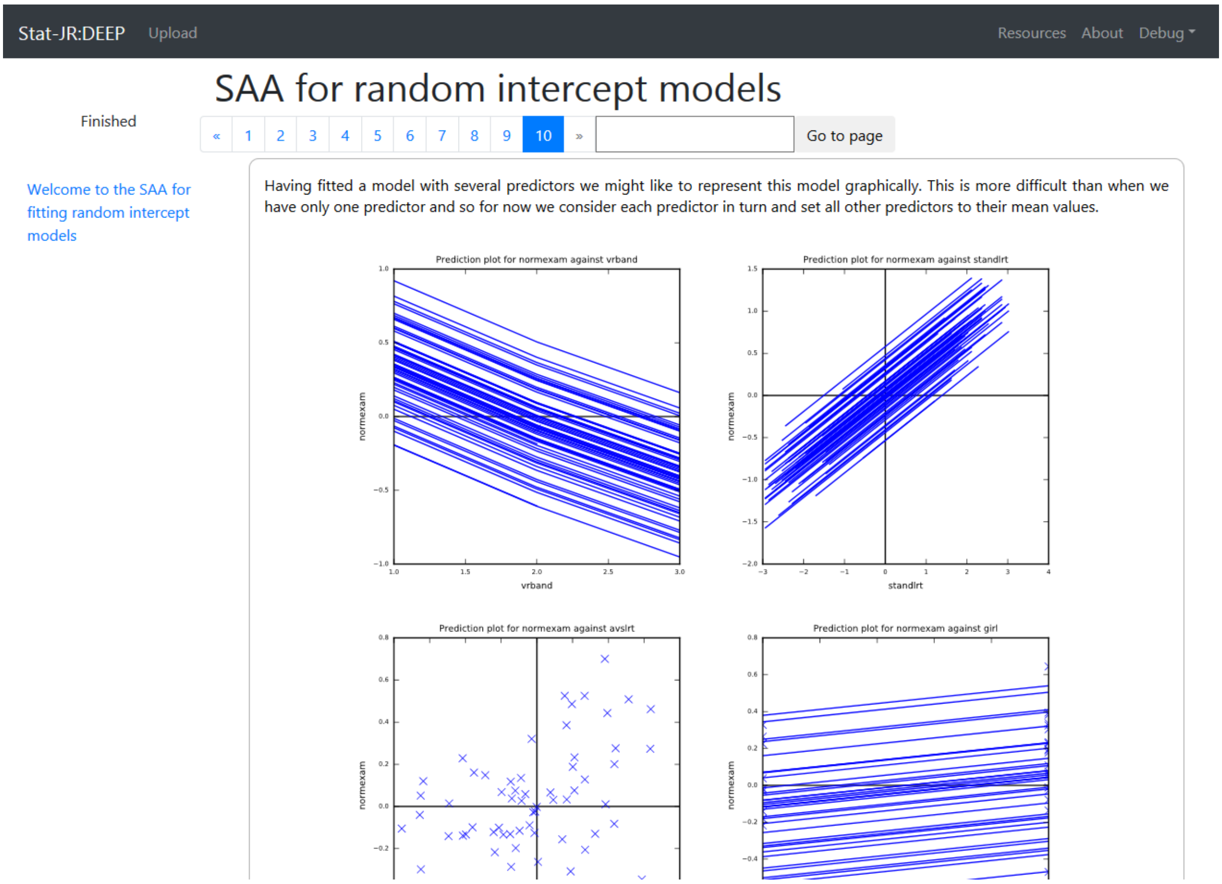
SAA for random intercept models

Next for level 2 residuals:

Here the distribution is reasonably symmetric with skewness value -0.17.

There are no obvious outliers in the residuals.

Finally we create predictions from this final model for each predictor whilst holding the others at their mean values as indicated below:



Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 9 10 »
Go to page
Welcome to the SAA for fitting random intercept models
Prediction plot for normexam against vrband
normexam
vrband
Prediction plot for normexam against standlrt
normexam
standlrt
Prediction plot for normexam against avslrt
normexam
Prediction plot for normexam against girl
normexam

SAA for random intercept models

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

5.2 - The Combined SAA

Our approach thus far has been to consider each model family and extension in turn and construct a new SAA that will work with that family. This approach allows us to tailor the SAA to precisely that family and require minimal user inputs but the disadvantages are that in creating a whole suite of SAAs we have to support them all and also the onus is on the user to select the SAA they require. To circumvent these issues we have constructed a "Combined SAA" which fits most of the models that we cover in this manual and represents all the features that we have integrated together so far in our work on SAAs to date. Some features like missing data are not yet incorporated in this SAA and so we leave discussion of these to the end.

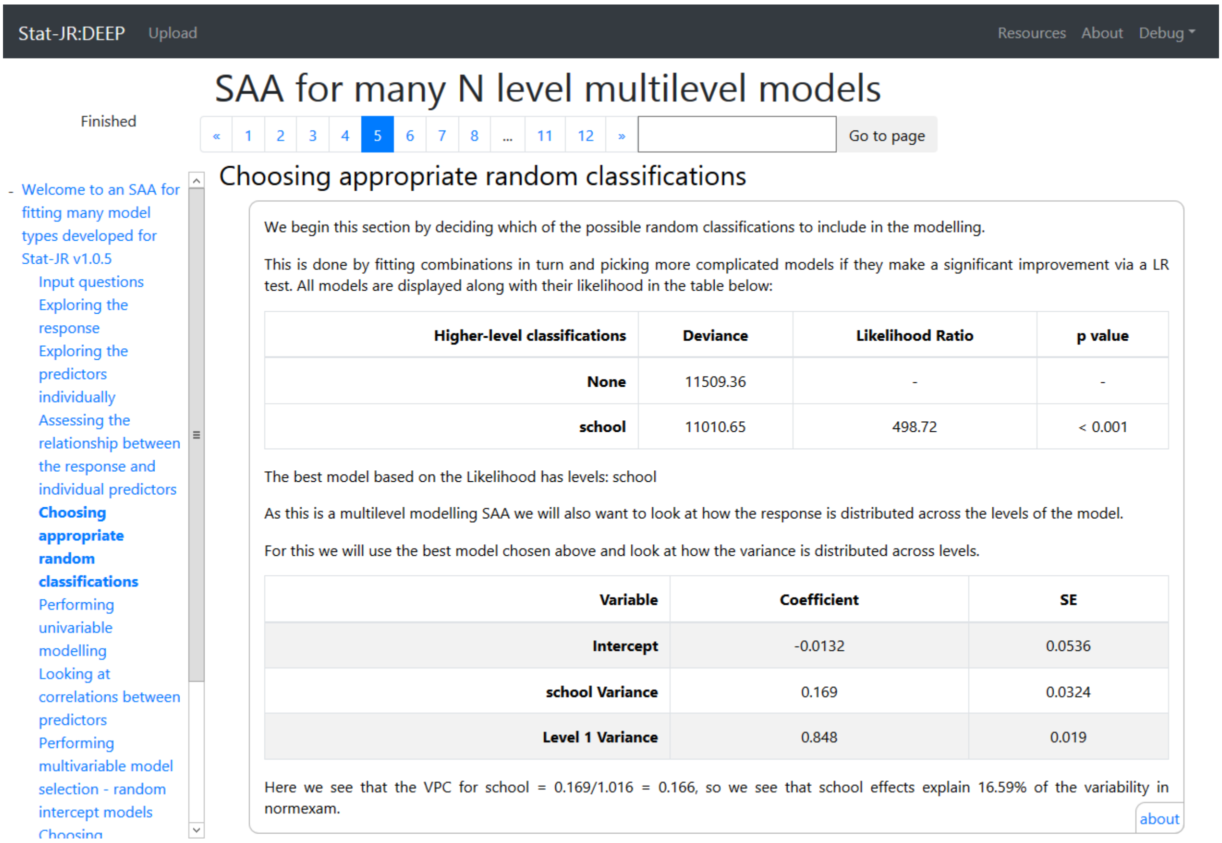
To give a basic idea of what the Combined SAA can handle it will fit models with a choice of Normal, Binomial or Poisson responses. It will fit the models with a choice of classical (IGLS) or MCMC estimation and in terms of multilevel modelling can handle multiple levels of clustering using MCMC for cross-classified models or either approach for nested models. It allows random intercepts and random slopes, and will also fit (pairwise) interactions between predictors including interactions with themselves to give quadratic terms. It also offers several approaches for selecting the best model.

5.3 - Random intercepts model using the Combined SAA

To illustrate the Combined SAA in action we will attempt to roughly replicate the analysis we have just performed using it. To do this we need to load up the DEEP interface of Stat-JR and find the eBook which has the name *combined.zip*. Given the SAA offers far more models than the random intercept eBook there will be several more inputs to include.

To fit the models we have just seen for the random intercepts eBook choose *tutorial* for the dataset, *IGLS* for the estimation method, *normexam* for the response variable, *Normal* for the distribution and *school* only from the possible classifications. The template allows us to force some predictors into the model via the next 2 questions so say *No* for both questions. Then say *Yes* to include continuous predictors and select (*standlrt*, *avslrt*) and *Yes* for categorical predictors and this time select (*girl*, *schgend, vrband*). We will use the selection type *Forward pass* to be in line with the random intercepts eBook and say *No* to *including random slopes* and *interactions*. We are then offered a choice of the *Likelihood Ratio* test or the *Wald* test for comparing models. Here we will choose *Likelihood Ratio* but note that *Wald* is quicker as it can work out p values direct from a particular model whilst the likelihood ratio test needs to compare pairs of models. Once again we are required to say *No* to logging the response variable on page 2 after which the SAA will whirl away fitting many models and eventually finish (as indicated by the *Running Workflow* timer being replaced by *Finished* in the top left of the screen). This may take a few minutes.

Many of the pages will look similar to those in the random intercepts eBook although a couple of pages are swapped in order (pages 4 and 5). Page 5 is the equivalent of page 4 in the earlier eBook and is shown below:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing
about

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting combinations in turn and picking more complicated models if they make a significant improvement via a LR test. All models are displayed along with their likelihood in the table below:

Higher-level classifications	Deviance	Likelihood Ratio	p value
None	11509.36	-	-
school	11010.65	498.72	< 0.001

The best model based on the Likelihood has levels: school

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

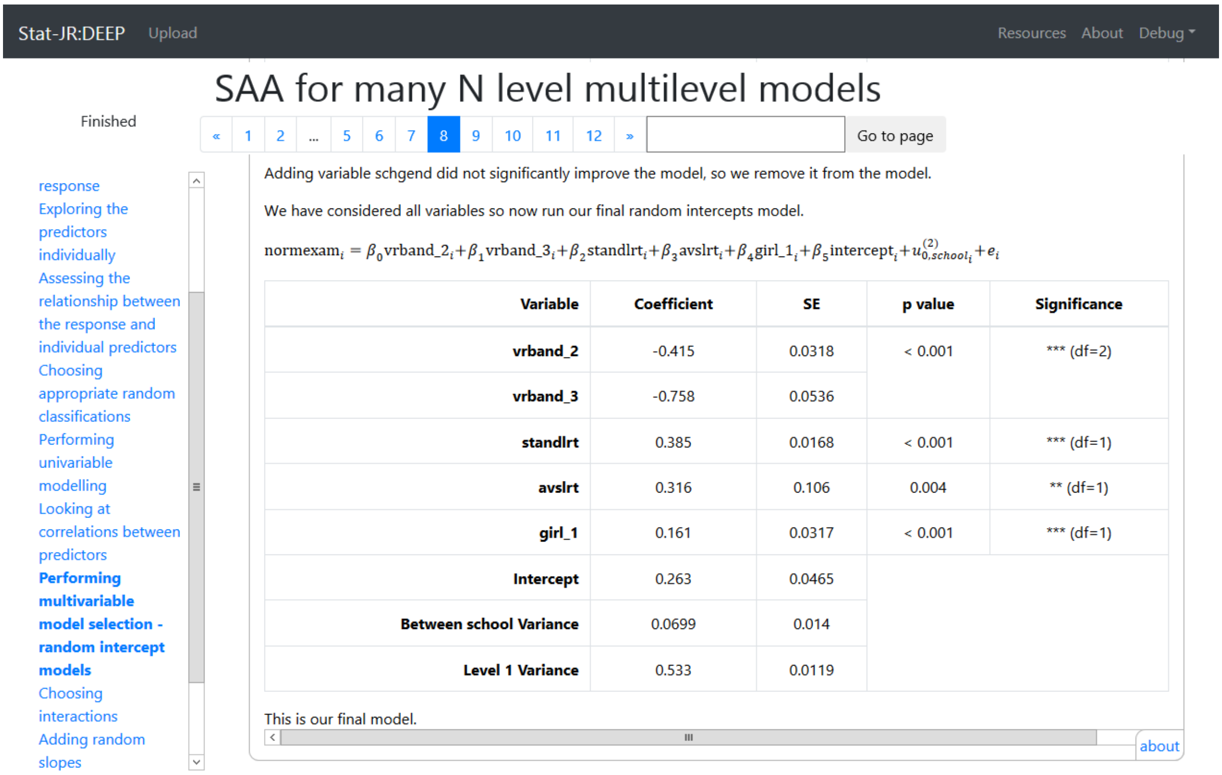
For this we will use the best model chosen above and look at how the variance is distributed across levels.

Variable	Coefficient	SE
Intercept	-0.0132	0.0536
school Variance	0.169	0.0324
Level 1 Variance	0.848	0.019

Here we see that the VPC for school = 0.169/1.016 = 0.166, so we see that school effects explain 16.59% of the variability in normexam.

Here we see that the Combined SAA doesn’t take it for granted that we should include the school level of clustering but instead tests the model against a simpler model with no random effects. We find that indeed school random effects are important, and the lower table is identical to that from the earlier eBook (though note that the Combined SAA doesn’t in addition show VPCs for predictors as well)

As expected the Combined SAA comes up with the same best model on page 8 as we saw with the random intercept eBook:


Stat-JR:DEEP
Upload
Resources
About
Debug
SAA for many N level multilevel models
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
about

Adding variable schgend did not significantly improve the model, so we remove it from the model.

We have considered all variables so now run our final random intercepts model.

$$\text{normexam}_i = \beta_0\text{vrband\_2}_i + \beta_1\text{vrband\_3}_i + \beta_2\text{standlrt}_i + \beta_3\text{avslrt}_i + \beta_4\text{girl\_1}_i + \beta_5\text{intercept}_i + u^{(2)}_{0,school_i} + e_i$$

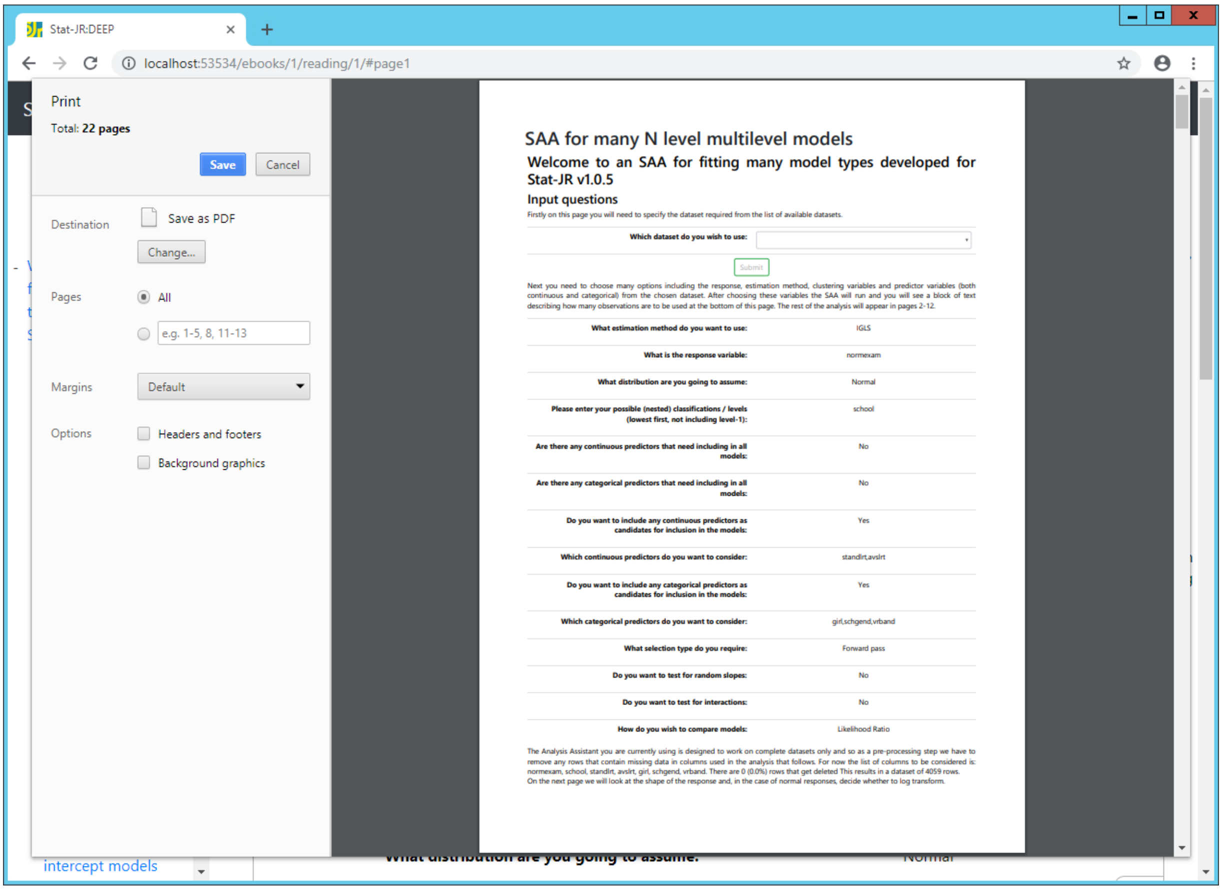
Variable	Coefficient	SE	p value	Significance
vrband_2	-0.415	0.0318	< 0.001	*** (df=2)
vrband_3	-0.758	0.0536		
standlrt	0.385	0.0168	< 0.001	*** (df=1)
avslrt	0.316	0.106	0.004	** (df=1)
girl_1	0.161	0.0317	< 0.001	*** (df=1)
Intercept	0.263	0.0465		
Between school Variance	0.0699	0.014		
Level 1 Variance	0.533	0.0119		

This is our final model.

The residuals and predictions (now on pages 11 and 12) should also correspond to those from the earlier eBook. This illustrates that although the Combined SAA can do several more advanced features it can also, via appropriate inputs, replicate the random intercepts SAA.

5.4 - Saving SAAs as pdfs

Before looking at how we can extend our modelling using the Combined SAA we might wonder how best to look at the output from the eBook and indeed how to share the eBook with others. Clearly another researcher can use the eBook with the same dataset and inputs to verify that they get the same answers for the model but sometimes a hardcopy is useful. As the DEEP interface is interactive and only shows some of the content on screen at once the easiest way to save a hardcopy of the eBook is to print it to the *"Save as PDF"* printer. To do this right mouse click in the browser window and select the *Print* option. From the *Destination list* click on *Change* and select *"Save as PDF"* from the list of printers. The screen should look as follows:



Print

Total: 22 pages

Destination: Save as PDF

Pages: ● All ○ e.g. 1-5, 8, 11-13

Margins: Default

Options: □ Headers and footers □ Background graphics

SAA for many N level multilevel models

Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5

Input questions

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:

Next you need to choose many options including the response, estimation method, clustering variables and predictor variables (both continuous and categorical) 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 2-12.

What estimation method do you want to use:	IGLS
What is the response variable:	normexam
What distribution are you going to assume:	Normal
Please enter your possible (nested) classifications / levels (lowest first, not including level-1):	school
Are there any continuous predictors that need including in all models:	No
Are there any categorical predictors that need including in all models:	No
Do you want to include any continuous predictors as candidates for inclusion in the models:	Yes
Which continuous predictors do you want to consider:	standlrt,avslrt
Do you want to include any categorical predictors as candidates for inclusion in the models:	Yes
Which categorical predictors do you want to consider:	girl,schgend,vrband
What selection type do you require:	Forward pass
Do you want to test for random slopes:	No
Do you want to test for interactions:	No
How do you wish to compare models:	Likelihood Ratio

The Analysis Assistant 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: normexam, school, standlrt, avslrt, girl, schgend, vrband. There are 0 (0.0%) rows that get deleted This results in a dataset of 4059 rows.
On the next page we will look at the shape of the response and, in the case of normal responses, decide whether to log transform.

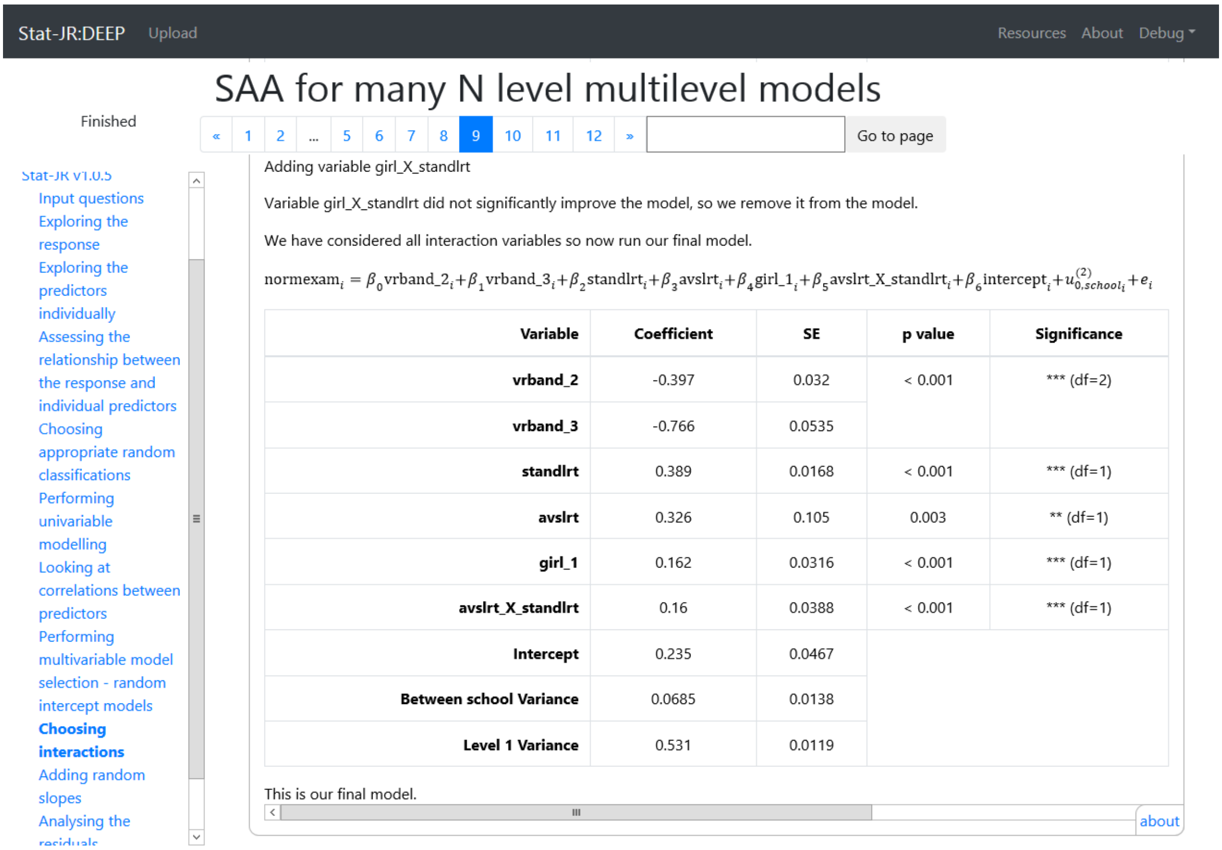
Here we see that when saved as a PDF the eBook will be a 35 page document. The document is not perfect and includes the inputs on the first page and some of the objects will not render exactly as on screen but it gives a good idea of the analysis undertaken.

Clicking on the Save button and choosing an appropriate directory and using the filename *tutrandint.pdf* will save the file. We will include all the pdfs for the eBooks in the remainder of the manual on the web with this manual.

5.5 - Adding in Interaction terms and Polynomials

When we looked at linear models we showed an additional eBook that considered the possibility of polynomial terms for the predictor in question. Polynomial terms are formed by multiplying a predictor by itself once (for a quadratic term) and repeatedly for cubic and higher order terms and including the terms created in the list of possible predictors to use. More generally when one multiplies predictors together they form interaction terms between predictors which allows one to let the effect of a predictor depend on the value of other predictors. Of course if we allow interactions this increases (often dramatically) the number of variables to be considered as predictors in a model. As a compromise in the Combined SAA we restrict our attention to pairwise interactions (between two variables) and quadratic terms. It is a straightforward task to do this by simply changing the answer to the question *Do you want interactions* to *Yes* but to do this in DEEP we have to proceed from the start of the inputs on page 1 and include the same inputs we used in section 5.4 apart from the changed question (remembering to say *No* to the logging question on page 2).

The majority of the eBook produced will be identical to that which we saw without interactions as the eBook firstly finds the best model containing only main effects on page 8. Then on page 9 this model is added to by constructing the set of all pairwise interactions (and quadratic terms) from the set of main effects and using the chosen (in this case forward pass) estimation procedure to test whether they improve the model. For this example the end of page 9 can be seen below:


Stat-JR:DEEP
Upload
Resources
About
Debug
SAA for many N level multilevel models
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
about

Adding variable girl_X_standlrt

Variable girl_X_standlrt did not significantly improve the model, so we remove it from the model.

We have considered all interaction variables so now run our final model.

$$\text{normexam}_i = \beta_0\text{vrband\_2}_i + \beta_1\text{vrband\_3}_i + \beta_2\text{standlrt}_i + \beta_3\text{avslrt}_i + \beta_4\text{girl\_1}_i + \beta_5\text{avslrt\_X\_standlrt}_i + \beta_6\text{intercept}_i + u^{(2)}_{0,school_i} + e_i$$

Variable	Coefficient	SE	p value	Significance
vrband_2	-0.397	0.032	< 0.001	*** (df=2)
vrband_3	-0.766	0.0535		
standlrt	0.389	0.0168	< 0.001	*** (df=1)
avslrt	0.326	0.105	0.003	** (df=1)
girl_1	0.162	0.0316	< 0.001	*** (df=1)
avslrt_X_standlrt	0.16	0.0388	< 0.001	*** (df=1)
Intercept	0.235	0.0467		
Between school Variance	0.0685	0.0138		
Level 1 Variance	0.531	0.0119		

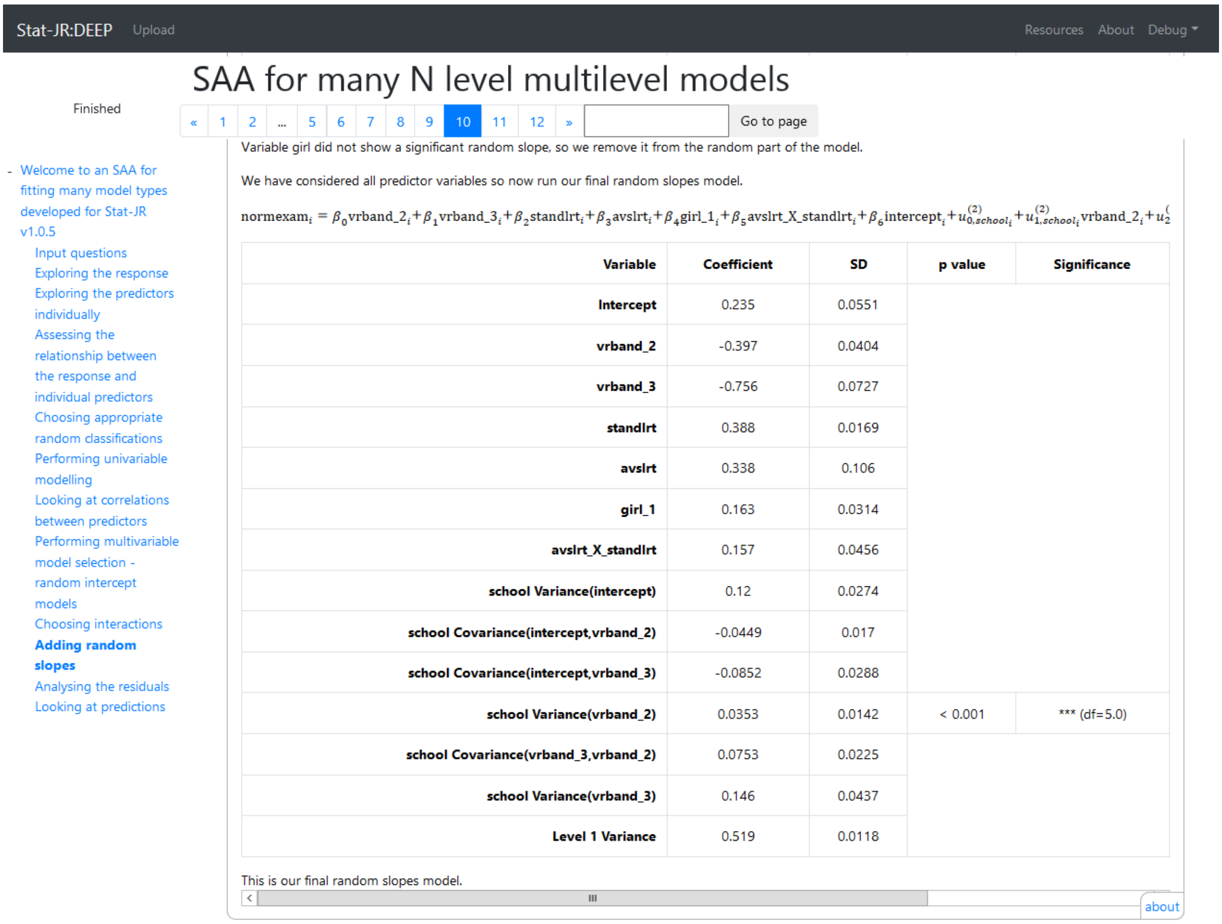
This is our final model.

Here we see that the final model contains only one interaction (between avslrt and standlrt). This model will then be used in the final pages to show the residuals and predictions for the final model that includes the interaction. For simplicity we currently only show predictions against each variable in turn holding all the other variables constant at their mean values, and we do not include other plots that might better illustrate the interactions. We have also saved this as a pdf entitled *tutinter.pdf*.

Chapter 6 – Random slopes models

For our next step we will consider expanding the model family to allow the effect of predictor variables to vary between different clusters. Such a model is often described as a random slopes (or random coefficients) model, as for continuous predictor variables if we plot the fitted values from the model against the predictors the resulting lines now have different slopes for each cluster as well as different intercepts.

We will first illustrate the random slopes model by simply expanding upon the modelling we have just done in chapter 5 where we used the tutorial dataset and looked at interactions. In fact, if you choose both interactions and random slopes the SAA will firstly find the best random intercept model including interactions and then move onto random slopes using the random intercept model as a base model and testing in turn whether allowing random slopes for each fixed predictor variable results in a better model. The random slopes model fitting will be shown on page 10 of the eBook as we show below:



Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Variable girl did not show a significant random slope, so we remove it from the random part of the model.

