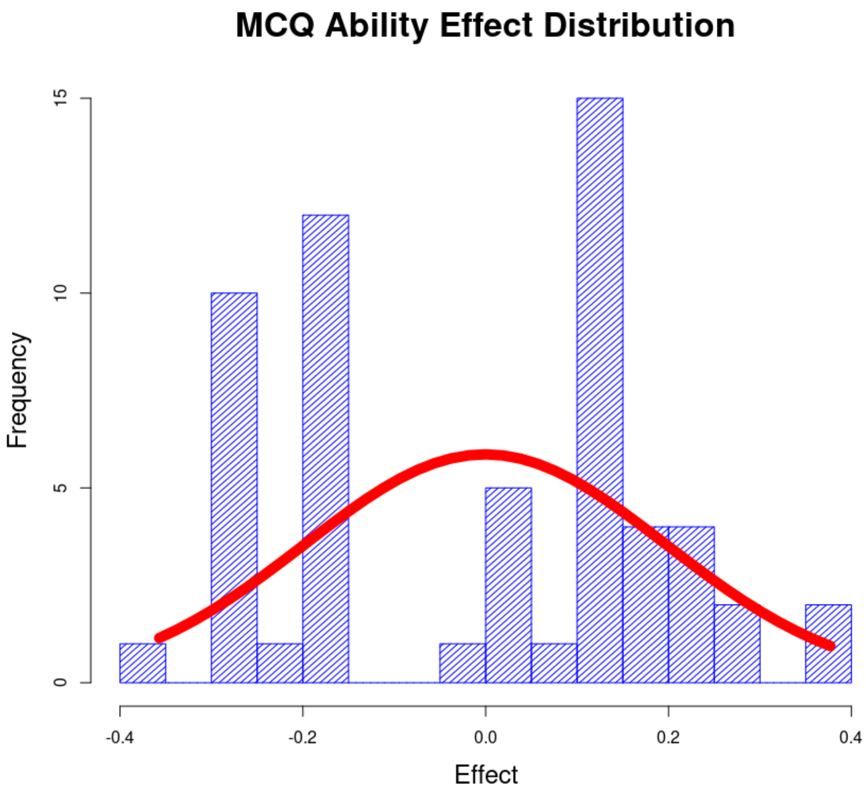


G_String GS_M

Manual

May 16 2023

Table of Contents

Introduction

G_String is a software tool, using ‘Generalizability Analysis’[*], to assess the performance of performance assessment tools. It answers the question:

> “What fraction of a test score* measures a specific
> quality of the test subject being tested,
> rather than some idiosyncrasies of the test itself?”

G_String is intended for the use by Performance Assessment practitioners, rather than for expert statisticians with their own preferred statistical software package, such as R or SPSS, yet G_String performs the same Generalizability Analysis* as the more general programs, and, additionally, allows for ‘D Studies’*, while leading the user step by step through the process.

Besides routine G_Studies*, GS_M also permits G_Studies with replications*, and the generation of synthetic* test score files. For each of these three calculations, the user has the choice of entering the parameters by hand, or to automatically read them in from a prepared script. Such scripts, in turn, can be saved automatically, once a design has been entered manually.

This manual also explains the rationale of Generalizability Analysis with a focus on plausibility rather than on formal stringency. It will start with explaining the central concept of ‘Variance Components’*. Where jargon terms are used, they are marked with an asterisk (*), and explained in Appendix 1. Although Generalizability Analysis has applications far beyond the area of performance, and learning assessment, we confine the terminology appropriate to the earlier field in order to avoid confusion.

The software name ‘G_String’ is derived from the fact that it performs **G**eneralizability Analysis by lexically transforming effect **Strings**. The current incarnation of G_String is GS_M, completed in 2023, using Java* and JavaFX* versions 17. GS_M runs on Windows, Mac and Linux(Ubuntu), and is not dependent on whatever Java libraries are already installed in the user’s operating system.

GS_M is still based on the original G_String design, developed at McMaster in 2007, but it has acquired further functionality. It is supported on the internet at:

https://github.com/G-String-Legacy/GS_MV/wiki

No commercial body ensures the future of G_String. But the above website provides the tools for the user community to keep the software up-to-date.

[*] Explanations for jargon terms in Appendix I. (in performance assessment context)

Estimating Variance Components

Most of G_String's 'heavy lifting' is done in step 9 by a C++ routine called urGenova designed by Prof. Robert L. Brennan, who has kindly given his permission to include it in this software. From the raw score data, and the full set of design parameters urGenova extracts estimates of the so called 'Variance Components' for each facet and their combinations. The variance components, in turn, are then used to calculate the G-Coefficients and perform D-Analysis.

The concept of variance components stems from a mathematical theory called '**General Linear Model**'. The model is called 'linear', because the the result is equal to the simple sum of a mean and various effects that determine the outcome. Although its general form is simple to write as a mathematical formula, it is somewhat obscure for non-mathematicians to understand. Instead we will first apply the concept to the example of a plain multiple choice exam[1], and then chat loosely about its extension to more complex cases. Fine details are not that important here, and G_String knows how to do it.

So, let's start with the 58 students answering 30 MCQ questions either correctly (score 1), or incorrectly (score 0). Thus, each student can potentially receive a minimum total score of 0, and a maximum of 30. The data file will have 58 rows and 31 columns. The first column contains the corresponding student index (to be ignored), the subsequent fields the individual question scores (0 or 1). The general mean score over all the students and questions is **0.58966 = μ_0**, which also corresponds to the probability of an average student getting the answer to an average question right.

But 'average' student, and 'average' question are theoretical constructs. In reality, students (s) differ by their overall ability, questions (q) by their easiness, and different students differ in their ability to handle different types of questions (*sq* interactions). In generalizability jargon these deviations from the mean are called '**effects**'*.

In fact, there is a set of effects (*v*) associated with each facet*, as well as with their various permitted combinations. The size of each set is determined by the sample size of its corresponding facet. Thus, the General Linear Model postulates that the score ($X_{i,j}$) for student *i* answering question *j* can be expressed as:

$$X_{i,j} = \mu_0 + \nu_i^s + \nu_j^q + \nu_{i,j}^{sq} \qquad \text{(Equation 1)}$$

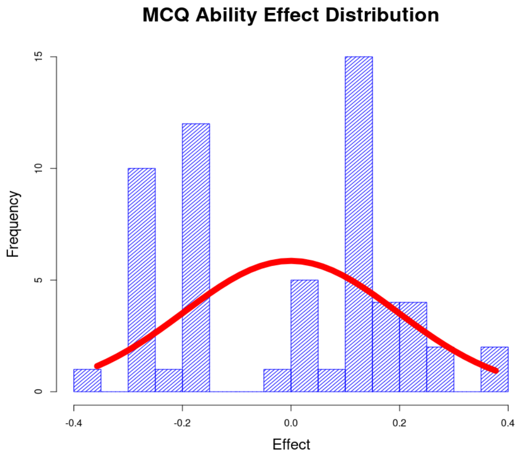
Where ν_i^s stands for the ability effect of student *i*, ν_j^q for the easiness effect of question *j*, and $\nu_{i,j}^{sq}$ for their cross-effect (interaction), with:

$$\sum_{i=0}^{57} \nu_i^s = 0 \quad (1a), \quad \sum_{j=0}^{29} \nu_j^q = 0 \quad (1b), \text{ and } \sum_{i=0}^{57} \sum_{j=0}^{29} \nu_{i,j}^{sq} = 0 \quad (1c)$$

Let us now examine the properties of effects for a given facet, or interaction of facets. Since the individual students' ability effects are essentially independent from each other, they must be considered

[1] https://github.com/G-String-Legacy/GS_MV/wiki/Clinical-Clerkship-Grading

to be random values, and because of equation (1a) they must be more or less symmetrically distributed around zero, which holds for the effect sets of all facets and of their respective interactions. As in most distributions, values get rarer, the further you deviate from zero. That, in fact, describes a bell-curve distribution. Each effect set distribution has its own variance*, and we call these the 'variance component'* of the corresponding facet. Most importantly, under the General Linear Model the variance of a sum is the sum of its component variances.



The variance components σ_s^2, σ_q^2, and σ_{sq}^2 are measures of the different effects' spread for their corresponding facets. On this diagram the distribution of the 58 students' ability effects, as extracted by urGenova from the data file, are shown as a histogram in blue. The histogram resembles a bactrian more than a dromedary. One might be tempted to infer that the class is split into strivers and slackers. But that would be a wrong conclusion. For the data were, in fact, created by a random number generator (G_String Simulation).

The variance component for students' performance comes out as $\sigma_s^2 \cong 0.04$, which corresponds to a standard deviation of 0.2. The red curve shows the corresponding ideal Gaussian distribution with a variance of 0.04, and a mean of 0.

There are two reasons for the discrepancy between actual and expected data. Firstly, 58 students represent a relatively small statistical sample, and secondly, the continuous effect values can only express themselves as either score 0, or score 1, which leads to large rounding errors.

G Coefficients

A good test has to measure what it claims to measure. Performance tests provide mean test scores for candidates $\overline{X}_i = \frac{1}{30}\sum_{j=0}^{29} X_{i,j}$, but claim to rate their ability.

The more the test scores of candidates reflect their ability, the more generalizable is the test.

The test results of record are the mean performance scores of each individual student achieved on the test. But desired ranking of students depends on their actual ability. As we have seen in equation 1, the performance depends on other factors as well, besides mere ability. So the indicator of test quality (G Coefficient) is the ratio of the variance of estimated student ability effect to the variance of their mean performance scores. In other words: "how much of the test score variance is explained by the ability effect variance?". Mathematically, that looks like this:

$$\text{G Coefficient} = \frac{\sigma_s^2}{\sigma^2(\overline{X_i})} \qquad \text{(Equation 2)}$$

More generally speaking, the literature recognizes two types of G Coefficients: Eρ^2, and Φ, depending on how the student score variance has been defined. Unfortunately we don't get around looking more carefully at 'variance', and how it is calculated.

Let us consider a set of N random numbers y_k, where k ranges from 0 to N-1.

