Supplemental Material

CBE—Life Sciences Education

Robert E. Furrow

Regression to the mean in pre-post testing

Using simulations and permutations to develop null expectations

Furrow, R.E.

Supplemental material, March 2019

S1. Regression to the mean in a simple model of pretest and posttest scores

A simple model of student test scores represents the overall variation in students scores as **among-student** variation and an independent source of **within-student** variation. Thinking of a student score as a random variable S, we can represent their total score as a sum of these two variables, $S = X + E$, where X and E are the among- and within-student contributions, respectively. Over two testing instances, we expect X to be the same while E differs and is independent and identically distributed for each testing instance. Consider pretest and posttest scores for a group of students.

$$S_1 = X + E_1, \qquad S_2 = X + E_2$$

In this case the change in scores (posttest minus pretest) is equal to $C = E_2 - E_1$, because the among-student contribution is identical for both testing instances. With this is mind, we can calculate the correlation between the pretest and posttest scores, and use that to derive a formula for the regression of pretest score on change in score. We define the variance in X (among-student variation) as σ_X^2 and the variance in E (within-student variation) as σ_E^2, so the variance in either testing instance is $\sigma_X^2 + \sigma_E^2$. The correlation, ρ between scores is

$$\rho = \frac{Cov(S_1, S_2)}{\sqrt{Var(S_1) * Var(S_2)}} = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_E^2},$$

and the regression of change in score on pretest score is

$$\frac{Cov(S_1, C)}{Var(S_1)} = \frac{Cov(X + E_1, E_2 - E_1)}{Var(S_1)} = \frac{-\sigma_E^2}{\sigma_X^2 + \sigma_E^2} = \rho - 1.$$

This regression coefficient is one way to quantify regression to the mean, and the coefficient becomes more negative as the pre-post correlation approaches zero (i.e. as the within-student variation becomes increasingly larger than the among-student variation).

S2. Simulating paired test scores on a 10 point scale

To simulate a 10 point test such as the EDAT, we use a binomial distribution with 10 trials. However, as specified above, we want to model both among-student variation and within-student variation across two testing instances. To do that, we can seperate the 10 trials into two smaller binomial variables with n_X and n_E trials, where $n_X + n_E = 10$. By sharing the draw from the former variable across instances, we can control the among-student variation, choosing n_X to simulate a pretest-posttest correlation of $\frac{n_X}{10}$. Below we demonstrate how to generate a simulation in the programming language R, using a mean score and number of students taken from Blumer and Beck's EDAT scores. We note that they did not report a pre-post correlation, so we chose 0.6. This correlation is relatively high, providing a somewhat conservative estimate of the strength of regression to the mean for EDAT data. The hashtags indicate comments that explain the code but do not affect it.

```r
p <- 0.375 # the overall test mean is 10*p = 3.75
n_X <- 6 # with n_X = 6, the expected correlation is 0.6
n_E <- 10 - n_X
numsamp <- 145 # we are considering 145 total students

X <- rbinom(numsamp, n_X, p) # the fixed, among-student variable, sampled for
# 145 students

# creating the pretest and posttest student scores by adding a different,
# random binomial variable (within-student variation) for each instance
S_1 <- X + rbinom(numsamp, n_E, p)
S_2 <- X + rbinom(numsamp, n_E, p)
```