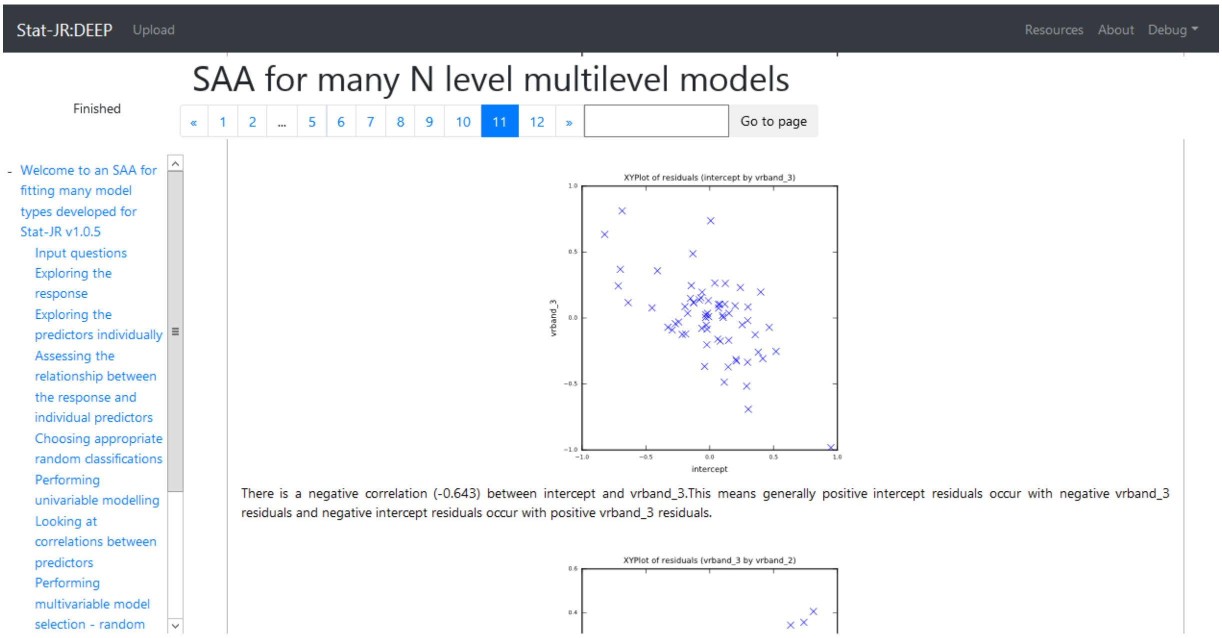
We have considered all predictor variables so now run our final random slopes model.

$$\mathrm{normexam}_i = \beta_0\mathrm{vrband\_2}_i + \beta_1\mathrm{vrband\_3}_i + \beta_2\mathrm{standlrt}_i + \beta_3\mathrm{avslrt}_i + \beta_4\mathrm{girl\_1}_i + \beta_5\mathrm{avslrt\_X\_standlrt}_i + \beta_6\mathrm{intercept}_i + u^{(2)}_{0,school_i} + u^{(2)}_{1,school_i}\mathrm{vrband\_2}_i + u_2^{(}$$

Variable	Coefficient	SD	p value	Significance
Intercept	0.235	0.0551		
vrband_2	-0.397	0.0404		
vrband_3	-0.756	0.0727		
standlrt	0.388	0.0169		
avslrt	0.338	0.106		
girl_1	0.163	0.0314		
avslrt_X_standlrt	0.157	0.0456		
school Variance(intercept)	0.12	0.0274		
school Covariance(intercept,vrband_2)	-0.0449	0.017		
school Covariance(intercept,vrband_3)	-0.0852	0.0288		
school Variance(vrband_2)	0.0353	0.0142	< 0.001	*** (df=5.0)
school Covariance(vrband_3,vrband_2)	0.0753	0.0225		
school Variance(vrband_3)	0.146	0.0437		
Level 1 Variance	0.519	0.0118		

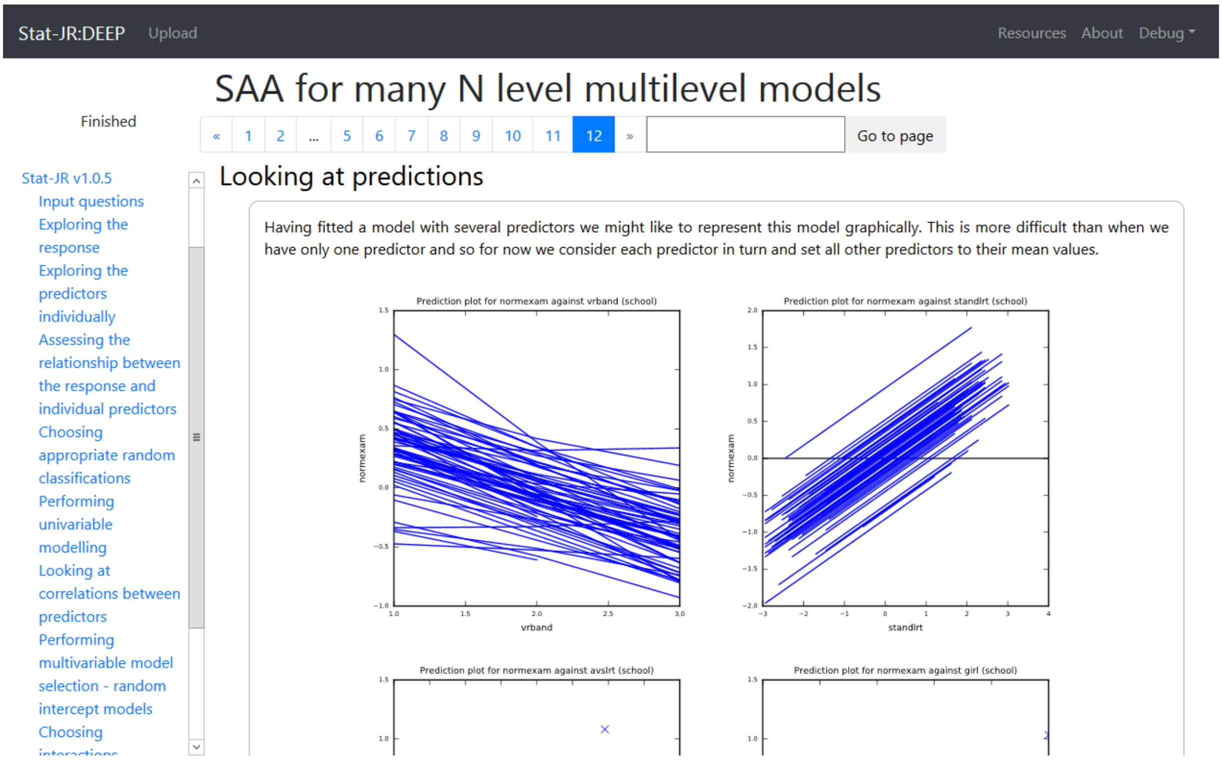
This is our final random slopes model.

Here we see that the model fitting decides that the best model includes random slopes for the *vrband* variable meaning that the predicted differences in exam score for the different VR bands differ between schools. On page 11 the residual plots also include plots of the random slope residuals and indeed pairwise residual plots as shown below:


Stat-JR:DEEP
Upload
Resources
About
Debug
SAA for many N level multilevel models
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random
XYPlot of residuals (intercept by vrband_3)
vrband_3
intercept
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
XYPlot of residuals (vrband_3 by vrband_2)
0.6
0.4

There is a negative correlation (-0.643) between intercept and vrband_3.This means generally positive intercept residuals occur with negative vrband_3 residuals and negative intercept residuals occur with positive vrband_3 residuals.

The prediction plots on page 12 also show the differing slopes thus:


Stat-JR:DEEP
Upload
Resources
About
Debug
SAA for many N level multilevel models
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Looking at predictions
Prediction plot for normexam against vrband (school)
normexam
vrband
1.5
1.0
0.5
0.0
−0.5
−1.0
1.0
1.5
2.0
2.5
3.0
Prediction plot for normexam against standlrt (school)
normexam
standlrt
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−3
−2
−1
0
1
2
3
4
Prediction plot for normexam against avslrt (school)
1.5
1.0
Prediction plot for normexam against girl (school)
1.5
1.0

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

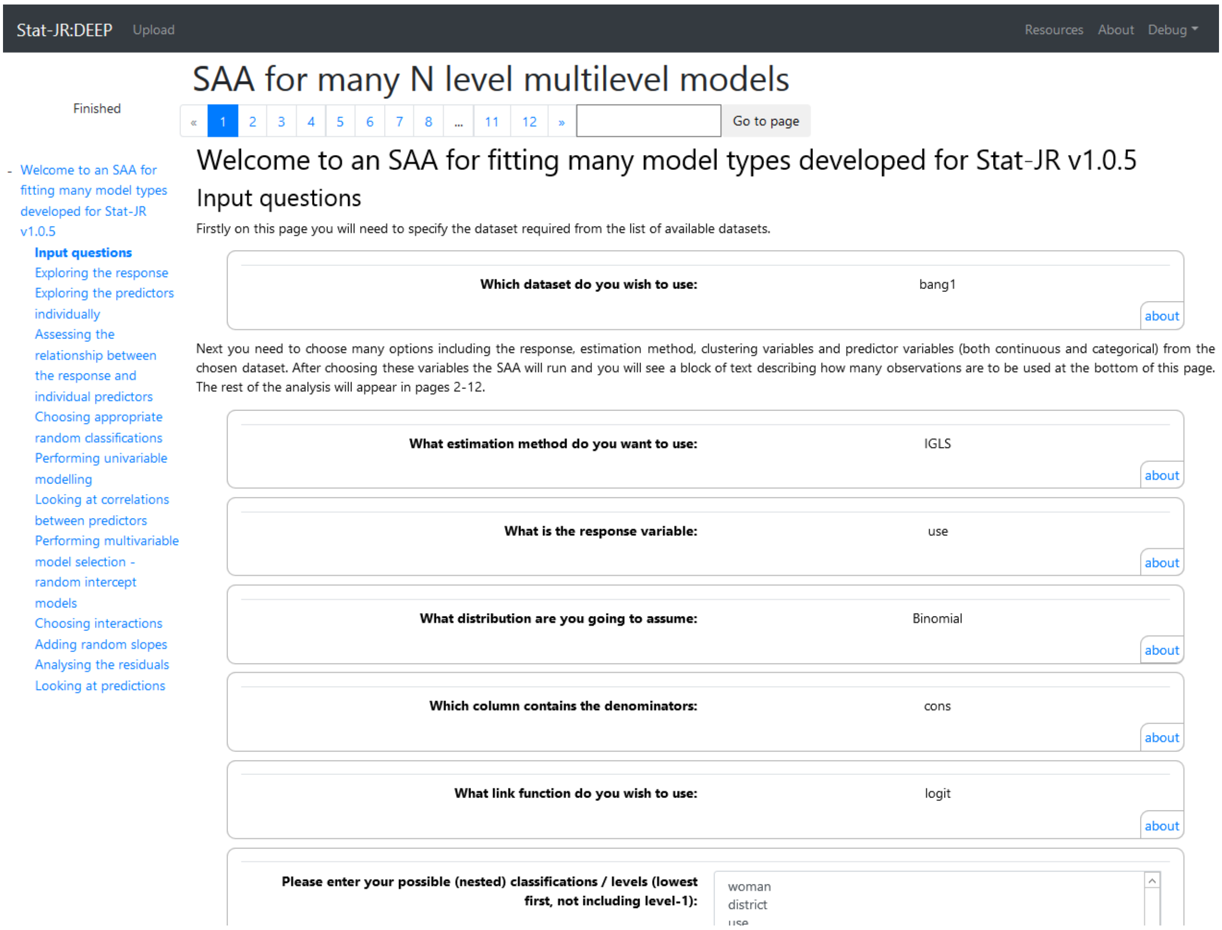
We have also saved the pdf as *tutrandslope.pdf*.

Chapter 7 – Logistic regression and multilevel logistic regression models

7.1 Single level logistic regression

Not all research questions result in a response model that can be easily fitted using normal distributions and one case in point is where the response is a binary variable. We will here look at using our Combined SAA on the dataset *bang1* (Huq & Cleland, 1990) which looks at the use of contraceptives in Bangladesh with the response being a yes / no variable as to whether women use contraceptives or not. This dataset has a multilevel structure with the women nested within districts within Bangladesh but for simplicity we will look first at a simpler one level logistic regression.

To do this we choose the Combined SAA in DEEP and then choose *bang1* as our dataset. We will again use *IGLS* (which will default to the 1st order MQL method of estimation for non-normal responses) for the estimation method. Our response is *use* and we will assume a *Binomial* distribution. As a result we will be asked for a denominator column and here we choose *cons* as our response is binary rather than a proportion which would be fit with a more general Binomial distribution. We will also be asked for a link function and here we choose *logit* for a logistic regression. The inputs should then look as follows:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5

Input questions

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: bang1

Next you need to choose many options including the response, estimation method, clustering variables and predictor variables (both continuous and categorical) 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 2-12.

What estimation method do you want to use: IGLS

What is the response variable: use

What distribution are you going to assume: Binomial

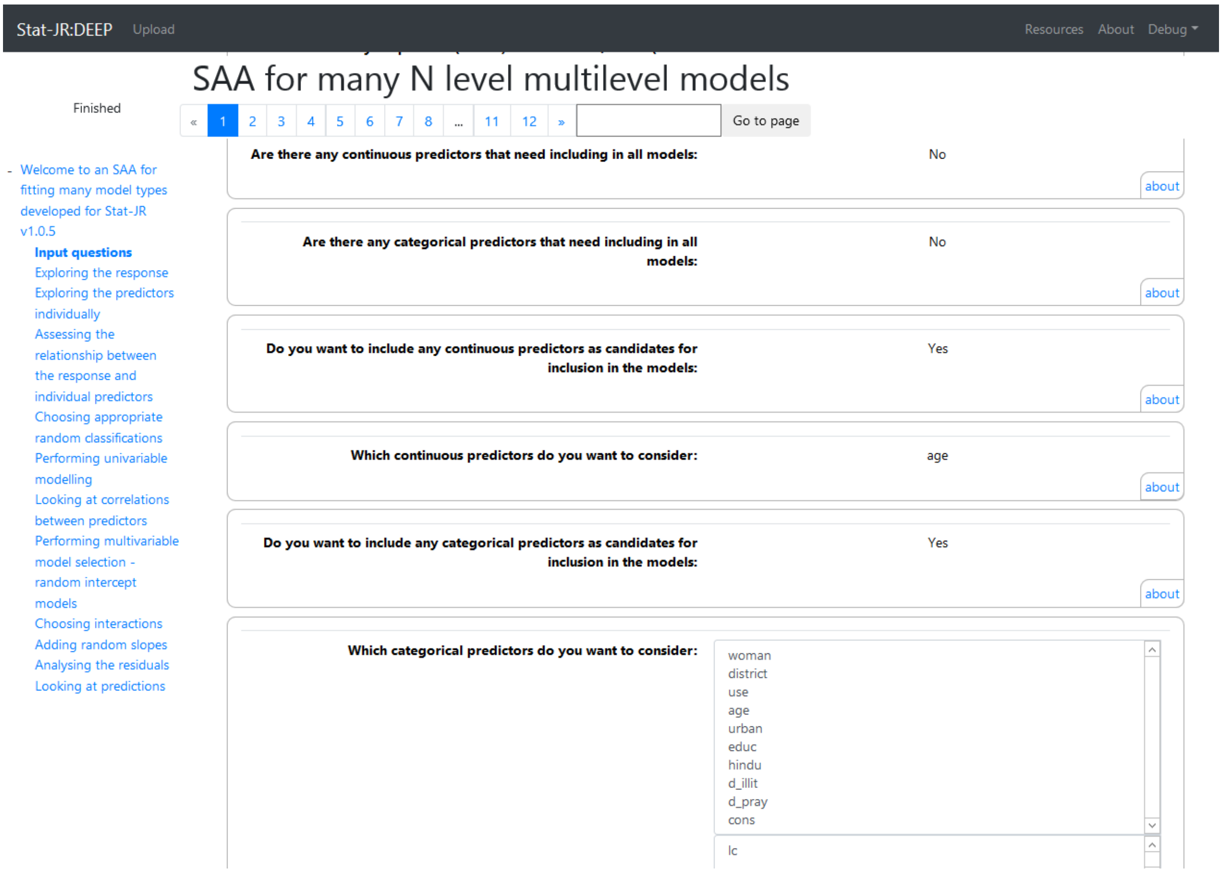
Which column contains the denominators: cons

What link function do you wish to use: logit

Please enter your possible (nested) classifications / levels (lowest first, not including level-1): woman
district
use

The Combined SAA is really designed for multilevel models and so now asks for classifications but if we choose *cons* then this will attempt to use *cons* as a higher-level identifier which will mean all observations in one cluster and this will be rejected and so we will then be left with a 1-level model.

For now we will have no variables (continuous or categorical) that are always included but we will test for the importance of 2 predictors, firstly a continuous predictor which is the woman's age (*age*) and second a categorical predictor the number of living children she has (*lc*). These inputs will look as follows:



Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Are there any continuous predictors that need including in all models: No

Are there any categorical predictors that need including in all models: No

Do you want to include any continuous predictors as candidates for inclusion in the models: Yes

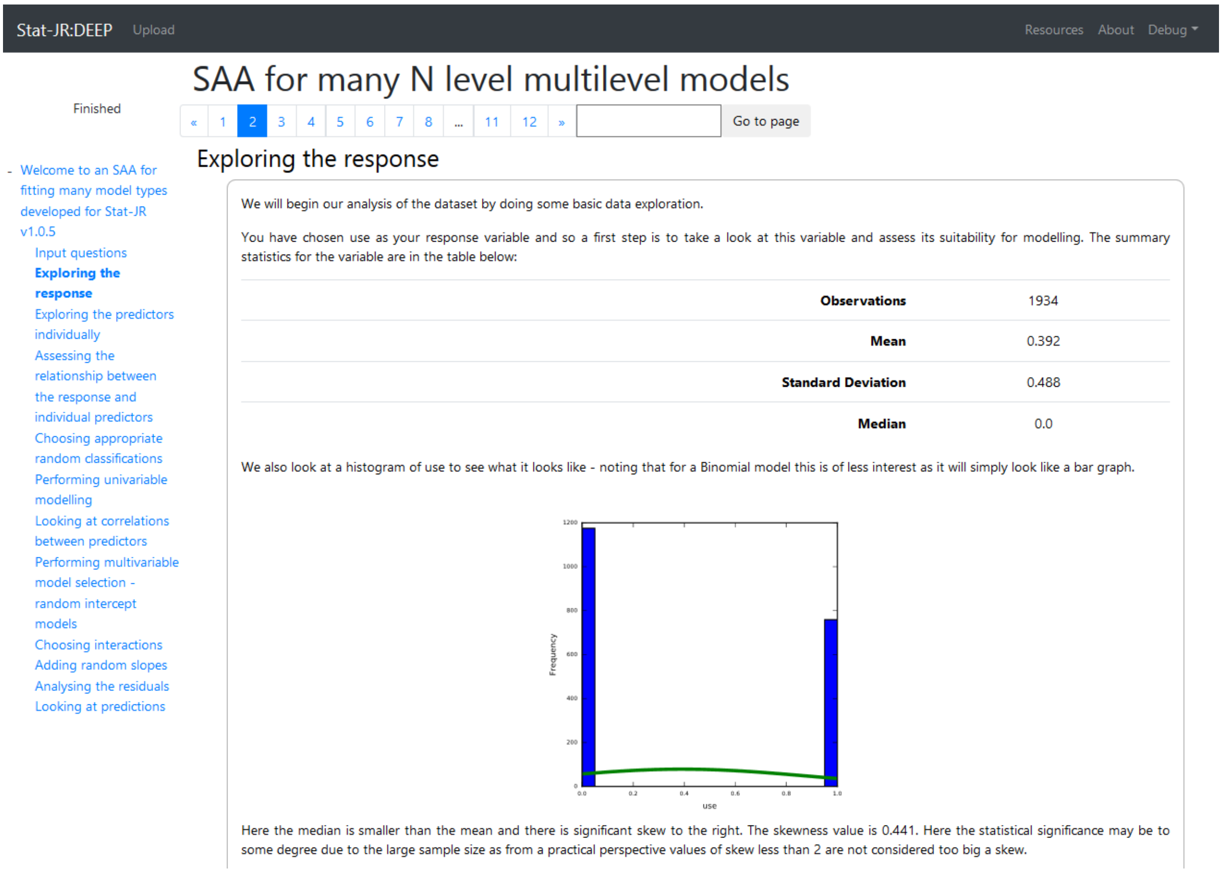
Which continuous predictors do you want to consider: age

Do you want to include any categorical predictors as candidates for inclusion in the models: Yes

Which categorical predictors do you want to consider: woman, district, use, age, urban, educ, hindu, d_illit, d_pray, cons; lc

We will use the *Forward Pass* method once again and for now not test for either *random slopes* or *interactions*. Having entered these last 3 inputs the eBook will execute and after a few minutes the models will all be fitted.

If we look through the pages in turn we see on page 2 some summary statistics for the response and a histogram which in this case as noted in the text is merely two bars of the 0s and 1s:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
Frequency
use
0
200
400
600
800
1000
1200
0.0
0.2
0.4
0.6
0.8
1.0

SAA for many N level multilevel models

Exploring the response

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

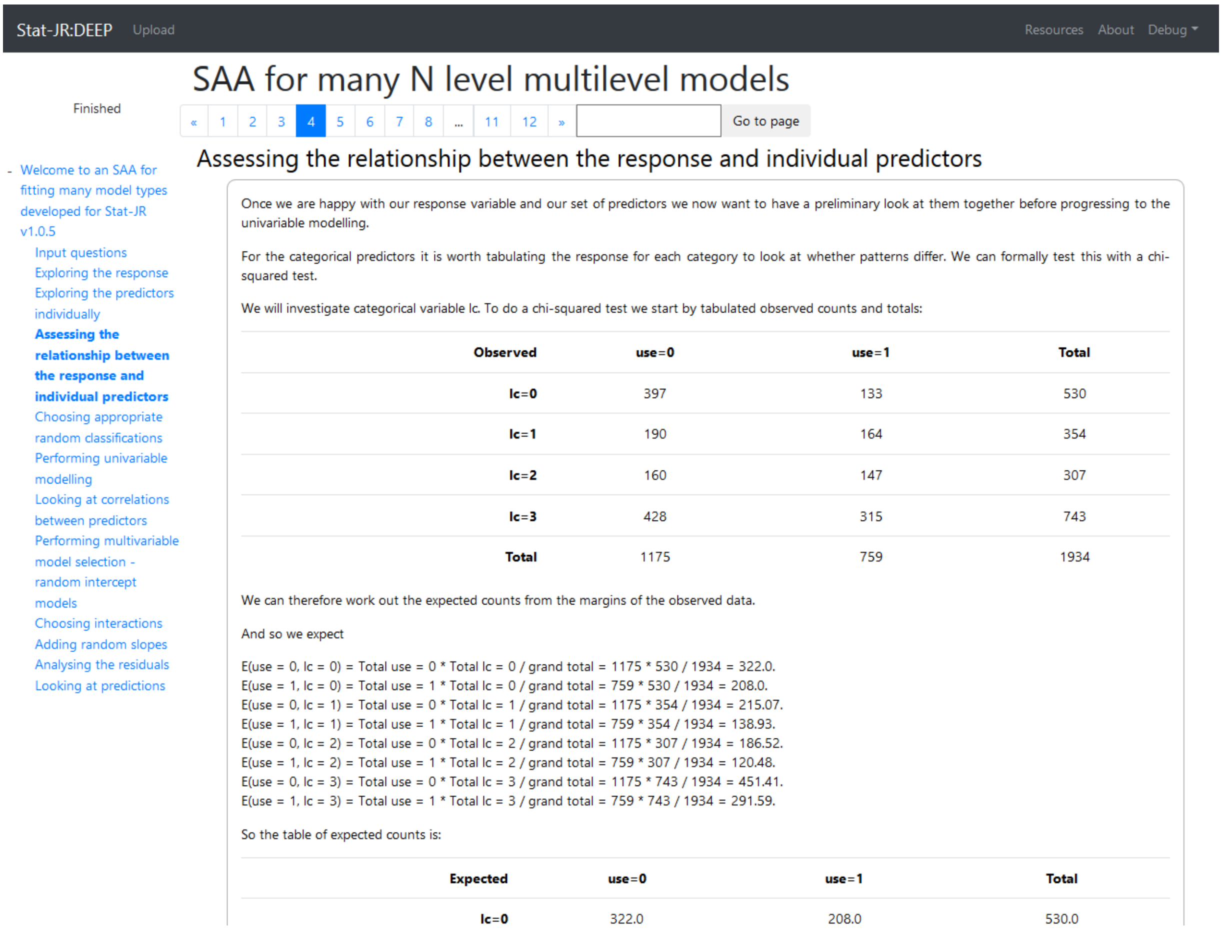
You have chosen use as your response variable and so a first step is to take a look at this variable and assess its suitability for modelling. The summary statistics for the variable are in the table below:

Observations	1934
Mean	0.392
Standard Deviation	0.488
Median	0.0

We also look at a histogram of use to see what it looks like - noting that for a Binomial model this is of less interest as it will simply look like a bar graph.

Here the median is smaller than the mean and there is significant skew to the right. The skewness value is 0.441. 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.

Page 3 looks at the two predictor variables on their own and here this is no different than when we have a normal response. Page 4 looks simply at the relationship between the response and the two predictors – for the categorical predictor (*lc*) the SAA now performs a chi-squared test (as shown in the screenshot below) and for the continuous predictor (*age*) a t-test to look at any differences in the mean ages of women using/not using contraceptives.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions

SAA for many N level multilevel models

Assessing the relationship between the response and individual predictors

Once we are happy with our response variable and our set of predictors we now want to have a preliminary look at them together before progressing to the univariable modelling.

For the categorical predictors it is worth tabulating the response for each category to look at whether patterns differ. We can formally test this with a chi-squared test.

We will investigate categorical variable lc. To do a chi-squared test we start by tabulated observed counts and totals:

Observed	use=0	use=1	Total
lc=0	397	133	530
lc=1	190	164	354
lc=2	160	147	307
lc=3	428	315	743
Total	1175	759	1934

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

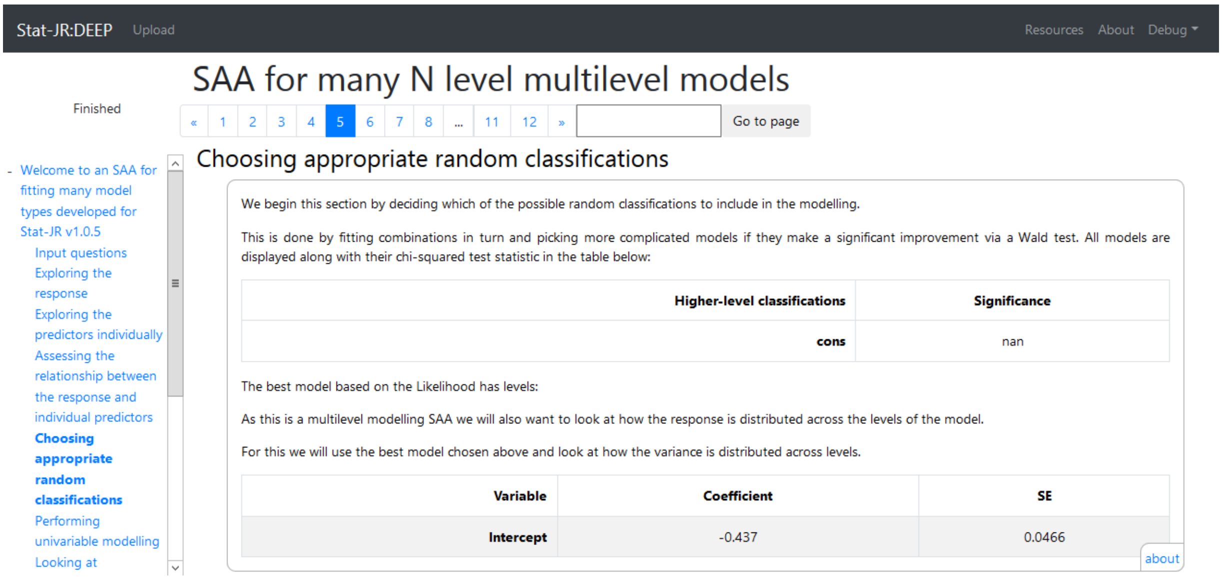
And so we expect

E(use = 0, lc = 0) = Total use = 0 * Total lc = 0 / grand total = 1175 * 530 / 1934 = 322.0.
E(use = 1, lc = 0) = Total use = 1 * Total lc = 0 / grand total = 759 * 530 / 1934 = 208.0.
E(use = 0, lc = 1) = Total use = 0 * Total lc = 1 / grand total = 1175 * 354 / 1934 = 215.07.
E(use = 1, lc = 1) = Total use = 1 * Total lc = 1 / grand total = 759 * 354 / 1934 = 138.93.
E(use = 0, lc = 2) = Total use = 0 * Total lc = 2 / grand total = 1175 * 307 / 1934 = 186.52.
E(use = 1, lc = 2) = Total use = 1 * Total lc = 2 / grand total = 759 * 307 / 1934 = 120.48.
E(use = 0, lc = 3) = Total use = 0 * Total lc = 3 / grand total = 1175 * 743 / 1934 = 451.41.
E(use = 1, lc = 3) = Total use = 1 * Total lc = 3 / grand total = 759 * 743 / 1934 = 291.59.

So the table of expected counts is:

Expected	use=0	use=1	Total
lc=0	322.0	208.0	530.0

Page 5 then tests for the best higher classifications and so as expected it rejects using *cons* as a classification in favour of a 1 level model:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at
about

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting combinations in turn and picking more complicated models if they make a significant improvement via a Wald test. All models are displayed along with their chi-squared test statistic in the table below:

Higher-level classifications	Significance
cons	nan

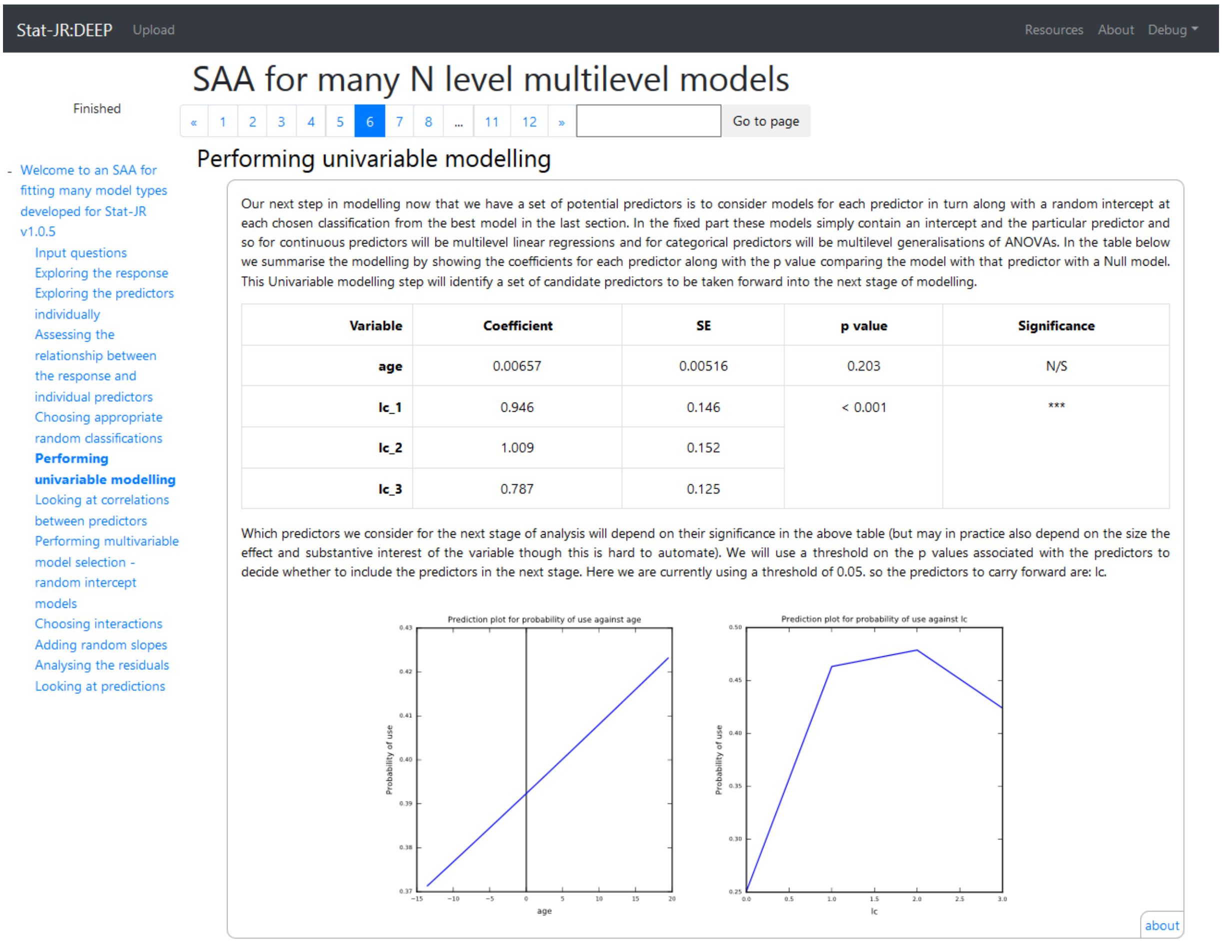
The best model based on the Likelihood has levels:

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

For this we will use the best model chosen above and look at how the variance is distributed across levels.