As we have seen, its arithmetic mean is calculated as:

$$\overline{y} = \frac{1}{N}\sum_{k=0}^{N-1} y_k \qquad \text{(Equation 3)}$$

while the harmonic mean results from:

$$\tilde{y} = \frac{N}{\sum_{k=0}^{N-1}\frac{1}{y_k}} \qquad \text{(Equation 4)}$$

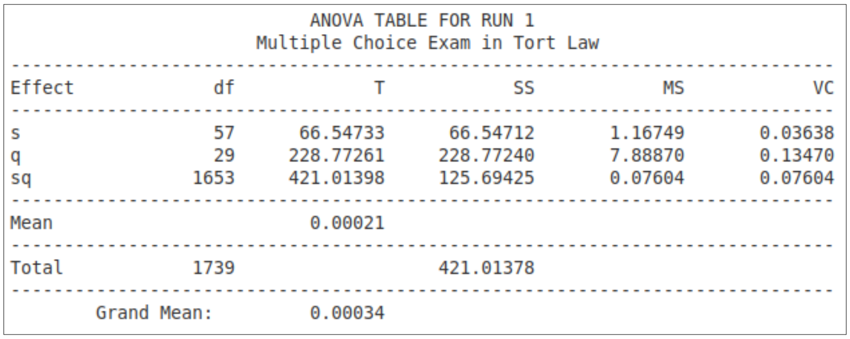
and its variance:

$$\sigma^2(y) = \frac{1}{N-1}\sum_{k=0}^{N-1}(y_k - \overline{y})^2 \qquad \text{(Equation 5)}$$

It turns out, however, that the mean $\overline{y}$ no longer has a fixed value, it now has its own, admittedly smaller variance (variance of the mean):

$$\sigma^2(\overline{y}) = \frac{\sigma^2(y)}{N}$$ (Equation 6)

Let us now get back to the output of urGenova, after it has processed the data entered, we are looking at the variance components for the two facets - student, question, and their interaction:



```
                          ANOVA TABLE FOR RUN 1
                     Multiple Choice Exam in Tort Law
----------------------------------------------------------------------------
Effect              df            T            SS            MS            VC
----------------------------------------------------------------------------
s                   57     66.54733      66.54712       1.16749       0.03638
q                   29    228.77261     228.77240       7.88870       0.13470
sq                1653    421.01398     125.69425       0.07604       0.07604
----------------------------------------------------------------------------
Mean                        0.00021
----------------------------------------------------------------------------
Total             1739                  421.01378
----------------------------------------------------------------------------
        Grand Mean:         0.00034
```

Where the variance components for the different facets and combination are to be found in column ‘VC’.

We are left with figuring out, how to express $\sigma^2(\overline{X_i})$ in terms of the calculated variance components. We must keep in mind that we are interested in the variance of the ‘facet of differentiation’* in relationship to the total score variance, i.e. averaged over all ‘facets of generalization’*. This leads us to the following expression:

$$\text{G Coefficient} = \frac{\sigma^2(\tau)}{\sigma^2(\tau)+\sigma^2(....)}$$ (Equation 7)

where $\sigma^2(\tau)$; stands for the sum of all variance components that contain the facet of differentiation, but no facets of generalization (**Brennan's Rule I.**).

Depending on the type of G Coefficient we want to calculate, formula (7) becomes:

Generalization coefficient: $$E\rho^2 = \frac{\sigma^2(\tau)}{\sigma^2(\tau)+\sigma^2(\delta)}$$ (Equation 7a)

or

Index of dependability: $$\Phi = \frac{\sigma^2(\tau)}{\sigma^2(\tau)+\sigma^2(\Delta)}$$ (Equation 7b)

where **$\sigma^2(\Delta)$** is the sum of all variance components, except for $\sigma^2(\tau)$ itself, divided by the size product of the respective facets of generalization (**Brennan's Rule II.**), and **$\sigma^2(\delta)$** the sum of all variance components that contains the facet of differentiation and at least one facet of generalization, divided by the sample size product of such. (**Brennan's Rule III.**)

Both Brennan's Rule II, and Brennan's Rule III require summing variances of means for facets of generalization. According to equation 6 this requires division of the sample variance by the total number of sample items. For set variances of single facet sets, this is simply the sample size of this facet. For purely crossed facets, it becomes the product of the facet sample sizes. But in the case of nested facets, there are multiple sample sizes, depending on the value of the nesting facet. In this case it becomes necessary to first calculate a mean of sample sizes for the nested facet. But it is no longer the arithmetic mean, the harmonic mean is required.

By general convention, $E\rho^2$ should have at least a value of 0.80 for 'High Stakes Exams'.

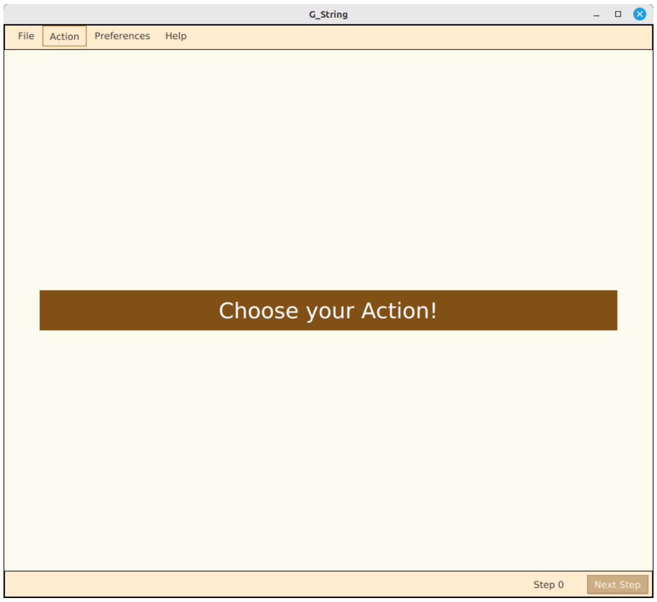
As it turns out, $E\rho^2 = 0.93$ for this case with 30 questions. But as we have seen, once we know the test design, the sample sizes, and the variance components, estimating $E\rho^2$ and Φ for a combination of different generalization sample sizes becomes a simple arithmetic operation. That’s what D Analysis does.

Elements of a Performance Test

Performance tests are an essential part of an educator's job. Just as a carpenter can not create quality furniture without a ruler, educators need to regularly assess the performance of their charges, and adapt the teaching strategies accordingly. But performance tests also play a role in rewarding, holding back, qualifying, or even dismissing students. High stakes tests, therefore, have to be defensible for both ethical and legal reasons. That's where test quality comes in.

A performance test consist of a set of subjects encountering a set of tasks that challenge the subjects to perform at their optimal level. Often it is necessary to classify both subjects (e.g. by class, gender or age), and tasks (e.g. by content area, or required skill set), and sometimes by raters. These different classification systems represent additional facets beyond subjects and tasks.

It is also the purpose of Generalizability Analysis to unravel the contribution of the various facets to performance quality.



Choose your Action!

Installing G_String GS_M

GS_MV installers can be freely downloaded as .deb (for Linux.ubuntu), .msi (for Windows), or .dmg (for mac) from https://github.com/G-String-Legacy/GS_MV/releases.

Installation is straight forward for .deb and .msi. Just click on the downloaded installer. The installation of GS_MV-2.0.7.dmg is a bit trickier because of Apple's monopolist policy. The installer has to be downloaded first on a non-Apple computer, then copied on an usb-stick. The usb-stick can then be plugged into a Mac and installed from there.

For all three installations, the first time G_String is started, it will automatically create a working directory called 'G_String_Working_Directory', and copy the appropriate machine language version of urGenova into this working directory. Should you accidentally erase that directory, just start G_String again.

G_String looks like this, when you first start it. At the very top is the menu bar:

- **File**
  - Start over
  - Save Results
  - Exit
- **Action**
  - Manual G Study
  - Scripted G Study
  - Manual Replications
  - Scripted Replications
  - Manual Synthesis
  - Scripted Synthesis
- **Preferences**
  - Change Preferences
- **Help**
  - Contextual Help
  - Background
  - urGenova Manual
  - About G_String_M
  - About urGenova

Standard Manual G Study

Next we run through all 10 steps of a regular G Study using the example of an ‘Objective Structured Clinical Examination’ (OSCE)[2]:

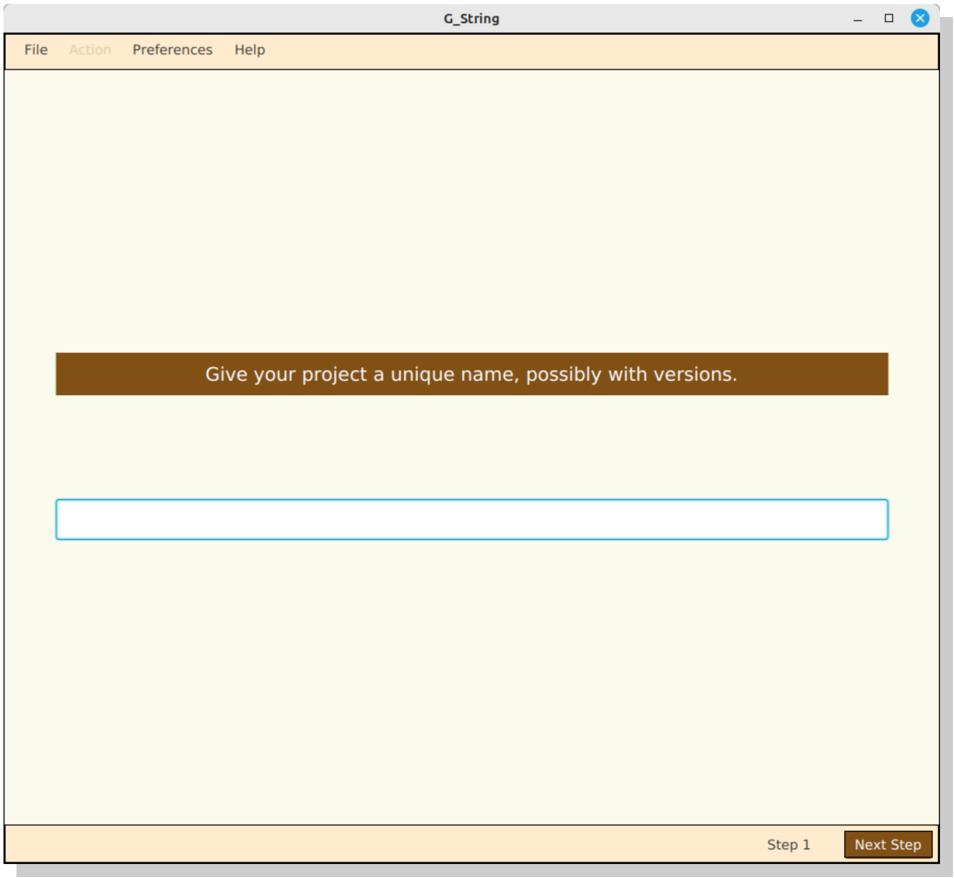
The Story

> The Salerno Medical School runs an 'Objective Structured Clinical Exam' at the end of the first clinical year. The OSCE is organized in 10 station circuits run over three days, one circuit each on morning, afternoon and evening. A total of 85 students will take the exam. The stations have been carefully designed and reviewed by a panel of 18 professors. At each station students are rated for competence in 5 areas by 2 independent raters on a 4 point scale.