With this approach, we have scores that are binomially distributed for each testing instance, but maintain some correlation between the pretest and posttest score. Simulating a large number of data sets, we can generate a distribution of mean change in scores for each pretest quartile. This provides a null distribution to compare with the results analyzed by Blumer and Beck, although we would need their pre-post correlation to match the simulations more closely to their data. Note that we are not writing maximally efficient code, but instead trying to make the purpose of each command clear. We use some tools from the `dplyr` R package to make the code more readable.

```r
set.seed(50) # by setting a seed, you can replicate our exact results

# in our loop below, we use functions from this package
# to compactly organize the data
library(dplyr)

p <- 0.375
n_X <- 6
n_E <- 10 - n_X
numsamp <- 145

# creating an empty matrix to store our per-quartile mean change
change_sim <- matrix(0,nrow = 10000,ncol = 4)

# simulating 10000 student samples by looping 10000 times
for(i in 1:10000)
{
    X <- rbinom(numsamp,n_X,p) # the among-student variable

    # creating a data.frame with scores
    summary <- data.frame(S_1 = X + rbinom(numsamp, n_E, p),
                          S_2 = X + rbinom(numsamp, n_E, p)) %>%
        mutate(quartile = ntile(S_1,4), # adding a quartile column
               C = S_2 - S_1) %>% # adding a change column
        group_by(quartile) %>% # grouping by quartile...
        summarize(mean_change = mean(C)) # ...to calculate per-quartile means

    # adding these means to our matrix
    change_sim[i,] <- summary$mean_change
}
```

This code has generated per-quartile means for 10,000 simulated samples of 145 students. Visualizing the distribution of these means helps provide intuition and quantitative expectations for the strength of

regression to the mean, assuming this as the null model (see Figure 1 in main manuscript). However, it is worth interpreting cautiously, as there may be a better null model to approximate the original data. For example, rubric-evaluated test scores have different probabilities of success for different criteria, so a stronger model might approximate scores as the sum of several Bernoulli random variables with different probabilities. This would yield a slightly lower overall variance.

S3. Using randomization to generate a null distribution

Instead of using a simulated data set, one can generate a null distribution by randomly permuting which scores are "pre" versus "post", while preserving score pairs. This maintains the same mean and correlation as the original data. Regression to the mean will still occur, but any real effect related to pretest score will be removed, on average. Below we look at two examples.

S3A. A joint binomial distribution with no real effect

Following the simulation approach above, we first generate a single focal sample.

```
set.seed(80) # set this seed to reproduce our exact sample

library(dplyr)

p <- 0.375 # the overall test mean is 10*p = 3.75
n_X <- 6 # with n_X = 6, the expected correlation is 0.6
n_E <- 10 - n_X
numsamp <- 145 # we are considering 145 total students

X <- rbinom(numsamp, n_X, p) # the fixed, among-student variable

# creating the pretest and posttest scores by adding a different,
# random binomial variable (within-student variation) for each instance
sample_A <- data.frame(S_1 = X + rbinom(numsamp, n_E, p),
                       S_2 = X + rbinom(numsamp, n_E, p)) %>%
        mutate(quartile = ntile(S_1,4), # adding a quartile column
               C = S_2 - S_1) # adding a change column
```

These data have a pre-post correlation of 0.57, and the pretest and posttest scores have similar mean and variance. Although there are packages in R with packages for restricted permutation testing, we use the base `sample()` function to demostrate the logic of permutation testing in the next block of code.

```
# creating an empty matrix to store our per-quartile mean change
change_perm_A <- matrix(0,nrow = 10000,ncol = 4)

library(dplyr)

for(i in 1:10000) # performing 10,000 permutations
{
    perm_temp <- sample(1:2,numsamp,replace=TRUE) # randomly generating 1s or 2s
    # to determine which element of a pair gets labeled as "pre"

    sample_A_scores <- sample_A %>%
        select(S_1, S_2) # selecting only the scores from our data.frame
```

```
    # indices of the permuted pre data, in the sample_A_scores data.frame
    inds_pre <- cbind(1:numsamp,perm_temp)

    # indices of the permuted post data, in the sample_A_scores data.frame
    inds_post <- cbind(1:numsamp,3-perm_temp)

    # creating the permuted data as a data.frame
    perm_df <- data.frame(S_1_perm = sample_A_scores[inds_pre],
                          S_2_perm = sample_A_scores[inds_post])

    summary <-  perm_df %>%
        mutate(quartile = ntile(S_1_perm,4), # binning by quartile
               C_perm = S_2_perm - S_1_perm) %>% # calculating change
        group_by(quartile) %>%
        summarize(mean_change = mean(C_perm)) # calculating per-quartile mean change

    # adding these means as a row in our matrix
    change_perm_A[i,] <- summary$mean_change
}
```

In this case, the original mean change in each quartile is near the middle of the randomization distributions (Figure S1, panel A), suggesting that any differences across quartile are due solely to regression to the mean.

