Generalizability Theory in R

Alan Huebner and Marisa Lucht, *University of Notre Dame*

Generalizability theory is a modern, powerful, and broad framework used to assess the reliability, or dependability, of measurements. While there exist classic works that explain the basic concepts and mathematical foundations of the method, there is currently a lack of resources addressing computational resources for those researchers wishing to apply generalizability in practice. This tutorial illustrates the efficient application of generalizability theory in the statistical software environment R, including data formatting, computing key quantities, and tabulating and visualizing results for four different study designs. The tutorial is entirely self-contained, as it includes R code to generate the example data thus allowing the reader to reproduce all computations, tables, and plots.

Generalizability theory (G-theory) is a statistical method that characterizes the reliability (or dependability or consistency) of measurements. G-theory has been referred to as a "conceptual breakthrough" (Suen & Ary, 1989) and "perhaps the most broadly defined measurement model currently in existence" (Brennan, 2001). G-theory has been applied to a wide variety of fields, including, for example, athletic training (Heitman, Kovaleski, & Pugh, 2009), educational assessment (Gugiu, Gugiu, & Baldus, 2012), and social behavior (Vispoel, Morris, & Kilinc, 2018).

There are several excellent resources for learning the fundamentals of G-theory. Shavelson and Webb (1991) provide an excellent primer suitable for readers wishing to use G-theory in practical situations. The classic text of Brennan (2001) covers introductory and advanced topics in greater mathematical detail. Suen and Ary (1989), de Vet et al. (2011), and Thompson (2003) give shorter treatments. These sources are essential for understanding the underlying concepts of G-theory. However, for practitioners who have mastered the basic concepts, there is a relative scarcity of resources for actually applying G-theory—i.e. starting with raw data and performing computations, visualizing results, and preparing reports in a seamless manner. This paper will illustrate how to conduct a G-theory analysis in R, a free software environment for statistical computing and graphics.

We emphasize that the present article is not a substitute for the outstanding references listed above; it is intended as a practical companion to them. This tutorial assumes some basic prerequisite knowledge of both G-theory (i.e. familiarity with chapters 1-7 of Shavelson and Webb [1991] and/or chapters 1-4 of Brennan [2001]) and R (i.e. defining variables, subsetting using brackets, using functions, and loading packages). Section 2 will introduce the two parts of a G-theory analysis, the G study and D study, as well as discuss currently available computational resources for applying G-theory. Section 3 will provide examples for obtaining G-theory results in R for a number of different experimental designs commonly used with G-theory. Section 4 will illustrate how to create visualizations of G-theory results as well as how to format the data. Section 5 will conclude by pointing out limitations of the current paper and suggesting avenues for further study.

Overview of G-theory

We present a brief review of G-theory to establish definitions and mathematical notation to facilitate the subsequent discussion. Every application of G-theory deals with a sample of objects of measurement from a population. This paper will refer to these as "persons", but the subjects may also be nonhuman. In addition, the G-theory framework defines "facets" as "sources of measurement error" (Shavelson & Webb, 1991). For

example, if a sample of people are administered a set of items that are graded by different raters on multiple occasions, then items, raters, and occasions would all be considered facets in the study. Most examples and applications of G-theory have one or two facets, but any number is possible, in theory. This paper will focus on study designs with one or two facets. For the one-facet and two-facet examples, we will use items and items and occasions, respectively, though other facets could be substituted.

G-theory also takes into account the design of a given study by specifying whether the facets are crossed or nested. In a crossed design, all conditions of one facet are observed in all conditions of every other facet. For example, in a crossed one-facet design with items as the only facet, every person is measured on each item, as this is referred to as a $p \times i$ design. The measurement for person p on item i is notated as X_{pi} and is expressed as

$$X_{pi} = \mu + \nu_p + \nu_i + \nu_{pi} \qquad (1)$$

Here, μ is the grand mean and the ν's are referred to as effects, or components, in the language of G-theory (Brennan, 2001). In particular, ν_p and ν_i are the main effects and ν_{pi} is the pi interaction effect. This interaction effect is sometimes denoted as $\nu_{pi,e}$ to express that it is confounded with unmeasured or unsystematic variability (Shavelson & Webb, 1991). For an example of a two-facet crossed design, consider an application with two facets, items and occasions. Then, all persons are administered the same items on all occasions, and this is notated as a $p \times i \times o$ design. In this case, X_{pio}, the measurement for person p on item i for occasion o, is given by

$$X_{pio} = \mu + \nu_p + \nu_i + \nu_o + \nu_{pi} + \nu_{po} + \nu_{io} + \nu_{pio} \qquad (2)$$

In addition, G-theory accommodates designs in which facets are nested within one another. For example, if the single-facet scenario described above was modified so that each person was administered a unique set of items (rather than all persons receiving the same items), then items would be nested in persons. This design is referred to as a $i{:}p$ design and results in a modified version of equation (1):

$$X_{pi} = \mu + \nu_p + \nu_{i:p} \qquad (3)$$

Comparing to equation (2), there is no distinct ν_i term in (3), and the residual effect is different. Both differences are because the nested design confounds the

ν_i and ν_{pi} effects (Brennan, 2001). An example of a nested design with two facets is the $p \times (i{:}o)$ design, in which all persons answer the same items, but the items are different on each occasion. This design is notated as

$$X_{pio} = \mu + \nu_p + \nu_o + \nu_{i:o} + \nu_{po} + \nu_{pi:o} \qquad (4)$$

Appendix 4.2 of Shavelson and Webb (1991) gives the mathematical formulations and intuitive descriptions for all possible nesting configurations for two-facet designs.