We start by selecting ‘Action’ from the menu, and pick ‘Manual G Study’, because we don’t have a corresponding script as yet.

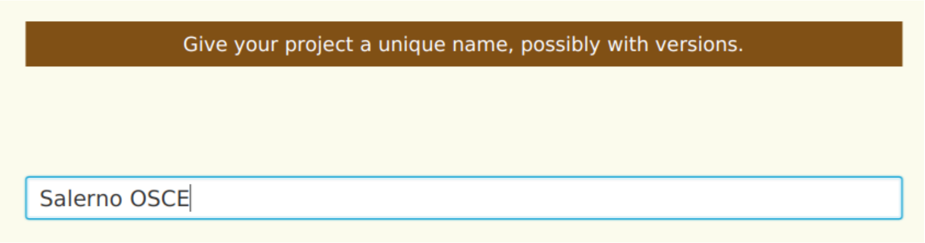
[2] See details at: https://github.com/G-String-Legacy/GS_MV/wiki/Objective-Structured-Clinical-Exam

Step 1 – Project Name



Give your project a unique name, possibly with versions.

The field for the project name is still empty, so fill it in,

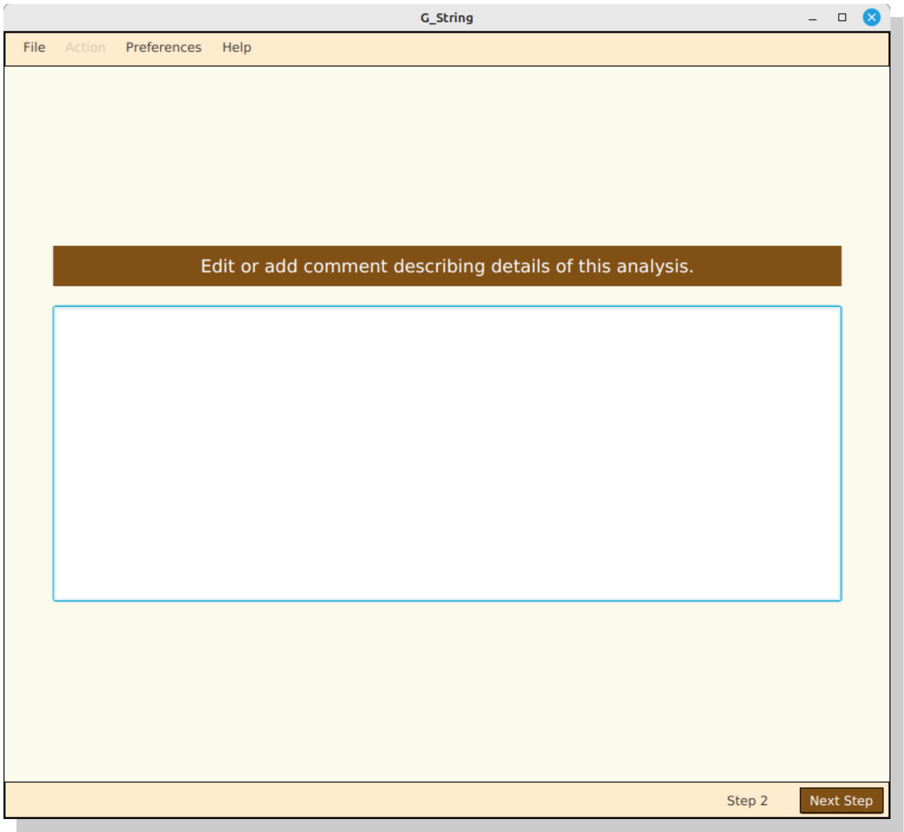


Give your project a unique name, possibly with versions.

Salerno OSCE

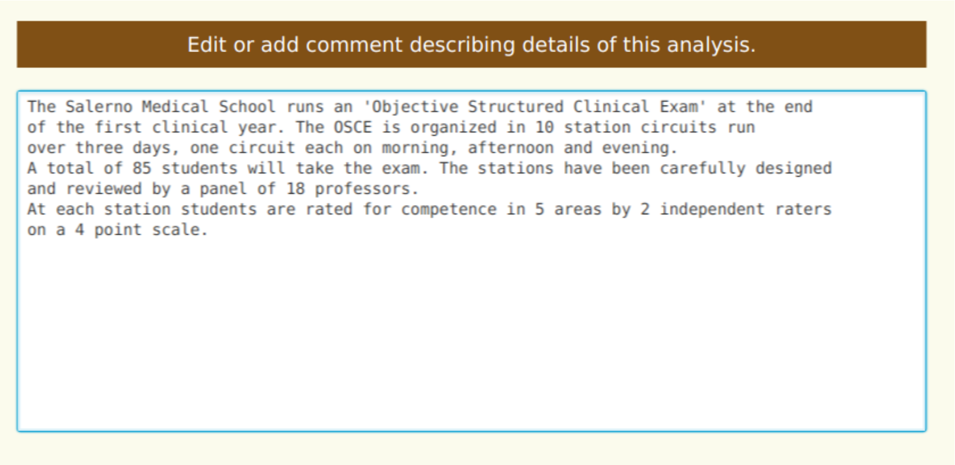
and go to the **Next Step**.

Step 2 – Project Description



Edit or add comment describing details of this analysis.

Enter a project description as detailed, as you like. But remember, adding the details does not cost you. At some future date, when you will review the project and results, the details will be handy.

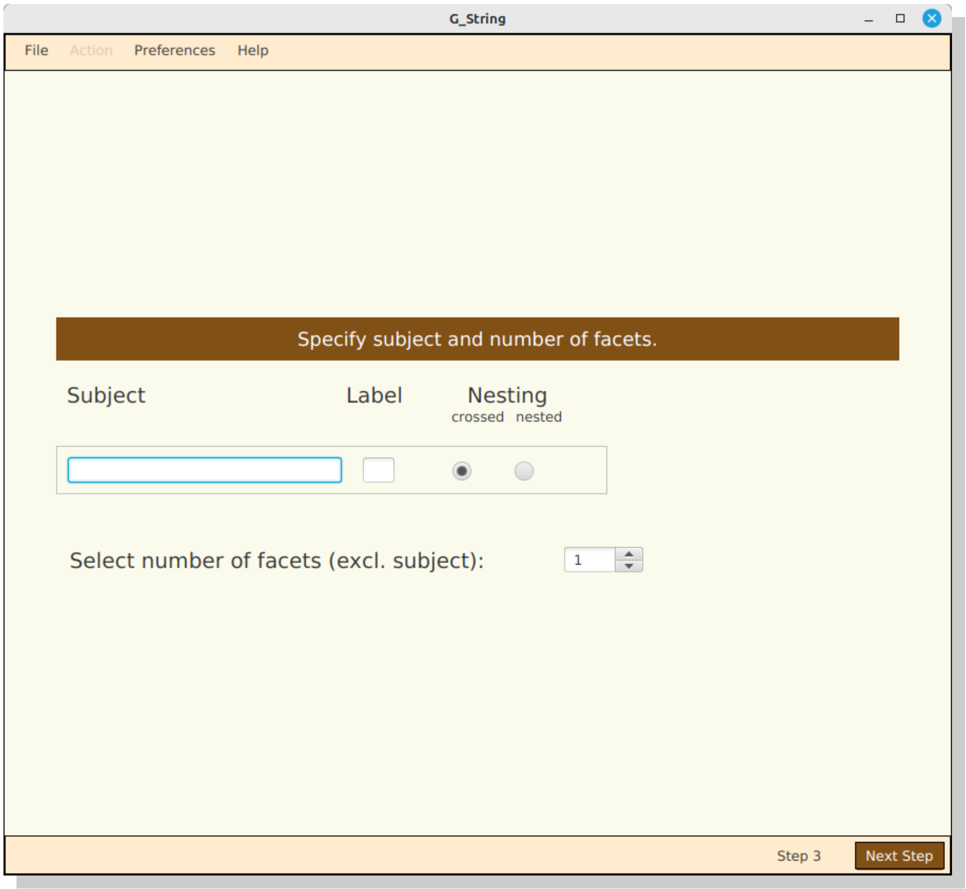


Edit or add comment describing details of this analysis.

```
The Salerno Medical School runs an 'Objective Structured Clinical Exam' at the end
of the first clinical year. The OSCE is organized in 10 station circuits run
over three days, one circuit each on morning, afternoon and evening.
A total of 85 students will take the exam. The stations have been carefully designed
and reviewed by a panel of 18 professors.
At each station students are rated for competence in 5 areas by 2 independent raters
on a 4 point scale.
```

And go to the **Next Step**.

Step 3 – Facet of Differentiation

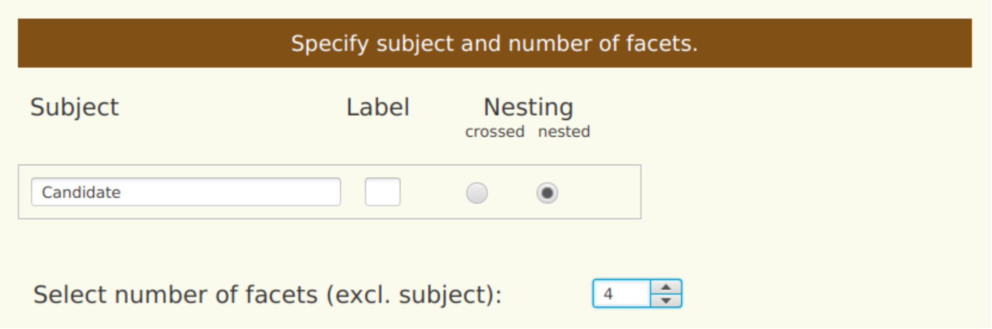


Specify subject and number of facets.