S3B. A simulated sample with a real effect mediated by student preparation

Here we simulate a sample with no overall change in mean (as in Blumer and Beck's data), but where where the weakest quartile of students truly did improve by one more point than expected, and the strongest quartile of students truly did improve by one point fewer than expected.

```
set.seed(25) # set this seed to reproduce our exact sample

library(dplyr)

p <- 0.375 # the overall test mean is 10*p = 3.75

# we start with a higher n_X, because our added effect will lower the correlation
# as we generate our real-effect data

n_X <- 8

n_E <- 10 - n_X
numsamp <- 145 # we are considering 145 total students

X <- rbinom(numsamp, n_X, p) # the fixed, among-student variable

# creating the pretest and posttest scores by adding a different,
# random binomial variable (within-student variation) for each instance
sample_B <- data.frame(S_1 = X + rbinom(numsamp, n_E, p),
                       S_2 = X + rbinom(numsamp, n_E, p)) %>%
        mutate(quartile = ntile(S_1,4), # adding a quartile column
               true_quartile = ntile(X,4), # the real skill quartiles
               S_2 = S_2 + (true_quartile == 1) - (true_quartile == 4),
               C = S_2 - S_1) # adding a change column
```

In this sample, the students in the bottom quartile for true ability (low scores for the among-student variable X) gained 1 point more than expected by chance, and the students in the top quartile gained 1 point fewer than expected. This is the kind of effect that Blumer and Beck argue is present in their data. In these data we see a pre-post correlation of 0.59, and similar pretest and posttest mean scores (3.52 and 3.56, respectively). However, as intuitively expected, the variance is substantially lower in the posttest scores (1.10) than the pretest scores (2.38).

Permutations of these data can be generated exactly as above, using `sample_B` instead of `sample_A`.

```
# creating an empty matrix to store our per-quartile mean change
change_perm_B <- matrix(0,nrow = 10000,ncol = 4)

for(i in 1:10000) # performing 10,000 permutations
{
    perm_temp <- sample(1:2,numsamp,replace=TRUE) # randomly generating 1s or 2s
    # to determine which element of a pair gets labeled as "pre"

    sample_B_scores <- sample_B %>%
        select(S_1, S_2) # selecting only the scores from our data.frame

    # indices of the permuted pre data, in the sample_A_scores data.frame
    inds_pre <- cbind(1:numsamp,perm_temp)

    # indices of the permuted post data, in the sample_A_scores data.frame
    inds_post <- cbind(1:numsamp,3-perm_temp)

    # creating the permuted data as a data.frame
    perm_df <- data.frame(S_1_perm = sample_B_scores[inds_pre],
                          S_2_perm = sample_B_scores[inds_post])

    summary <-  perm_df %>%
        mutate(quartile = ntile(S_1_perm,4), # binning by quartile
               C_perm = S_2_perm - S_1_perm) %>% # calculating change
        group_by(quartile) %>%
        summarize(mean_change = mean(C_perm)) # calculating per-quartile mean change

    # adding these means as a row in our matrix
    change_perm_B[i,] <- summary$mean_change
}
```

Figure S1 panel B shows that, using this model with a real effect for the least and most prepared students, the permuted data tends towards per-quartile mean changes that are smaller than those observed in the original simulated data. For the lowest and highest quartile, the mean change from the original data is far outside the middle 95% of permutations. In fact, for the lowest quartile the original mean is the most extreme observation, suggesting that these data are very unlikely to have occurred by chance alone.