Variable	Coefficient	SE
Intercept	-0.437	0.0466

On page 6 we have the univariable models for each predictor along with plots of the model fitted. Here the plots have been transformed back to the probability scale. Below we see the estimates and predicted plots. The plot for the living children variable shows an interesting pattern of low use for women with no children with the probability increasing for 1 and then 2 children before reducing again for 3 or more.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
Prediction plot for probability of use against age
Probability of use
0.43
0.42
0.41
0.40
0.39
0.38
0.37
−15
−10
−5
0
5
10
15
20
age
Prediction plot for probability of use against lc
Probability of use
0.50
0.45
0.40
0.35
0.30
0.25
0.0
0.5
1.0
1.5
2.0
2.5
3.0
lc
about

SAA for many N level multilevel models

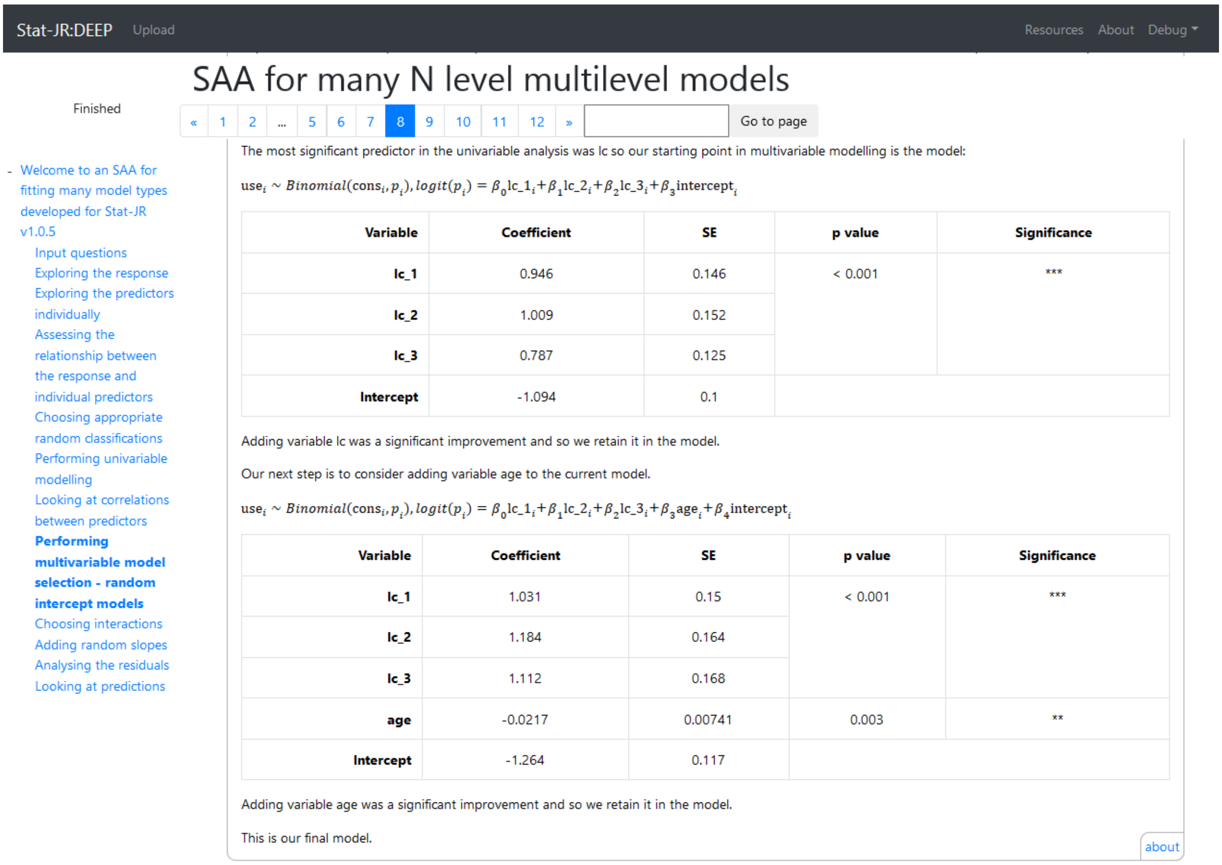
Performing univariable modelling

Our next step in modelling now that we have a set of potential predictors is to consider models for each predictor in turn along with a random intercept at each chosen classification from the best model in the last section. In the fixed part these models simply contain an intercept and the particular predictor and so for continuous predictors will be multilevel linear regressions and for categorical predictors will be multilevel generalisations of ANOVAs. In the table below we summarise the modelling by showing the coefficients for each predictor along with the p value comparing the model with that predictor with a Null model. This Univariable modelling step will identify a set of candidate predictors to be taken forward into the next stage of modelling.

Variable	Coefficient	SE	p value	Significance
age	0.00657	0.00516	0.203	N/S
lc_1	0.946	0.146	< 0.001	***
lc_2	1.009	0.152		
lc_3	0.787	0.125		

Which predictors we consider for the next stage of analysis will depend on their significance in the above table (but may in practice also depend on the size the effect and substantive interest of the variable though this is hard to automate). We will use a threshold on the p values associated with the predictors to decide whether to include the predictors in the next stage. Here we are currently using a threshold of 0.05. so the predictors to carry forward are: lc.

Moving on to page 8 we see the model building steps (below) where the software decides that the best model has both predictors together:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

The most significant predictor in the univariable analysis was lc so our starting point in multivariable modelling is the model:

$$\text{use}_i \sim Binomial(\text{cons}_i, p_i), logit(p_i) = \beta_0 \text{lc\_1}_i + \beta_1 \text{lc\_2}_i + \beta_2 \text{lc\_3}_i + \beta_3 \text{intercept}_i$$

Variable	Coefficient	SE	p value	Significance
lc_1	0.946	0.146	< 0.001	***
lc_2	1.009	0.152		
lc_3	0.787	0.125		
Intercept	-1.094	0.1		

Adding variable lc was a significant improvement and so we retain it in the model.

Our next step is to consider adding variable age to the current model.

$$\text{use}_i \sim Binomial(\text{cons}_i, p_i), logit(p_i) = \beta_0 \text{lc\_1}_i + \beta_1 \text{lc\_2}_i + \beta_2 \text{lc\_3}_i + \beta_3 \text{age}_i + \beta_4 \text{intercept}_i$$

Variable	Coefficient	SE	p value	Significance
lc_1	1.031	0.15	< 0.001	***
lc_2	1.184	0.164		
lc_3	1.112	0.168		
age	-0.0217	0.00741	0.003	**
Intercept	-1.264	0.117		

Adding variable age was a significant improvement and so we retain it in the model.

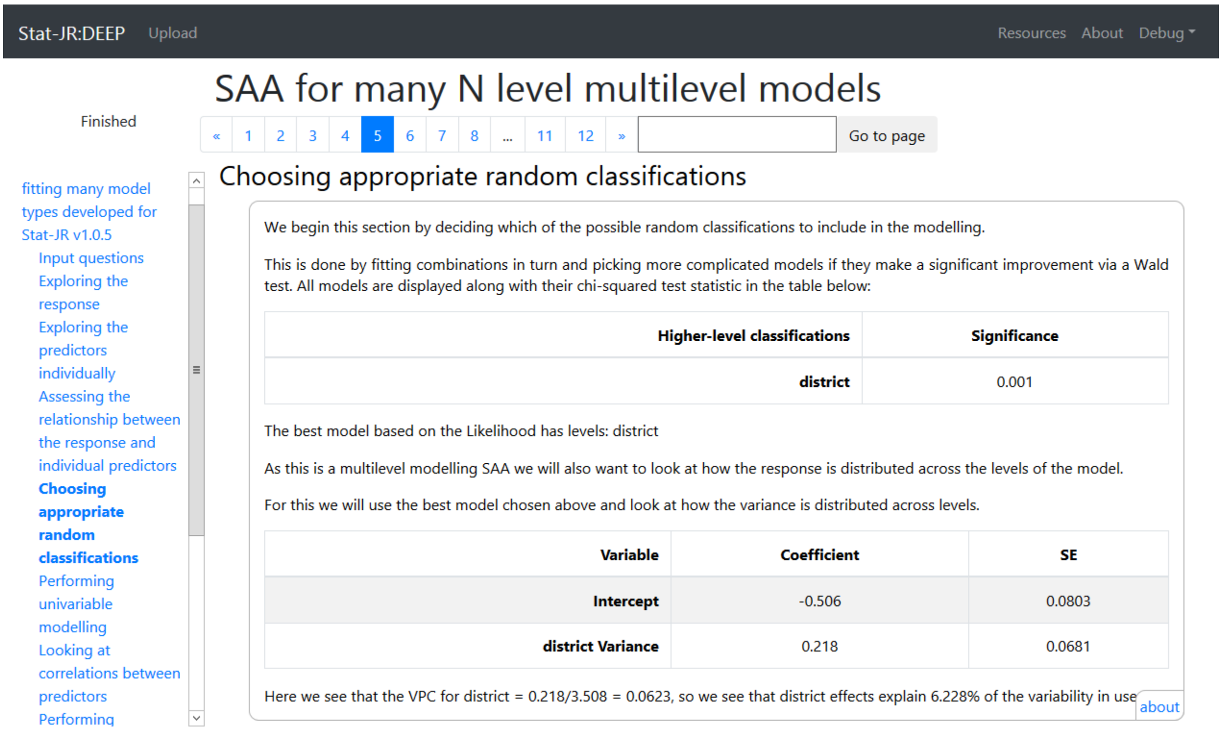
This is our final model.

The eBook ends with residual plots (although these are not plotted for level 1 in non-normal models so nothing appears here) and prediction plots from the final model. We have saved this eBook as a pdf file called *bang1lev.pdf*.

7.2 Multilevel logistic regression

To show more of the features of the Combined SAA we will next look again at the Bangladesh dataset but this time put the correct *district* identifier in for the classifications question and expand our continuous and categorical possible predictor variables set to *age*, *d_illit* and *d_pray* (for continuous) and *lc*, *urban*, *educ*, and *hindu* (for categorical). We will also test for random slopes and interactions. These inputs will result in more models to fit and so you will need to wait a little longer for the outputs. For non-normal models and IGLS estimation you are not offered a choice of comparison method as the software uses quasi-likelihood and so models are compared using Wald tests.

When the eBook finally finishes running (which takes a while) we will first look at page 5 where we check if the district level clustering is important. Here we see that a Wald test suggests it is, and also that districts explain 6.2% of the variability. Note that we use $\pi^2/3$ for the level 1 variance as suggested in (Goldstein, Browne, & Rasbash, 2002) for binomial models.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing
about

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting combinations in turn and picking more complicated models if they make a significant improvement via a Wald test. All models are displayed along with their chi-squared test statistic in the table below:

Higher-level classifications	Significance
district	0.001

The best model based on the Likelihood has levels: district

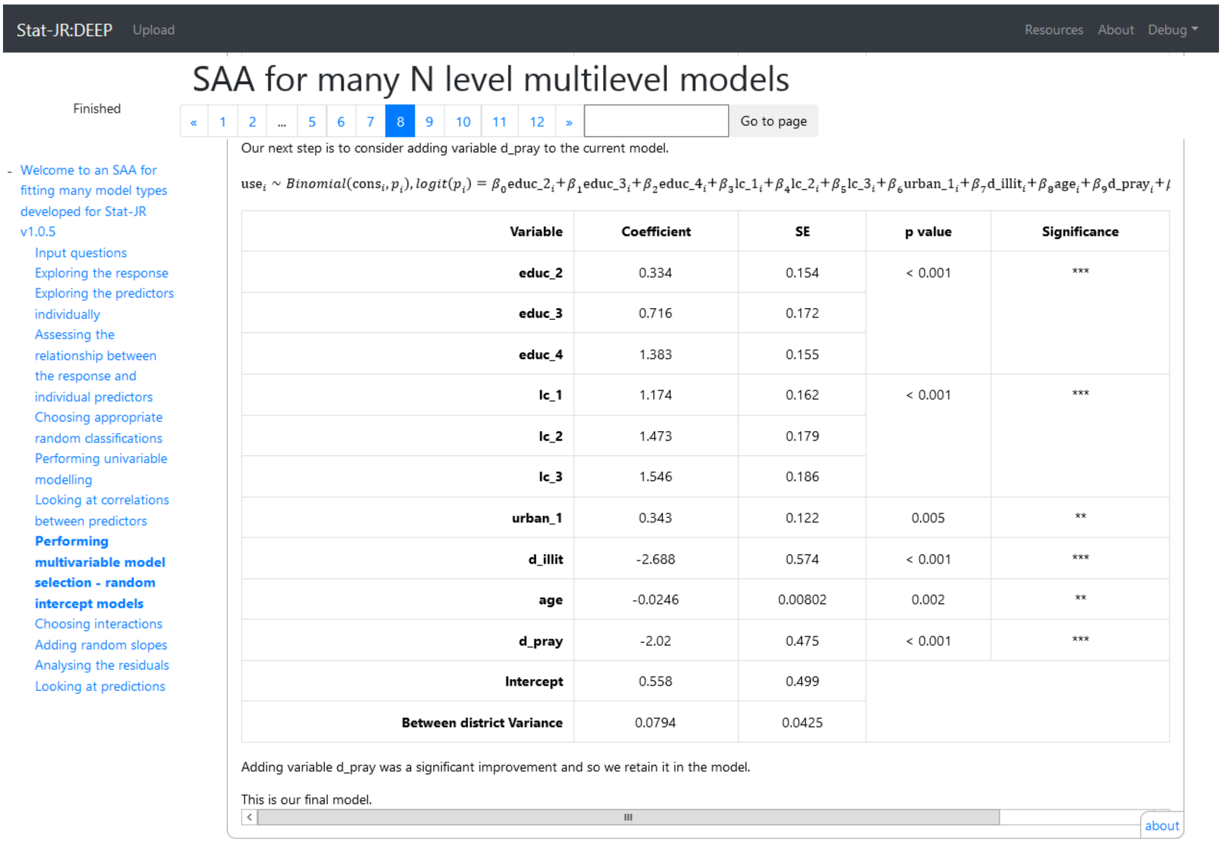
As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

For this we will use the best model chosen above and look at how the variance is distributed across levels.

Variable	Coefficient	SE
Intercept	-0.506	0.0803
district Variance	0.218	0.0681

Here we see that the VPC for district = 0.218/3.508 = 0.0623, so we see that district effects explain 6.228% of the variability in use

Looking at page 8 we can see the building up of a series of random intercept models with at each step predictors being retained or dropped depending on Wald tests. The only predictor that gets dropped in this model building is *hindu* and we see the final model below:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Our next step is to consider adding variable d_pray to the current model.

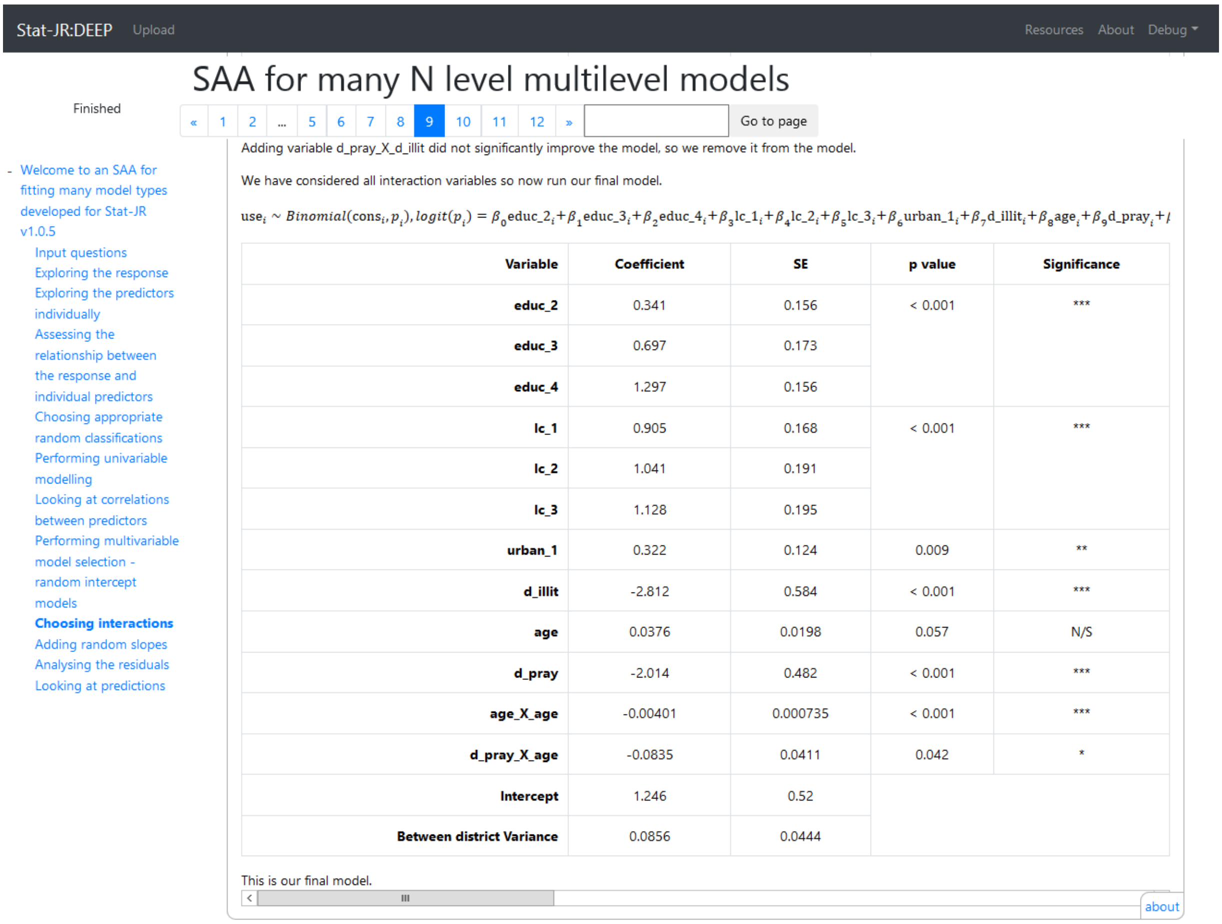
$\mathrm{use}_i \sim Binomial(\mathrm{cons}_i, p_i), logit(p_i) = \beta_0\mathrm{educ\_2}_i + \beta_1\mathrm{educ\_3}_i + \beta_2\mathrm{educ\_4}_i + \beta_3\mathrm{lc\_1}_i + \beta_4\mathrm{lc\_2}_i + \beta_5\mathrm{lc\_3}_i + \beta_6\mathrm{urban\_1}_i + \beta_7\mathrm{d\_illit}_i + \beta_8\mathrm{age}_i + \beta_9\mathrm{d\_pray}_i +$

Variable	Coefficient	SE	p value	Significance
educ_2	0.334	0.154	< 0.001	***
educ_3	0.716	0.172		
educ_4	1.383	0.155		
lc_1	1.174	0.162	< 0.001	***
lc_2	1.473	0.179		
lc_3	1.546	0.186		
urban_1	0.343	0.122	0.005	**
d_illit	-2.688	0.574	< 0.001	***
age	-0.0246	0.00802	0.002	**
d_pray	-2.02	0.475	< 0.001	***
Intercept	0.558	0.499		
Between district Variance	0.0794	0.0425		

Adding variable d_pray was a significant improvement and so we retain it in the model.

This is our final model.

We have also asked for interactions so they are considered in another model building step on page 9 starting from our final model on page 8. This time the final model chosen includes 2 additional terms – a quadratic term for *age* and an interaction between the *d_pray* and *age* variables. Note that although the linear term in *age* is no longer significant it is retained in the model as there are interactions involving *age*. The final model can be seen below:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Adding variable d_pray_X_d_illit did not significantly improve the model, so we remove it from the model.

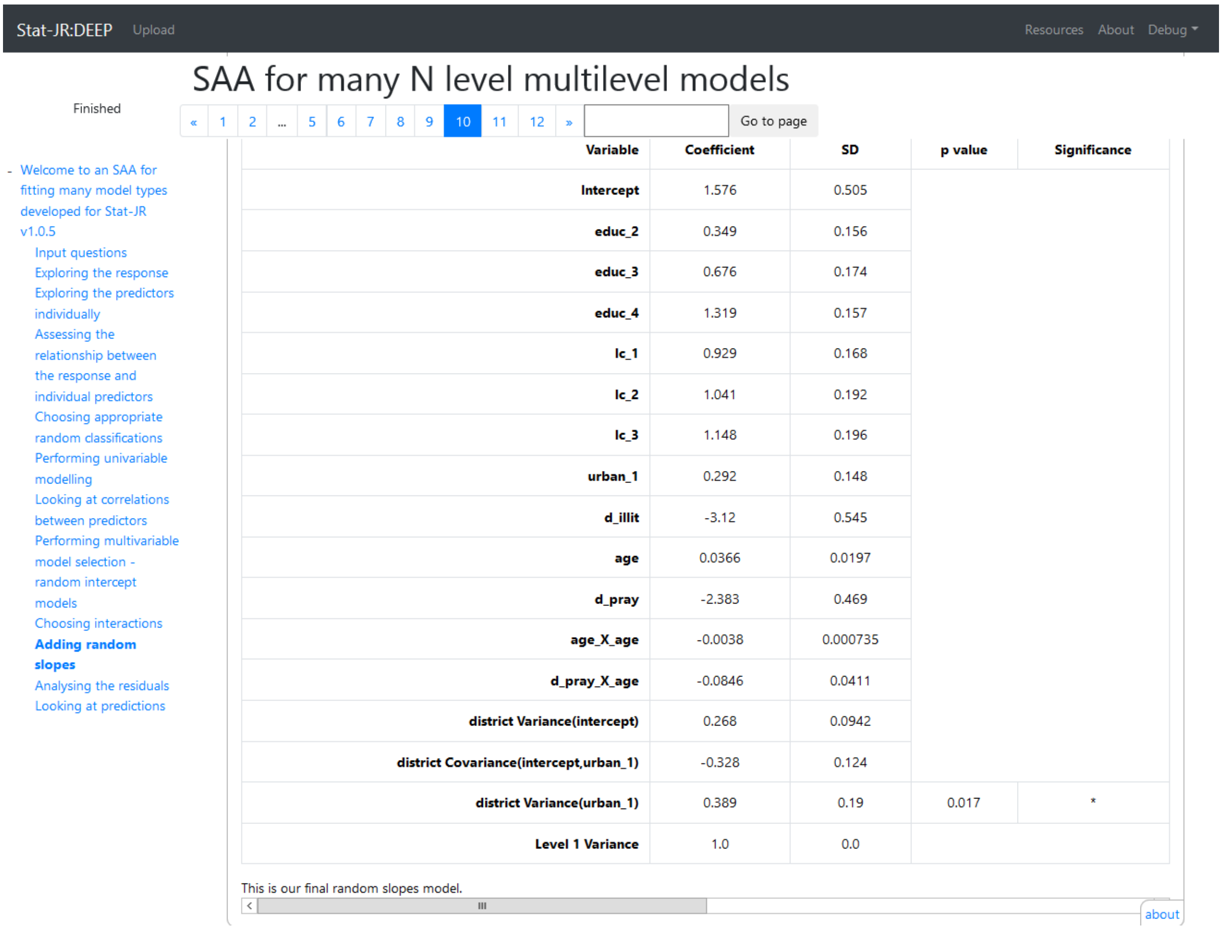
We have considered all interaction variables so now run our final model.

$\text{use}_i \sim Binomial(\text{cons}_i, p_i), logit(p_i) = \beta_0\text{educ\_2}_i + \beta_1\text{educ\_3}_i + \beta_2\text{educ\_4}_i + \beta_3\text{lc\_1}_i + \beta_4\text{lc\_2}_i + \beta_5\text{lc\_3}_i + \beta_6\text{urban\_1}_i + \beta_7\text{d\_illit}_i + \beta_8\text{age}_i + \beta_9\text{d\_pray}_i +$

Variable	Coefficient	SE	p value	Significance
educ_2	0.341	0.156	< 0.001	***
educ_3	0.697	0.173		
educ_4	1.297	0.156		
lc_1	0.905	0.168	< 0.001	***
lc_2	1.041	0.191		
lc_3	1.128	0.195		
urban_1	0.322	0.124	0.009	**
d_illit	-2.812	0.584	< 0.001	***
age	0.0376	0.0198	0.057	N/S
d_pray	-2.014	0.482	< 0.001	***
age_X_age	-0.00401	0.000735	< 0.001	***
d_pray_X_age	-0.0835	0.0411	0.042	*
Intercept	1.246	0.52		
Between district Variance	0.0856	0.0444		

This is our final model.

On page 10 we further expand our modelling by considering whether the effects of any of our terms varies across districts by fitting random slopes. It transpires that the only term that requires a random slope is *urban* and so the impact of being in an urban versus rural area on use of contraceptive varies from district to district. See the model below:

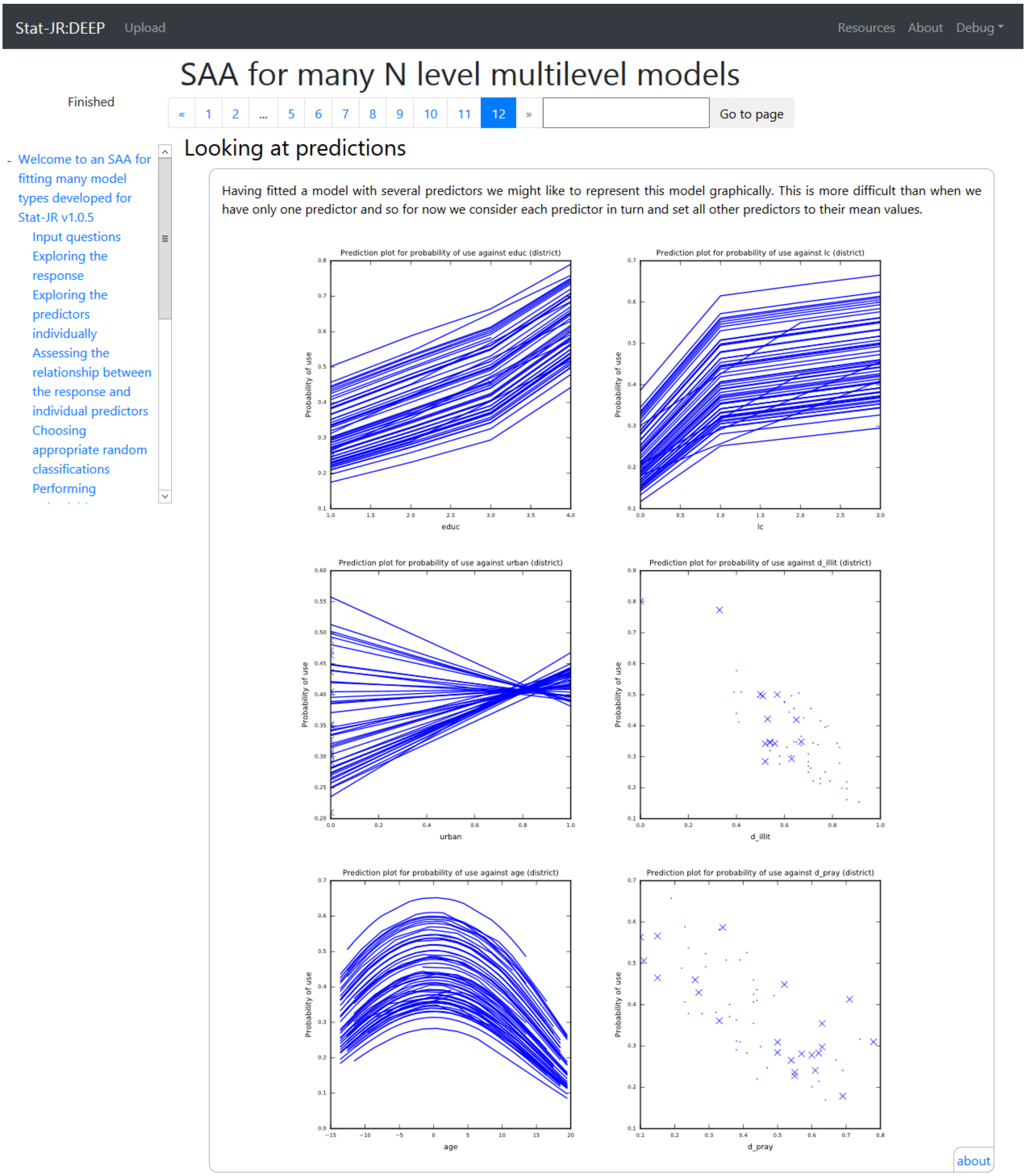

Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Variable	Coefficient	SD	p value	Significance
Intercept	1.576	0.505		
educ_2	0.349	0.156		
educ_3	0.676	0.174		
educ_4	1.319	0.157		
lc_1	0.929	0.168		
lc_2	1.041	0.192		
lc_3	1.148	0.196		
urban_1	0.292	0.148		
d_illit	-3.12	0.545		
age	0.0366	0.0197		
d_pray	-2.383	0.469		
age_X_age	-0.0038	0.000735		
d_pray_X_age	-0.0846	0.0411		
district Variance(intercept)	0.268	0.0942		
district Covariance(intercept,urban_1)	-0.328	0.124		
district Variance(urban_1)	0.389	0.19	0.017	*
Level 1 Variance	1.0	0.0		

This is our final random slopes model.

Page 11 includes residual plots at the district level for both intercepts and urban effects and on page 12 we see the predicted probabilities for each district for each predictor variable:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing
Prediction plot for probability of use against educ (district)
Prediction plot for probability of use against lc (district)
Prediction plot for probability of use against urban (district)
Prediction plot for probability of use against d_illit (district)
Prediction plot for probability of use against age (district)
Prediction plot for probability of use against d_pray (district)
Probability of use
educ
lc
urban
d_illit
age
d_pray
about

SAA for many N level multilevel models

Looking at predictions

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

Here you will see some interesting patterns: *urban* has different effects for each district and the strong negative correlation here is obvious; *age* has a quadratic relationship and clearly in each district the use of contraceptive increases with age before peaking and then decreasing; *educ* and *lc* are categorical predictors and so although we have joined the points for districts this is not really meaningful as for example you can't have 1.2 living children (plus the lines sometimes cross if there do not exist any women with a particular number of living children in some districts); finally *d_illit* and *d_pray* are district level variables and hence we have point plots rather than lines.