After the appropriate design is identified, the G-theory analysis may proceed. There are two elements to G-theory, generalizability (G) studies and decision (D) studies. Though the distinction is not always precise, G studies ordinarily involve estimating the variance of the measurements, which is decomposed into the variance of the components; e.g. for the $p \times i \times o$ design, the G study obtains the variance estimates

$$\hat{\sigma}^2(X_{pio}) = \hat{\sigma}^2(p) + \hat{\sigma}^2(i) + \hat{\sigma}^2(o) + \hat{\sigma}^2(pi) \\ + \hat{\sigma}^2(po) + \hat{\sigma}^2(io) + \hat{\sigma}^2(pio) \qquad (5)$$

This information allows the researcher to identify the sources contributing the greatest variability to the measurements. The G study variance components for the four designs described above are listed in the second column of Table 1. Formulas to compute these quantities are based on results from analysis of variance (ANOVA); see Shavelson and Webb (1991) and Brennan (2001) for their derivations.

The D study is used to identify the optimal number of conditions of each facet in order to maximize reliability. The D study variance components can be derived from the G study variance components listed in Table 1 by dividing the G study components by the proposed number of facets n'_i and n'_o. In order to indicate D study variance components, the facets are denoted with uppercase letters. Note, for all designs the $\sigma^2(p)$ component is the same in the G study and D study because it is assumed the number of objects of measurement will not be changed.

The D study also involves the computation of two reliability coefficients, $\boldsymbol{E\rho^2}$, the "generalizability coefficient" and $\boldsymbol{\Phi}$, the "index of dependability" (Brennan, 2001), given by

$$\boldsymbol{E\rho^2} = \frac{\sigma^2(\tau)}{\sigma^2(\tau) + \sigma^2(\delta)} \qquad (6)$$

$$\Phi = \frac{\sigma^2(\tau)}{\sigma^2(\tau) + \sigma^2(\Delta)} \qquad (7)$$

Both formulas depend on $\sigma^2(\tau)$, the variance component for persons (or objects). Note, for the examples in this paper, $\sigma^2(\tau) = \sigma^2(p)$, because all facets are considered "random", i.e. the sample size is much smaller than the population. The $\sigma^2(\delta)$ and $\sigma^2(\Delta)$ in equations (6) and (7) are referred to, respectively, as the "relative" and "absolute" error variances. The same general expressions for $\boldsymbol{E\rho^2}$ and $\boldsymbol{\Phi}$ seen in (6) and (7) are the same for various designs, but the computational formulas for $\sigma^2(\delta)$ and $\sigma^2(\Delta)$ vary. Table 2 shows the formulas for $\sigma^2(\delta)$ and $\sigma^2(\Delta)$ for the four designs considered above.

Existing Computational resources for G-theory

While there are excellent resources available that address the statistical concepts of G-theory, there is currently limited information on methods to perform computations and present results efficiently. Vispoel et al. (2018) comment that "a major stumbling block to using G-theory has been the lack of readily available software to conduct the analyses". For instance, Shavelson and Webb (1991) provide example code for the BMDP software, which is no longer available

(Statsols, 2019). Furthermore, Suen and Ary (1989) make no mention of software, and Thompson (2003) supplies a few lines of SPSS example code. Brennan (2001) references the use of the GENOVA Suite of programs for G-theory analyses written in C and FORTRAN, which are available from the University of Iowa College of Education website. The programs are free and include detailed instruction manuals. Vispoel et al. (2018) discuss a number of computational resources for G-theory, but some such as SAS and SPSS are not freely available.

On the other hand, the open-source and freely available statistical software R (R Core Team, 2019) has the advantages of offering a rich environment in which data cleaning, modeling, visualization, and reporting may be performed seamlessly. The package `gtheory` (Moore, 2016) performs G-theory analyses via two functions, `gstudy()` and `dstudy()`. The function `gstudy()` has several arguments, but for this paper, we will only use the first two:

```
gstudy(data,formula)
```

The data argument is simply the name of the data set in R, and formula is an expression indicating the study design and facets. Like `gstudy()`, the

Table 1. G study and D study variance components and computational formulas for the four study designs.