Subject | Label | Nesting: crossed ● nested ○

Select number of facets (excl. subject): 1

The Facet of Differentiation in our case is the ‘Candidate’. We label it with the lower case character ‘c’. ‘Candidate’ is nested (in Circuit), and there are 4 more facets to come:



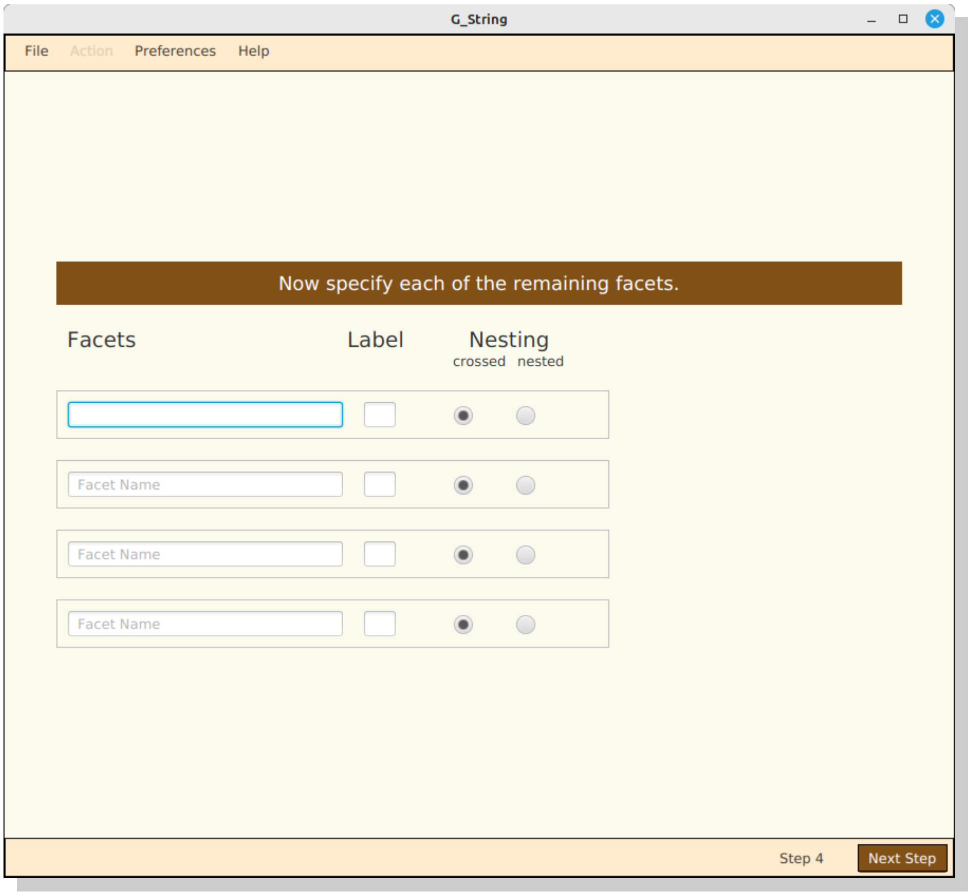
Specify subject and number of facets.

Subject: Candidate | Label | Nesting: crossed ○ nested ●

Select number of facets (excl. subject): 4

Then go on to **Next Step**.

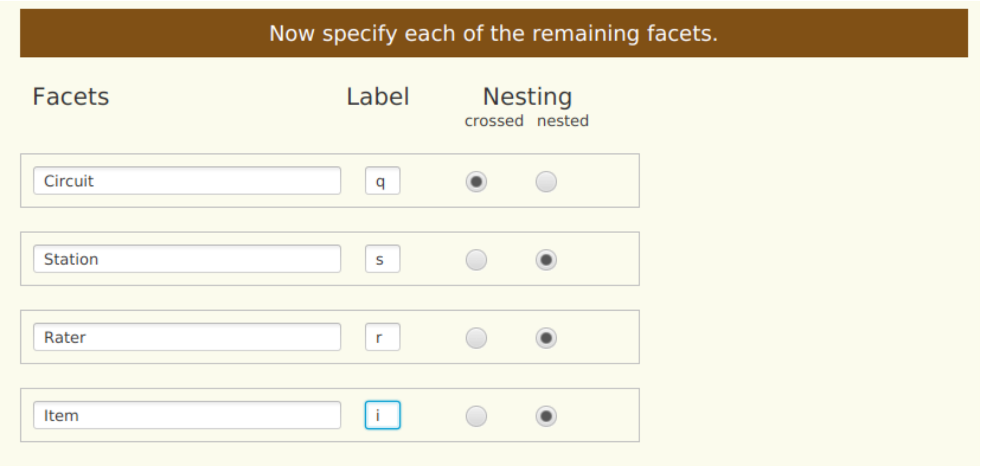
Step 4 – Facets of Generalization and Stratification



Now specify each of the remaining facets.

Facets	Label	Nesting crossed	Nesting nested
		●	○
Facet Name		●	○
Facet Name		●	○
Facet Name		●	○

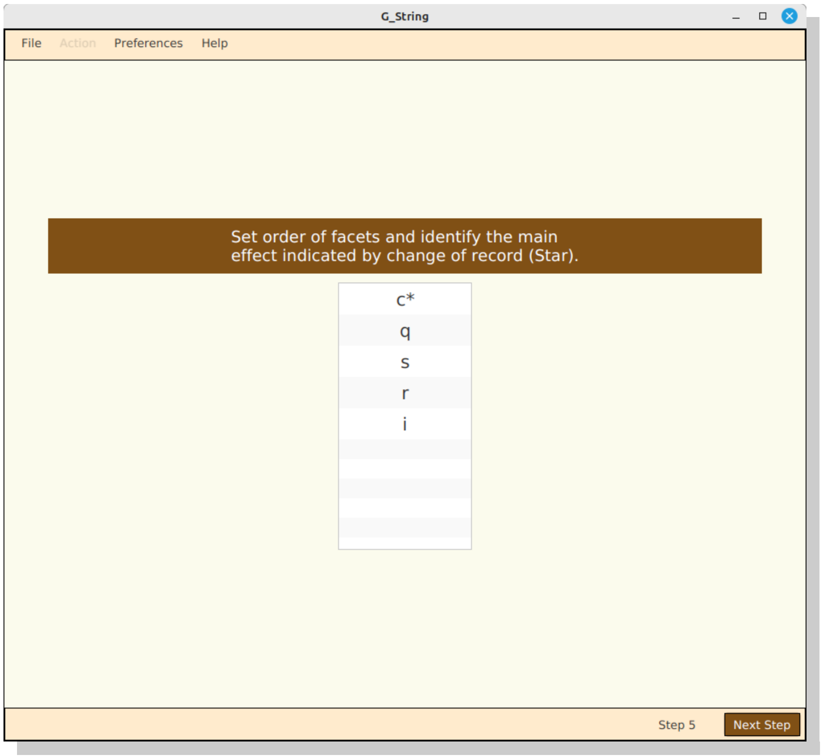
And now enter the other four facets:



Now specify each of the remaining facets.

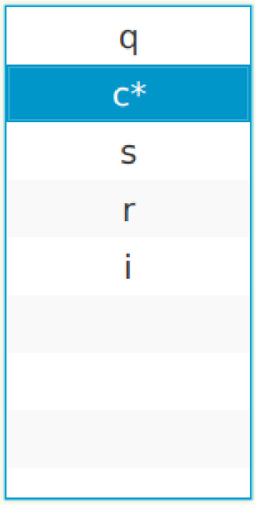
Facets	Label	Nesting crossed	Nesting nested
Circuit	q	●	○
Station	s	○	●
Rater	r	○	●
Item	i	○	●

Step 5 – Sequence of Facets in the Scores-Data File

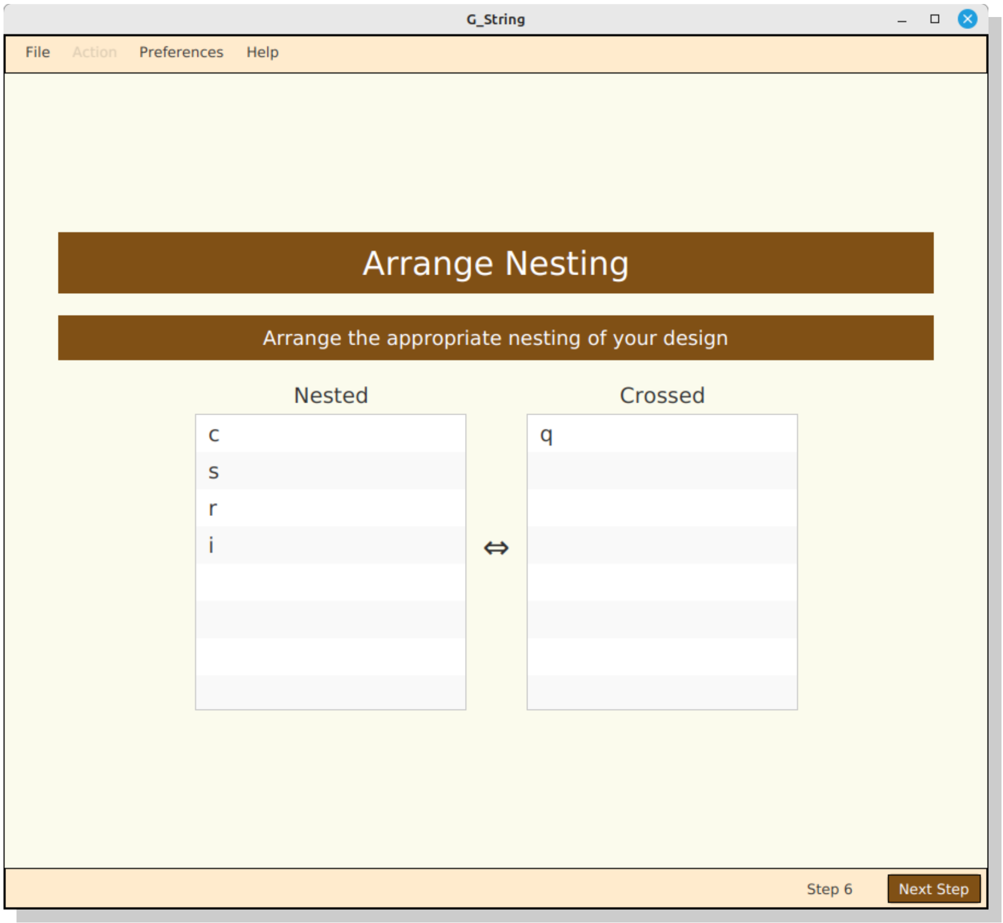


Set order of facets and identify the main effect indicated by change of record (Star).