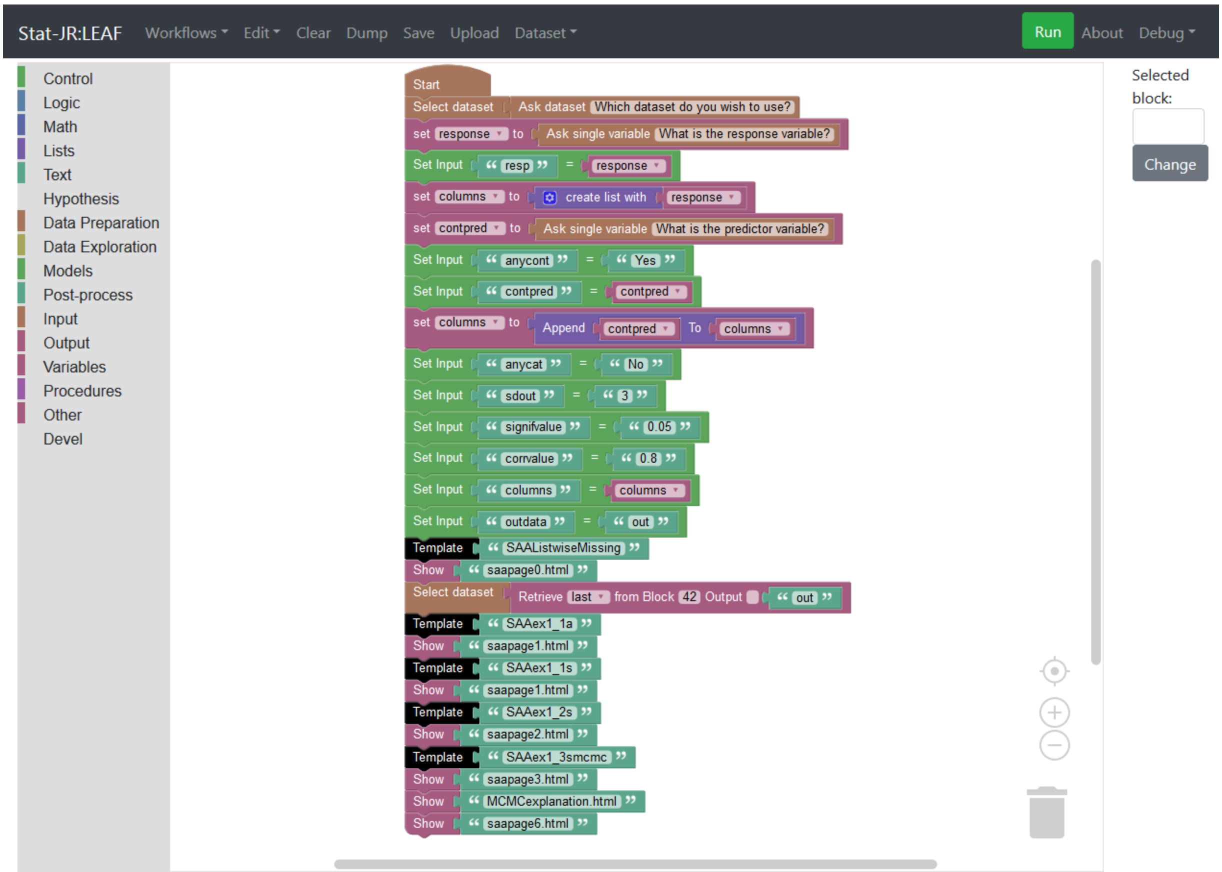
If you want to look at more of the eBook we have saved it in the pdf file, *bang2lev.pdf*.

Chapter 8 – MCMC estimation

8.1 - MCMC for a linear regression

MCMC estimation is an alternative, simulation-based method for fitting models that is generally cast in a Bayesian framework. If you are unfamiliar with MCMC and Bayesian methods we recommend the book ‘MCMC Estimation in MLwiN’ (Browne W. , MCMC Estimation in MLwiN v3.00, 2017) which will take you through the basic concepts involved in MCMC and Bayesian statistics using the MLwiN software which is the estimation engine being used in the SAAs in Stat-JR. Constructing SAAs using MCMC is more challenging than using the classical IGLS algorithm in particular as it is easier to determine when IGLS has converged and thus a model fit has finished. In MCMC how long one runs a model (in terms of numbers of iterations) is harder to decide upon. In addition the fact that MCMC runs in a Bayesian framework means that each model requires prior distributions for all parameters and this is also harder to incorporate in an automatic system. We will move on to look at how we have incorporated MCMC estimation in our Combined SAA but first we will start with something similar which is to revisit the simple linear regression model and approach this using MCMC.

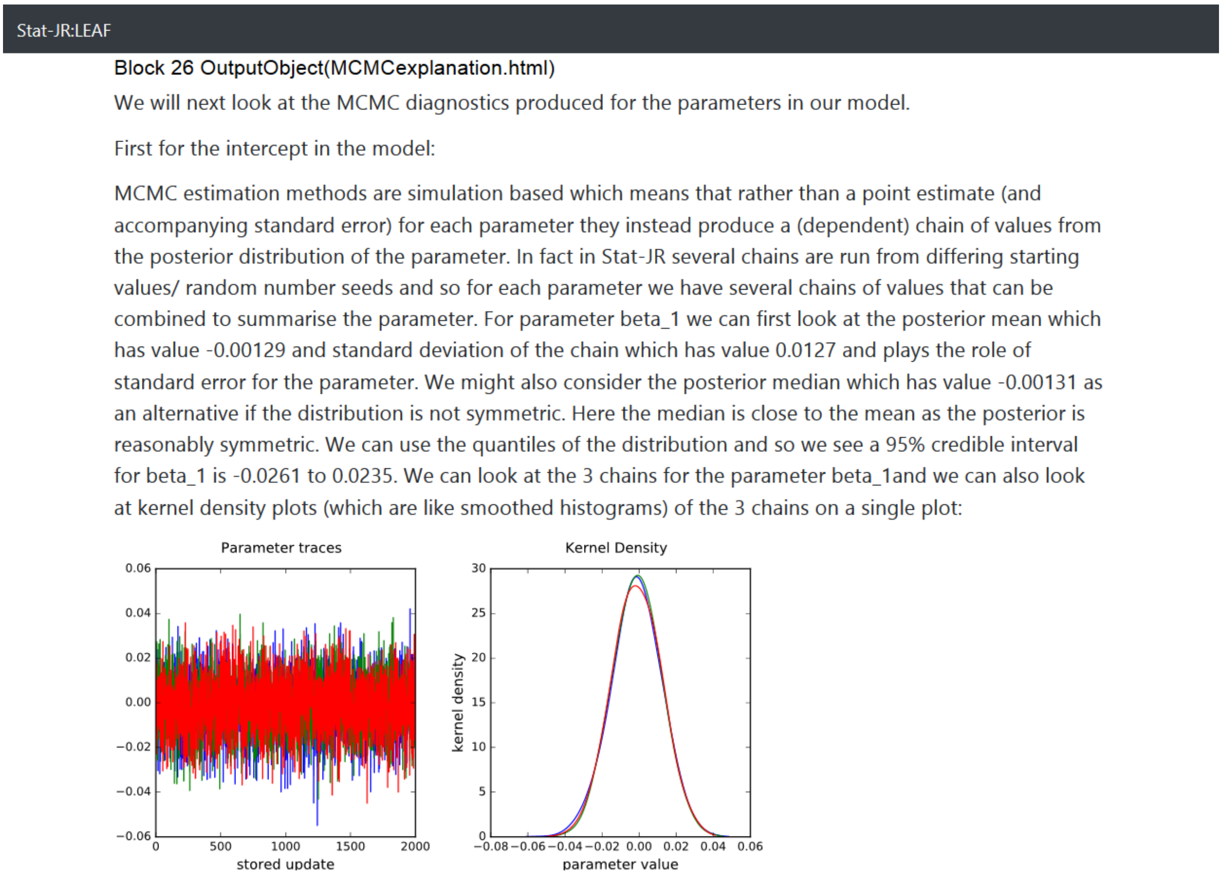
We will look at this using the LEAF interface first so start up LEAF and choose the workflow *linregMCMC* from the *Stat_Assistant* directory. This workflow looks as follows:



```
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_3smcmc"
Show "saapage3.html"
Show "MCMCexplanation.html"
Show "saapage6.html"
```

As we saw for earlier regression templates we have the standard inputs that are required: a dataset (for which we will choose *tutorial*), response variable (*normexam*) and predictor variable (*standlrt*). Looking at the end of the workflow we see a template block for a template *SAAex1_3smcmc* and this template is followed by 3 show blocks. Basically the *SAAex1_3mcmc* template performs all the model fitting and post-processing. It runs the *Regression1* template which will fit the regression model using MCMC and the in-built eStat engine (rather than MLwiN) within StatJR for 3 chains with a burnin of 500 iterations and main run of 2000 iterations. It will then take the outputs and for each of three parameters (intercept, slope and residual variance) call another template *MCMCExplanation* which post processes MCMC chains for a parameter and discusses diagnostics. The 3 Show blocks therefore show 3 separate html outputs that cover – the model and predicted line, the MCMC diagnostics and residuals.

You can run this workflow and the outputs from the show blocks will be produced as shown below (zooming in on the top of the *MCMCexplanation* output):



Block 26 OutputObject(MCMCexplanation.html)

We will next look at the MCMC diagnostics produced for the parameters in our model.

First for the intercept in the model:

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 -0.00129 and standard deviation of the chain which has value 0.0127 and plays the role of standard error for the parameter. We might also consider the posterior median which has value -0.00131 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 -0.0261 to 0.0235. 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:

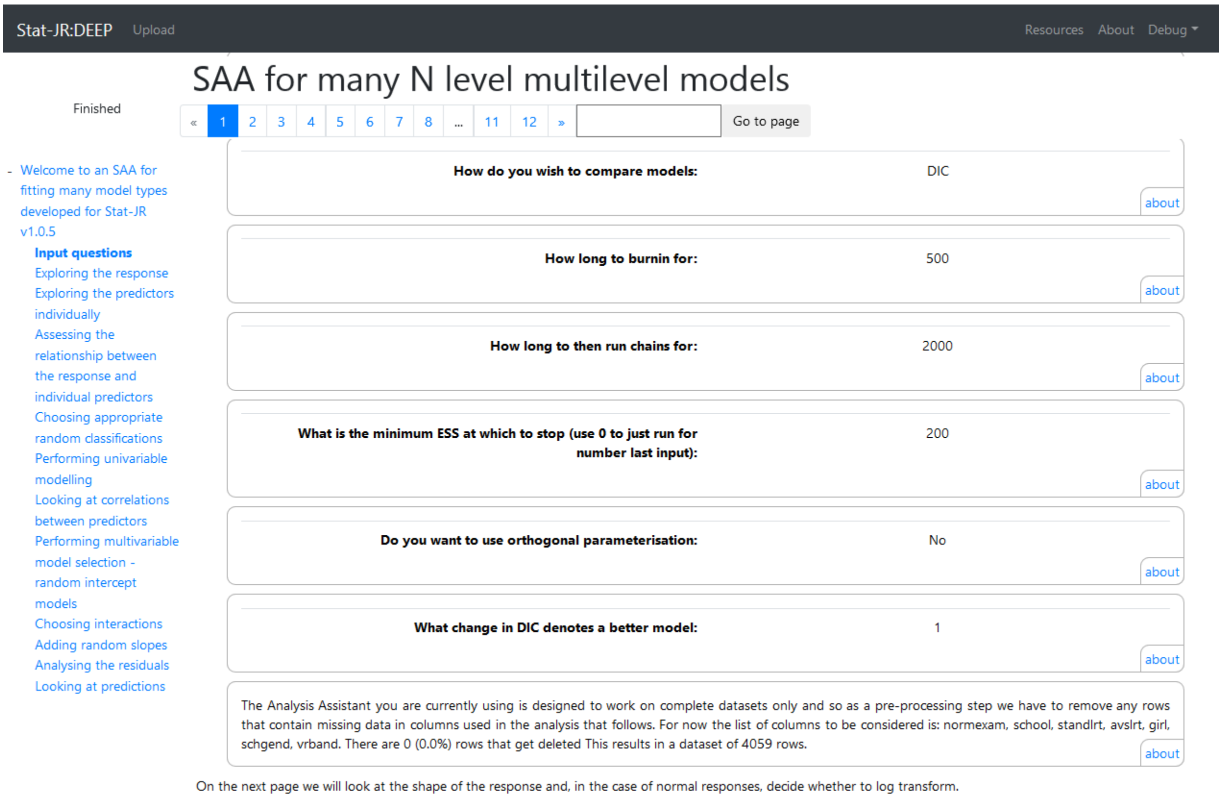
Basically this output looks at the diagnostic plots produced by the MCMC estimation and attempts to explain them to the reader. One thing to note is that the *SAAex1_3mcmc* template has been hardwired with fixed burnin and chain lengths and so if these aren't long enough then we might want to run for longer but can't via the eBook. We will revisit this next when we consider how we have incorporated MCMC estimation into our Combined SAA. It's also clear that even though it is interesting to see the details with regard MCMC diagnostics here they take up rather a lot of space and this example eBook only fits one model that contains only 3 parameters and so from a practical perspective having these diagnostics reported for each model in a complete analysis that is automatically choosing between many models is rather unwieldy. We will therefore not do this in the Combined SAA but this is not to suggest that checking diagnostics is not important! There is an eBook *linregmcmc.zip* that uses this workflow and we have stored in pdf format in the file *tutlrmcmc.pdf* the output eBook for the regression of *standlrt* on *normexam* for the tutorial dataset.

8.2 - Using MCMC in the Combined SAA

We will now consider using MCMC via our Combined SAA. We will start by looking at the random intercepts model that we used for the tutorial dataset as a first demonstration of the SAA. The inputs are therefore fairly similar to before i.e. the dataset is *tutorial*, this time estimation method is *MCMC*, response in *normexam*, distribution is *Normal*, and higher-level classifications to consider is (only) *school.* Answer *No* to both questions about always including variables (continuous and categorical) and then choose to include *standlrt* and *avslrt* as continuous predictors and *girl, schgend* and *vrband* as categorical predictors. We will as before use *Forward Pass* to choose models and have no random slopes or interactions.

For MCMC we are offered again two methods to compare models DIC and Wald. DIC is perhaps the most commonly used method with MCMC for model selection and we will use this here – the Wald method basically just performs Wald tests using the MCMC estimates which is not strictly speaking something that a Bayesian statistician would recommend but which seems to give reasonable results. We next need to approach the issue of how long to run the procedure for. This requires 3 inputs – the first two are a *burnin* (which we will set to 500) which is the number of iterations to throw away before storing the estimates and a *run length* (which we will set to 2000) which is how long to then store the estimates for. These are standard inputs in MCMC and the software will in the background run a single MCMC chain using the MCMC engine within MLwiN. The third input is more interesting as we are asked for a *minimum ESS value*. The ESS or Effective Sample Size is a measure of how many independent iterations the dependent iterations we get out of MCMC for each parameter are equivalent to and so this is a method of automating convergence detection as the MCMC routine will continue running for blocks of the indicated run length until all ESS values are greater than this minimum. If this minimum is set to 0 then effectively the software will run each model for the indicated run length only and then finish with that model. We will set this to the value 200 for now so that we don't end up with any models returning with really small ESS values. A couple of words of warning here however are that if the SAA ends up fitting a poorly defined model then it is possible that the software will get stuck never reaching ESS values of 200; secondly this approach does not affect the burnin chosen so if the chains haven't reached their equilibrium distribution prior to the end of the burnin then those iterations post-burnin that are not converged will still be retained.

We are asked next whether to use *orthogonal parameterisation* which is a method (Browne, Steele, Golalizadeh, & Green, 2009) within MLwiN for improving MCMC estimation but it only really works for non-normal models so we say *No* here. Finally we are asked what value of DIC we deem to be a better model and here we choose *1*. In practice any smaller DIC is a better model but the DIC diagnostic is a stochastic quantity and so we will say a difference of 1 is required. Our final inputs can be seen in the window below and then after clicking *Submit* we need to move onto page 2 and say *No* to the logging question.



SAA for many N level multilevel models

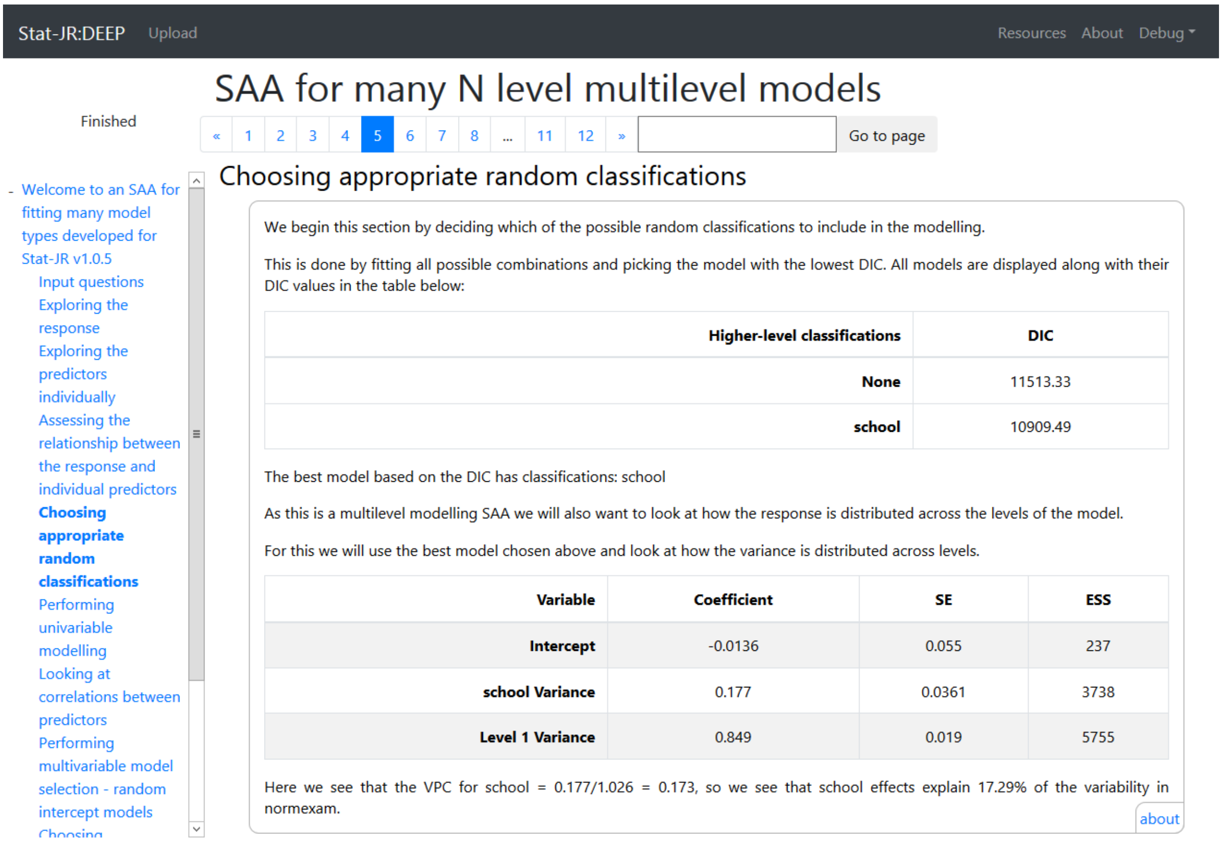
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
  - **Input questions**
  - Exploring the response
  - Exploring the predictors individually
  - Assessing the relationship between the response and individual predictors
  - Choosing appropriate random classifications
  - Performing univariable modelling
  - Looking at correlations between predictors
  - Performing multivariable model selection - random intercept models
  - Choosing interactions
  - Adding random slopes
  - Analysing the residuals
  - Looking at predictions

How do you wish to compare models:	DIC
How long to burnin for:	500
How long to then run chains for:	2000
What is the minimum ESS at which to stop (use 0 to just run for number last input):	200
Do you want to use orthogonal parameterisation:	No
What change in DIC denotes a better model:	1

The Analysis Assistant 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: normexam, school, standlrt, avslrt, girl, schgend, vrband. There are 0 (0.0%) rows that get deleted This results in a dataset of 4059 rows.

On the next page we will look at the shape of the response and, in the case of normal responses, decide whether to log transform.

The model fitting will then begin. In fact for this set of inputs the model fitting should not take that long though generally MCMC is slower than IGLS and the DIC method of comparison can take a while. Looking at what is different from the IGLS approach on page 5 we see the following:



SAA for many N level multilevel models

- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
  - Input questions
  - Exploring the response
  - Exploring the predictors individually
  - Assessing the relationship between the response and individual predictors
  - **Choosing appropriate random classifications**
  - Performing univariable modelling
  - Looking at correlations between predictors
  - Performing multivariable model selection - random intercept models
  - Choosing

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting all possible combinations and picking the model with the lowest DIC. All models are displayed along with their DIC values in the table below:

Higher-level classifications	DIC
None	11513.33
school	10909.49

The best model based on the DIC has classifications: school

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

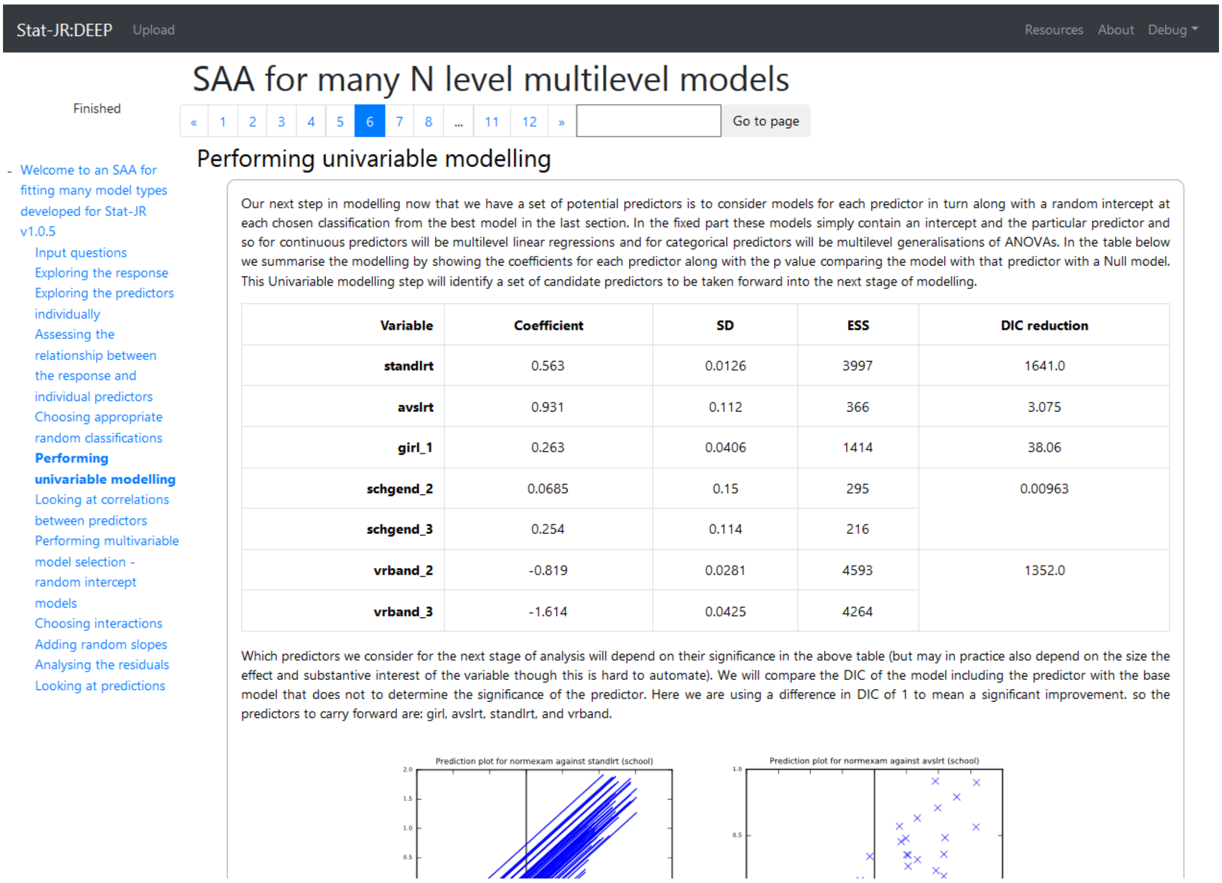
For this we will use the best model chosen above and look at how the variance is distributed across levels.

Variable	Coefficient	SE	ESS
Intercept	-0.0136	0.055	237
school Variance	0.177	0.0361	3738
Level 1 Variance	0.849	0.019	5755

Here we see that the VPC for school = 0.177/1.026 = 0.173, so we see that school effects explain 17.29% of the variability in normexam.

Here we see the DIC value for a single level and a multilevel model (school) and note the school DIC is smaller meaning a better model. We then see the estimates of this simple variance components model and note that the VPC at 0.173 or 17.3% of variation at the school level is comparable with the 16.6% seen in the IGLS approach. Note here that the MCMC approach uses uniform priors throughout which tend to give larger higher-level variances. You will see the ESS column in the table and note that all values are greater than 200 as required.

On page 6 we then have a table of the estimates from various univariable models along with ESS scores. This time whether variables are important is determined from the reduction in DIC and we see that all variables apart from school gender improve DIC which agrees with IGLS again.



SAA for many N level multilevel models

- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
  - Input questions
  - Exploring the response
  - Exploring the predictors individually
  - Assessing the relationship between the response and individual predictors
  - Choosing appropriate random classifications
  - **Performing univariable modelling**
  - Looking at correlations between predictors
  - Performing multivariable model selection - random intercept models
  - Choosing interactions
  - Adding random slopes
  - Analysing the residuals
  - Looking at predictions

Performing univariable modelling

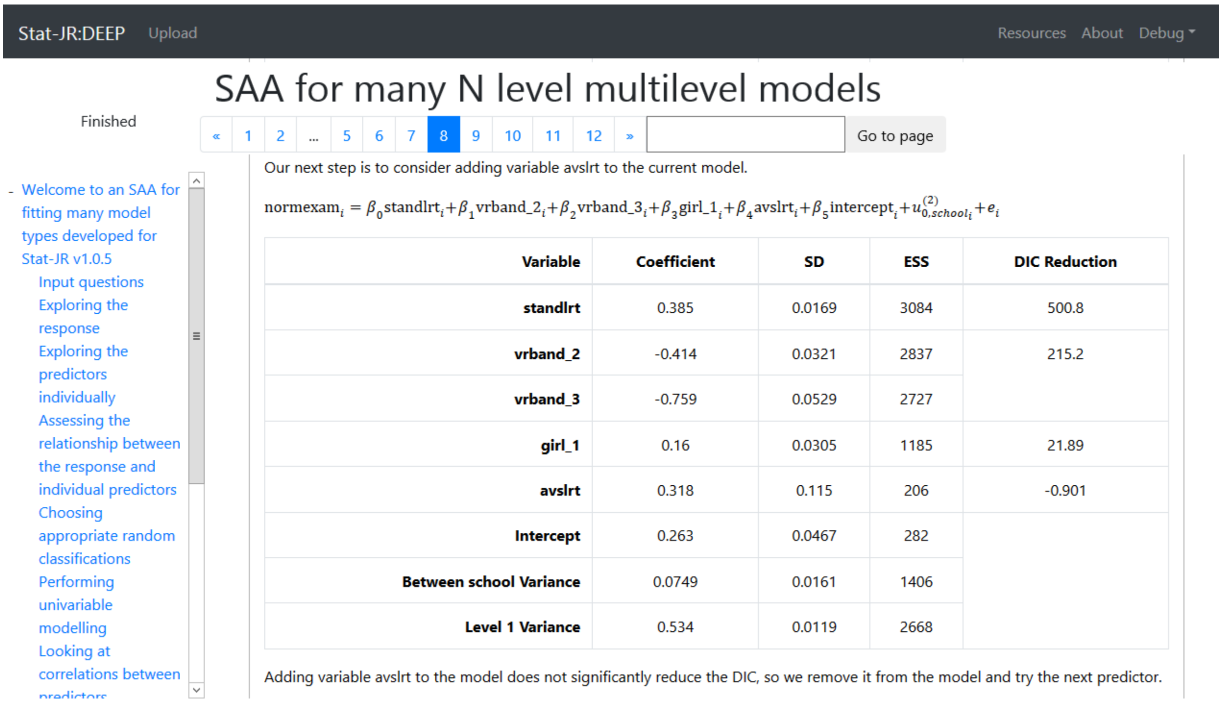
Our next step in modelling now that we have a set of potential predictors is to consider models for each predictor in turn along with a random intercept at each chosen classification from the best model in the last section. In the fixed part these models simply contain an intercept and the particular predictor and so for continuous predictors will be multilevel linear regressions and for categorical predictors will be multilevel generalisations of ANOVAs. In the table below we summarise the modelling by showing the coefficients for each predictor along with the p value comparing the model with that predictor with a Null model. This Univariable modelling step will identify a set of candidate predictors to be taken forward into the next stage of modelling.

Variable	Coefficient	SD	ESS	DIC reduction
standlrt	0.563	0.0126	3997	1641.0
avslrt	0.931	0.112	366	3.075
girl_1	0.263	0.0406	1414	38.06
schgend_2	0.0685	0.15	295	0.00963
schgend_3	0.254	0.114	216	
vrband_2	-0.819	0.0281	4593	1352.0
vrband_3	-1.614	0.0425	4264	

Which predictors we consider for the next stage of analysis will depend on their significance in the above table (but may in practice also depend on the size the effect and substantive interest of the variable though this is hard to automate). We will compare the DIC of the model including the predictor with the base model that does not to determine the significance of the predictor. Here we are using a difference in DIC of 1 to mean a significant improvement. so the predictors to carry forward are: girl, avslrt, standlrt, and vrband.

Prediction plot for normexam against standlrt (school)
2.0
1.5
1.0
0.5
Prediction plot for normexam against avslrt (school)
1.0
0.5

On page 8 we look at various models where we include additional terms and then test whether the model is better. Here we see one of the challenges of using the DIC to judge better models. If we look at the model where *avslrt* is added we see the following table. The table shows a model with *avslrt* included along with the already included predictors and for each predictor an estimated coefficient with posterior SD and ESS.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors

SAA for many N level multilevel models

Our next step is to consider adding variable avslrt to the current model.

$$\text{normexam}_i = \beta_0\text{standlrt}_i + \beta_1\text{vrband\_2}_i + \beta_2\text{vrband\_3}_i + \beta_3\text{girl\_1}_i + \beta_4\text{avslrt}_i + \beta_5\text{intercept}_i + u^{(2)}_{0,school_i} + e_i$$

Variable	Coefficient	SD	ESS	DIC Reduction
standlrt	0.385	0.0169	3084	500.8
vrband_2	-0.414	0.0321	2837	215.2
vrband_3	-0.759	0.0529	2727	
girl_1	0.16	0.0305	1185	21.89
avslrt	0.318	0.115	206	-0.901
Intercept	0.263	0.0467	282	
Between school Variance	0.0749	0.0161	1406	
Level 1 Variance	0.534	0.0119	2668	

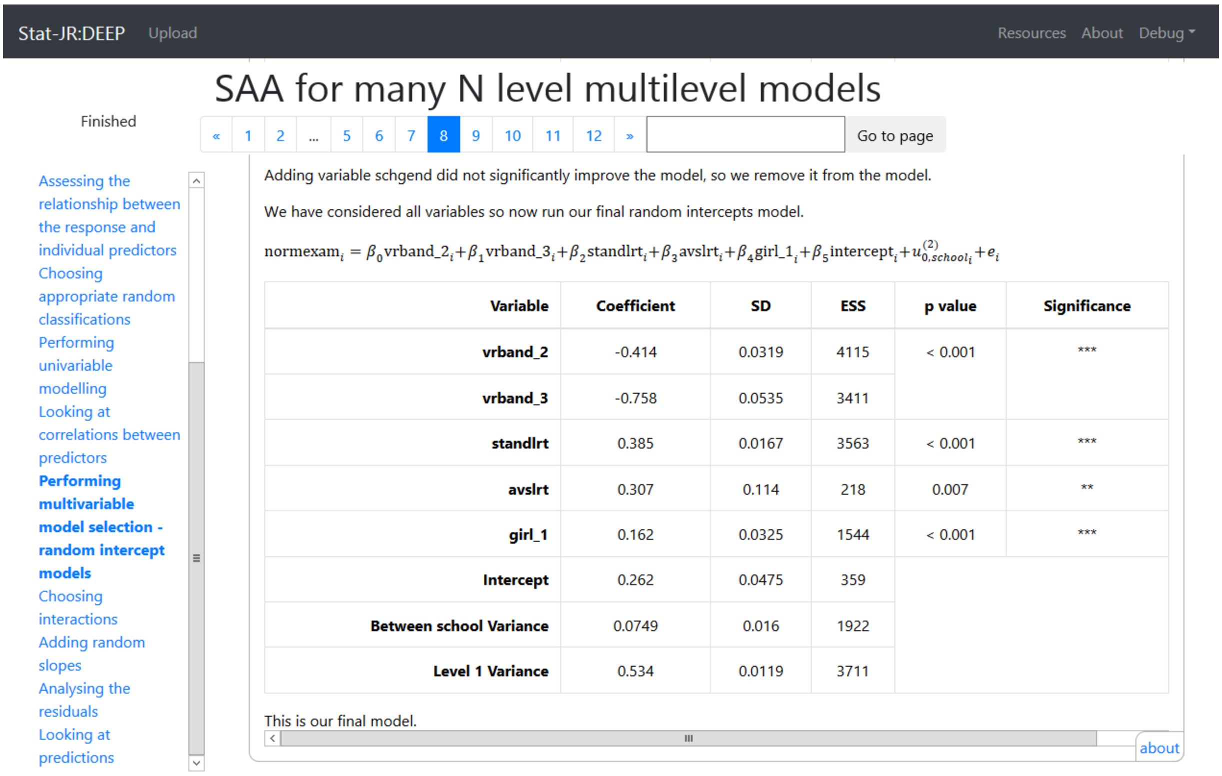
Adding variable avslrt to the model does not significantly reduce the DIC, so we remove it from the model and try the next predictor.