Design	G Study Variance Components	D Study Variance Components and Formulas
$p \times i$	$\sigma^2(pi)$ $\sigma^2(p)$ $\sigma^2(i)$	$\sigma^2(pI) = \sigma^2(pi)/n_i'$ $\sigma^2(p) = \sigma^2(p)$ $\sigma^2(I) = \sigma^2(i)/n_i'$
$p \times i \times o$	$\sigma^2(pio)$ $\sigma^2(pi)$ $\sigma^2(po)$ $\sigma^2(io)$ $\sigma^2(p)$ $\sigma^2(i)$ $\sigma^2(o)$	$\sigma^2(pIO) = \sigma^2(pio)/n_i'n_o'$ $\sigma^2(pI) = \sigma^2(pi)/n_i'$ $\sigma^2(pO) = \sigma^2(po)/n_o'$ $\sigma^2(IO) = \sigma^2(io)/n_i'n_o'$ $\sigma^2(p) = \sigma^2(p)$ $\sigma^2(I) = \sigma^2(i)/n_i'$ $\sigma^2(O) = \sigma^2(o)/n_o'$
$i : p$	$\sigma^2(i, pi)$ $\sigma^2(p)$	$\sigma^2(I, pI) = \sigma^2(i, pi)/n_i'$ $\sigma^2(p) = \sigma^2(p)$
$p \times (i : o)$	$\sigma^2(pi, pio)$ $\sigma^2(i, io)$ $\sigma^2(po)$ $\sigma^2(p)$ $\sigma^2(o)$	$\sigma^2(pI, pIO) = \sigma^2(pi, pio)/n_i'n_o'$ $\sigma^2(I, IO) = \sigma^2(i, io)/n_i'n_o'$ $\hat\sigma^2(pO) = \sigma^2(po)/n_o'$ $\sigma^2(p) = \sigma^2(p)$ $\sigma^2(O) = \sigma^2(o)/n_o'$

Table 2. D study relative and absolute variance computational formulas for four study designs.

Design	Error Variance	Computational Formulas
$p \times i$	$\sigma^2(\delta)$	$\sigma^2(pI)$
	$\sigma^2(\Delta)$	$\sigma^2(I) + \sigma^2(pI)$
$p \times i \times o$	$\sigma^2(\delta)$	$\sigma^2(pI) + \sigma^2(pO) + \sigma^2(pIO)$
	$\sigma^2(\Delta)$	$\sigma^2(I) + \sigma^2(O) + \sigma^2(IO) + \sigma^2(pI) + \sigma^2(pO) + \sigma^2(pIO)$
$i{:}p$	$\sigma^2(\delta)$	$\sigma^2(I, pI)$
	$\sigma^2(\Delta)$	$\sigma^2(I, pI)$
$p \times (i{:}o)$	$\sigma^2(\delta)$	$\sigma^2(pO) + \sigma^2(pI, pIO)$
	$\sigma^2(\Delta)$	$\sigma^2(O) + \sigma^2(I, IO) + \sigma^2(pO) + \sigma^2(pI, pIO)$

`dstudy()` function has several arguments, only a subset of which will be used for this paper:

```
dstudy(x,colname.objects,data,colnam
        e.scores = NULL)
```

The argument `x` is the name of the object created by the `gstudy()` function, `colname.objects` is the name of the data column containing the people or objects of measurement, data is the name of the data set, and `colname.scores` is the name of the column containing the scores, or measurements. A complete G-theory analysis can be completed using the `gstudy()` and `dstudy()` functions; however, these functions do not provide the raw ANOVA quantities, i.e. the sums of squares (SS), degrees of freedom (DF), and mean squares (MS) that are intermediary steps in computing the G study variance component estimates. These fundamental quantities are presented in many illustrative examples in Shavelson and Webb (1991) and Brennan (2001) and can be obtained using the `aov()` function in R. (Note, the `aov()` function is in the base R package that is loaded automatically upon startup.) Thus, for the sake of completeness, we will also demonstrate the use of this function.

While performing G-theory analysis in R has a number of advantages, specifying the formulas for various experimental designs within the `gstudy()` function and extracting results from `dstudy()` are nontrivial tasks, and there are very few examples currently available. Moreover, to our knowledge there are no freely available, modern resources on providing high-quality visualizations for reporting G-theory results. The following will aim to remedy this situation.

Tutorial Using R

This tutorial will demonstrate the process of completing G and D studies in R for four study designs described above: the $p \times i$, $p \times i \times o$, $i{:}p$, and $p \times (i{:}o)$ designs. The data for all examples is read into R by executing the code in the Appendix. Then, the data can be viewed by simply typing the name of the data set into the R console, e.g. `pi_dat`. We note that all three data sets are entirely fictitious, created via trial and error to yield easy-to-follow results (i.e., nonnegative variance estimates and realistic reliability coefficients). Also, for all examples in this paper, the `gtheory` package must be loaded by running the following snippet of code: `library(gtheory)`. Finally, we have prepared the following code using R version 3.5.3. If readers encounter errors when running the code, a general first step for troubleshooting is to check their R version by typing `R.version` into the R console and then updating if an older version is being used.

One-facet $p \times i$ design

A one-facet $p \times i$ crossed design is the simplest study design and is thus suitable for a first example using the `pi_dat` data set. First, an ANOVA analysis is conducted on the data in order to determine the DF, SS, and MS for the data, shown in columns two through four of Table 3, where α denotes a generic effect.

```
summary(aov(Score~Person+Item, data
= pi_dat))
```

Note, in the resulting R output, the row corresponding to pi is labeled as "Residuals", due to the fact that this interaction is confounded with the error, as mentioned in the previous section. Formulas for computing the estimates of G study variance components from these ANOVA results are provided

by Brennan (2001) and Shavelson and Webb (1991). However, in practice a researcher would most likely wish to perform computations efficiently; thus, we demonstrate the use of the `gstudy()` function to obtain the estimates. The model must be specified for the `formula` argument; for the one-facet $p \times i$ crossed design, the formula is written as follows:

```
formula1 <- Score ~ (1|Person)+(1|Item)
```

As for the `aov()` function call, specifying the interaction term explicitly is not required. (Experienced R users may recognize this notation as that of specifying a linear random effects model using the lme4 [Bates & Maechler, 2015] package. The `gtheory` package is essentially a wrapper for `lme4`.) This formula is passed as an argument to the `gstudy()` function to determine the variance component estimates and the percentage of total variation for each term.

```
g1 <- gstudy(data = pi_dat, formula1)
```

The results of the G study, the variance components estimates and percentage of the total variation, are displayed in the fifth and sixth columns of Table 3. This portion of Table 3 may be obtained by typing `g1$components`.

