

AURÉLIEN ALLARD

"Recovering data from summary statistics: Sample Parameter Reconstruction via Iterative TEchniques (SPRITE)" has comments on PubPeer



Analytic-GRIMMER: a new way of testing the possibility of standard deviations

2018-05-20 · 11 min read · 2 Comments



You may already be familiar with the GRIM test, invented by Nick Brown and James Heathers (if not, you can see their original paper [here](#)). It's a very simple technique, which can be used to check the possibility of means reported in papers using discrete and integer measures (like likert scales). Brown and Heathers used this tool to check the possibility of means reported in the psychological literature. Out of the 71 papers that used likert scales and could be tested using GRIM, about half (36) reported at least one mean that was incompatible with the reported sample size.

GRIM is a great technique, but it can only be applied to test the possibility of means, not standard deviations. A few months after Brown and Heathers posted their discovery as a pre-print, Jordan Anaya discovered a new technique, which he appropriately called **GRIMMER**, to test whether standard deviations were compatible with the reported mean and sample sizes. While the technique discovered by Anaya is ingenious and has been put to profit in discovering many impossible results (most notably during the Brian Wansink investigation), the technique is complicated and rely on extensive simulations. Here, I show how basic math can be used to create a new technique, and to provide a simpler way to test the compatibility of means, standard deviations, and sample size. I call this new technique Analytic-GRIMMER, or A-GRIMMER for short.

Full disclaimer: this new technique is not a revolution compared to the current GRIMMER technique invented by Anaya. Indeed, I suspect that it would give exactly the same answer for all practical purpose! This post is rather another example of the use of simple maths derivations, in the spirit of [this blog post](#) by Alexander Etz. In addition, I offer R code to implement A-GRIMMER, while the original GRIMMER is only available [online](#) or on Python.¹

GRIM and GRIMMER: the basics

AURÉLIEN ALLARD

participants. The participants answered the following question: “How happy are you these days?”, and could answer on a 1 to 7 scale, from 1: “very unhappy”, to 7: “very happy”. After having analyzed the data, the researcher claimed that he found a mean of 3.45. Is this result compatible with the fact that all participants gave answers that were whole numbers? As Brown and Heathers noticed, all means have to conform to the following (simple) equations:

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

where μ is the sample mean, n is the number of participants, and x_i refers to the participant’s answer on the likert scale. If we modify the equation slightly, we can see that

$$n \times \mu = \sum_{i=1}^n x_i$$

What’s interesting here is that the equation on the right hand-side is the sum of whole numbers; in other words, *the sample size (n) times the mean must be equal to a whole number*. In our toy example, 3.45×10 must be equal to a whole number; in this case, obviously, 34.5 is not a whole number, so this mean is incompatible with the reported sample size and the use of a likert scale².

This simple mathematical derivation is powerful: it is easy to understand, simple to implement in any software, and it is simple to check that the logic makes sense. After GRIM was posted as a pre-print, Jordan Anaya proposed a new technique, GRIMMER, to test the consistency between means, standard deviations and variances, and sample sizes. However, contrary to Brown and Heathers, Anaya didn’t use a mathematical derivation, but simulated gazillions of samples to test the empirical properties of standard deviations. In the process, he found that variances and SD exhibit some strong regularities, and that any reported SD violating these regularities is, most probably, a mistake, or a misreported number. Given the extensive simulations that Anaya has done, it seems very likely that, for all practical purposes, the empirical regularities he found will give identical results with those derived from a mathematical argument. However, in this case, there exists a simple mathematical derivation, that I will present below.

An easier way to check the consistency of standard deviations: Analytic-GRIMMER.

To show how we can check the compatibility of mean, SD, and sample size, we’ll use a simple trick which is only marginally more difficult than the one Brown and Heathers used for GRIM. As

AURÉLIEN ALLARD

$$n - 1$$

where μ is the mean, n the sample size, and σ^2 is the sample variance. As you may know, and as can be easily derived³, an alternative formula for the sample variance is

$$\sigma^2 = \frac{\sum_{i=1}^n x_i^2 - n \times \mu^2}{n - 1}$$

And we've done most of the work! Indeed, you can probably already see where we are going: we already know σ^2 , μ and n , because they are reported in the paper; and $\sum_{i=1}^n x_i^2$ is a sum of whole numbers! (since the square of a whole number is, obviously, a whole number)

Rearranging the whole thing leads to:

$$\sum_{i=1}^n x_i^2 = (n - 1) \times \sigma^2 + n \times \mu^2$$

And we're done! In other words, the degree of freedom ($n - 1$) times the sample variance, to which we add the sample size (n) times the mean, must be a whole number.

