Reproducibility Analysis for 'NIH Peer Review: Criterion Scores Completely Account for Racial Disparities in Overall Impact Scores'

Elena A. Erosheva, Sheridan Grant, Mei-Ching Chen, Mark D. Lindner, Richard K. Nakamura, Carole J. Lee

April 3, 2020

This R Markdown (RMD) document can be used to reproduce the main results in "NIH Peer Review: Criterion Scores Completely Account for Racial Disparities in Overall Impact Scores", by Elena A. Erosheva, Sheridan Grant, Mei-Ching Chen, Mark Lindner, Richard Nakamura, and Carole J. Lee.

Before compilation, download the data ("NIH-public-data_Erosheva-et-al.csv"), move it to a folder of your choice and set your working directory to that folder.

Environment Setup

Required R packages:

```
require(readxl)
require(readr)
require(dplyr)
require(lme4)
require(knitr)
require(kableExtra)
require(Hmisc)
```

Data Handling

In this section, we read in the data, recode variables for clarity in the analyses, and define variable subsets and data subsets for use in the subsequent analyses.

Read data into tibble "long":

```
long <- read_csv("NIH-public-data_Erosheva-et-al.csv")
n <- dim(long)[1]
```

Make sure factor columns are factor-type objects:

```
long_factors <- c('GROUP_ID', 'PI_RACE', 'PI_ID', 'REVIEWER_ID',
                  'APPLICATION_ID', 'IRG', 'ADMIN_ORG', 'SRG',
                  'APPLICATION_TYPE', 'PI_GENDER', 'PI_ETHNICITY',
                  'CAREER_STAGE', 'DEG_CATEGORY', 'INSTITUTION_BIN')
long[long_factors] <- lapply(long[long_factors], as.factor)
```

We reorder the factors to set up baseline categories for convenience. For example, we make “White” the default level of the PI_RACE variable so that the PI_RACE coefficient represents the conditional expected score difference for a PI being black *relative to* being white.

```
# White is considered the default level of the race variable
long$GROUP_ID <- factor(long$GROUP_ID,
                        levels = c('Matched White',
                                   'Matched Black',
                                   'Random White',
                                   'All Black'))

# "Experienced" is the default career stage
long$CAREER_STAGE <- factor(long$CAREER_STAGE,
                            levels = c('Experienced',
                                       'ESI',
                                       'Non-ESI NI'))

# "Male" is considered the default gender in this study
long$PI_GENDER <- factor(long$PI_GENDER,
                             levels = c('Male', 'Female'))

# "Non-Hispanic" is considered the default ethnicity in this study
long$PI_ETHNICITY <- factor(long$PI_ETHNICITY,
                            levels = c('Non-Hispanic', 'Hispanic/Latino'))

# White is considered the default level of the race variable
long$PI_RACE <- factor(long$PI_RACE,
                       levels = c('White', 'Black'))

# PhD is considered the default degree
long$DEG_CATEGORY <- factor(long$DEG_CATEGORY,
                            levels = c('PHD', 'MD' ,'MD/PHD', 'Others'))
```

Since all SEPs (Special Emphasis Panels) are coded as a single SRG (a single level of the “SRG” variable), we recode this factor to be a concatentation of the IRG and SRG, so that SEPs in different IRGs are identified as different by the model.

```
long <- long %>% mutate(SRG = paste(SRG, IRG, sep = '_'))
long$SRG <- as.factor(long$SRG)
```

Variables for Models

Defining variable subsets for easier specification of models:

```
criteria <- c('SIGNIFICANCE_INIT', 'INVESTIGATOR_INIT', 'INNOVATION_INIT',
              'APPROACH_INIT', 'ENVIRONMENT_INIT')

ID_clusters <- c('PI_ID', 'REVIEWER_ID')

org_clusters <- c('ADMIN_ORG', 'IRG', 'SRG')

matching <- c('APPLICATION_TYPE', 'AMENDED', 'PI_GENDER', 'PI_ETHNICITY',
              'CAREER_STAGE', 'DEG_CATEGORY', 'INSTITUTION_BIN', 'IRG')

# Application- and applicant-specific variables for the public data are all matching variables e
xcept for IRG, which is a structural variable
app_app_vars <- matching[1:7]
```

