Brennan 2.2

Ralph Bloch

November 26, 2023

Contents

Introduction

Celebrating yet another birthday surrounded by my family, I was reminiscing about another birthday 70 years ago that directed me towards a life in the exact sciences. It was the hoped-for birthday present: a 12" scientific slide rule. It allowed me to do complex calculations efficiently and transparently.

At present I am similarly awed by the slide rule's 21. century successor - **R**, a tool, no modern scientist should be without. In my dotage, I spend much time popularizing Generalizability Theory, particularly a corresponding software tool G-String.

Generalizability Analysis (GA) is a statistical method for analyzing the validity of psychometric assessment tools, such as structured performance examinations. GA is an extension of "Analysis of Variance" (ANOVA), and applying GA to any meaningful examination or test examining a non-trivial group of subjects, could involve hundreds, if not thousands of arithmetic operations and thus would be highly error-prone for manual processing. Almost universally, assessment practitioners today use specialized software tools for GA, such as G-String. These programs are highly efficient but completely non-transparent!

In this R Notebook I would like to demonstrate how to perform the ANOVA calculations required for GA transparently and efficiently using **R**. Here is a simple (p x i) design from "Generalizability Theory" by R. L. Brennan: Tables 2.2 and 2.3 on page 28.

TABLE 2.2. Synthetic Data Set No. 1 and the $p \times i$ Design

	Item Scores (X_{pi})												
Person	1	2	3	4	5	6	7	8	9	10	11	12	$\overline{X}_p$
1	1	0	1	0	0	0	0	0	0	0	0	0	.1667
2	1	1	1	0	0	1	0	0	0	0	0	0	.3333
3	1	1	1	1	1	0	0	0	0	0	0	0	.4167
4	1	1	0	1	1	0	0	1	0	0	0	0	.4167
5	1	1	1	1	1	0	1	0	0	0	0	0	.5000
6	1	1	1	0	1	1	1	0	0	0	0	0	.5000
7	1	1	1	1	1	1	1	0	0	0	0	0	.5833
8	1	1	1	1	0	1	1	1	1	1	0	0	.7500
9	1	1	1	1	1	1	1	1	1	1	1	0	.9167
10	1	1	1	1	1	1	1	1	1	1	1	1	1.0000
$\overline{X}_i$	1.0	.9	.9	.7	.7	.6	.6	.4	.3	.3	.2	.1	

$\overline{X} = .5583$ $\quad \sum_p \overline{X}_p^2 = 3.7292$ $\quad \sum_i \overline{X}_i^2 = 4.71$ $\quad \sum_p \sum_i X_{pi}^2 = 67$

TABLE 2.3. G Study $p \times i$ Design For Synthetic Data Set No. 1

Effect(α)	$df(\alpha)$	$SS(\alpha)$	$MS(\alpha)$	$\hat{\sigma}^2(\alpha)$
p	9	7.3417	.8157	$\hat{\sigma}^2(p) = .0574$
i	11	9.6917	.8811	$\hat{\sigma}^2(i) = .0754$
pi	99	12.5583	.1269	$\hat{\sigma}^2(pi) = .1269$

Maybe, I should quickly explain the nomenclature used in the **R** calculations, since 'Rmd', the code used for the calculations, uses a slightly different set of characters. But it is actually quite simple to understand the symbolic equivalences from these examples:

dfi -> df_i

Xpbar -> $\overline{X}^2$

SumXibar -> $\sum_i \overline{X}_i^2$

Erho2 -> $E\rho^2$, etc.

Set Table 2.2

```
scores <- read.csv("scores.csv", header=FALSE)
# scores <- read.csv("scoresMCQgh.csv", header=FALSE)
```

Calculate row, col, and gross mean

```
Xibar <- colMeans(scores)
Xpbar <- rowMeans(scores)
Xbar <- mean(Xpbar)
```

Calculate sample sizes (n) and degrees of freedom (df)

```
np <- nrow(scores)
ni <- ncol(scores)
dfp <- np -1
dfi <- ni -1
dfpi <- dfp * dfi
```

Calculate Sum Squares

```
SumXpbar2 <- sum(Xpbar^2)
SumXibar2 <- sum(Xibar^2)
SumXpibar2 <- sum(scores^2)
```