Use the Mouse 'grab-move-drop' function to change the order of facets to match their order in the data file, and make sure that the label 'c' carries an asterisk, since the data file adds a new row, every time the candidate index changes. Any label can carry the asterisk, just click on it.



Step 6 – Crossed and Nested Facets

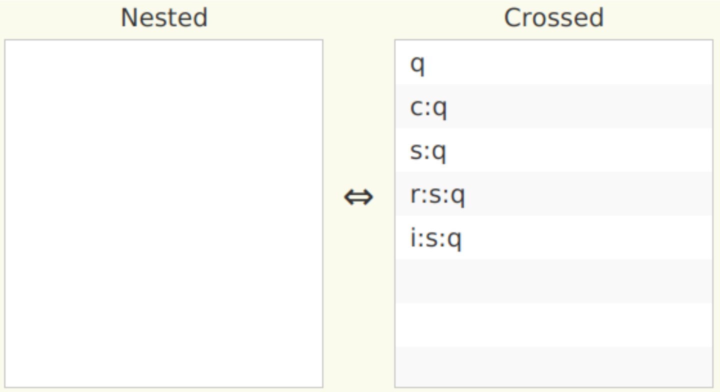


Arrange Nesting

Arrange the appropriate nesting of your design

Nested: c, s, r, i ⇔ Crossed: q

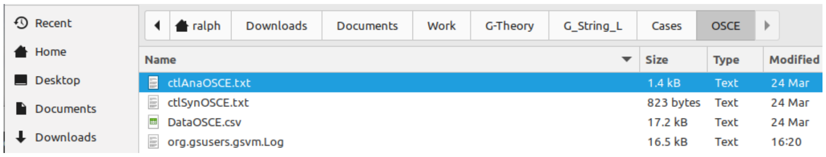
Again with ‘grab-move-drop’ establish the correct study design:



Nested: ⇔ Crossed: q, c:q, s:q, r:s:q, i:s:q

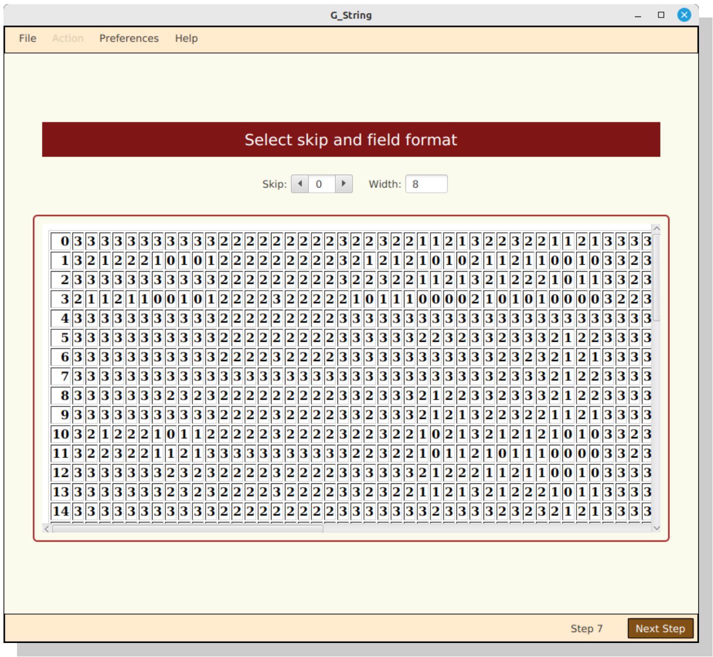
Step 7 – Scores-Data File Adjustment

After clicking on Next Step, a file selection box appears:



Name	Size	Type	Modified
ctlAnaOSCE.txt	1.4 kB	Text	24 Mar
ctlSynOSCE.txt	823 bytes	Text	24 Mar
DataOSCE.csv	17.2 kB	Text	24 Mar
org.gsusers.gsvm.Log	16.5 kB	Text	16:20

Pick the desired Scores-Data file.



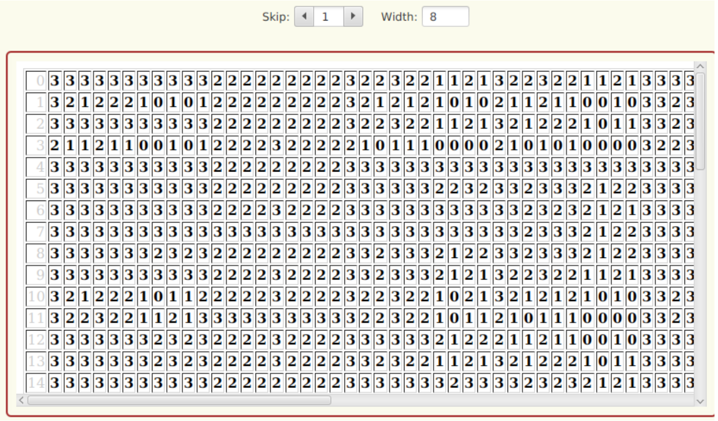
Select skip and field format

Skip: 0 Width: 8

0	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	2	2	3	2	2	1	1	2	1	3	2	2	3	2	2	1	1	2	1	3	3	3	3
1	3	2	1	2	2	2	1	0	1	0	1	2	2	2	2	2	2	2	2	2	3	2	1	2	1	2	1	0	1	0	2	1	1	2	1	1	0	0	1	0	3	3	2	3
2	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	2	2	3	2	2	1	1	2	1	3	2	1	2	2	2	1	0	1	1	3	3	2	3
3	2	1	1	2	1	1	0	0	1	0	1	2	2	2	2	3	2	2	2	2	2	1	0	1	1	1	0	0	0	0	2	1	0	1	0	1	0	0	0	0	3	2	2	3
4	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	2	2	3	2	3	3	2	3	3	3	2	1	2	2	3	3	3	3
6	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	2	3	2	3	2	1	2	1	3	3	3	3
7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	2	1	2	2	3	3	3	3
8	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	2	2	2	2	2	3	3	2	3	3	3	2	1	2	2	3	3	2	3	3	3	2	1	2	2	3	3	3	3
9	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	3	2	2	2	2	3	3	2	3	3	3	2	1	2	1	3	2	2	3	2	2	1	1	2	1	3	3	3	3
10	3	2	1	2	2	2	1	0	1	1	2	2	2	2	2	3	2	2	2	2	3	2	2	3	2	2	1	0	2	1	3	2	1	2	1	2	1	0	1	0	3	3	2	3
11	3	2	2	3	2	2	1	1	2	1	3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2	1	0	1	1	2	1	0	1	1	1	0	0	0	0	3	3	2	3
12	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	3	2	2	2	2	3	3	3	3	3	3	2	1	2	2	2	1	1	2	1	1	0	0	1	0	3	3	3	3
13	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	3	2	2	2	2	3	3	2	3	2	2	1	1	2	1	3	2	1	2	2	2	1	0	1	1	3	3	3	3
14	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	2	3	3	3	3	2	3	2	3	2	1	2	1	3	3	3	3

Step 7

Skip the first column to tell G_String that the ‘Candidate’ index should be ignored:

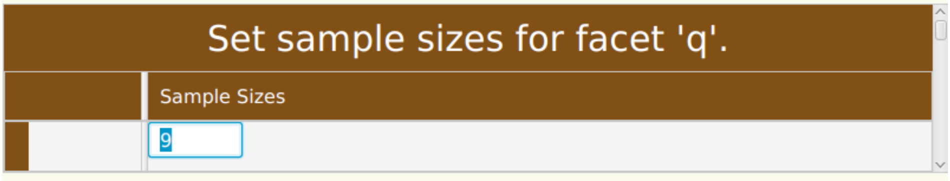


Skip: 1 Width: 8

0	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	2	2	3	2	2	1	1	2	1	3	2	2	3	2	2	1	1	2	1	3	3	3	3
1	3	2	1	2	2	2	1	0	1	0	1	2	2	2	2	2	2	2	2	2	3	2	1	2	1	2	1	0	1	0	2	1	1	2	1	1	0	0	1	0	3	3	2	3
2	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	2	2	3	2	2	1	1	2	1	3	2	1	2	2	2	1	0	1	1	3	3	2	3
3	2	1	1	2	1	1	0	0	1	0	1	2	2	2	2	3	2	2	2	2	2	1	0	1	1	1	0	0	0	0	2	1	0	1	0	1	0	0	0	0	3	2	2	3
4	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	2	2	3	2	3	3	2	3	3	3	2	1	2	2	3	3	3	3
6	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	3	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	2	3	2	3	2	1	2	1	3	3	3	3
7	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	2	1	2	2	3	3	3	3
8	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	2	2	2	2	2	3	3	2	3	3	3	2	1	2	2	3	3	2	3	3	3	2	1	2	2	3	3	3	3
9	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	3	2	2	2	2	3	3	2	3	3	3	2	1	2	1	3	2	2	3	2	2	1	1	2	1	3	3	3	3
10	3	2	1	2	2	2	1	0	1	1	2	2	2	2	2	3	2	2	2	2	3	2	2	3	2	2	1	0	2	1	3	2	1	2	1	2	1	0	1	0	3	3	2	3
11	3	2	2	3	2	2	1	1	2	1	3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2	1	0	1	1	2	1	0	1	1	1	0	0	0	0	3	3	2	3
12	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	3	2	2	2	2	3	3	3	3	3	3	2	1	2	2	2	1	1	2	1	1	0	0	1	0	3	3	3	3
13	3	3	3	3	3	3	3	2	3	2	3	2	2	2	2	3	2	2	2	2	3	3	2	3	2	2	1	1	2	1	3	2	1	2	2	2	1	0	1	1	3	3	3	3
14	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	2	3	3	3	3	2	3	2	3	2	1	2	1	3	3	3	3