Table 3. G study results for the $p \times i$ design.

Effect (α)	DF (α)	SS (α)	MS (α)	$\hat{\sigma}^2(\alpha)$	Percent
p	5	44.50	8.90	1.91	26.4
i	3	76.33	25.44	4.03	55.9
pi	15	19.17	1.28	1.28	17.7

A D study can be performed by using the `dstudy()` function with the same formula. The object

of measurement is always the person or group being measured, and the score is the numeric value of the measured variable. For this example, the results of the `dstudy()` function are stored in the variable `d1`.

```
d1 <-
dstudy(g1,colname.objects="Person",
colname.scores="Score",
data= pi_dat)
```

The output of the `dstudy()` function are used to create the results shown in Table 4. The `dstudy()` function provides all the values in the column of Table 4 for $n_i = n_i' = 4$, the number of items actually used. The values of the estimated variance components, $\hat{\sigma}^2(p)$, $\hat{\sigma}^2(I)$, and $\hat{\sigma}^2(pI)$, are given in the `d1$components` attribute. Connecting the mathematical notation used by Brennan (2001) to the R attributes, the universe score variance, $\hat{\sigma}^2(\tau)$, is given by `d1$var.universe`, the relative error variance, $\hat{\sigma}^2(\delta)$, is given by `d1$var.error.rel`; the generalizability coefficient, $E\hat{\rho}^2$, is given by `d1$generalizability`; and the dependability coefficient, Φ, is given by `d1$dependability`.

The values in Table 4 for alternative values of n_i' (i.e. columns four through seven) can be determined by using the formulas in Tables 1 and 2 and equations (6) and (7). For example, the D study variance components $\hat{\sigma}^2(I)$ and $\hat{\sigma}^2(pI)$, for $n_i'=2$ are determined by dividing the G study variance components, $\hat{\sigma}^2(i)$ and $\hat{\sigma}^2(pi)$, by 2. The computation of the error variances and coefficients may be automated to some extent using vectorized arithmetic in R. For example, the values for $\hat{\sigma}^2(\delta)$, $\hat{\sigma}^2(\Delta)$, $E\hat{\rho}^2$, and $\hat{\Phi}$ for

Table 4. D study results for the $p \times i$ design

$\hat{\sigma}^2(\alpha)$	n_i'	4	1	2	5	10
$\hat{\sigma}^2(p) = 1.91$	$\hat{\sigma}^2(p)$	1.91	1.91	1.91	1.91	1.91
$\hat{\sigma}^2(i) = 4.03$	$\hat{\sigma}^2(I)$	1.01	4.03	2.01	0.81	0.40
$\hat{\sigma}^2(pi)= 1.28$	$\hat{\sigma}^2(pI)$	0.32	1.28	0.64	0.26	0.13
	$\hat{\sigma}^2(\delta)$	0.32	1.28	0.64	0.26	0.13
	$\hat{\sigma}^2(\Delta)$	1.33	5.31	2.65	1.06	0.53
	$E\hat{\rho}^2$	0.86	0.60	0.75	0.88	0.94
	$\hat{\Phi}$	0.59	0.26	0.42	0.64	0.78

columns corresponding to n_i'=1, 2, 5, and 10 can be obtained using this code:

```
n_i <- c(1,2,5,10)
#relative error variance
rel_err_var <- g1$components[3,2]/n_i
#absolute error variance
abs_err_var <-
g1$components[2,2]/n_i+g1$components[3,
2]/n_i
#calculate generalizability coefficient
gen_coef <-
g1$components[1,2]/(g1$components[1,2]
+ rel_err_var)
#calculate dependability coefficient
dep_coef <-
g1$components[1,2]/(g1$components[1,2]
+ abs_err_var)
round(rel_err_var,2)
round(abs_err_var,2)
round(gen_coef,2)
round(dep_coef,2)
```

Note, the above code uses brackets to extract the appropriate values from the `g1$components` object. Specifically, `g1$components[1,2]` obtains the quantity $\hat{\sigma}^2(p)$ in the 1st row, 2nd column of `g1$components`, `g1$components[2,2]` obtains the quantity $\hat{\sigma}^2(i)$ in the 2nd row, 2nd column, and so on.

It is important to note that as the value of n_i' increases, the generalizability and dependability increase. When designing a study, the researcher must consider their desired reliability as well as the marginal cost for increasing the number of items, used in the study. For this reason, it can be helpful to visualize the diminishing returns of increasing the value of n_i', which is aided by creating plots of the key D study components as shown in the next section.