That's cool, isn't it? But there's more: this whole thing (the sum of the squared numbers) must not only be a whole number, it must also be a whole number whose parity (whether it's even or odd) must be equal to the parity of the sum of the whole numbers. That's because, if a sum of numbers is even (or odd), then the sum of the same numbers, but squared, will also be even or odd⁴. But we know whether the sum of the non-squared numbers is odd or even, because that's what the GRIM test is all about! Our reconstruction of the sum of the numbers (via the GRIM test) must agree with our reconstruction of our sum of the squared numbers (via the A-GRIMMER test).

Of course, in most cases, things are a bit more complicated, since the standard deviation may have been rounded down (or up). Because of that, you have to compute lower and upper bounds of a standard deviation; for instance, a reported SD of 3.89 could have been anything between 3.885 and 3.895. If you use both of these numbers in the formula above (and thus square each SD to get the variance, and then multiply the variance by the degree of freedom), there might be several integers that are squeezed between the lower and the upper bounds. To test whether any of these integers can really be the sum of squared items, you would have to test each of them, dividing them by the degree of freedom, to test whether these possible sum of squared numbers are indeed compatible with the reported standard deviation. It's more complicated, but still gives a valid technique to check the possibility of a reported SD.

AURÉLIEN ALLARD

same for GRIMMER. So I think this is really a newly invented technique.

Sanity check, R code, and a short example

As a sanity check, I did an R simulation to generate 100000 samples of integers, with varying sample sizes (from 5 to 99). From these samples, I stored the means and SD, always rounded with two decimal digits (I tried to round them up, down, and to round them via the R function `round()`). I then used the code below to test what A-GRIMMER would say about them.

Fortunately, in all cases, A-GRIMMER recognized that these were indeed legitimate means and SD.

To test whether A-GRIMMER would spot impossible means and SD, I used it on the pizzagate paper, to check whether it would give the same answer as the original GRIMMER. In all cases, the answer was the same.

Thirdly, I generated all combinations of means, SD, and sample sizes, with means ranging from 1 to 7, SD from 0 to 4, and sample size from 5 to 50. This generates about 11000000 possible combinations. In this case, the R-code below spotted that 88% of combinations were impossible (76% via the GRIM test, 12% via the A-GRIMMER test. This means that, in this sample, about half of the means and SD combinations that were considered possible by the GRIM test were considered impossible by the A-GRIMMER test. This is a notable increase in efficiency).

Finally, a short example: imagine that you find in a published article that participants gave answers to the following likert-type scale: "I feel guilty about how much I ate". 18 participants gave an answer, and you see that the reported mean is 3.44, and the reported standard deviation is 2.47⁵. Are the reported mean and standard deviation compatible with the sample size? To test this, you would simply have to use the function below, and to enter: `aGrimmer(n = 18, mean = 3.44, SD = 2.47)`. `aGrimmer` would then give you the following message: "GRIMMER inconsistent". And this is sweet.

Here is the R code for A-GRIMMER:

```
aGrimmer <- function(n, mean, SD, decimals_mean = 2, decimals_SD = 2){  
  
  if(n>10^decimals_mean){  
    print("The sample size is too big compared to the precision of the reported mean, it is not possible to apply GRIM.")  
  }  
  
  #Applies the GRIM test, and computes the possible mean.
```

AURÉLIEN ALLARD

```

# Creates functions to round a number consistently up or down, when the last digit is 5

round_down <- function(number, decimals=2){
  is_five <- number*10^(decimals+1)-floor(number*10^(decimals))*10
  number_rounded <- ifelse(is_five==5, floor(number*10^decimals)/10^decimals, round(number, digits = decimals))
  return(number_rounded)
}

round_up <- function(number, decimals=2){
  is_five <- number*10^(decimals+1)-floor(number*10^(decimals))*10
  number_rounded <- ifelse(is_five==5, ceiling(number*10^decimals)/10^decimals, round(number, digits = decimals))
  return(number_rounded)
}

# Applies the GRIM test, to see whether the reconstituted mean is the same as the reported mean (with both down and up rounding)

consistency_down <- round_down(number = realmean, decimals = decimals_mean)==mean
consistency_up <- round_up(number = realmean, decimals = decimals_mean)==mean

if(consistency_down+consistency_up==0){
  return("GRIM inconsistent")
}

#Computes the lower and upper bounds for the sd.

Lsigma <- ifelse(SD<5/(10^decimals_SD), 0, SD-5/(10^decimals_SD))
Usigma <- SD+5/(10^decimals_SD)

#Computes the lower and upper bounds for the sum of squares of items.

Lowerbound <- (n-1)*Lsigma^2+n*realmean^2
Upperbound <- (n-1)*Usigma^2+n*realmean^2

#Checks that there is at least an integer between the lower and upperbound

FirstTest<- ifelse(ceiling(Lowerbound)>floor(Upperbound), FALSE, TRUE)

if(FirstTest==FALSE){
  return("GRIMMER inconsistent")
}

#Takes a vector of all the integers between the lowerbound and upperbound