Note that the magnitude of this effect (seen as the difference between the original data's means and the means of the randomization distribution) appears to be lower than what we specified in our model. This occurs because because the bins used for the analysis (and for real data) are created by observed pretest scores, which are a noisy estimate of actual skill with the material. But the difference between the observed change and the mean of the distribution does provide an estimate of the average effect for a student who scored into that pretest quartile.

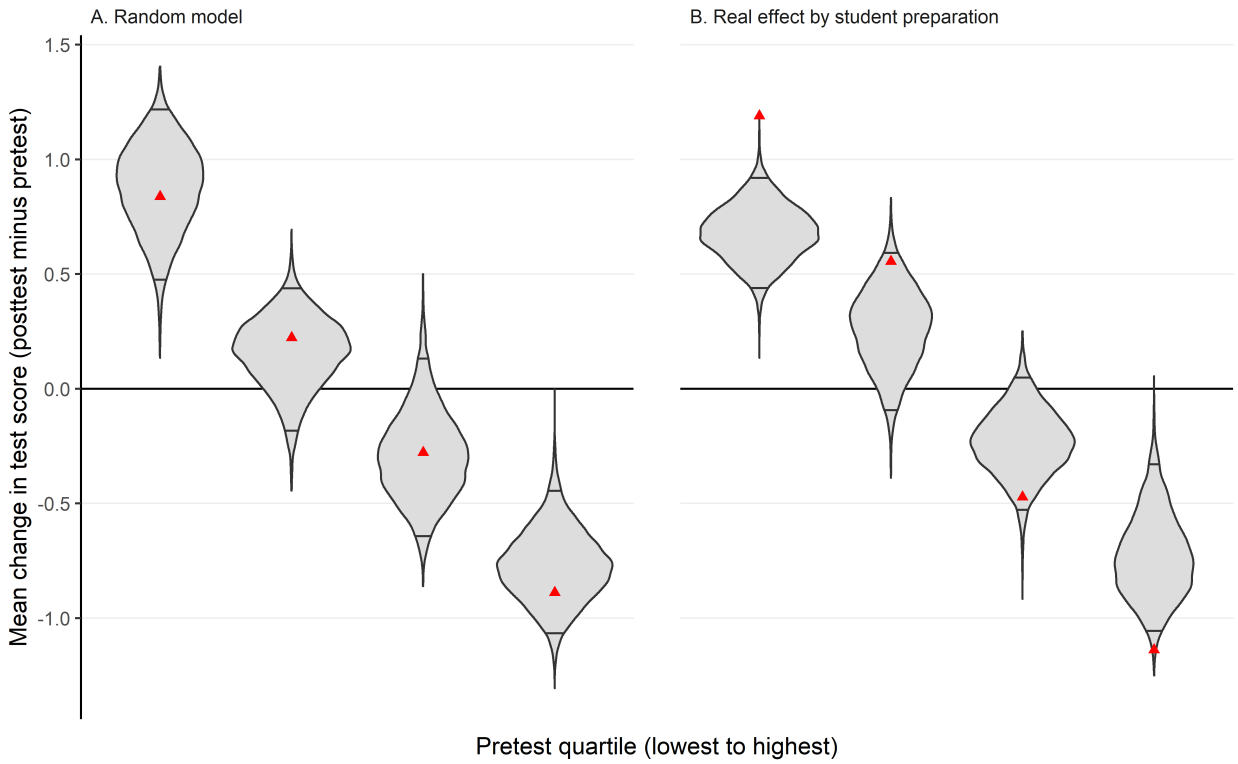

Figure S1. Per-quartile mean change calculated from simulated data, overlaid on a randomization distribution of these means, for two different models of student scores. Panel A corresponds to null model of correlated pretest and posttest scores, with only random chance determining these means. Panel B corresponds to a model in which the weakest students truly did improve more, and the strongest students scored worse on average on the posttest. The red triangles indicate the means calculated from the original simulated data. The distributions were created from 10,000 random permutations of "pre" versus "post" score.