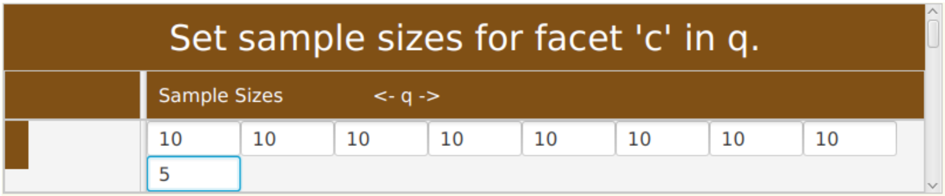
Step 8 – Sample Sizes

Next you have to step through all 5 facets to select the sample sizes for the current project by entering the desired values:



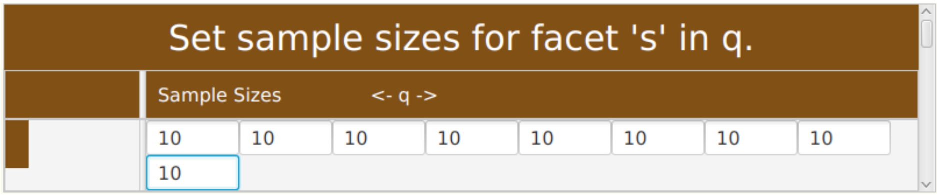
Set sample sizes for facet 'q'.

	Sample Sizes
	9



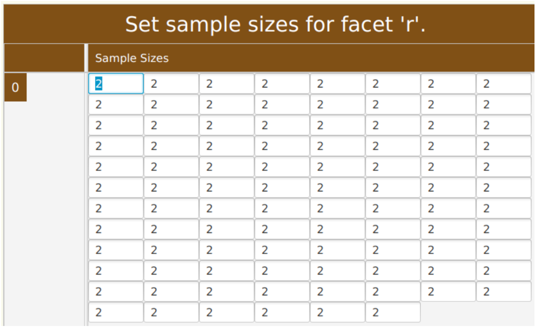
Set sample sizes for facet 'c' in q.

	Sample Sizes	<- q ->						
	10	10	10	10	10	10	10	10
	5							



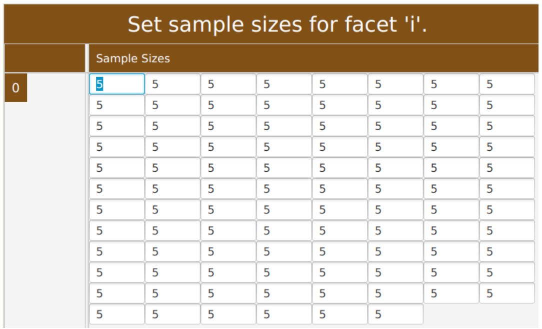
Set sample sizes for facet 's' in q.

	Sample Sizes	<- q ->						
	10	10	10	10	10	10	10	10
	10							



Set sample sizes for facet 'r'.

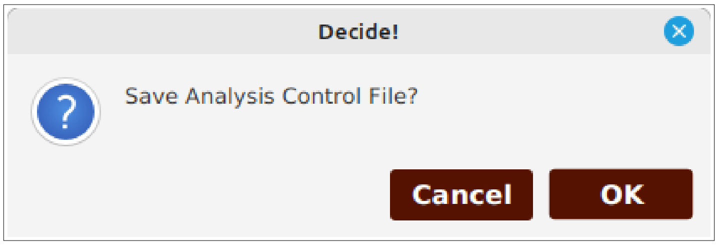
	Sample Sizes							
0	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2	2	2
	2	2	2	2	2	2		



Set sample sizes for facet 'i'.

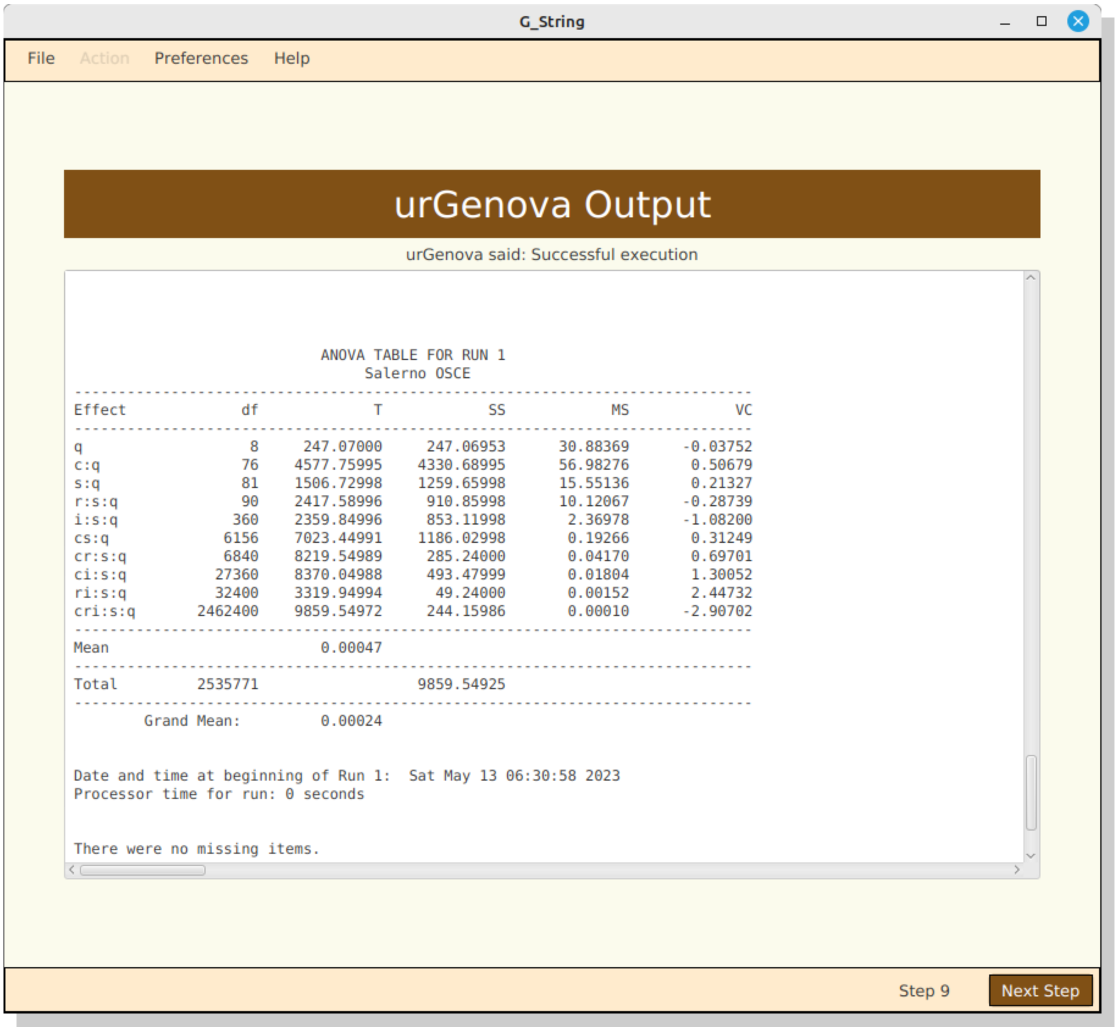
	Sample Sizes							
0	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5	5	5
	5	5	5	5	5	5		

Step 9 – urGenova Results



Save Analysis Control File?

Decide whether you want the parameters, you have entered, as a script, or not.