There is then a DIC reduction column which compares the fitted model with a model with that term removed (this is why DIC is quite computationally expensive as for this model it had to run an additional 4 models to assess change in DIC!). Now removing *avslrt* has very little impact on DIC (in fact it reduces it) and so using DIC as a criterion for model fitting means we remove *avslrt* from the model. This is in contrast with what we found using IGLS but note that the coefficient for *avslrt* is 0.318 with associated SD of 0.115 (and so the estimate is more than 2 SDs in magnitude) and so using the MCMC estimates and Wald estimation we would also include the term. The reason for this discrepancy is that the DIC has what is known as a focus which loosely means the level at which we are trying to maximise fit and in the SAA this focus is the level of the observation (level 1). Now *avslrt* is a level 2 predictor being constant for a school and so when added to the model although it is significant it is essentially explaining some of the between school variance which is reduced from 0.0872 to 0.0749 as a result here while not affecting the level 1 variance which is 0.533 and 0.534 respectively. This means that adding *avslrt* has not improved things at level 1 and so hence is rejected. This is the motivation behind offering the Wald test alternative which we will visit in a minute. Note that another alternative not offered is the concept of a Bayesian P value which is effectively a non-parametric version of the Wald test based on the chain. The reason we haven’t implemented this is the challenge that adding groups of variables e.g. a categorical predictor presents as a Bayesian P value is associated with a single parameter rather than a group. This is a topic for further research.

The model fitting continues and on pages 11 and 12 we see residuals and prediction plots as usual. We have saved this eBook as *tutmcmcridic.pdf* and now we will illustrate the alternative method of using Wald tests. Basically we use all the same inputs but change *DIC* to *Wald* and remember to say *No* to logging on page 2. The SAA should then run a little quicker than when DIC was chosen and eventually finish executing.

It should be noted that even when you choose Wald that for the section on page 5 where the best set of classifications is required DIC is used as the normal assumption for parameters is less plausible for variances and also when there are several classifications there is not necessarily a hierarchy of models to compare and find a best model.

If we now look at page 8 we see the following final model:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Adding variable schgend did not significantly improve the model, so we remove it from the model.

We have considered all variables so now run our final random intercepts model.

$$\text{normexam}_i = \beta_0\text{vrband\_2}_i + \beta_1\text{vrband\_3}_i + \beta_2\text{standlrt}_i + \beta_3\text{avslrt}_i + \beta_4\text{girl\_1}_i + \beta_5\text{intercept}_i + u^{(2)}_{0,school_i} + e_i$$

Variable	Coefficient	SD	ESS	p value	Significance
vrband_2	-0.414	0.0319	4115	< 0.001	***
vrband_3	-0.758	0.0535	3411		
standlrt	0.385	0.0167	3563	< 0.001	***
avslrt	0.307	0.114	218	0.007	**
girl_1	0.162	0.0325	1544	< 0.001	***
Intercept	0.262	0.0475	359		
Between school Variance	0.0749	0.016	1922		
Level 1 Variance	0.534	0.0119	3711		

This is our final model.

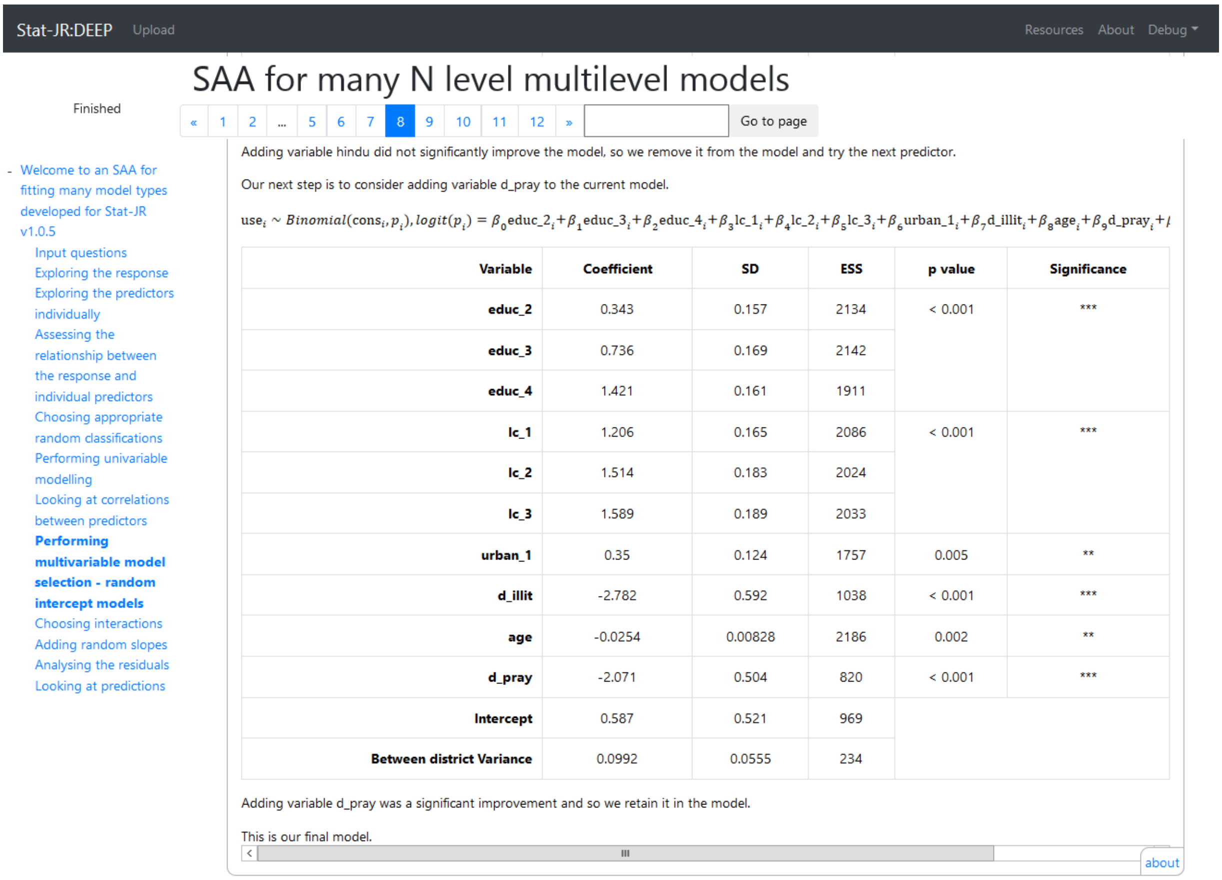
This model is now similar to that which we got earlier from IGLS. We will save this eBook run as *tutmcmcriwald.pdf*.

8.3 - Using MCMC for logistic models

We will continue by fitting models to the Bangladesh dataset using MCMC estimation. We will not consider quite the same inputs here and remove the option for random slopes models. This is because there can be issues with random slopes models with categorical predictors which come to light when using MCMC estimation. To illustrate the issue consider fitting a model with one binary predictor say *urban* and fitting a random slopes model so we have a random intercept and a random urban effect (as effectively happens in the earlier IGLS fitting of the Bangladesh dataset). Now the SAA fits full covariance matrices so we will have 3 terms to represent variation at level 2: the intercept variance, the urban slope variance and the covariance between them. In reality what we are trying to fit is a model with different variances at the district level for women in urban and women in rural areas and as women cannot be in both we really only need 2 terms to do this and so the covariance is redundant. In a normal model IGLS will realise this and set one of the terms to be 0 but in a Binomial model quasi-likelihood is used and so some values are given for all 3 terms but note that if you look at the correlation between the two sets of residuals it comes out as greater than 1! MCMC is less forgiving and rather than recognise the difficulty here will get stuck. It is perfectly acceptable to use random slopes for continuous predictors and so there are ways to do just this which we will program into the SAA later but for now we will simply not include the slopes.

So returning to the modelling here we choose dataset *bang1*; estimation method *MCMC;* response variable *use*; distribution *Binomial*; denominators *cons*; link function *logit*; higher classifications *district* (only); *No* to the two include in all models questions; for continuous predictors *Yes* and *age, d_illit* and *d_pray* and for categorical predictors *Yes* and *lc, urban, educ* and *hindu*; *Forward pass* for selection type; *No* for random slopes but *Yes* for interactions; *Wald* for model comparison, *500* burnin, *2000* iterations and minimum ESS of *200*; *Yes* for orthogonal parameterisation and finally *1* for change of DIC. This eBook run takes a very long time (an hour or so!) compared to the ones done thus far.

The final random intercept model on page 8 has exactly the same predictors as for IGLS and is as follows:



SAA for many N level multilevel models

- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
  - Input questions
  - Exploring the response
  - Exploring the predictors individually
  - Assessing the relationship between the response and individual predictors
  - Choosing appropriate random classifications
  - Performing univariable modelling
  - Looking at correlations between predictors
  - **Performing multivariable model selection - random intercept models**
  - Choosing interactions
  - Adding random slopes
  - Analysing the residuals
  - Looking at predictions

Adding variable hindu did not significantly improve the model, so we remove it from the model and try the next predictor.

Our next step is to consider adding variable d_pray to the current model.

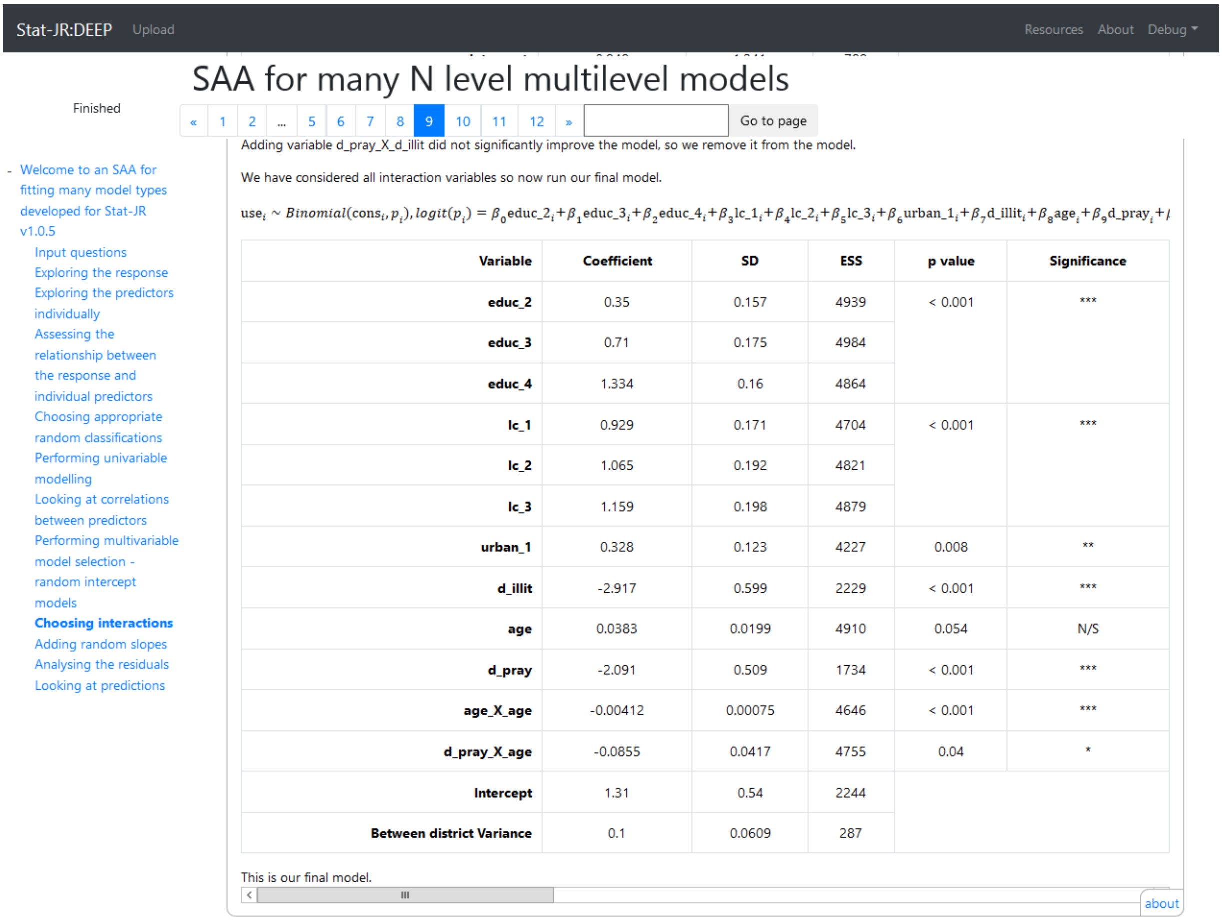
$\text{use}_i \sim Binomial(\text{cons}_i, p_i), logit(p_i) = \beta_0\text{educ\_2}_i + \beta_1\text{educ\_3}_i + \beta_2\text{educ\_4}_i + \beta_3\text{lc\_1}_i + \beta_4\text{lc\_2}_i + \beta_5\text{lc\_3}_i + \beta_6\text{urban\_1}_i + \beta_7\text{d\_illit}_i + \beta_8\text{age}_i + \beta_9\text{d\_pray}_i +$

Variable	Coefficient	SD	ESS	p value	Significance
educ_2	0.343	0.157	2134	< 0.001	***
educ_3	0.736	0.169	2142		
educ_4	1.421	0.161	1911		
lc_1	1.206	0.165	2086	< 0.001	***
lc_2	1.514	0.183	2024		
lc_3	1.589	0.189	2033		
urban_1	0.35	0.124	1757	0.005	**
d_illit	-2.782	0.592	1038	< 0.001	***
age	-0.0254	0.00828	2186	0.002	**
d_pray	-2.071	0.504	820	< 0.001	***
Intercept	0.587	0.521	969		
Between district Variance	0.0992	0.0555	234		

Adding variable d_pray was a significant improvement and so we retain it in the model.

This is our final model.

Then when adding in interactions on page 9 we again end up with a model with a quadratic term for age and one interaction between *age* and *d_pray* as shown below:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Adding variable d_pray_X_d_illit did not significantly improve the model, so we remove it from the model.

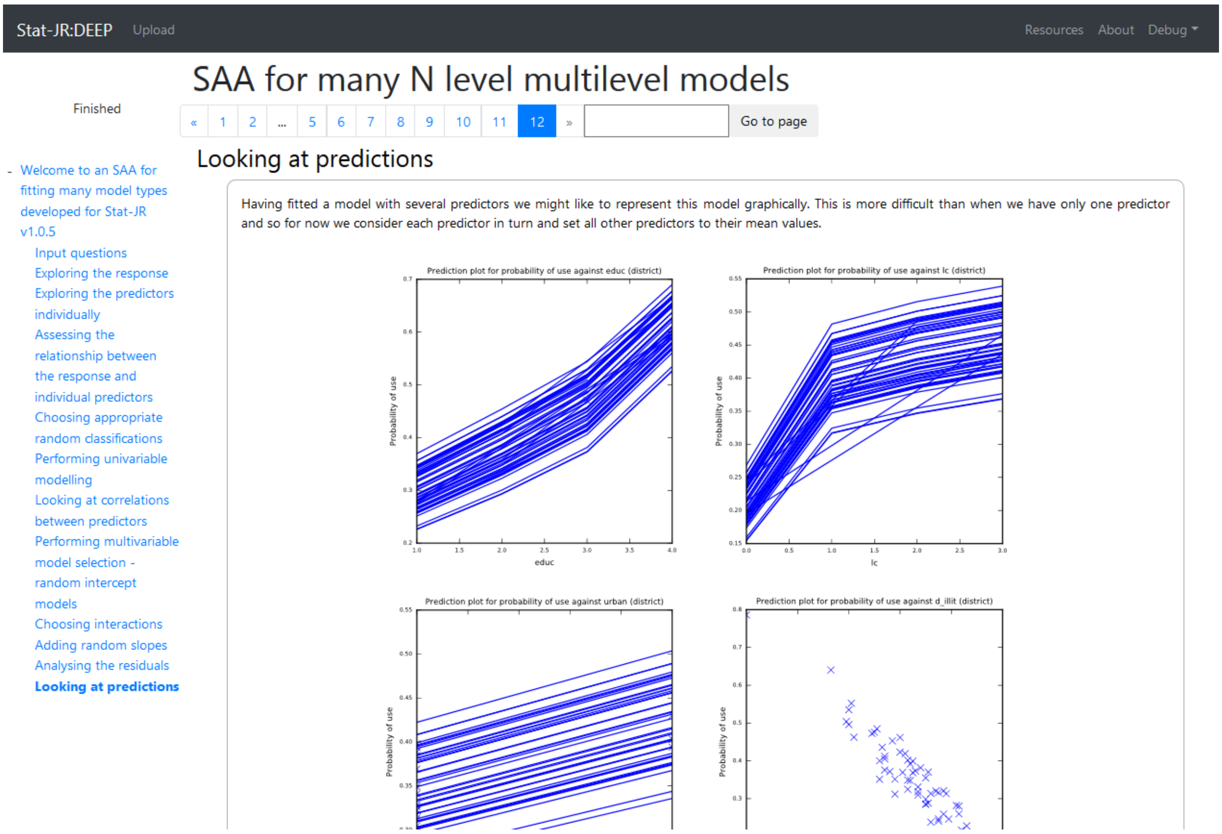
We have considered all interaction variables so now run our final model.

$\text{use}_i \sim Binomial(\text{cons}_i, p_i), logit(p_i) = \beta_0\text{educ\_2}_i + \beta_1\text{educ\_3}_i + \beta_2\text{educ\_4}_i + \beta_3\text{lc\_1}_i + \beta_4\text{lc\_2}_i + \beta_5\text{lc\_3}_i + \beta_6\text{urban\_1}_i + \beta_7\text{d\_illit}_i + \beta_8\text{age}_i + \beta_9\text{d\_pray}_i +$

Variable	Coefficient	SD	ESS	p value	Significance
educ_2	0.35	0.157	4939	< 0.001	***
educ_3	0.71	0.175	4984		
educ_4	1.334	0.16	4864		
lc_1	0.929	0.171	4704	< 0.001	***
lc_2	1.065	0.192	4821		
lc_3	1.159	0.198	4879		
urban_1	0.328	0.123	4227	0.008	**
d_illit	-2.917	0.599	2229	< 0.001	***
age	0.0383	0.0199	4910	0.054	N/S
d_pray	-2.091	0.509	1734	< 0.001	***
age_X_age	-0.00412	0.00075	4646	< 0.001	***
d_pray_X_age	-0.0855	0.0417	4755	0.04	*
Intercept	1.31	0.54	2244		
Between district Variance	0.1	0.0609	287		

This is our final model.

Finally the prediction plots look similar aside from the one for *urban* where we do not allow random slopes:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
Prediction plot for probability of use against educ (district)
Probability of use
educ
0.2
0.3
0.4
0.5
0.6
0.7
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Prediction plot for probability of use against lc (district)
Probability of use
lc
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Prediction plot for probability of use against urban (district)
Probability of use
0.35
0.40
0.45
0.50
0.55
Prediction plot for probability of use against d_illit (district)
Probability of use
0.3
0.4
0.5
0.6
0.7
0.8

SAA for many N level multilevel models

Looking at predictions

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

This eBook run has been stored as *bangmcmc.pdf.*

Chapter 9 – Three level models and cross-classified models

In the examples we have looked at thus far we have restricted ourselves to one random classification in our dataset. In practice the Combined SAA can handle any number of random classifications and these classifications can be either nested in a hierarchy or more generally crossed with each other.

For nested classifications we have the option to use either IGLS or MCMC estimation whereas for crossed modelling we are restricted to MCMC estimation only. We will therefore begin with an example of a 3-level nested model – the A-level chemistry dataset.

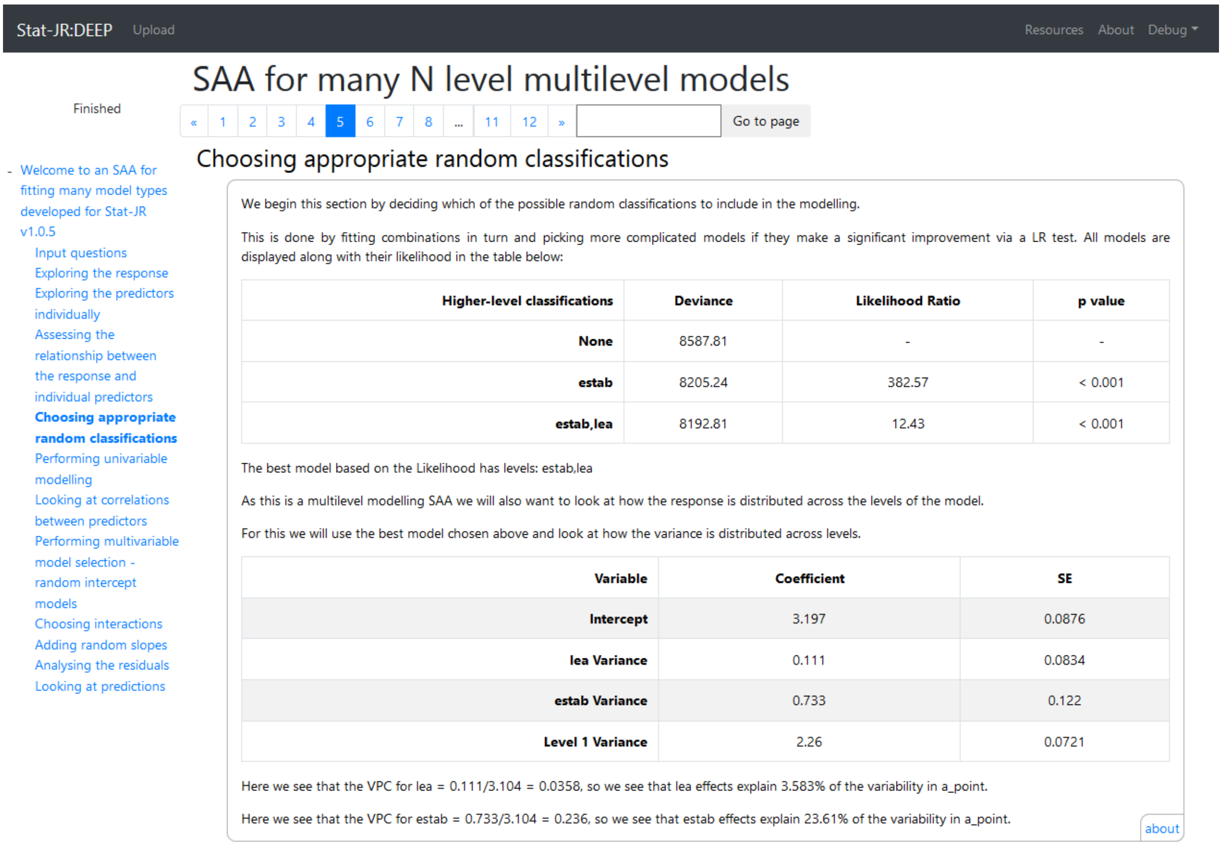
9.1 - A level Chemistry 3-level model

The A level chemistry dataset (Yang & Woodhouse, 2001) is an education dataset that consists of data on 2166 pupils from 219 schools and 70 local education authorities (LEAs) from 1997. The response is the score on a chemistry exam taken at age 18 by the students and in fact we often use this dataset to illustrate ordered category models as the response *a-point* only takes values 1-6 corresponding to grades F to A here (so our assumption of Normality here is a little suspect!). We have as possible predictor variables the number of GCSE exams (exams at age 16) the pupil took *gcse-no* and their total points on these exams *gcse-tot* as well as the pupil's *gender*. There are other things one might do with these data for example construct *gcse-av* but for now we will just use these 3 raw variables (see (Browne W. , MCMC Estimation in MLwiN v3.00, 2017) for an ordered response analysis of the data using MCMC).

Our inputs for the Combined SAA in DEEP are then as follows:

Dataset – *alevchem*; Estimation method – *IGLS*; response – *a_point*; distribution – *Normal*; classifications (in this order) – *estab, lea*; always include continuous – *No*; always include categorical – *No*; include continuous – *Yes*; continuous predictors – *gcse_tot, gcse_no*; include categorical – *Yes*; categorical predictors – *gender*; selection – *Forward pass*; random slopes – *Yes*; interactions – *Yes*; Compare Models – *Likelihood Ratio*; log variable (on page 2) – *No*.

The SAA will take a little while to run but we will see on page 5 the start of the main analysis:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting combinations in turn and picking more complicated models if they make a significant improvement via a LR test. All models are displayed along with their likelihood in the table below:

Higher-level classifications	Deviance	Likelihood Ratio	p value
None	8587.81	-	-
estab	8205.24	382.57	< 0.001
estab,lea	8192.81	12.43	< 0.001

The best model based on the Likelihood has levels: estab,lea

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

For this we will use the best model chosen above and look at how the variance is distributed across levels.

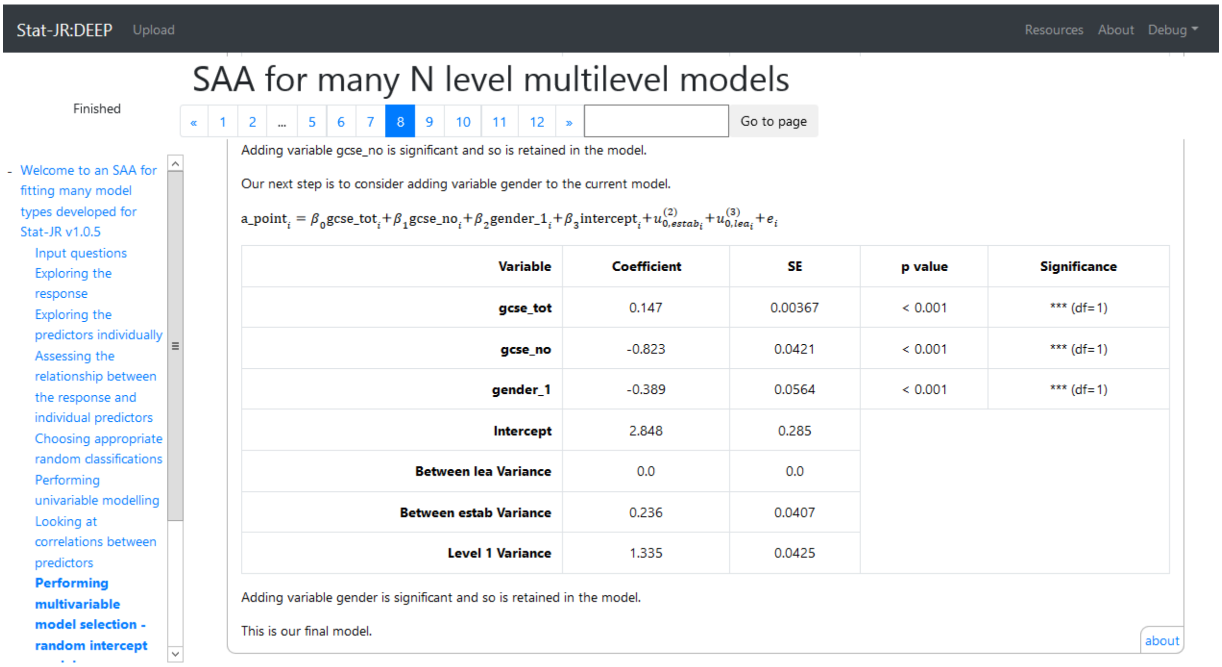
Variable	Coefficient	SE
Intercept	3.197	0.0876
lea Variance	0.111	0.0834
estab Variance	0.733	0.122
Level 1 Variance	2.26	0.0721

Here we see that the VPC for lea = 0.111/3.104 = 0.0358, so we see that lea effects explain 3.583% of the variability in a_point.

Here we see that the VPC for estab = 0.733/3.104 = 0.236, so we see that estab effects explain 23.61% of the variability in a_point.

Here we see that the levels are added in turn and that there is a big improvement in adding school (*estab*) effects and a smaller but still significant improvement from adding in LEA (*lea*) effects. This is backed up in the second table where we see that school and LEA explain 23.6% and 3.6% of the variability respectively. Having decided that the best model has both classifications the SAA then continues as with the earlier normal models but with the 3-level structure as the base model.

On page 8 we find that the best model includes all 3 predictor variables:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept
about

SAA for many N level multilevel models

Adding variable gcse_no is significant and so is retained in the model.

Our next step is to consider adding variable gender to the current model.

$$\text{a\_point}_i = \beta_0\text{gcse\_tot}_i + \beta_1\text{gcse\_no}_i + \beta_2\text{gender\_1}_i + \beta_3\text{intercept}_i + u^{(2)}_{0,estab_i} + u^{(3)}_{0,lea_i} + e_i$$

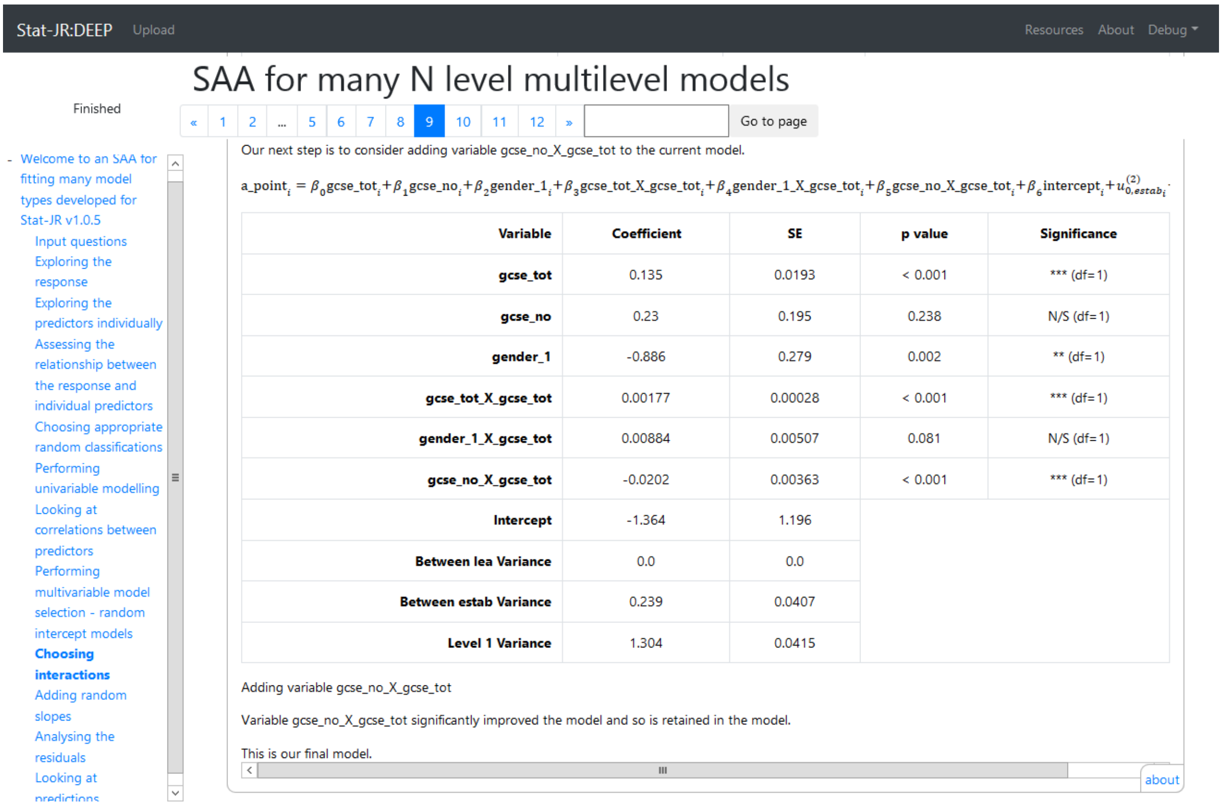
Variable	Coefficient	SE	p value	Significance
gcse_tot	0.147	0.00367	< 0.001	*** (df=1)
gcse_no	-0.823	0.0421	< 0.001	*** (df=1)
gender_1	-0.389	0.0564	< 0.001	*** (df=1)
Intercept	2.848	0.285		
Between lea Variance	0.0	0.0		
Between estab Variance	0.236	0.0407		
Level 1 Variance	1.335	0.0425		

Adding variable gender is significant and so is retained in the model.