Two-facet $p \times i \times o$ design

The variance in a two-facet $p \times i \times o$ design can be attributed to the seven variance components as shown in Table 1. We use the `pio_cross_dat` to illustrate. Again, the first step is an ANOVA analysis:

```
summary(aov(Score~Person*Item*
Occasion, data = pio_cross_dat))
```

Note the only difference between this command and the ANOVA command in the previous section is the addition of 'Occasion' to the formula call. The syntax

remains the same because this is another crossed design. The resulting values are displayed in columns two through four of Table 5. Then, for the G study, the formula reflects the two-facet $p \times i \times o$ crossed design:

```
formula2 <-
Score~(1|Person)+(1|Item)+
(1|Occasion)+
(1|Person:Item)+(1|Person:Occasion)+
(1|Occasion:Item)
```

In contrast to the formula for the one-facet $p \times i$ design, the paired interaction terms are included. It is important to note that the interaction term for pio is not included because this is the error term for this design. The model formula is then used in the `gstudy()` function from the `gstudy` package to determine the variance components.

```
g2 <- gstudy(data = pio_cross_dat,
             formula2)
```

The results of the G study are displayed in columns five and six of Table 5 and can be obtained from R by typing `g2$components`. Then, the D study results are obtained similarly to the previous example and are displayed in Table 6.

```
d2 <-
dstudy(g2,colname.objects="Person",
colname.scores="Score",data=
pio_cross_dat)
```

Table 5. G study results for $p \times i \times o$ design.

Effect (α)	DF (α)	SS (α)	MS (α)	$\hat{\sigma}^2(\alpha)$	Percent
p	5	112.94	22.59	2.48	31.7
i	3	117.9	39.3	3.07	39.2
o	1	15.19	15.19	0.58	7.4
pi	15	35.48	2.37	0.77	9.8
po	5	5.94	1.19	0.09	1.1
io	3	2.9	0.97	0.02	0.3
pio	15	12.48	0.83	0.83	10.6

For a $p \times i \times o$ design, the output of the `dstudy()` function includes all of the values in the third column of Table 6, i.e. values corresponding $n_i = n_i' = 4$ and $n_o = n_o' = 2$, the number of items and occasions actually used. The D study output with alternative values of n_i' and n_o' can be determined by using the equations in Table 1. The R code to automate the calculation of

Table 6: D study results for the $p \times i \times o$ design

$\hat{\sigma}^2(\alpha)$	n'_i	4	1	2	3	4	5
	n'_o	2	1	2	3	3	3
$\hat{\sigma}^2(p)$=2.48	$\hat{\sigma}^2(p)$	2.48	2.48	2.48	2.48	2.48	2.48
$\hat{\sigma}^2(i)$=3.07	$\hat{\sigma}^2(I)$	0.77	3.07	1.54	1.02	0.77	0.61
$\hat{\sigma}^2(o)$=0.58	$\hat{\sigma}^2(O)$	0.29	0.58	0.29	0.19	0.19	0.19
$\hat{\sigma}^2(pi)$=0.77	$\hat{\sigma}^2(pI)$	0.19	0.77	0.38	0.26	0.19	0.15
$\hat{\sigma}^2(po)$=0.09	$\hat{\sigma}^2(pO)$	0.04	0.09	0.04	0.03	0.03	0.03
$\hat{\sigma}^2(io)$=0.02	$\hat{\sigma}^2(IO)$	0.00	0.02	0.01	0.00	0.00	0.00
$\hat{\sigma}^2(pio)$=0.83	$\hat{\sigma}^2(pIO)$	0.10	0.83	0.21	0.09	0.07	0.06
	$\hat{\sigma}^2(\delta)$	0.34	1.69	0.64	0.38	0.29	0.24
	$\hat{\sigma}^2(\Delta)$	1.40	5.36	2.47	1.60	1.25	1.05
	$E\hat{\rho}^2$	0.88	0.60	0.80	0.87	0.90	0.91
	$\hat{\Phi}$	0.64	0.32	0.50	0.61	0.66	0.70

$\hat{\sigma}^2(\delta)$, $\hat{\sigma}^2(\Delta)$, $E\hat{\rho}^2$, and $\hat{\Phi}$ changes in two respects. First, we must define vectors for both n'_i and n'_o; second, we must refer to Table 2 for the appropriate expressions for $\hat{\sigma}^2(\delta)$ and $\hat{\sigma}^2(\Delta)$, which depend on the computational formulas in Table 1. For example, the code below computes the values of $\hat{\sigma}^2(\delta)$ and $E\hat{\rho}^2$ for the five rightmost columns of Table 6.

```
n_i <- c(1,2,3,4,5)
n_o <- c(1,2,3,3,3)
#relative error variance
rel_err_var <- g2$components[1,2]/n_i +
g2$components[2,2]/n_o
+g2$components[7,2]/(n_i*n_o)
#calculate generalizability coefficient
gen_coef <-
g2$components[4,2]/(g2$components[4,2]
+ rel_err_var)
round(rel_err_var,2)
round(gen_coef,2)
```

One-facet $i:p$ design

Another important type of design is the nested design, of which the simplest is the one-facet $i:p$ design. The pi_dat data set will be reused for this example. Again, the analysis begins with ANOVA; however, there is a key difference in the syntax compared to the previous uses of the aov() function: above. When heteroskedasticity is present, the variance of the estimated regression coefficients becomes:

```
summary(aov(Score~(Person/Item),
        data = pi_dat))
```

Note, the nested facet follows the forward slash, and this ensures this component is included in the residual. For this simple example, this produces the same results as when the Item facet is excluded from the formula since the pi interaction is part of the residual. The ANOVA results are displayed in columns two through four of Table 7.