urGenova said: Successful execution

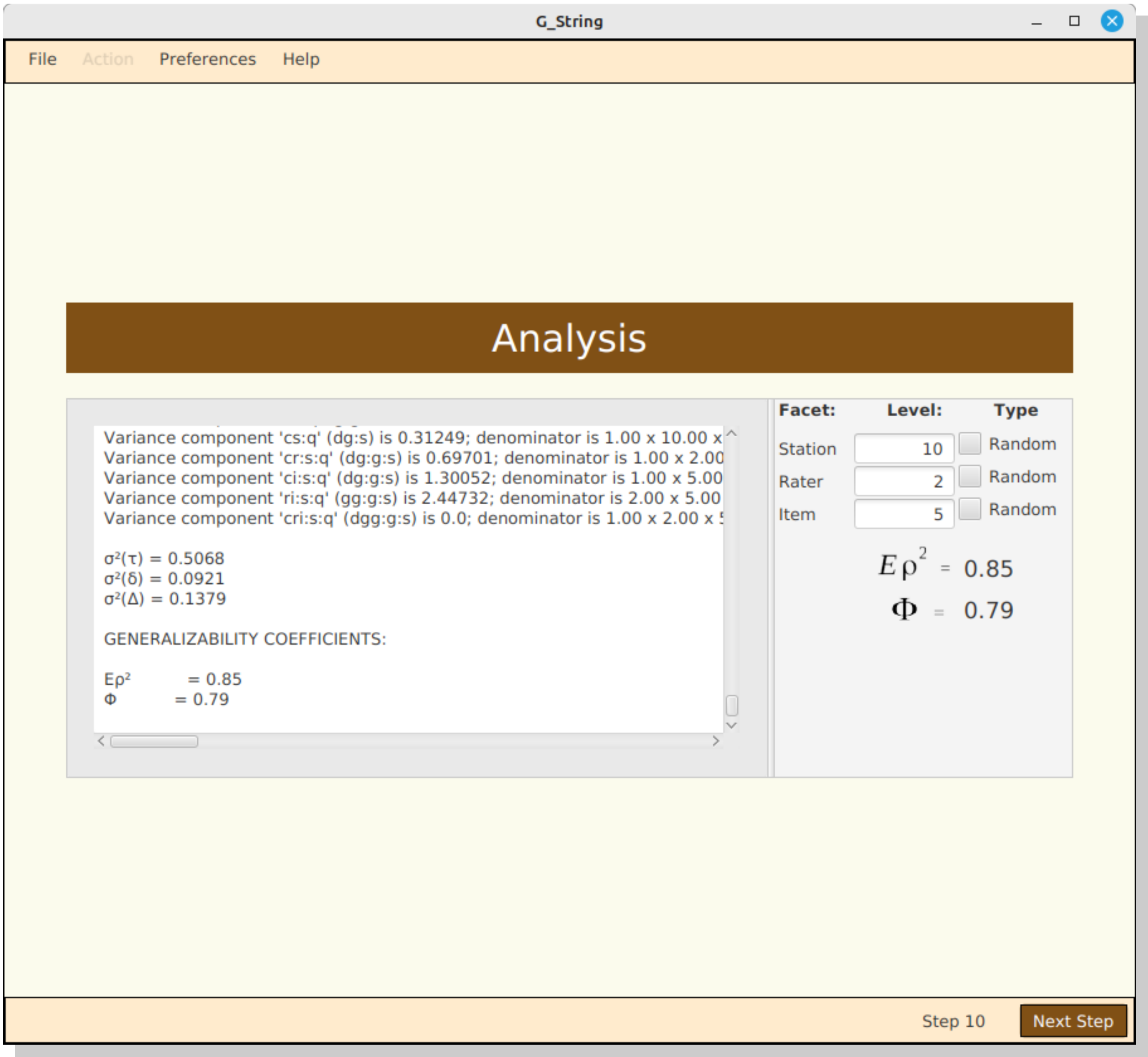
```
                            ANOVA TABLE FOR RUN 1
                                 Salerno OSCE
-----------------------------------------------------------------------------
Effect             df             T            SS            MS            VC
-----------------------------------------------------------------------------
q                   8     247.07000     247.06953      30.88369      -0.03752
c:q                76    4577.75995    4330.68995      56.98276       0.50679
s:q                81    1506.72998    1259.65998      15.55136       0.21327
r:s:q              90    2417.58996     910.85998      10.12067      -0.28739
i:s:q             360    2359.84996     853.11998       2.36978      -1.08200
cs:q             6156    7023.44991    1186.02998       0.19266       0.31249
cr:s:q           6840    8219.54989     285.24000       0.04170       0.69701
ci:s:q          27360    8370.04988     493.47999       0.01804       1.30052
ri:s:q          32400    3319.94994      49.24000       0.00152       2.44732
cri:s:q       2462400    9859.54972     244.15986       0.00010      -2.90702
-----------------------------------------------------------------------------
Mean                        0.00047
-----------------------------------------------------------------------------
Total         2535771                  9859.54925
-----------------------------------------------------------------------------
        Grand Mean:         0.00024


Date and time at beginning of Run 1:  Sat May 13 06:30:58 2023
Processor time for run: 0 seconds


There were no missing items.
```

Scroll the window almost to the bottom, in order to be able to inspect the Variance Component for each facet (in column ‘VC’).

Step 10 – G Coefficients



```
Variance component 'cs:q' (dg:s) is 0.31249; denominator is 1.00 x 10.00 x
Variance component 'cr:s:q' (dg:g:s) is 0.69701; denominator is 1.00 x 2.00
Variance component 'ci:s:q' (dg:g:s) is 1.30052; denominator is 1.00 x 5.00
Variance component 'ri:s:q' (gg:g:s) is 2.44732; denominator is 2.00 x 5.00
Variance component 'cri:s:q' (dgg:g:s) is 0.0; denominator is 1.00 x 2.00 x

σ²(τ) = 0.5068
σ²(δ) = 0.0921
σ²(Δ) = 0.1379

GENERALIZABILITY COEFFICIENTS:

Eρ²         = 0.85
Φ         = 0.79
```

Facet:	Level:	Type
Station	10	□ Random
Rater	2	□ Random
Item	5	□ Random

$E\rho^2 = 0.85$

$\Phi = 0.79$

In this window you get a further view on the G_String results so far. You have the option to now perform **D Studies** by entering various values into the mean sample size level fields at the top right, and get the new G Coefficients after clicking **Next Step** each time.

When you are done, don’t forget to save all the results with **File > Save Results**.

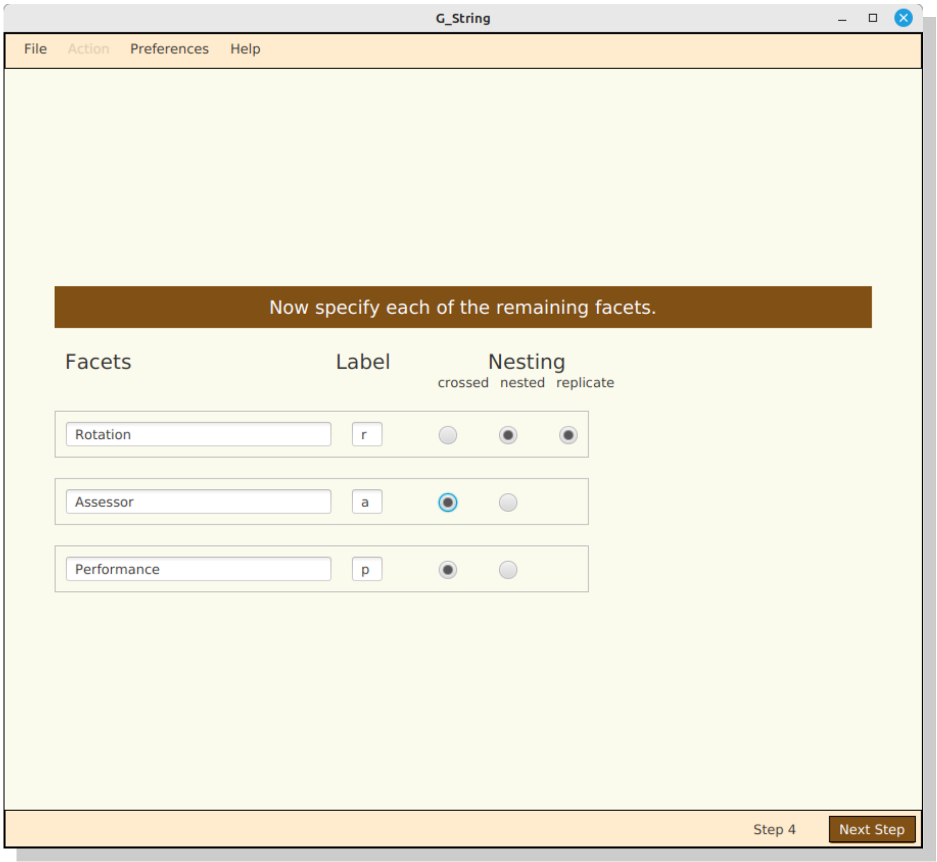
G Study with Replication

Most of the time all the sample sizes are fixed as part of the study design, or a few have to be adjusted at the time, the G Study is run. However, a major exception exists, It occurs in situations, when the sample sizes of a facet of generalization nested in the facet of differentiation differ from subject to subject, such as in the evaluation of resident, or clinical clerk rotations, or of teachers by students.

Even though urGenova would allow specifying such a situation in the script, this is very cumbersome, and error prone extra work. The 'Replication' option of G_String makes short shrift of it. The sample size is extracted directly from the scores-data file. But only one facet can be replicating.

When you choose the replication option as action, only step 4 differs from that of the simple G Study, and the program won't prompt you for the corresponding sample sizes.

Let us consider Clinical Clerkship Rotations[3]. In replication mode, the 'replicate' option is offered for a facet, when 'nested' has been selected. But the asterisk must have been set either on the facet of differentiation, or its directly nested facet of generalization.



Now specify each of the remaining facets.

Facets	Label	crossed	nested	replicate
Rotation	r	○	●	●
Assessor	a	●	○	
Performance	p	●	○	

[3] https://github.com/G-String-Legacy/GS_MV/wiki/Clinical-Clerkship-Grading

Synthetic Data Files

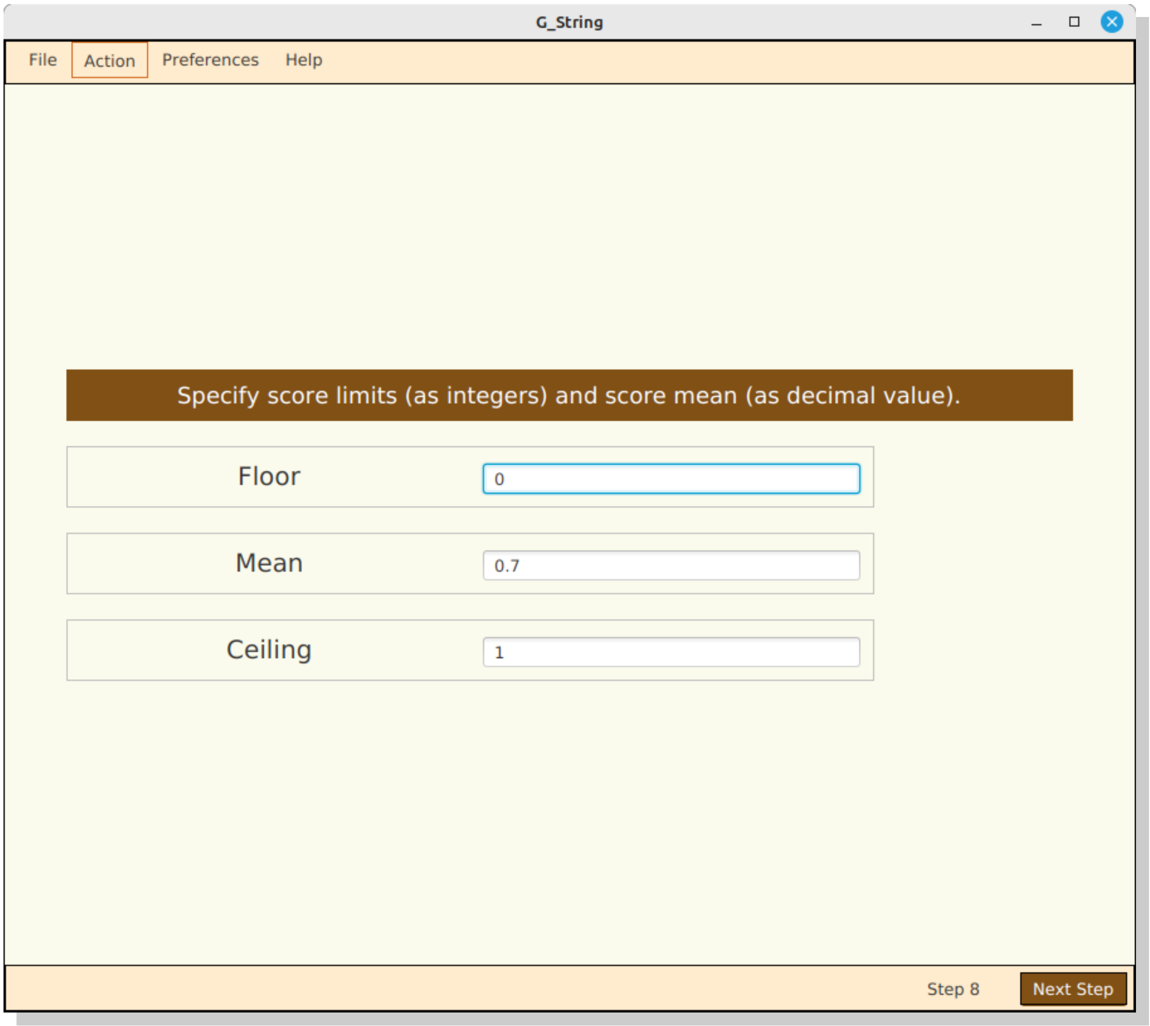
The ‘Synthesis’ action can be of interest to designers of complex studies, or to educators of Generalizability Analysis. Rather than having to feed G_String a data-score file, the program generates a synthetic data-score file, based on parameter data entered.