```

AURÉLIEN ALLARD

```

Predicted_Variance <- (Possible_Integers-n*realmean^2)/(n-1)
Predicted_SD <- sqrt(Predicted_Variance)

#Computes whether one Predicted_SD matches the SD (trying to round both down and up)

Rounded_SD_down <- round_down(Predicted_SD, decimals_SD)
Rounded_SD_up <- round_up(Predicted_SD, decimals_SD)

Matches_SD <- Rounded_SD_down==SD | Rounded_SD_up==SD

if(sum(Matches_SD)==0){
  return("GRIMMER inconsistent")
}

#Computes first whether there is any integer between lower and upper bound, and then whether there is
#an integer of the correct oddness between the lower and upper bounds.
oddsness <- realsum%%2
Matches_Oddness <- Possible_Integers%%2==oddsness
Third_Test <- Matches_SD&Matches_Oddness
return(ifelse(
  sum(Third_Test)==0, "GRIMMER inconsistent", "The mean and SD are consistent.")
)
}

```

And here is the R code for the simulations mentioned above:

```

#Simulation that generates 100000 simulations of distributions from likert scales, generates the means and sds (rounded up or down), and tests whether all simulations are considered valid by A-GRIMMER.

# Creates empty vectors for the loop below

nsim <- 1e5
result <- vector(length = nsim)
mean <- vector(length = nsim)
sd <- vector(length = nsim)
n <- vector(length = nsim)

# Creates functions that can round up or down numbers, in case the digits before the last decimal is 5 (for instance, round_down rounds 8.005 to 8.00; round_up rounds 8.005 to 8.01. Both functions round 8.0051 to 8.01)

round_down <- function(number, decimals=2){
  is_five <- number*10^(decimals+1)-floor(number*10^(decimals))*10
  number_rounded <- ifelse(is_five==5, floor(number*10^decimals)/10^decimals, round(number, digits = decimals))
  return(number_rounded)
}

```

AURÉLIEN ALLARD

```
und(number, digits = decimals))
  return(number_rounded)
}

# Loop that generates samples, stores their mean, SD, and stores whether A-GRIMMER
# considered the mean and sd to be possible.

for(i in 1:nsim){
  n[i] <- sample(5:99, 1)
  sampleI <- sample(1:7, size = n[i], replace = TRUE)
  mean[i] <- mean(sampleI)
  sd[i] <- sd(sampleI)
  roundedmean <- round_up(mean[i], 2)
  roundedsd <- round_up(sd[i], 2)
  result[i] <- aGrimmer(n=n[i], mean=roundedmean, SD=roundedsd)
}

sum(result=="The mean and SD are consistent.")
```

```
## [1] 100000
```

Simulations I did to test the efficiency of A-GRIMMER. It takes a lot of time to run all simulations, so I have commented everything.

```
# library(tidyverse)
#
# n <- 5:50
#
# mean <- seq(from = 1, to = 7, by=0.01)
#
# SD <- seq(from = 0, to = 4, by = 0.01)
#
# test <- crossing(n, mean, SD)
#
# results <- vector(mode = "character", length = nrow(test))
#
# for(i in 1:nrow(test)){
#   results[i] <- aGrimmer(n=test$n[i], mean=test$mean[i], SD=test$SD[i])
# }
#
# new_data <- cbind(test, results)
```

1. In practice, you should not limit yourselves to techniques such as GRIM or GRIMMER, but you should also try to generate possible values for the raw data via [SPRITE](#) or [CORVIDS](#). ↩

AURÉLIEN ALLARD

3. The trick is to see that $\sum_{i=1}^n (x_i - \mu)^2 = \sum_{i=1}^n x_i^2 - 2 \times \mu \times \sum_{i=1}^n x_i + n \times \mu^2$.
Since $-2 \times \mu \times \sum_{i=1}^n x_i$ is equal to $-2 \times n \times \mu^2$, we get the formula above. ↩
4. This is true because parity is preserved when we square numbers. If a is an even number, then a can also be written under the form $2 \times b$, where b is another whole number. Then $a^2 = 4 \times b^2$, which is again an even number. If a is odd, then it can be written as $2 \times b + 1$, which, once squared, will make $4b^2 + 4b + 1$, or $4 \times (b^2 + b) + 1$, which is obviously odd. ↩
5. As you may guess, this is taken from the original pizzagate paper. ↩

1 comment on PubPeer (by: Jordan Anaya)