The formula for use in the gstudy() package is specified as follows:

```
formula3 <- Score ~ (1|Person)
```

The syntax for more complex nested designs includes the nested facets (see the next section); however, for this simple case, the Item component is omitted. The formula is then used in the gstudy() function to determine the variance components, and the results are displayed in columns five and six of Table 7.

```
g3 <- gstudy(data = pi_dat, formula3)
```

Table 7. G study results for $i:p$ design

Effect (α)	DF (α)	SS (α)	MS (α)	$\hat{\sigma}^2(\alpha)$	Percent
p	5	44.5	8.90	0.90	14.5
i, pi	18	95.5	5.31	5.31	85.5

The D study is performed similarly as in the previous examples, and the results are displayed in Table 8.

Table 8. D study results for the $i{:}p$ design

$\hat{\sigma}^2(\alpha)$	n_i'	4	1	5	10	15	20
$\hat{\sigma}^2(p)$=0.899	$\hat{\sigma}^2(p)$	0.90	0.90	0.90	0.90	0.90	0. 90
$\hat{\sigma}^2(i,pi)$=5.31	$\hat{\sigma}^2(I,pI)$	1.33	5.31	1.06	0.53	0.35	0.27
	$\hat{\sigma}^2(\delta)$	1.33	5.31	1.06	0.53	0.35	0.27
	$\hat{\sigma}^2(\Delta)$	1.33	5.31	1.06	0.53	0.35	0.27
	$E\hat{p}^2$	0.40	0.15	0.46	0.63	0.72	0.77
	$\hat{\Phi}$	0.40	0.15	0.46	0.63	0.72	0.77

Two-facet $p \times (i{:}o)$ design

Performing G-theory analysis on a two-facet $p \times (i{:}o)$ design is an extension of the $i{:}p$ design analysis. The `pio_nest_dat` is used to illustrate, beginning with ANOVA:

```
summary(aov(Score~Person*
(Occasion/Item), data
= pio_nest_dat))
```

The resulting DF, SS, and MS are shown in columns two through four of Table 9. Notice that the nested syntax is once again used in the `aov()` function; Item follows the forward slash because it is nested within Occasion. The formula for the `gstudy()` function is written as follows:

```
formula4 <- Score ~
(1|Person)+(1|Person:Occasion)+
(1|Occasion/Item)
```

Again, we see the same syntax for nesting Item within Occasion. The estimated variance components and percent of total variation are shown in columns five and six of Table 9.

```
g4 <- gstudy(data = pio_nest_dat,
        formula4)
```

Table 9. G study results for $p \times (i{:}o)$ design.

Effect (α)	DF (α)	SS (α)	MS (α)	$\hat{\sigma}^2(\alpha)$	Percent
p	5	56.88	11.38	2.19	18.9
o	1	92.04	92.04	7.20	62.3
i,io	2	7.42	3.71	0.50	4.3
po	5	13.21	2.64	0.97	8.4
pi,pio	10	7.08	0.71	0.71	6.1

The results of the D study are shown in Table 10 and can be obtained in a similar manner as the previous examples.

```
d4 <- dstudy(g4,
colname.objects="Person",
colname.scores="Score", data
= pio_nest_dat)
```

Table 10. D study results for $p \times (i{:}o)$ design

$\hat{\sigma}^2(\alpha)$	n_i'	2	1	2	3	3	4
	n_o'	2	1	3	2	3	5
$\hat{\sigma}^2(p)$=2.19	$\hat{\sigma}^2(p)$	2.19	2.19	2.19	2.19	2.19	2.19
$\hat{\sigma}^2(o)$=7.20	$\hat{\sigma}^2(O)$	3.60	7.20	2.40	3.60	2.40	1.44
$\hat{\sigma}^2(po)$=0.97	$\hat{\sigma}^2(pO)$	0.48	0.97	0.32	0.48	0.32	0.19
$\hat{\sigma}^2(i,io)$=0.50	$\hat{\sigma}^2(I,IO)$	0.13	0.50	0.08	0.08	0.06	0.02
$\hat{\sigma}^2(pi,pio)$=0.71	$\hat{\sigma}^2(pI,pIO)$	0.18	0.71	0.12	0.12	0.08	0.04
	$\hat{\sigma}^2(\delta)$	0.66	1.67	0.44	0.60	0.40	0.23
	$\hat{\sigma}^2(\Delta)$	4.39	9.38	2.92	4.29	2.86	1.69
	$E\hat{p}^2$	0.77	0.57	0.83	0.78	0.85	0.91
	$\hat{\Phi}$	0.33	0.19	0.43	0.34	0.43	0.56

Table 11. Two-facet nested designs and their formulas

Two-facet Nested Design	R Random Effects Formula Syntax
$p \times (i{:}o)$	`Score~(1|Person)+(1|Person:Occasion)+(1|Occasion/Item)`
$(i{:}p) \times o$	`Score~(1|Occasion)+(1|Occasion:Person)+(1|Person/Item)`
$i{:}(p \times o)$	`Score~(1|Person)+(1|Occasion)+(1|Person:Occasion)`
$(i \times o){:}p$	`Score~(1|Person/Item)+(1|Person:Occasion)`
$i{:}o{:}p$	`Score~(1|Person/Occasion)`

While this example shows how to analyze the $p \times (i{:}o)$ design, there are a number of different types of two-facet nested designs. An exhaustive list of these designs and their corresponding formula syntax are shown in Table 11. Further details about these designs can be found on p. 57-61 of Shavelson and Webb (1991).