ANOVA

```
SS <- np * ni * Xbar ^ 2
SSp <- ni * SumXpbar2 - SS
SSi <- np * SumXibar2 - SS
SSpi <- SumXpibar2 - ni * SumXpbar2 - np * SumXibar2 + SS
MSp <- SSp / dfp
MSi <- SSi / dfi
MSpi <- SSpi / dfpi
sigma2p <- (MSp - MSpi)/ ni
sigma2i <- (MSi - MSpi)/ np
sigma2pi <- MSpi
```

Generalizability Coefficients

```
Erho2 <- sigma2p/(sigma2p + sigma2pi/ni)
Phi <- sigma2p/(sigma2p + (sigma2pi + sigma2i)/ni)
```

Results for Brennan's Table 2.2 data

R		Brennan	
Data			
scores	10 obs. of 12 variables		
Values			
dfi	11	df(I)	11
dfp	9	df(p)	9
dfpi	99	df(pI)	99
Erho2	0.844494892167991	$E\rho^2$	.8445
MSi	0.881060606060606	MS(p)	.8157
MSp	0.815740740740741	MS(I)	.8811
MSpi	0.126851851851852	MS(pI)	.1269
ni	12L	n_I	12
np	10L	n_p	10
Phi	0.773023519410598	Φ	.7730
sigma2i	0.0754208754208755	σ^2(I)	.0754
sigma2p	0.0574074074074074	σ^2(p)	.0574
sigma2pi	0.126851851851852	σ^2(pI)	.1269
SS	37.4083333333333		
SSi	9.69166666666667	SS(I)	9.6917
SSp	7.34166666666667	SS(p)	7.3417
SSpi	12.5583333333333	SS(pI)	12.5583
SumXibar2	4.71	$\Sigma\overline{X}_i^2$	4.71
SumXpbar2	3.72916666666667	$\Sigma\overline{X}_p^2$	3.7292
SumXpibar2	67	$\Sigma\Sigma\overline{X}_{pi}^2$	67
Xbar	0.558333333333333		
Xibar	Named num [1:12] 1 0.9 0.9 0.7 0.7 0.6 0.6 0.4 0.3 0.3 ...		
Xpbar	num [1:10] 0.167 0.333 0.417 0.417 0.5 ...		

In other words, the few R instructions give us the correct answers.

Results for Professor Hansen's Tort MCQ

But there is more to it. We simply replace Brennan's scores with those from the G-String website MCQ scores by commenting out 'scores <- read.csv("scores.csv", header=FALSE)', and removing the hash mark from 'scores <- read.csv("scoresMCQgh.csv", header=FALSE)'. It takes less than a second for the new results:

Data	
scores	58 obs. of 30 variables
Values	
dfi	29
dfp	57
dfpi	1653
Erho2	0.97235669092899
MSi	3.71652794292509
MSp	3.03948376688848
MSpi	0.0840213891844157
ni	30L
np	58L
Phi	0.952725512084247
sigma2i	0.0626294233403565
sigma2p	0.0985154125901356
sigma2pi	0.0840213891844157
SS	612.08275862069
SSi	107.779310344828
SSp	173.250574712644
SSpi	138.887356321839
SumXibar2	12.4114149821641
SumXpbar2	26.1777777777778
SumXpibar2	1032
Xbar	0.593103448275862
Xibar	Named num [1:30] 0.862 0.466 0.638 0.931 0.914 ...
Xpbar	num [1:58] 1 0.9 0.733 0.933 1 ...

Figure 1: R Analysis of website MCQ data

When we compare this with the results from G-String:

ANOVA TABLE FOR RUN 1
Multiple Choice Exam in Tort Law

Effect	df	T	SS	MS	VC
s	57	173.25059	173.25057	3.03948	0.09852
q	29	107.77933	107.77931	3.71653	0.06263
sq	1653	419.91725	138.88734	0.08402	0.08402
Mean		0.00002			
Total	1739		419.91723		

we see that sum scores and variance components are spot on! but let's now look at the Generalizability Coefficients:

$E\rho^2 = 0.97$
$\Phi = 0.95$

They too match the results from R. However, remember that this Multiple Choice Exam represents the simple (p x i) design, where the ANOVA was straight forward. When analyzing more complex designs, the calculations become much more complicated. G-String employs Brennan's urGENOVA subroutine that uses elaborate iterative approaches, we can not easily replicate in simple R algorithms.