As illustration, we use our familiar MCQ example. Most of the steps in Synthesis correspond to the G Study format. We only discuss here those steps that differ.

Up to, and including step 6 nothing has changed. The analytic step 7 read in the empirical data file, obviously, is no longer relevant here. Instead, we skip directly to the sample size sequence.

In step 8 we enter specifications for the score, such as lowest possible score (Floor), highest possible score (Ceiling), and the Mean (μ_0). Floor and Ceiling are entered as integers, while Mean is entered as decimal number.

We show here the forms with MCQ synthesis parameters already entered:



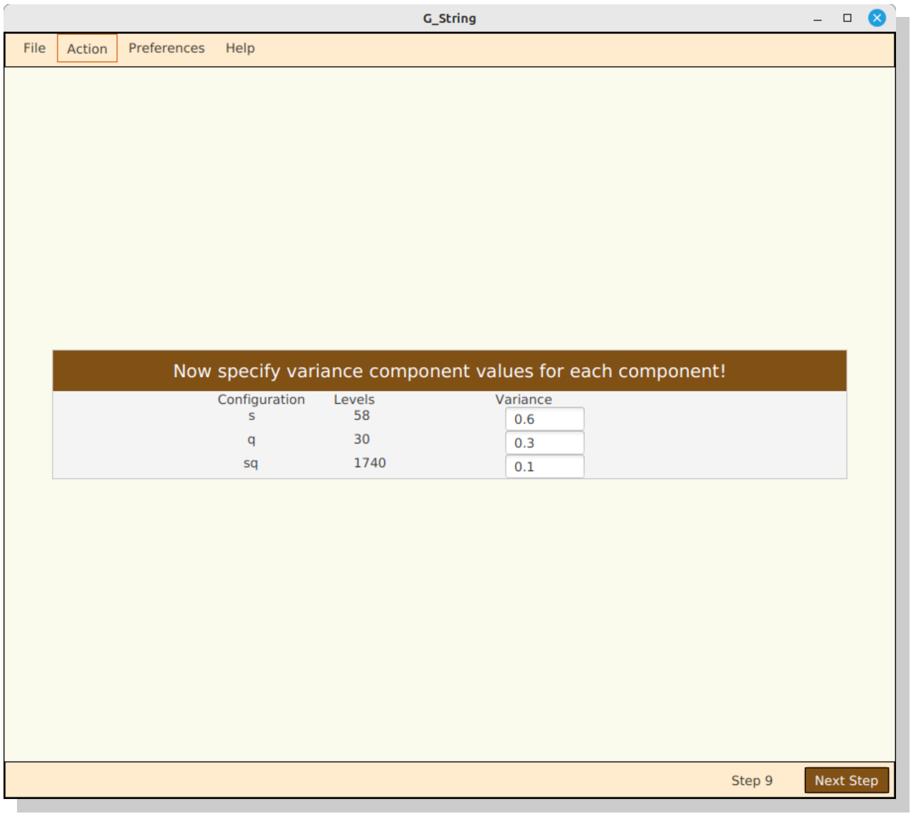
Specify score limits (as integers) and score mean (as decimal value).

Floor 0

Mean 0.7

Ceiling 1

In step 9 G_String has already figured out for which facets, and facet combinations variance components are required. We guess variance values and experiment, until we find a resulting data set that meets our requirement.

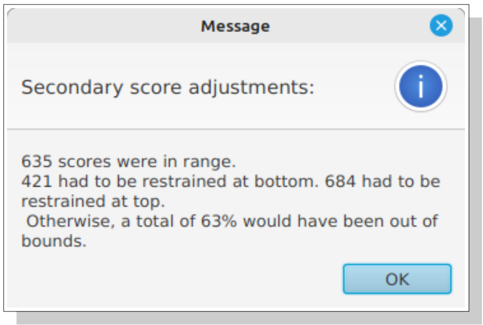


Now specify variance component values for each component!

Configuration	Levels	Variance
s	58	0.6
q	30	0.3
sq	1740	0.1

Next, G_String offers the option to save the synthesis script, and then prompts for where to save the resulting synthetic score-data file.

Finally, it provides a summary feedback of the synthesis:



Secondary score adjustments:

635 scores were in range.
421 had to be restrained at bottom. 684 had to be restrained at top.
Otherwise, a total of 63% would have been out of bounds.

To keep enterprising graduate students from generating fake experimental results, G_String weaves a signature into the synthetic data file. On analyzing as G Study, the saved results show a ‘Sig’ < 10 at the very end. In contrast, empiric studies show ‘Sig’ >> 10.

Problems

The current version of G_String has evolved over 16 years. It comprises over 7000 lines of source code, and compiles to about 80 Mb. But the author had little assistance in debugging the program. It is, therefore not surprising, when all kinds of little bugs may still raise their nasty heads from time to time. Many problems have been caught and fixed over the years, and some are probably still lurking. This, however, is not unusual in complex software, and it is the reason, why computer programs need to be updated periodically.

So don’t panic, if the program should suddenly behave in an unusual way. Here are some rational steps to help you with unexpected situations:

1. Exit, and restart the program, Then try to copy the previous sequence of steps, and check if the same problem happens again. If so, write down the step, where the problem occurred (bottom right).
2. Copy the most appropriate template[4] matching your problem, try to run it from the script, and check if your problem arises as well. If so, your installation might have become compromised.
3. If not, we might be dealing with a new bug. Open the most recent log file in the current data directory.
4. Go to the ‘Discussion forum’[5] and start a new discussion. Give a detailed explanation of what you were trying to do, what happened, in which step it occurred, and what the log file said. Hopefully, other members of the user group will help you solving the problem. Possibly, they may raise it to an ‘Issue’[6]. That means usually that a software expert has to become involved.

In order for this problem solving sequence to function, it is necessary that the user group coalesces, and provides mutual support, if necessary even contributing to the cost of hiring a programmer, when the author of G_String is no longer around.

The core of the G_String website[7] on GitHub contains all the technical information, an expert would need to solve the problem.

[4] https://github.com/G-String-Legacy/GS_MV/wiki/Hands-on-Examples
[5] https://github.com/G-String-Legacy/GS_MV/discussions
[6] https://github.com/G-String-Legacy/GS_MV/issues
[7] https://github.com/G-String-Legacy/GS_MV

Appendix I: Explanation of Jargon Terms

Where Generalizability Analysis is applied in fields other than performance assessment, such as dog shows, or price rating for art objects, the terminology needs to be adapted accordingly.

D Study: ‘what if’ analysis after performing a G Study by varying various sample sizes in order to reach a desired compromise between test effort and Eρ^2.

Effect: amount the task score of a Candidate deviates from the mean due to each facet.

Facets: are the various factors of a particular assessment design that affect actual candidate performance scores.

Facet of Differentiation: the candidate for whom we are attempting to estimate performance quality.

Facets of Generalization: the facets that define define the rating process.

Facets of Stratification: allow candidates to be rated within subcategories.

Generalizability Analysis, G Study: see chapters ‘Estimating Variance Components’ and ‘G Coefficients’.

Java: high level computer programming language requiring extensive libraries.

JavaFX: special syntax and library for graphic display, assisting Java.

Replications: special case of G Study where the sample size of tasks is not fixed, but varies from candidate to candidate. These sample sizes are extracted automatically from the data file.

Score: ordinal, numerical performance rating (per candidate and task), either criteria based, or by rater.

Synthesis: generating an artificial data file corresponding to a desired test design and variance. components.

Task: individual challenge on which the performance of a candidate is scored.

urGenova: Machine language routine created by Robert L. Brennan to efficiently calculate Variance Components.

Variance: amount of variation in data set; see definition in Equation 5.

Variance Components: a set of variances, one for each facet.

Appendix II : Bibliography

- **Brennan, R.L.** Generalizability Theory. New York, Springer, (2003) "*The "bible" of G theory)*"

- **Cronbach, L.J., Gleser, G.C., Nanda, H., & Rajaratnam, N.** (1972). The dependability of behavioral measurements: Theory of generalizability for scores and profiles. New York: John Wiley.

- **Shavelson, P., Webb, N.** Generalizability Theory: A Primer. (1991) Thousand Oaks CA, SAGE

- **Bloch R., Norman G.** Generalizability theory for the perplexed: a practical introduction and guide: AMEE Guide No. 68. Med Teach. 2012;34(11):960-92.

- **Andersen S.A.W., Nayahangan L.J., Park Y.S., Konge L.** Use of Generalizability Theory for Exploring Reliability of and Sources of Variance in Assessment of Technical Skills: A Systematic Review and Meta-Analysis. Acad Med, Vol. 96/11 2021.

- **Wikipedia:** Generalizability theory

Alphabetical Index