Further Topics

Now that the basic G and D study analyses have been completed, we focus on two additional topics, visualizing D study results and formatting data for use in the `gtheory` package.

Visualizing D Study Results

Choosing an optimal number of conditions to maximize reliability in the D study may be aided by producing plots such as seen in Table 2.4 of Brennan (2001). The R package `ggplot2` (Wickham, 2016) is

Figure 1. Generalizability and dependability for the $p \times i$ design.

used in the following example in which we return to the $p \times i$ example in a previous section. The code below begins very similarly to the code to produce the results in Table 4. The main difference is that the plot can illustrate many values of n_i'; thus, we define a vector of n_i' ranging from 1 to 25 and compute the corresponding $\hat{\sigma}^2(\delta)$, $\hat{\sigma}^2(\Delta)$, $E\hat{\rho}^2$, and $\hat{\Phi}$:

```
n_prime <- 1:25
#relative error variance
rel_err_var <-
g1$components[3,2]/n_prime
#absolute error variance
abs_err_var <-
g1$components[2,2]/n_prime +
g1$components[3,2]/n_prime
#calculate generalizability coefficient
gen_coef <-
g1$components[1,2]/(g1$components[1,2]
+ rel_err_var)
#calculate dependability coefficient
dep_coef <-
g1$components[1,2]/(g1$components[1,2]
+ abs_err_var)
```

Next, we load to `ggplot2` package and use the `ggplot()` function to construct the plot; as in Brennan (2001), the symbol ρ represents the generalizability coefficient, $E\hat{\rho}^2$.

```
(library(ggplot2)
label1 <- "rho"
label2 <- "phi"
ggplot(data.frame(x=n_prime,y=gen_coef,
label=label1)) +

geom_text(aes(n_prime,gen_coef,label=la
bel1), parse=TRUE) +

geom_text(aes(n_prime,dep_coef,label=la
bel2), parse=TRUE) +
  ggtitle("Plot") + xlab("D Study
Sample Sizes") + ylab("Cofficient")
```
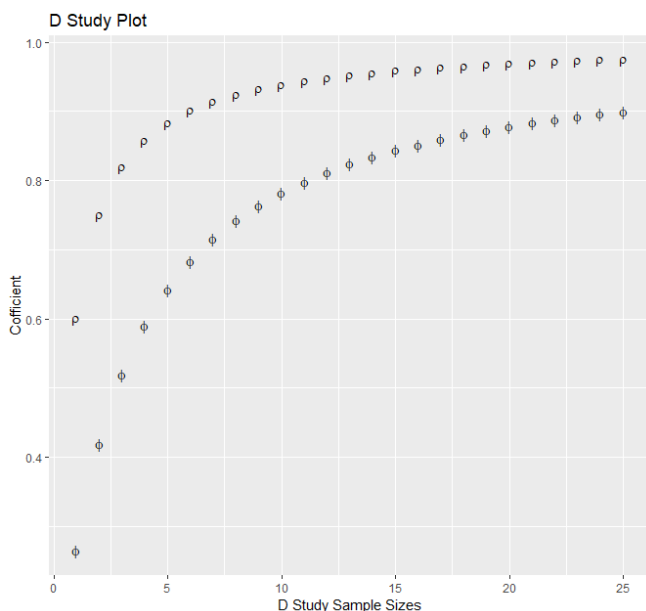
Formatting the Data: Wide Form vs. Long Form

After examining the data sets accompanying this paper and working through the tutorial, readers experienced with G-theory or other measures of reliability may notice that these data sets are not formatted in same the manner as is often seen in many other papers and texts on the subject. The data sets here are formatted in "long" form, which is required by the functions in the `gtheory` package. In contrast, many times G-theory example data sets (as well as data for other reliability methods such as the intra-class correlation coefficient) are arranged in "wide" form. As an example, Table 12 presents the `pi_dat` data set in wide form; contrast this to the structure of the version of the same data provided. Readers with data in wide form can convert it to long form using the example presented below.

Table 12. The `pi_dat` data arranged in wide form.

Person	Item 1	Item 2	Item 3	Item 4
1	9	9	7	4
2	9	8	4	6
3	8	8	6	2
.	.	.	.	.
.	.	.	.	.

Reformatting data from wide to long form and vice versa can be accomplished using functions from the `tidyr` package (Wickham & Henry, 2019). The `spread()` function changes data from long to wide format, while the `gather()` function changes the format from wide to long. Thus, the functions perform inverse operations. We will create a wide format version of the `pio_cross_dat` data using `spread()` and then switch it to long form using `gather()`. The code below creates the wide form version (To run the example, make sure to have the original `pio_cross_dat` read in first):

```
library(tidyr)
pio_cross_dat_wide <-
spread(pio_cross_dat,Item,Score)
```

The reader should visually examine the new data. Now, the `gather()` function can be used to reformat this wide data back to long form for use in the `gtheory` package:

```
gather(pio_cross_dat_wide, key=Item,
value=Score,'1','2','3','4', factor_key
= TRUE)
```

Though the order of the rows may be different, the resulting data set is the same as the original `pio_cross_dat`. The code is the same for one-facet designs; the reader may wish to verify this using the `pi_dat` data set.