This is our final model.

It is worth noting that here having added the 3 predictor variables we find the between LEA variance estimated as zero i.e. the predictors have explained all the variability between LEAs. One difficulty in writing an automatic SAA is that we have to have a set workflow underlying the SAA and we do not test the significance of the higher-level effects once we add predictor variables and so the LEA effects are retained despite the zero variance. That said the model fit will be same as if IGLS simply fitted a model with school random effects only and you could confirm this by repeating the analysis but only adding *estab*.

Moving on to the inclusion of interactions we end up with following model on page 9:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

Our next step is to consider adding variable gcse_no_X_gcse_tot to the current model.

$$\text{a\_point}_i = \beta_0\text{gcse\_tot}_i + \beta_1\text{gcse\_no}_i + \beta_2\text{gender\_1}_i + \beta_3\text{gcse\_tot\_X\_gcse\_tot}_i + \beta_4\text{gender\_1\_X\_gcse\_tot}_i + \beta_5\text{gcse\_no\_X\_gcse\_tot}_i + \beta_6\text{intercept}_i + u^{(2)}_{0,estab_i}$$

Variable	Coefficient	SE	p value	Significance
gcse_tot	0.135	0.0193	< 0.001	*** (df=1)
gcse_no	0.23	0.195	0.238	N/S (df=1)
gender_1	-0.886	0.279	0.002	** (df=1)
gcse_tot_X_gcse_tot	0.00177	0.00028	< 0.001	*** (df=1)
gender_1_X_gcse_tot	0.00884	0.00507	0.081	N/S (df=1)
gcse_no_X_gcse_tot	-0.0202	0.00363	< 0.001	*** (df=1)
Intercept	-1.364	1.196		
Between lea Variance	0.0	0.0		
Between estab Variance	0.239	0.0407		
Level 1 Variance	1.304	0.0415		

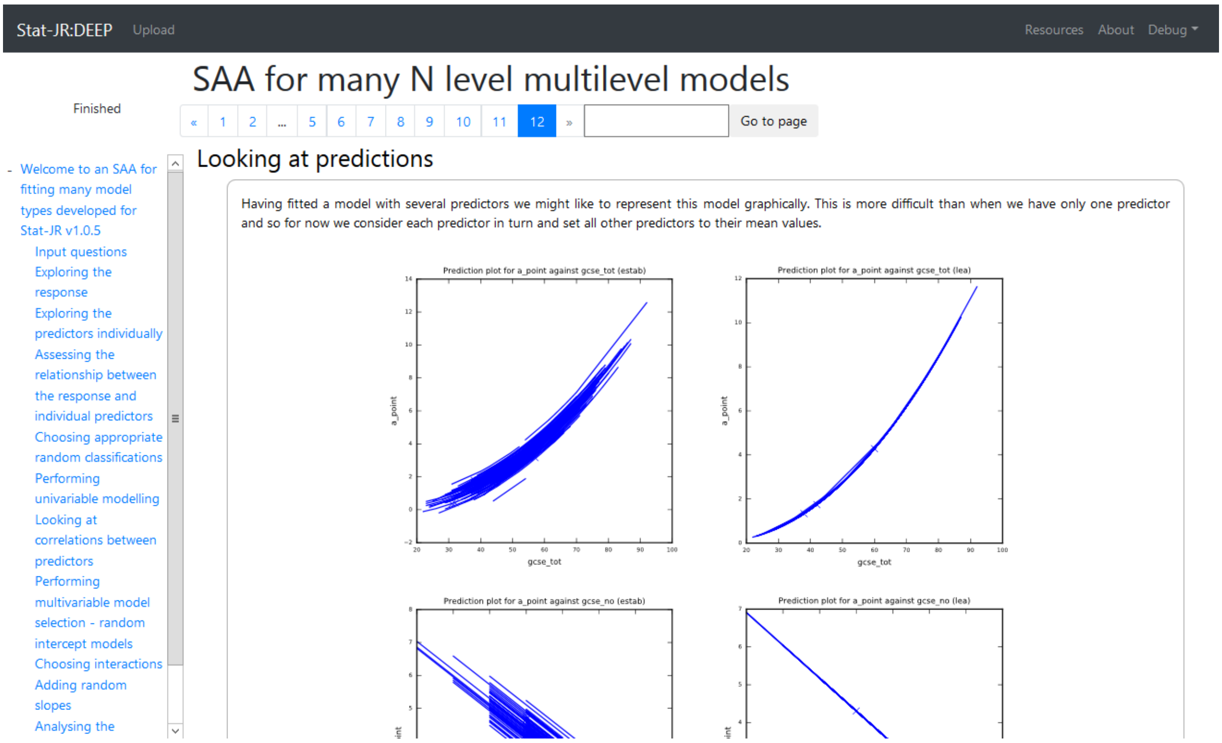
Adding variable gcse_no_X_gcse_tot

Variable gcse_no_X_gcse_tot significantly improved the model and so is retained in the model.

This is our final model.

Here the estimation results in including 3 interactions. It is noticeable that one of these turns out to be non-significant but as we are using the *forward-pass* method it is retained in the model as it was significant when it entered the model. If we had instead used in addition *Backward Elimination* then the model fitting would have continued and this term would have been removed from the model.

On page 10 we test for random slopes at either the school or LEA level but none are significant so we are left with the model from page 9. Page 11 gives the residual plots at all 3 levels but note that as the between LEA variance is zero at this level all residuals are zero! On page 12 we see the plots of the fitted values at each of the two levels:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
Prediction plot for a_point against gcse_tot (estab)
Prediction plot for a_point against gcse_tot (lea)
a_point
gcse_tot
Prediction plot for a_point against gcse_no (estab)
Prediction plot for a_point against gcse_no (lea)

SAA for many N level multilevel models

- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
  - Input questions
  - Exploring the response
  - Exploring the predictors individually
  - Assessing the relationship between the response and individual predictors
  - Choosing appropriate random classifications
  - Performing univariable modelling
  - Looking at correlations between predictors
  - Performing multivariable model selection - random intercept models
  - Choosing interactions
  - Adding random slopes
  - Analysing the

Looking at predictions

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

Note again that at the LEA level these are simply the fixed part prediction as there are no LEA effects. Note also the quadratic effect for GCSE total. In practice we see a positive effect of total score at GCSE while the number of GCSEs taken has a negative effect once accounting for the total score which makes sense i.e. if two pupils scored 40 points from 8 GCSEs and 40 points from 9 GCSES respectively then the first student has a better average score. Also in this case girls do worse than boys on average in chemistry.

The full SAA output can be found in pdf format in the file *alevchem.pdf*.

9.2 Cross-classified modelling

The other way that we might consider extending 2 level models is to add a second higher level classification that is not nested within the first. In our example here we look at the *xc* dataset (Paterson, 1991) which is another educational example, this time the data is from Fife in Scotland. The data consists of 3,435 children from 19 secondary schools who previously attended 148 primary schools. Our response variable is an attainment score, *attain* (from 1 to 10) from a test taken at school leaving at 16. We have several predictor variables to test: an earlier verbal reasoning test (*vrq*) and the child's social class (*sc*) which we will treat as continuous, and four categorical variables – gender (*sex*), father's education (*fed*), mother's education (*med*) and choice of secondary school (*choice*) where 1 is first choice etc. The dataset is cross-classified as primary schools and secondary schools are not nested classifications.

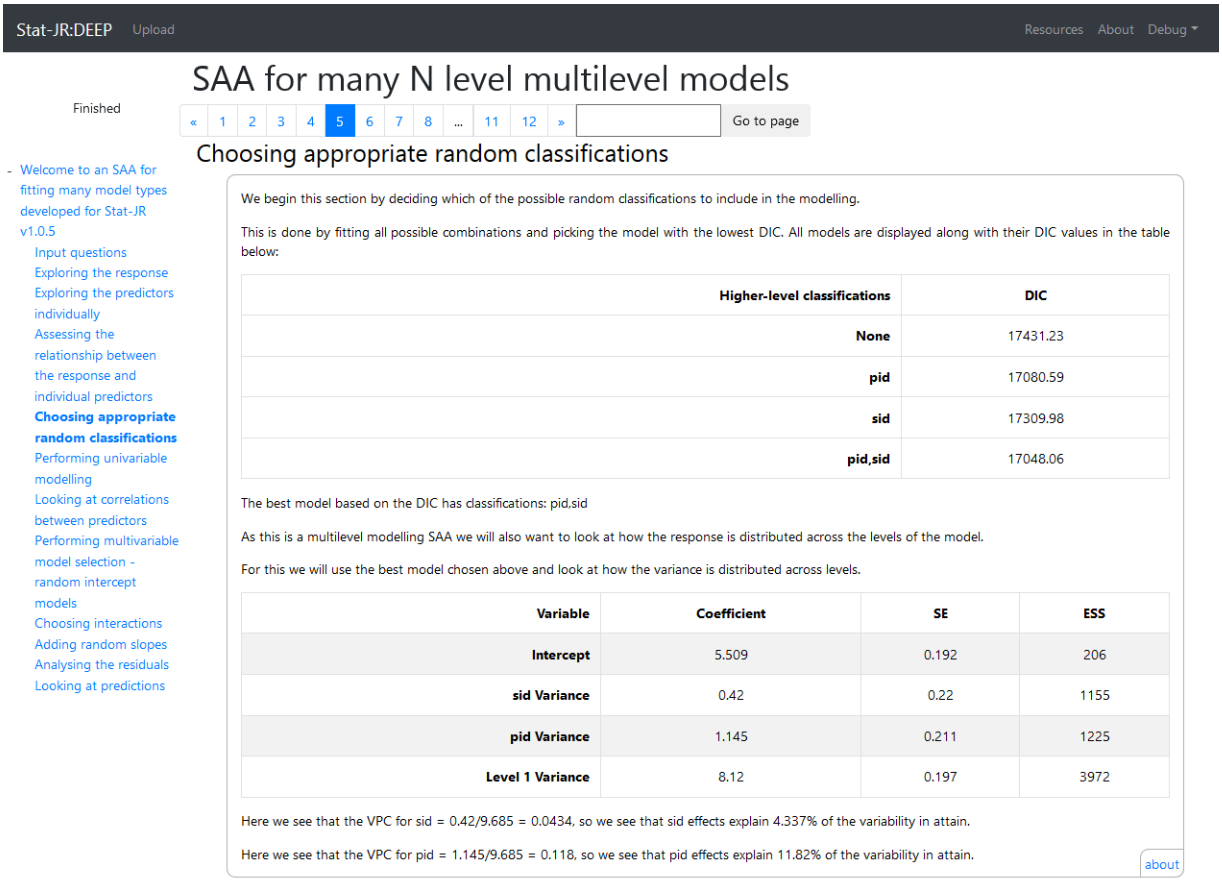
We will illustrate using MCMC estimation on this dataset but for speed we will only consider random intercept models without interactions. This will avoid the issue that MCMC can have with categorical random slopes but mainly is to fit the eBook more quickly.

The inputs we use are as follows:

Dataset – *xc*; Estimation method – *MCMC*; response – *attain*; distribution – *Normal*; classifications – *pid,sid*; always include continuous – *No*; always include categorical – *No*; include continuous – *Yes*; continuous predictors – *vrq,sc*; include categorical – *Yes*; categorical predictors – *sex,fed,choice,med*;

selection – *Forward pass*; random slopes – *No*; interactions – *No*; Compare Models – *Wald*; burnin – *500*; iterations – *2000*; minimum ESS – *200*; orthogonal parameterisation – *No*; Change in DIC – *1*; log variable (on page 2) – *No*.

After the SAA finishes running we can look at page 5 to see what it thinks is best base model:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
about

- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting all possible combinations and picking the model with the lowest DIC. All models are displayed along with their DIC values in the table below:

Higher-level classifications	DIC
None	17431.23
pid	17080.59
sid	17309.98
pid,sid	17048.06

The best model based on the DIC has classifications: pid,sid

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

For this we will use the best model chosen above and look at how the variance is distributed across levels.

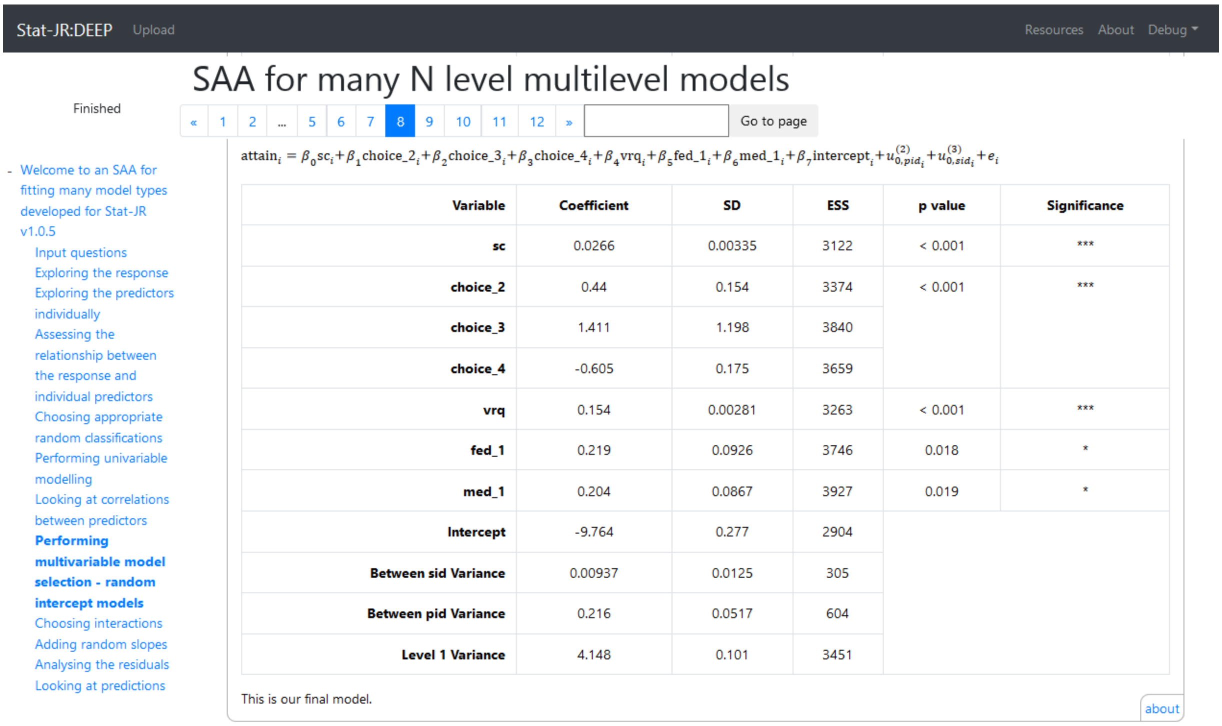
Variable	Coefficient	SE	ESS
Intercept	5.509	0.192	206
sid Variance	0.42	0.22	1155
pid Variance	1.145	0.211	1225
Level 1 Variance	8.12	0.197	3972

Here we see that the VPC for sid = 0.42/9.685 = 0.0434, so we see that sid effects explain 4.337% of the variability in attain.

Here we see that the VPC for pid = 1.145/9.685 = 0.118, so we see that pid effects explain 11.82% of the variability in attain.

Here we use DIC to determine the best model as we don't have a natural nesting of the models that we choose between and so we see that the lowest DIC is for *pid,sid* i.e. both primary and secondary school effects. We see below that primary school explains 11.9% of the variability with secondary school explaining 4.3%.

On page 8 we can then see the model fitting and we see that all predictors apart from gender are in our final model:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

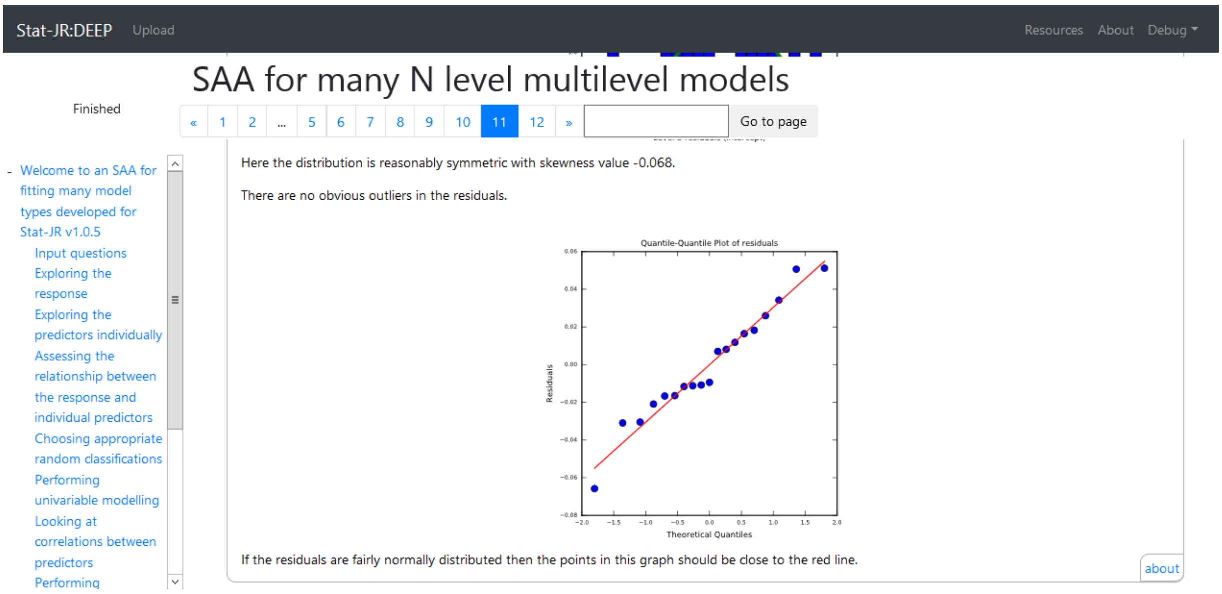
SAA for many N level multilevel models

$\text{attain}_i = \beta_0\text{sc}_i + \beta_1\text{choice\_2}_i + \beta_2\text{choice\_3}_i + \beta_3\text{choice\_4}_i + \beta_4\text{vrq}_i + \beta_5\text{fed\_1}_i + \beta_6\text{med\_1}_i + \beta_7\text{intercept}_i + u^{(2)}_{0,pid_i} + u^{(3)}_{0,sid_i} + e_i$

Variable	Coefficient	SD	ESS	p value	Significance
sc	0.0266	0.00335	3122	< 0.001	***
choice_2	0.44	0.154	3374	< 0.001	***
choice_3	1.411	1.198	3840		
choice_4	-0.605	0.175	3659		
vrq	0.154	0.00281	3263	< 0.001	***
fed_1	0.219	0.0926	3746	0.018	*
med_1	0.204	0.0867	3927	0.019	*
Intercept	-9.764	0.277	2904		
Between sid Variance	0.00937	0.0125	305		
Between pid Variance	0.216	0.0517	604		
Level 1 Variance	4.148	0.101	3451		

This is our final model.

Here we see all three variances are reduced by fitting the predictor variables and in fact the secondary school effects are greatly reduced. When Browne (Browne W. , MCMC Estimation in MLwiN v3.00, 2017) analyse this dataset they also fit the final model given here but looked at the residual plots for the secondary schools - given on page 11 of the SAA and shown below:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing
Quantile-Quantile Plot of residuals
Residuals
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
Theoretical Quantiles
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
about

SAA for many N level multilevel models

Here the distribution is reasonably symmetric with skewness value -0.068.

There are no obvious outliers in the residuals.

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

They go on to put in a dummy variable for the lowest performing school (school 19) and find that fitting this instead of the school effects results in a model with a lower DIC but the SAA as currently written cannot make such circuitous decisions.

We store the full output of this analysis in *xc1.pdf*.

Chapter 10 – Other features of the Combined SAA

10.1 - More response types – Poisson models

In our final modelling example using the Combined SAA we consider the third response distribution offered – Poisson responses that are used for fitting count data.

Here our example is the *mmmec* dataset (Langford, Bentham, & McDonald, 1998) which is a public health dataset and looks at melanoma mortality in the European community in the period 1971 to 1980. The response, *obs* is therefore the number of deaths in particular geographic areas – in fact we have three levels of geography with the observations measured at the lowest level (labelled *county*) nested within regions (*region*) of which there are 79 nested within countries (*nation*) of which there are only 9. Poisson models are typically fit with a log link and so the effects of the predictor variables then become multiplicative rather than additive which makes sense as we might anticipate a predictor variable doubling the count in areas rather than adding 2 to them. For Poisson models on geographical data we are often interested in rates rather than raw counts as otherwise the model might simply show that larger population centres have larger number of cancer deaths. To do this Poisson models as well as using a log link also introduce an offset parameter which is a fixed effect with coefficient fixed to 1 that is typically based on the expected counts adjusting for population size and makeup. Usually the offset will be the log of the expected counts and then the model is

Log(obs*) = log(exp) + XB

or using rules of logs,

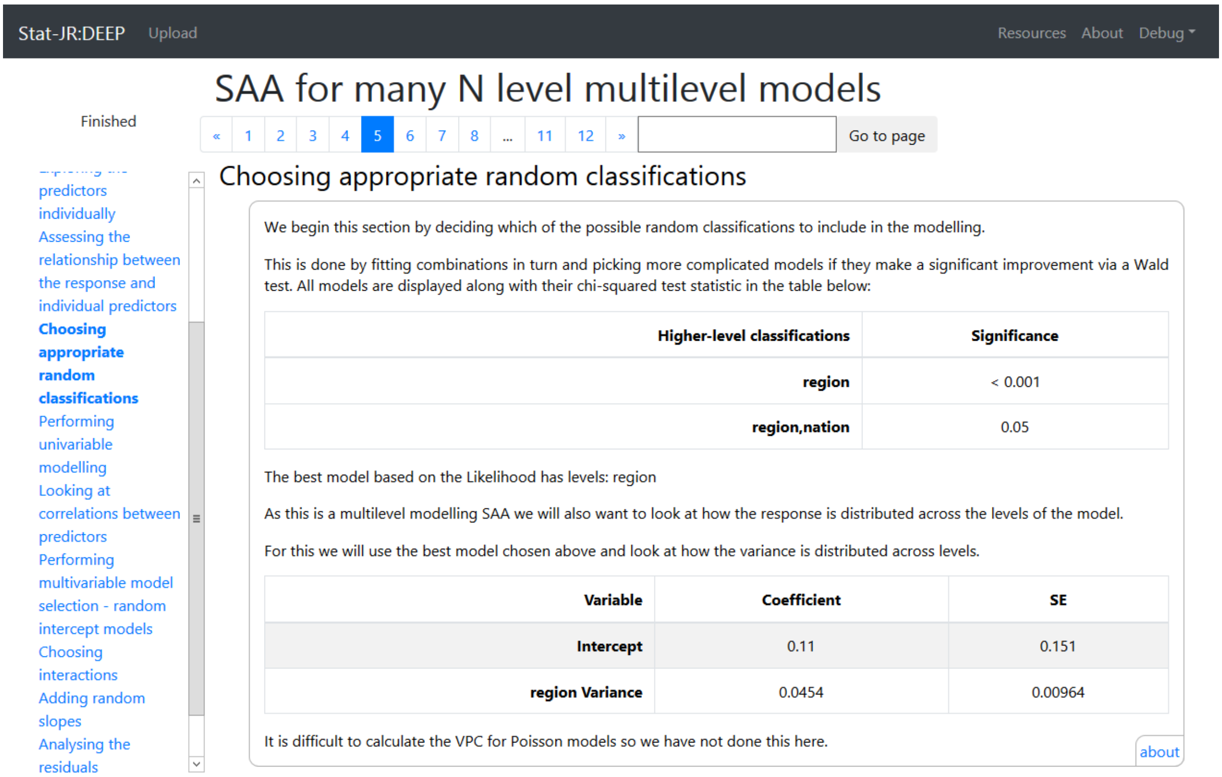
Log (obs*/exp) = XB

where obs* is the estimated observed counts and so we now have a response that is approximately the ratio of observed counts to expected. Let's see this in practice by using the SAA on this dataset.

The inputs are as follows:

Dataset –*mmmec*; Estimation method – *IGLS*; response – *obs*; distribution – *Poisson*; is offset – *Yes*; offset – *logexp*; classifications (in this order) – *region, nation*; always included continuous – *No*; always included categorical – *No*; include continuous – *Yes*; continuous predictors – *uvbi*; include categorical – *Yes*; categorical predictors – *nation*; selection – *Forward pass*; random slopes – *Yes*; interactions – *Yes*;

We will now wait for the SAA to run, noting that as the response is non-normal we are forced to use Wald tests for model comparison with IGLS and don't get asked about logging the response. Here also note we are taking the unusual step of trying to fit *nation* as both a random classification and as a categorical fixed effect (as this is how it is fitted in (Browne W. , MCMC Estimation in MLwiN v3.00, 2017)). This is because there are only 9 nations and so it will be hard to fit this as a level in the model as is evidenced on page 5 in the screenshot below.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
about

SAA for many N level multilevel models

Choosing appropriate random classifications

We begin this section by deciding which of the possible random classifications to include in the modelling.

This is done by fitting combinations in turn and picking more complicated models if they make a significant improvement via a Wald test. All models are displayed along with their chi-squared test statistic in the table below:

Higher-level classifications	Significance
region	< 0.001
region,nation	0.05

The best model based on the Likelihood has levels: region

As this is a multilevel modelling SAA we will also want to look at how the response is distributed across the levels of the model.

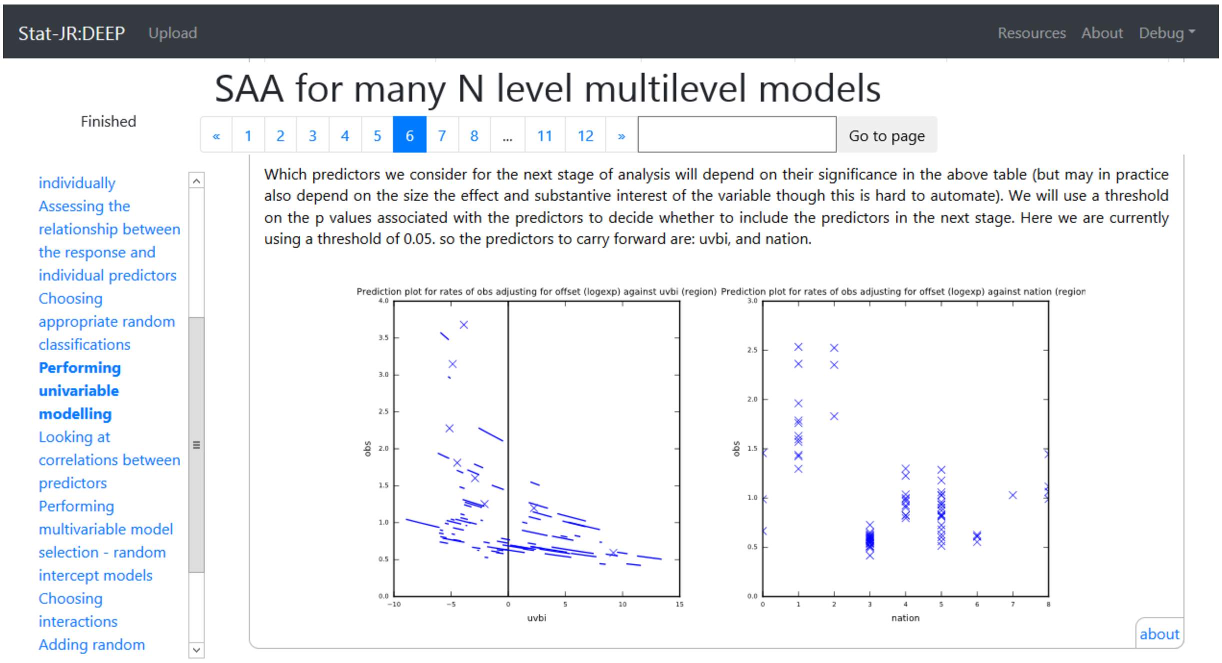
For this we will use the best model chosen above and look at how the variance is distributed across levels.

Variable	Coefficient	SE
Intercept	0.11	0.151
region Variance	0.0454	0.00964

It is difficult to calculate the VPC for Poisson models so we have not done this here.

Here we see the SAA decides to fit just region effects in terms of random effects though the decision is fairly marginal with the improvement of adding nations having a P value of 0.05 to 2 decimal places!

On page 6 we can see that models for each of the predictors in turn shows they are both significant and we get interesting prediction plots showing the exponentiated terms that then become the ratios of observed/expected

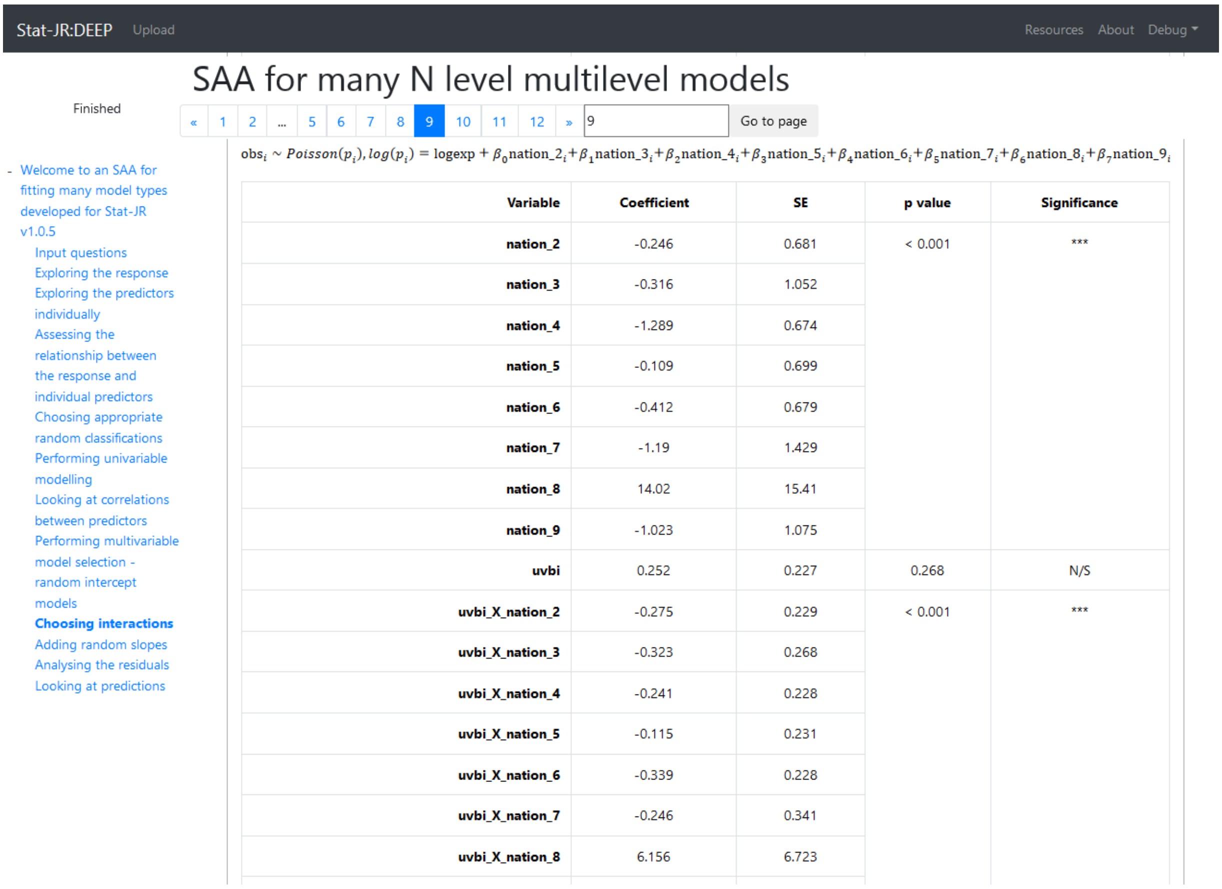

Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 3 4 5 6 7 8 ... 11 12 »
Go to page
individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random
Prediction plot for rates of obs adjusting for offset (logexp) against uvbi (region)
obs
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−10
−5
0
5
10
15
uvbi
Prediction plot for rates of obs adjusting for offset (logexp) against nation (region
obs
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
8
nation
about

SAA for many N level multilevel models

Which predictors we consider for the next stage of analysis will depend on their significance in the above table (but may in practice also depend on the size the effect and substantive interest of the variable though this is hard to automate). We will use a threshold on the p values associated with the predictors to decide whether to include the predictors in the next stage. Here we are currently using a threshold of 0.05. so the predictors to carry forward are: uvbi, and nation.

There are lines in the left-hand plot for each of the 79 regions whilst in the right-hand plot we get a point for each of the regions and so we can see the spread for each *nation*.

We can then see the modelling on pages 8-10. It transpires that there are no significant random slopes and so the final model is that at the end of page 9 with *nation* and *uvb* main effects and interactions between the two meaning different effects of UVBI in the different nations.

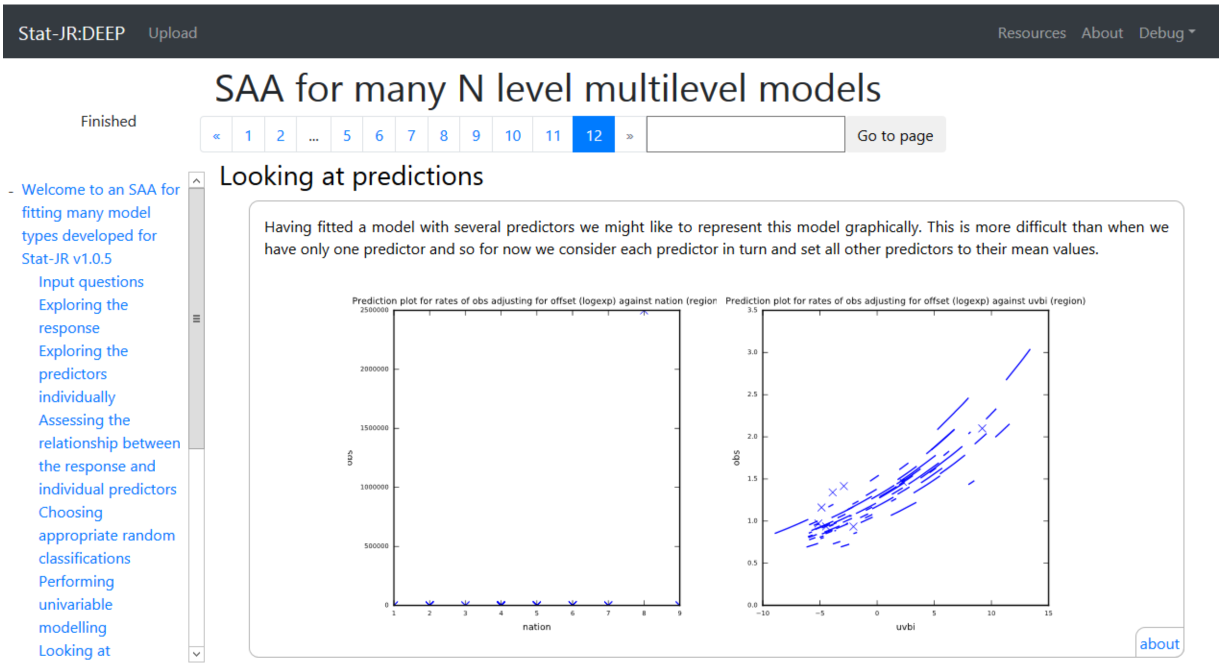

Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
9
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions

SAA for many N level multilevel models

$\mathrm{obs}_i \sim Poisson(p_i), log(p_i) = \mathrm{logexp} + \beta_0\mathrm{nation\_2}_i + \beta_1\mathrm{nation\_3}_i + \beta_2\mathrm{nation\_4}_i + \beta_3\mathrm{nation\_5}_i + \beta_4\mathrm{nation\_6}_i + \beta_5\mathrm{nation\_7}_i + \beta_6\mathrm{nation\_8}_i + \beta_7\mathrm{nation\_9}_i$

Variable	Coefficient	SE	p value	Significance
nation_2	-0.246	0.681	< 0.001	***
nation_3	-0.316	1.052		
nation_4	-1.289	0.674		
nation_5	-0.109	0.699		
nation_6	-0.412	0.679		
nation_7	-1.19	1.429		
nation_8	14.02	15.41		
nation_9	-1.023	1.075		
uvbi	0.252	0.227	0.268	N/S
uvbi_X_nation_2	-0.275	0.229	< 0.001	***
uvbi_X_nation_3	-0.323	0.268		
uvbi_X_nation_4	-0.241	0.228		
uvbi_X_nation_5	-0.115	0.231		
uvbi_X_nation_6	-0.339	0.228		
uvbi_X_nation_7	-0.246	0.341		
uvbi_X_nation_8	6.156	6.723		

On page 12 we see prediction plots for this model:


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
- Welcome to an SAA for fitting many model types developed for Stat-JR v1.0.5
Input questions
Exploring the response
Exploring the predictors individually
Assessing the relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at
Prediction plot for rates of obs adjusting for offset (logexp) against nation (region
2500000
2000000
1500000
1000000
500000
0
obs
1
2
3
4
5
6
7
8
9
nation
Prediction plot for rates of obs adjusting for offset (logexp) against uvbi (region)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
obs
−10
−5
0
5
10
15
uvbi
about

SAA for many N level multilevel models

Looking at predictions

Having fitted a model with several predictors we might like to represent this model graphically. This is more difficult than when we have only one predictor and so for now we consider each predictor in turn and set all other predictors to their mean values.

These prediction plots basically plot against one predictor holding the other predictor at its average value and so the left-hand plot is not very useful as we see nation 8 (Luxembourg) getting crazy predictions as the mean value for UVB across the whole dataset would be an extreme value in Luxembourg! The right-hand plot shows a general pattern of the rates observed increasing with UVB exposure.

The full output for this SAA run is available in the file *mmmec.pdf*

10.2 Always keeping variables in the model

Before moving on to looking at missing data we should briefly mention some of the other features of the Combined SAA that we have not talked about here. In all of the examples that we have mentioned in this guide we have not examined a couple of options that we will mention in this section and the next.

In our examples we have not used the options that are available from the two questions asking if you wish to include particular continuous and/or categorical variables in all models. There may be reasons why as a researcher you wish to always include variables irrespective of their significance and in fact you could use these questions to get the SAA to only fit a specific model that you are interested in and then see the residual and predictions plots for that model. Here you would answer *Yes* to the always include questions and *No* to the include questions which are used with significance testing.

We do not include an example of this here but mention it for completeness.

10.3 Alternative Model Selection routines

In this guide we have stuck to a rather simple model selection procedure that we describe as 'Forward Pass'. This builds on the fact that as a first step in our modelling we fit each predictor in turn to look at its significance in isolation and plot predictions for it. As part of this step we then establish an order of the predictors based on their significance. The forward pass method then uses this order of significance and does a single pass through the predictors based on this order. We start with a model with the most significant predictor and then for each predictor in turn we add it to the current model and keep it if it is significant or remove it if not. After passing through all the predictors this will result in a final model. If interactions are chosen then this procedure is repeated with the possible interactions terms and as a final step a variant of the procedure is used to look at random slopes.

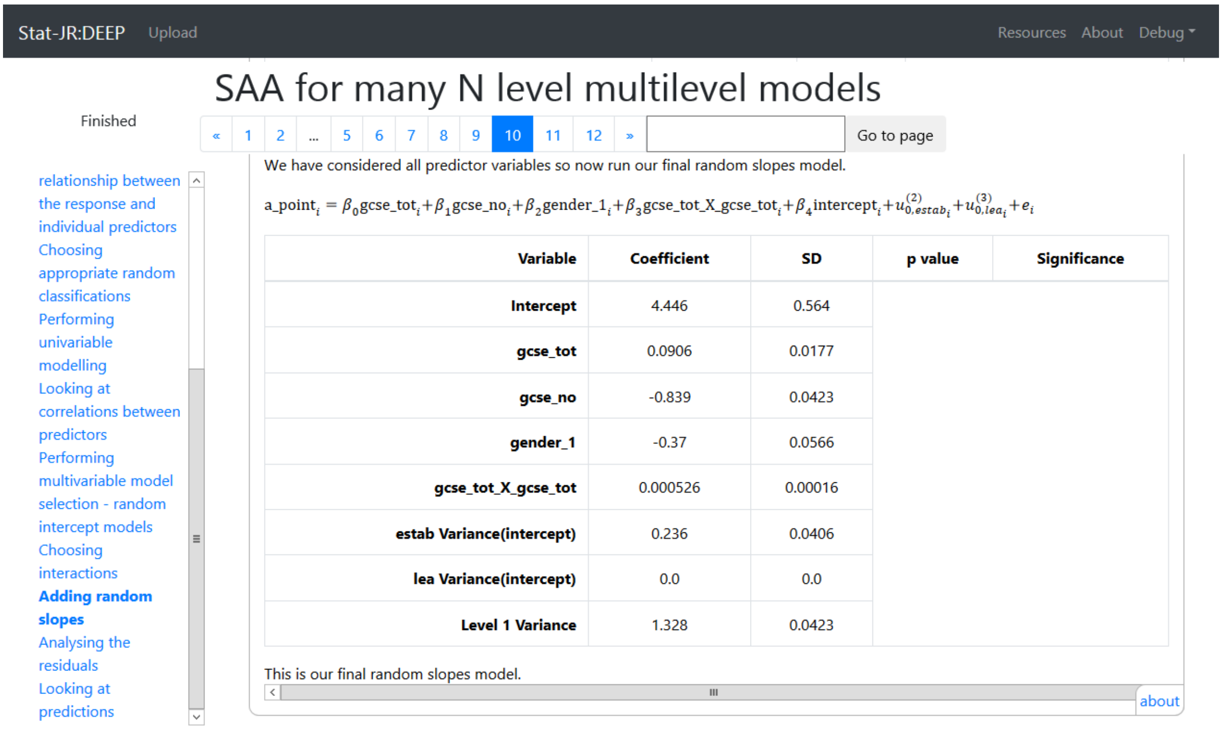
The forward pass method can be thought of as a quicker, watered down version of *Full forward selection* which we offer as an alternative. In *Full Forward* selection we begin with a model with the most significant predictor included. We then try out adding each of the remaining predictors to this model and choose the resulting model that makes the most significant improvement. We then repeat this procedure until adding none of the predictors makes an improvement. Here, if say we start with 10 predictors then we would select the best by fitting 10 models, then compare the 9 models formed by adding each other predictor to the first selected one, then 8, 7 etc. until no predictors to be added are significant and thus this method generally fits far more models than the forward pass method.

The third method offered is *backward elimination* and here we start with a model with all predictors included rather than none and then work in the opposite direction to find a best model. Here the significance of each predictor is investigated and the least significant (of those that are not deemed to be significant) is removed from the model and the new model fitted. This procedure is then repeated on this new model until we end up with a model that has only significant terms.

We also offer two combination methods that combine a forward method followed by a backward method so in these cases having run through the forward method we end up with a final model. Often this final model contains all significant terms and nothing more will be done but it is possible that terms that enter the model earlier in the method become non-significant as other terms are entered as the forward methods only guarantee that the last term entered is significant. If this is the case then the backward elimination method is then run to prune out the non-significant terms.

For speed we have stuck to forward pass estimation in this guide but you are very welcome to try the alternative methods and see their impact. For illustration we have stored the outputs from running the SAA on the A level chemistry dataset using Full Forward/Backward in the file *alevchemffb.pdf* to compare the outcomes.

Here perhaps as we might anticipate the non-significant interaction (*gender_1_X_gcse_tot*) that we noticed when fitting the forward pass method is not present but also the other significant term (*gcse_no_X_gcse_tot*) is not significant when the non-significant term is not present.


Stat-JR:DEEP
Upload
Resources
About
Debug
Finished
« 1 2 ... 5 6 7 8 9 10 11 12 »
Go to page
relationship between the response and individual predictors
Choosing appropriate random classifications
Performing univariable modelling
Looking at correlations between predictors
Performing multivariable model selection - random intercept models
Choosing interactions
Adding random slopes
Analysing the residuals
Looking at predictions
about

SAA for many N level multilevel models

We have considered all predictor variables so now run our final random slopes model.

$$\text{a\_point}_i = \beta_0\text{gcse\_tot}_i + \beta_1\text{gcse\_no}_i + \beta_2\text{gender\_1}_i + \beta_3\text{gcse\_tot\_X\_gcse\_tot}_i + \beta_4\text{intercept}_i + u^{(2)}_{0,estab_i} + u^{(3)}_{0,lea_i} + e_i$$

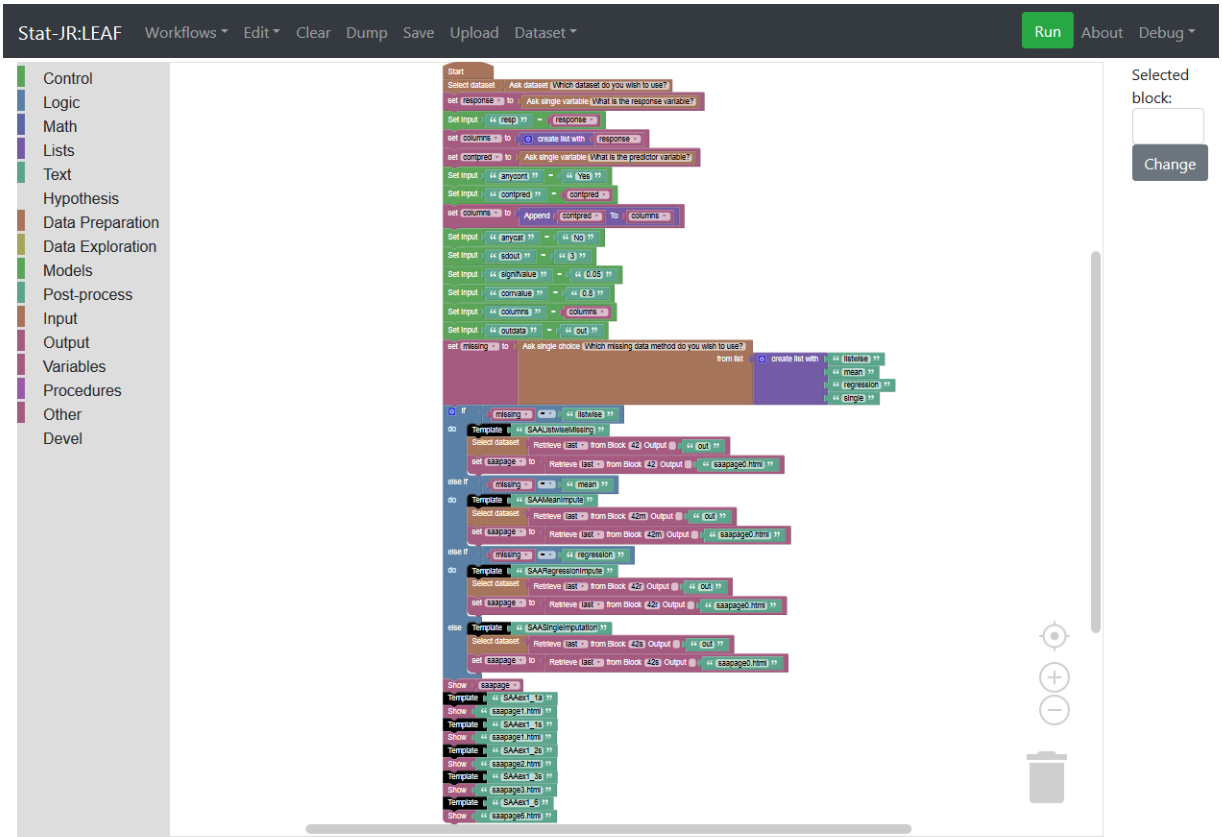
Variable	Coefficient	SD	p value	Significance
Intercept	4.446	0.564		
gcse_tot	0.0906	0.0177		
gcse_no	-0.839	0.0423		
gender_1	-0.37	0.0566		
gcse_tot_X_gcse_tot	0.000526	0.00016		
estab Variance(intercept)	0.236	0.0406		
lea Variance(intercept)	0.0	0.0		
Level 1 Variance	1.328	0.0423		

This is our final random slopes model.

Chapter 11 – Missing data

There is much literature on methods for dealing with missing data and in many ways dealing with missing data is a very important yet often under considered aspect of data analysis. The SAAs that you have used thus far will work on datasets that contain missing data because the first step in each of them is to list-wise delete any observations that contain any missing data before proceeding. This clearly reduces the sample of data that we work on and thus there are other less drastic approaches to dealing with missing data and we will consider 3 more methods here: mean imputation, regression imputation and single imputation. A more modern way of dealing with missing data, that also gives unbiased parameter estimates, is via 'multiple imputation' and variants of this approach and we will talk about this briefly at the end.

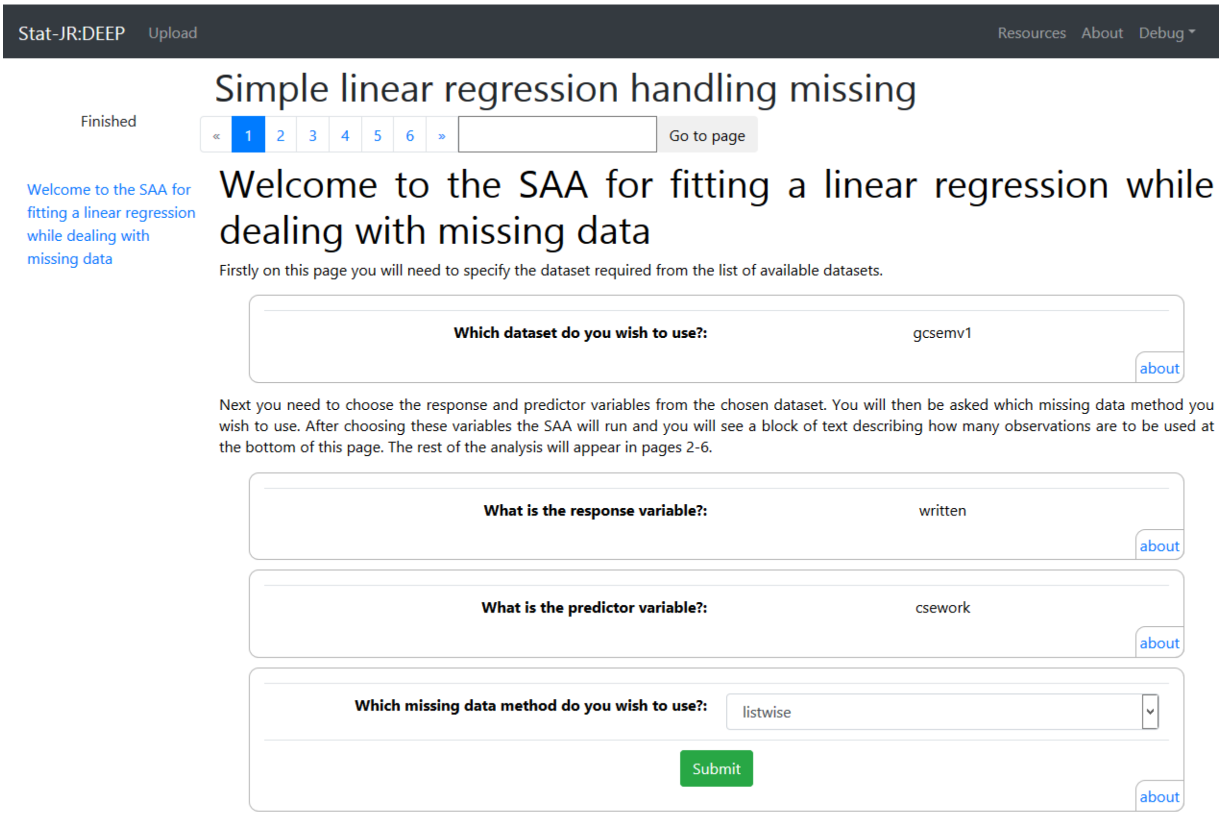
We will illustrate list-wise deletion and the other three methods using a simple linear regression model on a dataset with missing data (*gcsemv1* (Rasbash, Steele, Browne, & Goldstein, 2012)) and regress a written exam score on a coursework score. All these methods can be viewed as pre-processing procedures, so that we will modify the workflow for a linear regression so that an addition question is asked to establish which method to be used thus:



```
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"
set missing to  Ask single choice Which missing data method do you wish to use?
                from list  create list with "listwise"
                                            "mean"
                                            "regression"
                                            "single"
if missing = "listwise"
do    Template "SAAListwiseMissing"
      Select dataset  Retrieve last from Block 42 Output "out"
      set saapage to  Retrieve last from Block 42 Output "saapage0.html"
else if missing = "mean"
do    Template "SAAMeanImpute"
      Select dataset  Retrieve last from Block 42m Output "out"
      set saapage to  Retrieve last from Block 42m Output "saapage0.html"
else if missing = "regression"
do    Template "SAARegressionImpute"
      Select dataset  Retrieve last from Block 42r Output "out"
      set saapage to  Retrieve last from Block 42r Output "saapage0.html"
else  Template "SAASingleImputation"
      Select dataset  Retrieve last from Block 42s Output "out"
      set saapage to  Retrieve last from Block 42s Output "saapage0.html"
Show saapage
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"
```

This workflow (*linregmiss.xml*) then forms the main part of the eBook (*linregmiss.zip*).

Our dataset for this chapter is *gcsemv1* which is an examinations dataset containing amongst other variables the two components of a chemistry GCSE exam taken at age 16. The two components are a written exam (*written*) and some coursework (*csework*) that together produce the overall mark for the student. The dataset has missing data and so some students only have exam scores while other students only have coursework marks. We will begin by trying the standard method of list-wise deletion as shown below, Here the inputs are dataset being *gcsemv1*, response is *written* and predictor *csework*:



Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Welcome to the SAA for fitting a linear regression while dealing with missing data

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?: gcsemv1

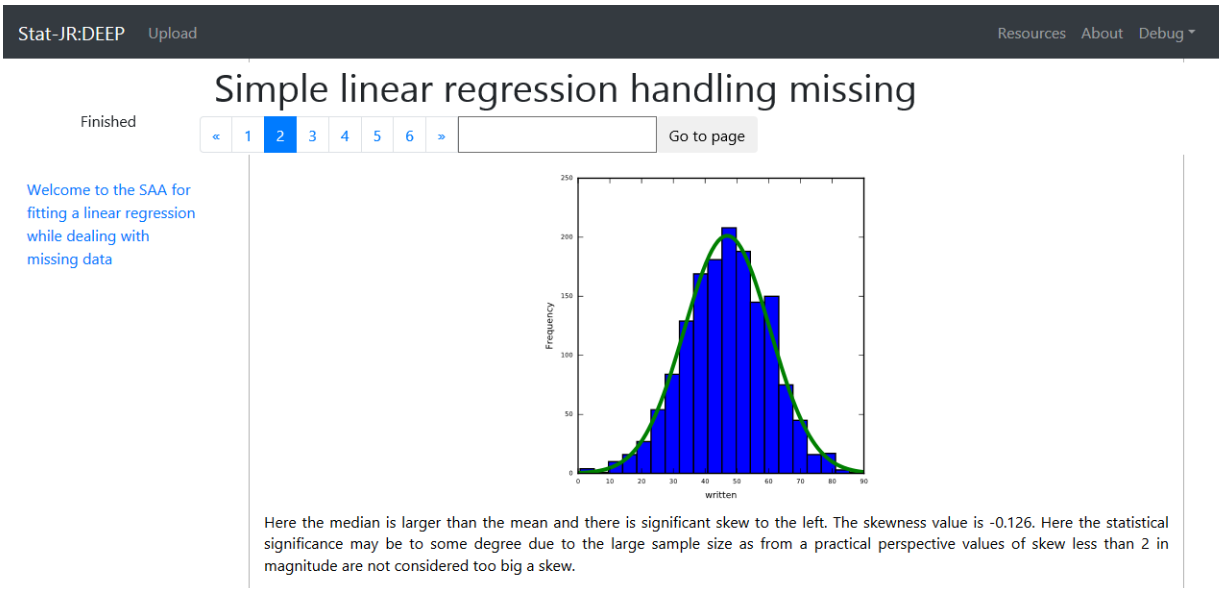
Next you need to choose the response and predictor variables from the chosen dataset. You will then be asked which missing data method you wish to use. After choosing these variables the SAA will run and you will see a block of text describing how many observations are to be used at the bottom of this page. The rest of the analysis will appear in pages 2-6.

What is the response variable?: written

What is the predictor variable?: csework

Which missing data method do you wish to use?: listwise

When we do this we see that we are only using 1523 rows but for example on page 2 we have a fairly normal looking histogram:

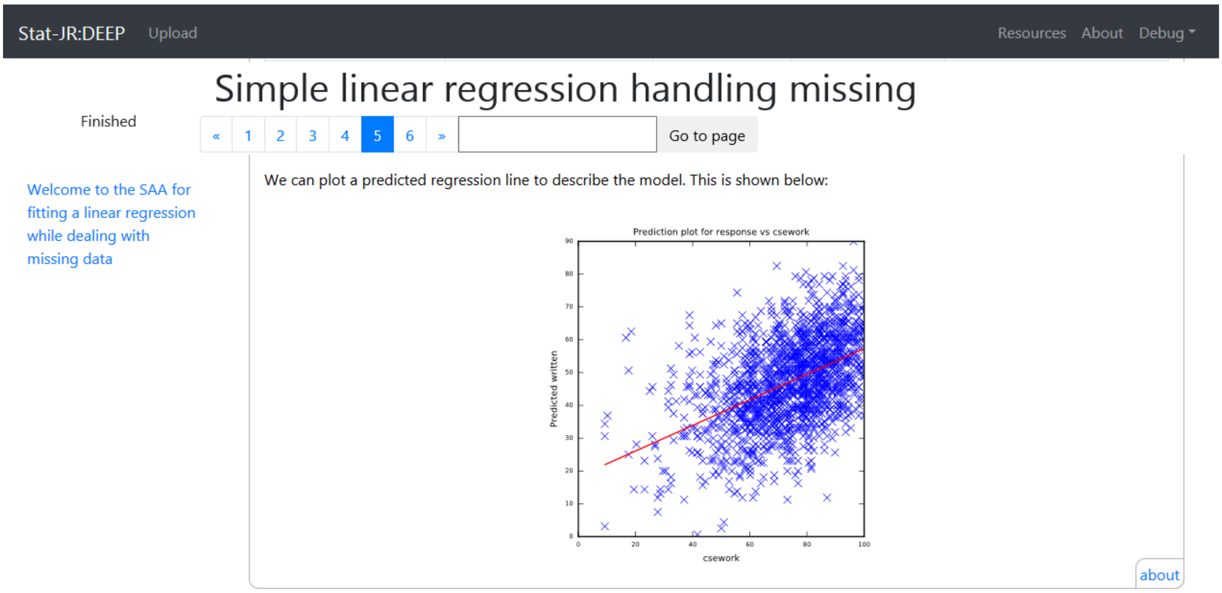


Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Here the median is larger than the mean and there is significant skew to the left. The skewness value is -0.126. 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 in magnitude are not considered too big a skew.

We can also see that on page 5 the modelling looks fairly straightforward with a regression line through the data:

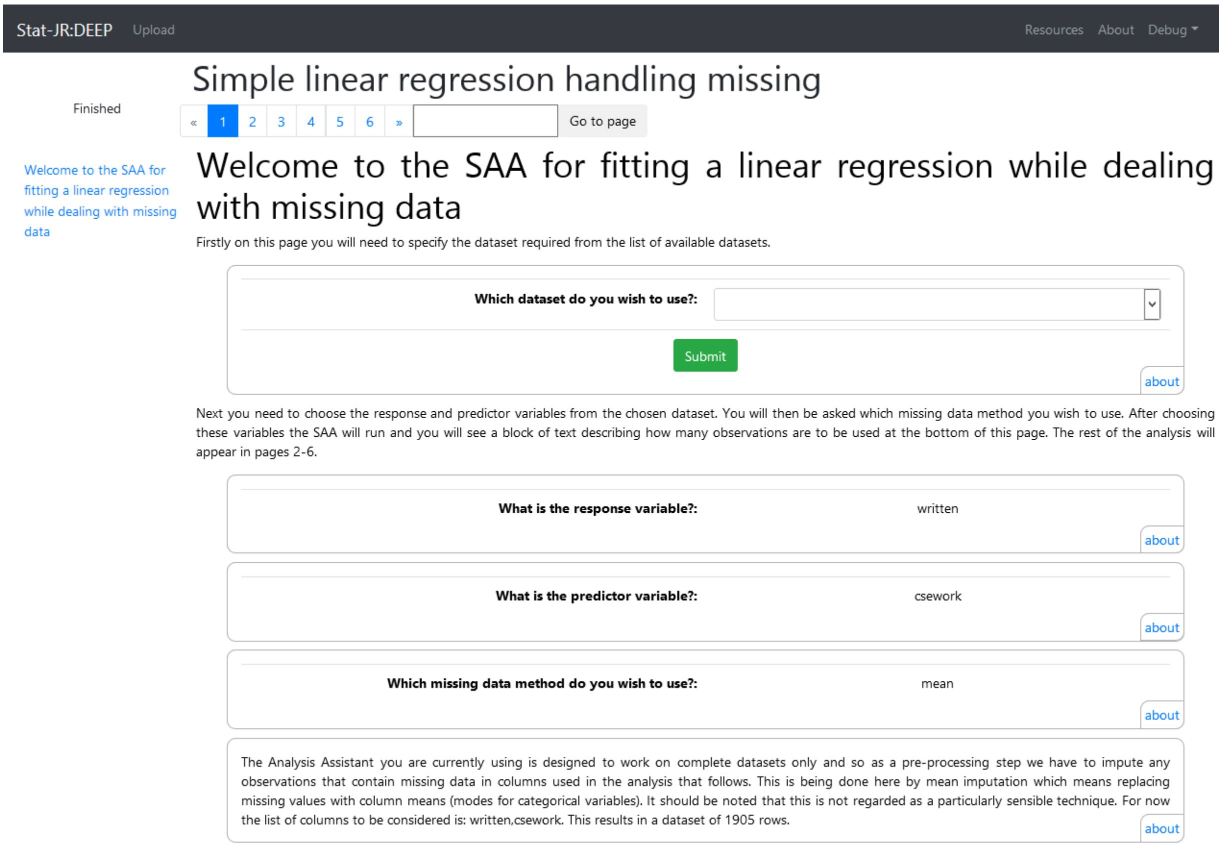


Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

We can plot a predicted regression line to describe the model. This is shown below:

If we now return to page 1 and select instead *mean imputation* we see the following:



Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Welcome to the SAA for fitting a linear regression while dealing with missing data

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?:

Next you need to choose the response and predictor variables from the chosen dataset. You will then be asked which missing data method you wish to use. 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.