Conclusion

This paper has illustrated the process to perform a complete G-theory analysis in R, including initial data formatting, computation of key quantities, preparation of tables, and visualizing results. The `gtheory` package in R provides a rich set of resources for applying G-theory, but its use is nontrivial due to the syntax required to specify different study designs and efficiently extract and prepare numerical results. We have focused mainly on computation; for more details on the interpretation and explanation of results, see the excellent resources cited throughout the paper.

A single paper cannot address the practical implementation of all possible topics in G-theory. For example, all the examples presented were for random facets, i.e. it was assumed that the items and occasions were randomly selected from a very large population. However, sometimes facets are fixed rather than random, e.g. the researcher does not wish to generalize beyond a certain set of items. The processes outlined above must be amended for mixed designs, i.e. those with at least one random and one fixed facet. Another topic of interest covered in depth by Brennan (2001) is computing the standard errors of estimated variance components via resampling methods such as the bootstrap or jackknife. Future papers may address the practical issues in the application of these topics.

References

Bates, D., Machler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67 (1), 1-48.

Brennan, R. L. (2001). Generalizability theory. New York, NY: Springer.

de Vet, H.C., Terwee, C.B., Mokknink, L. B., & Knol, D.L. (2011). Measurement in medicine: A practical guide. New York, NY: Cambridge University Press.

Gugiu, M.R., Gugiu, P.C., & Baldus, R. (2012). Utilizing generalizability theory to investigate the reliability of the grades assigned to undergraduate research papers. *Journal of Multidisciplinary Evaluation*, 8 (19), 26-40.

Heitman, R.J., Kovaleski, J.E., & Pugh, S.F. (2009). Application of generalizability theory in estimating the reliability of ankle-complex laxity measurement. *Journal of Athletic Training*, 44(1), 48-52.

Moore, C. T. (2016). gtheory: Apply Generalizability Theory with R. R package version 0.1.2. Retrieved from https://CRAN.R-project.org/package=gtheory .

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/ .

Shavelson, R. J., & Webb, N.M. (1991). Generalizability theory: A primer. Newbury Park, CA: Sage Publications.

Statsols. (2019). About the company. Retrieved May 14, 2019, from https://www.statsols.com/statsols-about

Suen, H.K., & Ary, D. (1989). Analyzing quantitative behavioral observation data. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Thompson, B. (2003). A brief introduction to generalizability theory. In B. Thompson (Ed.), Score reliability: contemporary thinking on reliability issues (pp. 43-58). Thousand Oaks, CA: Sage Publications.

Vispoel, W. P., Morris, C. A., & Kilinc, M. (2018). Applications of generalizability theory and their relations to classical test theory and structural equation modeling. *Psychological Methods*, 23(1), 1-26.

Wickham, H. (2016). ggplot2: Elegant graphics for data analysis. New York, NY: Springer-Verlag.

Wickham, H. & Henry L. (2019). tidyr: easily tidy data with 'spread()' and 'gather()' functions. R package version 0.8.3. Retrieved from https://CRAN.R-project.org/package=tidyr

Citation:

Huebner, A. & Lucht, M. (2019). Generalizability Theory in R. *Practical Assessment, Research & Evaluation*, 24(5). Available online: http://pareonline.net/getvn.asp?v=24&n=5

Corresponding Author

Alan Huebner
Associate Teaching Professor
University of Notre Dame
email: Alan.Huebner.10 [at] nd.edu

Appendix

Read in the `pi_dat` data:

```
Person <- as.factor(rep(1:6,each = 4))
Item <- as.factor(rep(1:4,times = 6))
Score <- c(9,9,7,4,9,8,4,6,8,8,6,2,
           9,8,6,3,10,9,8,7,6,4,5,1)
pi_dat <- data.frame(Person,Item,Score)
```

Read in the `pio_cross_dat` data:

```
Person <- as.factor(rep(1:6,each = 8))
Occasion <- as.factor(rep(1:2,each = 4,times = 6))
Item <- as.factor(rep(1:4,times = 12))
Score <- c(9,9,7,4,9,8,5,5,9,8,4,6,
           6,5,3,3,8,8,6,2,8,7,3,2,
           9,8,6,3,9,6,6,2,10,9,8,7,
           8,8,9,7,6,4,5,1,3,2,3,2)
pio_cross_dat <- data.frame(Person,Item,Score,Occasion)
```

Read in the `pio_nest_dat` data:

```
Person <- as.factor(rep(1:6,each = 4))
Occasion <- as.factor(rep(1:2,each = 2,times = 6))
Item <- as.factor(rep(1:4,times = 6))
Score <- c(9,9,5,5,9,8,3,3,8,8,3,2,
           9,8,6,2,10,9,9,7,6,4,3,2)
pio_nest_dat <- data.frame(Person,Item,Score,Occasion)
```