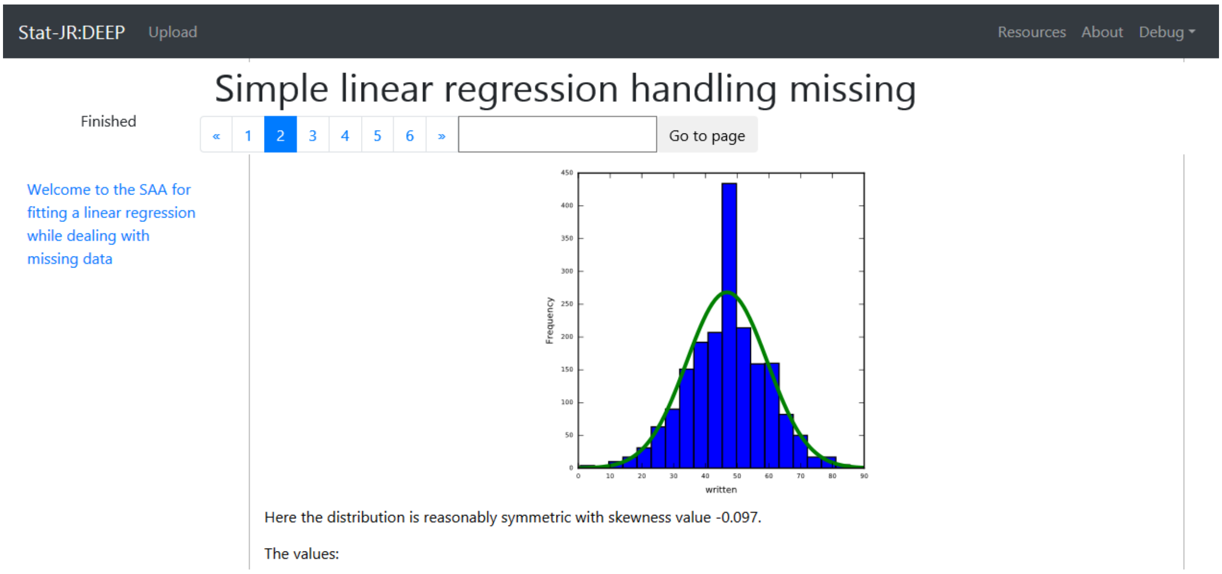
What is the response variable?: written

What is the predictor variable?: csework

Which missing data method do you wish to use?: mean

The Analysis Assistant you are currently using is designed to work on complete datasets only and so as a pre-processing step we have to impute any observations that contain missing data in columns used in the analysis that follows. This is being done here by mean imputation which means replacing missing values with column means (modes for categorical variables). It should be noted that this is not regarded as a particularly sensible technique. For now the list of columns to be considered is: written,csework. This results in a dataset of 1905 rows.

So aside from the caveat that this isn't a great method we see that we do now have a full dataset with 1,905 observations. If we now compare the histogram of the response on page 2 we see that there is a huge peak of values imputed in one histogram bin (at the mean) and this also has the effect of making the distribution more symmetric.

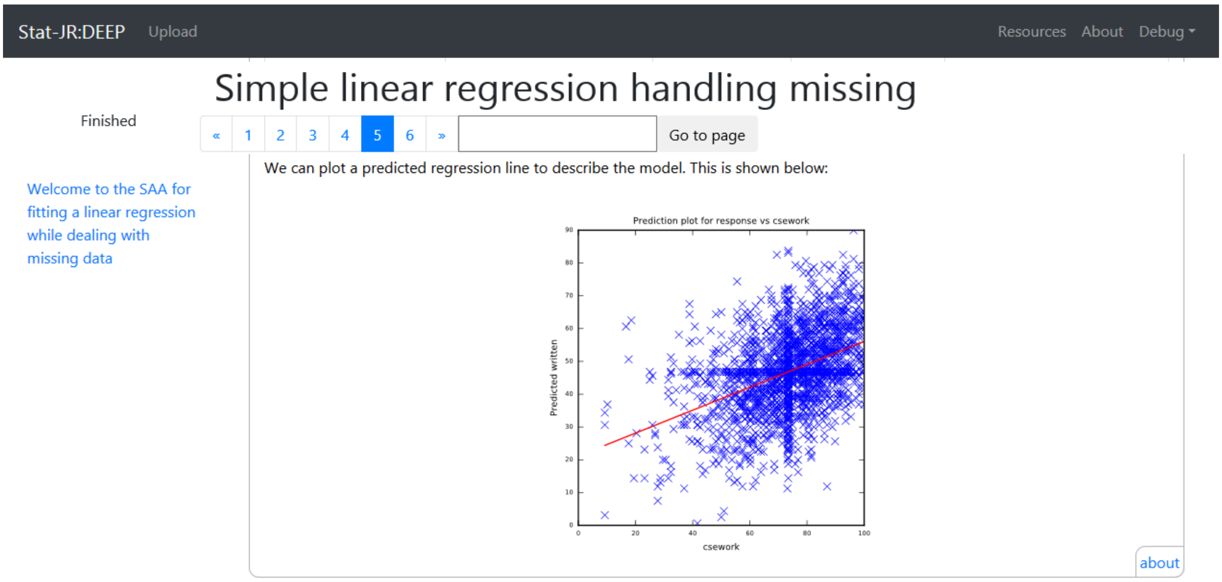


Simple linear regression handling missing

Here the distribution is reasonably symmetric with skewness value -0.097.

The values:

If we follow this on to the model fitting we can see the full extent of the mean imputation with the cross of imputed points visible in the graph:

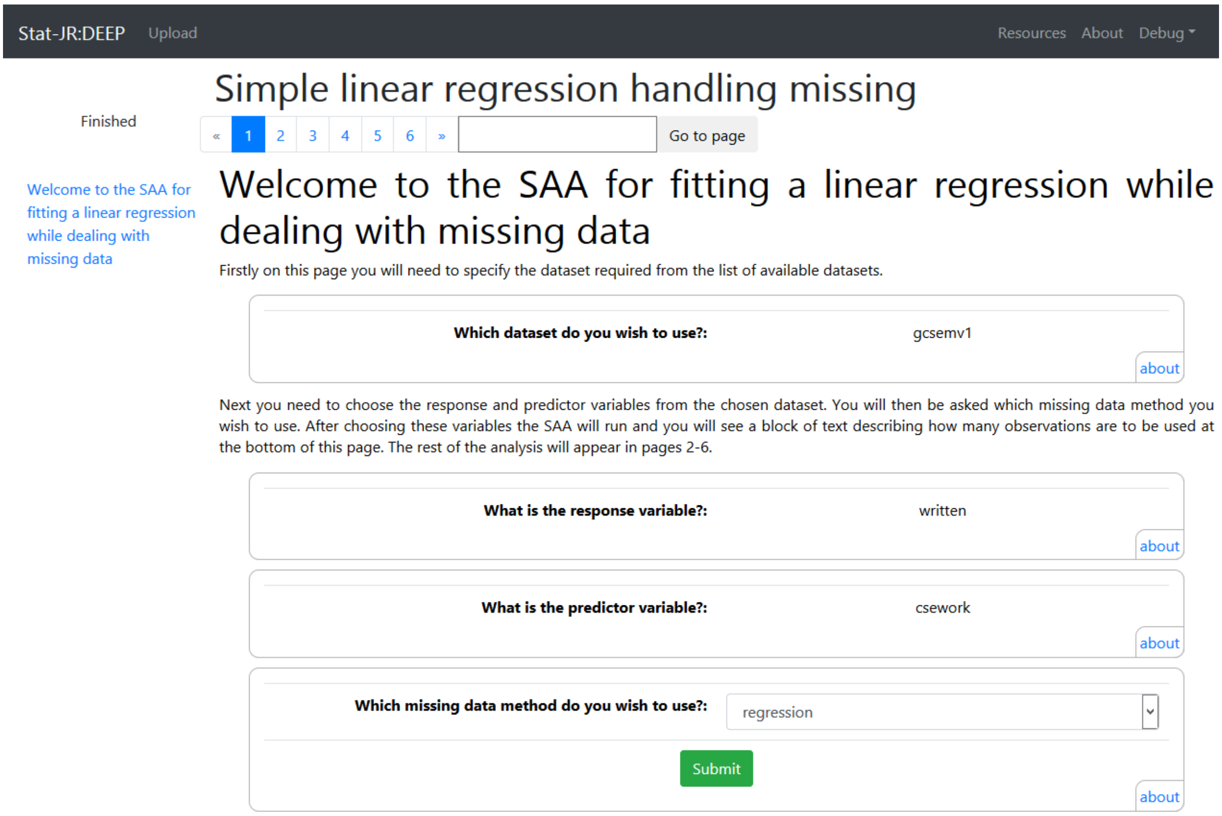


Simple linear regression handling missing

We can plot a predicted regression line to describe the model. This is shown below:

The regression line is similarly statistically significant here as in the list-wise case and in fact the size of the slope has reduced slightly but this is balanced by both a reduction in the standard error of the slope and the overall residual variance (*sigmasq*). This method is known to be biased, however, and is not recommended for routine use.

We next move on to *regression imputation*.



Simple linear regression handling missing

Finished

Welcome to the SAA for fitting a linear regression while dealing with missing data

Welcome to the SAA for fitting a linear regression while dealing with missing data

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?: gcsemv1

Next you need to choose the response and predictor variables from the chosen dataset. You will then be asked which missing data method you wish to use. 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.

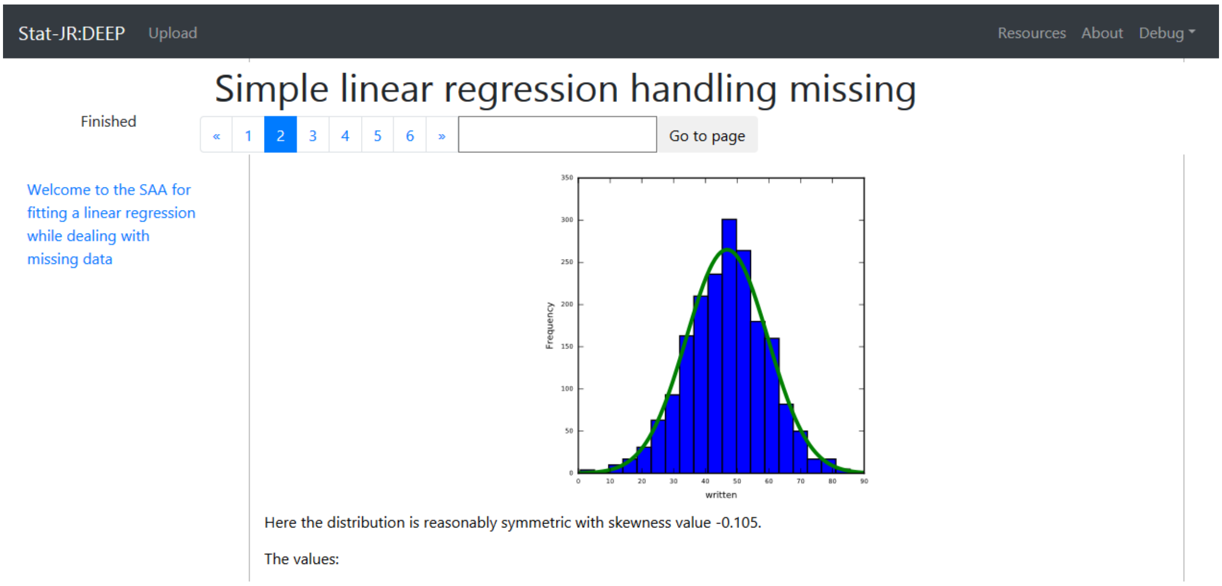
What is the response variable?: written

What is the predictor variable?: csework

Which missing data method do you wish to use?: regression

Submit

In this situation the missing response and predictor variables are replaced by the predicted values from respective regression models (regressing *written* on *csework* and *csework* on *written* respectively and in practice one could extend this approach to regress each variable on all the others in more general linear models). We will look at the impact the technique has on the same outputs as before:



Simple linear regression handling missing

Finished

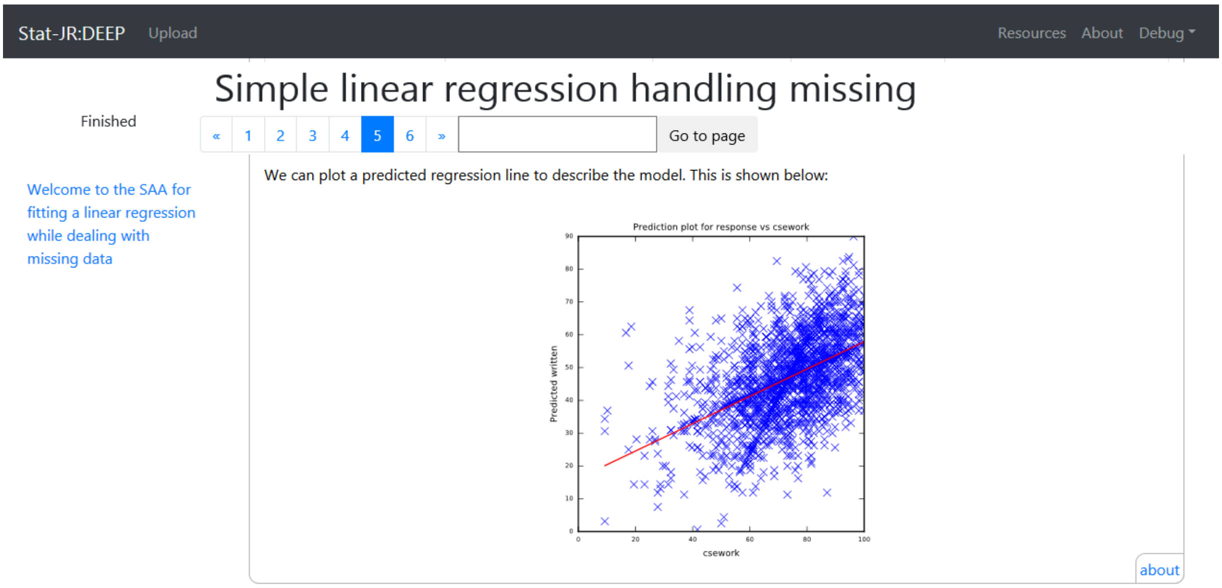
Welcome to the SAA for fitting a linear regression while dealing with missing data

Here the distribution is reasonably symmetric with skewness value -0.105.

The values:

The histogram of the response looks similar to that from complete cases as the imputed values take different values depending on the observed values for the other variables and come from the distribution of interest.

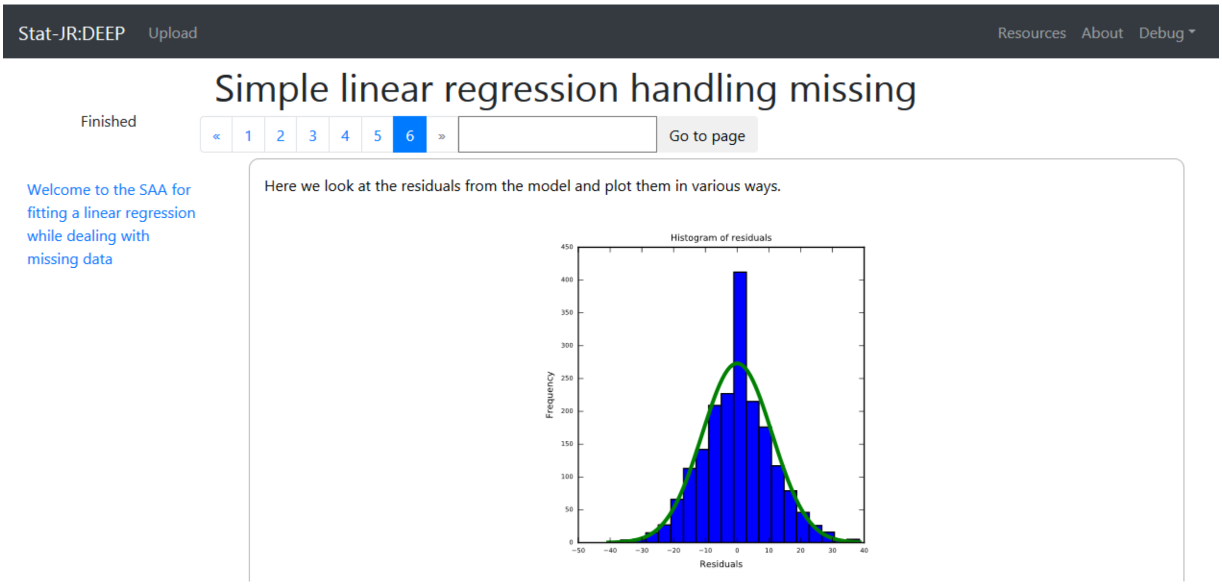
If we look at the fitted regression on page 5 we see the following:



Simple linear regression handling missing

We can plot a predicted regression line to describe the model. This is shown below:

Here you can see in the data the two regression lines (*written* on *csework* and *csework* on *written*) that the imputed values all lie upon so you can see how the imputation is reducing variability. If we in addition look at the model residuals we see the following:

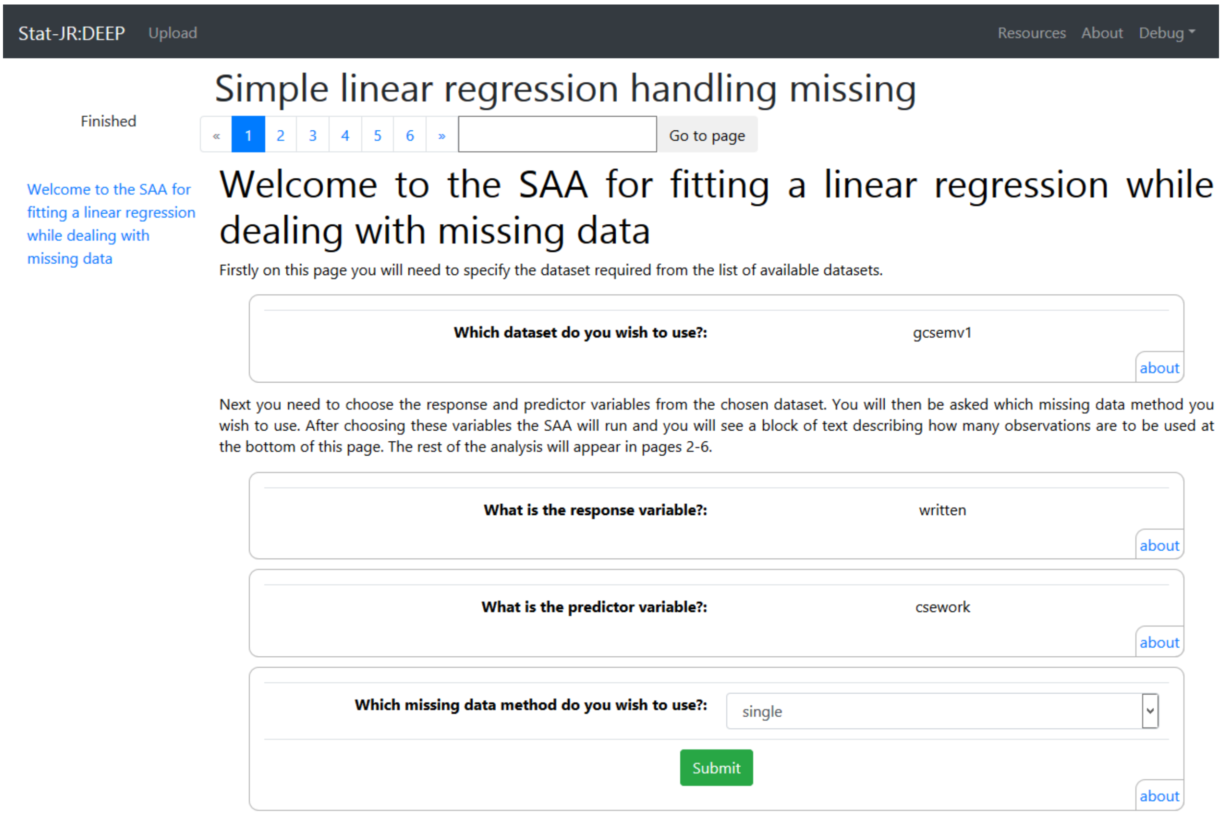


Simple linear regression handling missing

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

So now we have loads of zero residuals as the imputed values by definition lie on the regression line. In many scenarios the model of interest that we fit will differ from the imputation model and this patterning issue will be less obvious but still present.

A final solution for now is to fit a *single imputation* model:



Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Welcome to the SAA for fitting a linear regression while dealing with missing data

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?: gcsemv1

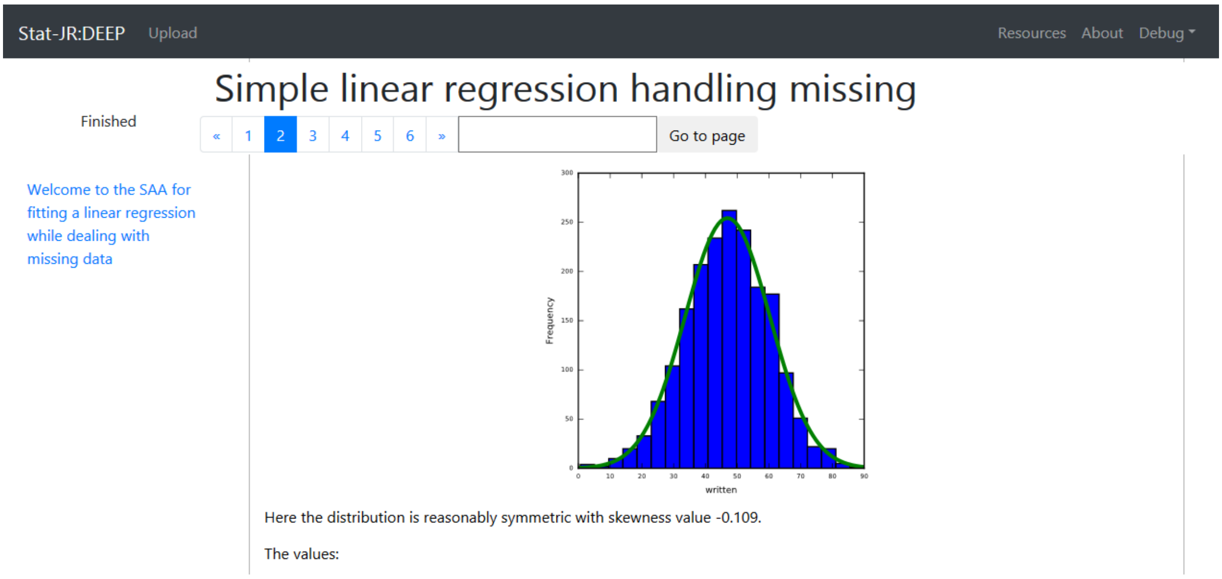
Next you need to choose the response and predictor variables from the chosen dataset. You will then be asked which missing data method you wish to use. After choosing these variables the SAA will run and you will see a block of text describing how many observations are to be used at the bottom of this page. The rest of the analysis will appear in pages 2-6.

What is the response variable?: written

What is the predictor variable?: csework

Which missing data method do you wish to use?: single

Here rather than placing the imputed values on the regression lines they are instead drawn from the distribution that surrounds the regression. Again the histogram of responses looks similar to the complete case data:



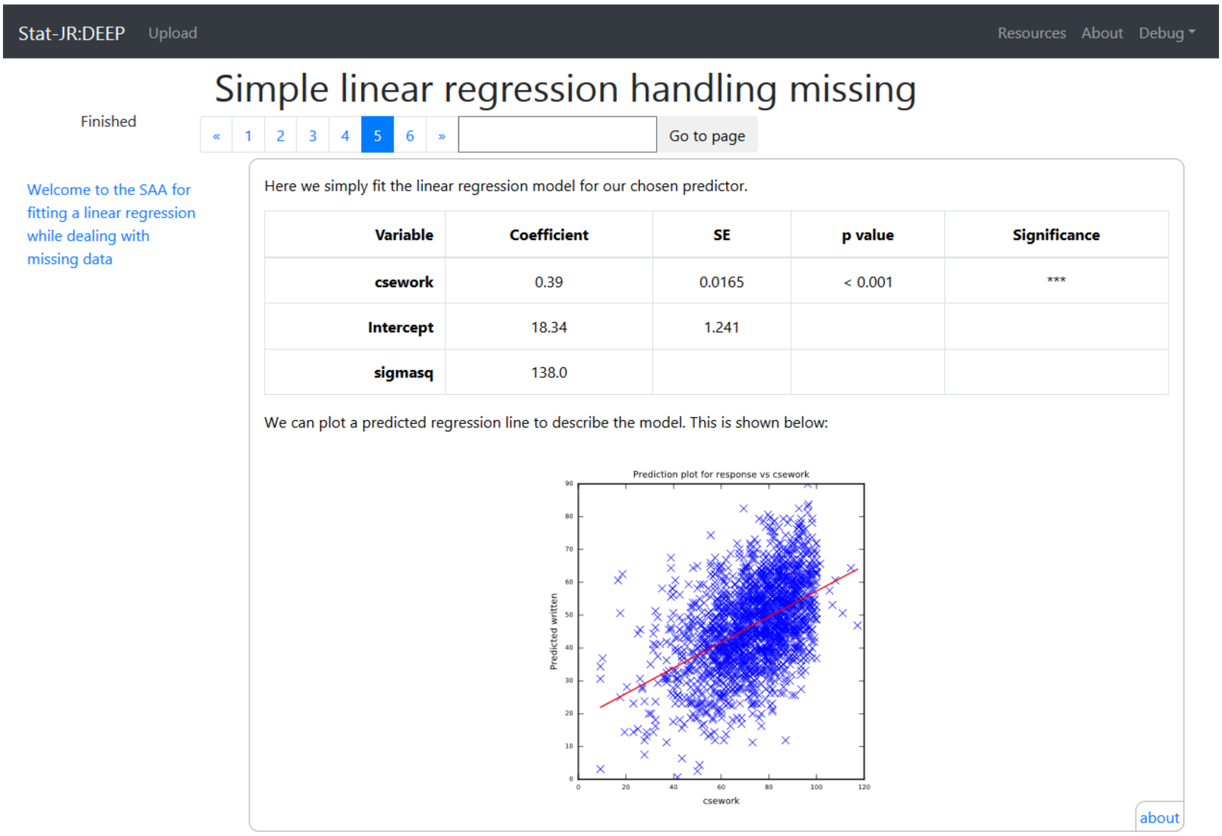
Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Here the distribution is reasonably symmetric with skewness value -0.109.

The values:

The predicted model below doesn't show any patterning and so we don't have the issues we have seen in the other imputation methods.



Simple linear regression handling missing

Welcome to the SAA for fitting a linear regression while dealing with missing data

Here we simply fit the linear regression model for our chosen predictor.

Variable	Coefficient	SE	p value	Significance
cswork	0.39	0.0165	< 0.001	***
Intercept	18.34	1.241		
sigmasq	138.0			

We can plot a predicted regression line to describe the model. This is shown below:

The model coefficients are similar to those from the complete case analysis although with smaller standard errors due to the inflated sample size. This points to single imputation not being a final solution as essentially having drawn random values from the distribution in the imputation the modelling has then assumed these values are the truth. We could of course repeat the random value generation with a different set of random numbers and get different data and different estimates. Such an approach of creating several imputed datasets forms the basis of multiple imputation and in this case the estimates produced can be combined (via a procedure known as Rubin's rules) to include the uncertainty in the estimates.

To do multiple imputation therefore a more complicated workflow is required that creates several imputed datasets and then fits the same model to each of them. This workflow will then need to combine the differing model estimates together so that the uncertainties are preserved (using Rubin's rule). At this point we have not combined multiple imputation into our more general SAA system but we have constructed a few Stat-JR templates that do multiple imputation for one model (see (Parker & Goldstein, 2017) for more details).

Chapter 12 – Future work in bringing it all together

In this guide we have described several statistical analysis assistants (SAAs) that can be used to fit several of the most commonly fitted models covered by the MLwiN software package. For each SAA we have attempted to go beyond the simple model fit and included the exploratory data analysis steps that proceed a model fit, some aspects of model selection and some post-processing steps including residual analysis and predictions.

The final (combined) SAA that we introduced first in chapter 7 is a very flexible SAA that will fit several model families by calling either the IGLS or MCMC engines in MLwiN. It is capable of dealing with normal, Binomial and Poisson distributed observations. It can include random intercepts and random slopes and (order 1) interaction and quadratic terms into models that can have both nested and crossed random effects. Although it offers quite a lot of flexibility it is of course deterministic and given particular inputs it will produce a particular analysis based on how it is programmed. Not everyone will agree with all its suggestions and of course in an ideal world there would be a particular SAA for any analyst that tries to mirror their decision making rather than be generic. We hope as we release this version of Stat-JR and the Combined SAA that we will get analysts using it and offering advice and feedback so that this customising of the SAA can be achieved.

Writing the SAAs described in this book helped us to discover what features of a statistical analysis it was feasible to get a computer to do and which were more difficult. One feature which we had hoped to include in this release was automating missing data imputation but as yet we haven't quite pieced together all the pieces to make this a reality. As mentioned in chapter 11 we have constructed several Stat-JR templates (see (Parker & Goldstein, 2017)) that use Stat-JRs in-built engines to fit specific models including imputation and other methods in isolation. To include these in an SAA we would have to plug in this engine in place of the MLwiN engine and come up with generic methods for imputing data for all models that form part of a statistical analysis. We would also need to consider how best to compare models with missing data. These are big challenges but we are as we speak considering the best approaches to use and we hope in future work to at least prototype methods to do this.

We very much hope you find the SAAs that we introduce in this book interesting and useful in your own data analysis. Please let us know how you get on and help us to improve the Stat-JR system.

WJB 2017

References

Browne, W. (2017). *MCMC Estimation in MLwiN v3.00.* Centre for Multilevel Modelling.

Browne, W., Parker, R., Charlton, C., Michaelides, D., & Moreau, L. (2016). *Stat-JR LEAF Workflow Guide.* Centre for Multilevel Modelling, University of Bristol & Electronics and Computer Science, University of Southampton.

Browne, W., Steele, F., Golalizadeh, M., & Green, M. (2009). The use of simple reparameterizations to improve the efficiency of Markov chain Monte Carlo estimation for multilevel models with applications to discrete time survival models . *Journal of Royal Statistical Society, Series A*(172), 579-598.

Charlton, C., Rasbash, J., Browne, W., Healy, M., & Cameron, B. (2017). MLwiN Version 3.00. Centre for Multilevel Modelling.

Gelfand, A., Hills, S., Racine-Poon, A., & Smith, A. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*(85), 972-985.

Goldstein, H., Browne, W., & Rasbash, J. (2002). Partitioning Variation in Multilevel Models. *Understanding Statistics, 1*(4), 223-231.

Goldstein, H., Rasbash, J., Yang, M., Woodhouse, G., Pan, H., Nuttall, D., & Thomas, S. (1993). A multilevel analysis of school examination results. *Oxford Review of Education*(19), 425–433.

Huq, N., & Cleland, J. (1990). Bangladesh fertility survey, 1989. Dhaka: National Institute of Population Research and Training (NIPORT).

Langford, I., Bentham, G., & McDonald, A.-L. (1998). Multi-level modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European Community. *Statistics in Medicine*(17), 41-57.

Parker, R., & Goldstein, H. (2017). Imputation for Multilevel Models with Missing Data Using Stat-JR. Bristol: Centre for Multilevel Modelling.

Paterson, L. (1991). Socio economic status and educational attainment: a multidimensional and multilevel study. *Evaluation and Research in Education*(5), 97-121.

Rasbash, J., Steele, F., Browne, W., & Goldstein, H. (2012). *A User's Guide to MLwiN.* Centre for Multilevel Modelling,.

Yang, M., & Woodhouse, G. (2001). Progress from GCSE to A and AS level: institutional and gender differences, and trends over time. *British Educational Research Journal*(27), 245-267.