Here, we define the matched and random subsets of the data by filtering the rows of “long” based on “GROUP_ID”. Then, we make the appropriate variables in “d_matched” and “d_random” factor types.

```
# Filter based on GROUP_ID
d_matched <- long[long$GROUP_ID %in% c('Matched White', 'Matched Black'),]
d_random <- long[long$GROUP_ID != 'Matched White',]

# Ensure factor variables have factor data type, remove unnecessary GROUP_ID factor levels after
filtering
d_matched$GROUP_ID <- factor(d_matched$GROUP_ID)
d_random$GROUP_ID <- factor(d_random$GROUP_ID)
for (i in 1:(dim(long)[2])) {
  if (is.factor(long[,i][[1]])) {
    d_matched[,i][[1]] <- factor(d_matched[,i][[1]])
    d_random[,i][[1]] <- factor(d_random[,i][[1]])
  }
}
```

Racial Disparity Models

We now generate the results of the racial disparity analysis as reported in Table S9 of the Reproducibility section of the supplement. These results are comparable to the full-data results in Table 5 of the main paper, discussed in the section titled “Racial Disparity in Preliminary Overall Impact Scores.”

Model specifications:

```
# Model 1: only structural variables and PI Race
structural <- paste('IMPACT_INIT ~ PI_RACE',
                    ' + (1|PI_ID)',
                    ' + (1|REVIEWER_ID)',
                    ' + IRG',
                    ' + ADMIN_ORG',
                    ' + (1|SRG)',
                    sep = ' ')

# Model 2: structural variables, application-/applicant-specific covariates, and PI race
structural_app_app <- paste(structural, ' + ',
                            paste(app_app_vars, collapse = ' + '))

# Model 3: structural variables, criterion scores, and PI race
structural_crit <- paste(structural, ' + ',
                         paste(criteria, collapse = ' + '))

# Model 4: structural variables, application-/applicant-specific covariates, criterion scores, a
nd PI race
structural_app_app_crit <- paste(structural, ' + ',
                                 paste(app_app_vars, collapse = ' + '),
                                 ' + ',
                                 paste(criteria, collapse = ' + '))
```

“Bobyqua” optimization algorithm used throughout:

```
optimizer <- lmerControl(optimizer='bobyqa')
```

We utilize a helper function for computing two-sided p-values based on a normal approximation (justified by the approximate model df of 8000+), which is the following:

```
twoSideP <- function(t) {
  return(2 * pnorm(-abs(t)))
}
```

Then we estimate the matched subset racial disparity models, below:

```
structural_matched <- lmer(as.formula(structural),
                           data = d_matched,
                           REML = T,
                           control = optimizer)

structural_app_app_matched <- lmer(as.formula(structural_app_app),
                                   data = d_matched,
                                   REML = T,
                                   control = optimizer)

structural_crit_matched <- lmer(as.formula(structural_crit),
                                data = d_matched,
                                REML = T,
                                control = optimizer)

structural_app_app_crit_matched <- lmer(as.formula(structural_app_app_crit),
                                        data = d_matched,
                                        REML = T,
                                        control = optimizer)
```

We also estimate the random subset racial disparity models, given below:

```
structural_random <- lmer(as.formula(structural),
                          data = d_random,
                          REML = T,
                          control = optimizer)

structural_app_app_random <- lmer(as.formula(structural_app_app),
                                  data = d_random,
                                  REML = T,
                                  control = optimizer)

structural_crit_random <- lmer(as.formula(structural_crit),
                               data = d_random,
                               REML = T,
                               control = optimizer)

structural_app_app_crit_random <- lmer(as.formula(structural_app_app_crit),
                                       data = d_random,
                                       REML = T,
                                       control = optimizer)
```

Effect sizes are computed by dividing coefficient estimates by residual standard deviations, per one of *multiple valid options for mixed-effects models* as detailed by Hedges (2007). We encourage readers to focus on coefficient estimates themselves (first line of table) rather than effect sizes, since many different effect sizes can be computed for mixed-effects models, each with their own subtle interpretations.

Table S9, racial disparity models: matched subset analysis, public-use data.

Structural	Structural + Applicant/Application-Specific	Structural + Criteria	Structural + Applicant/Application-Specific + Criteria

	Structural	Structural + Applicant/Application-Specific	Structural + Criteria	Structural + Applicant/Application-Specific + Criteria
Race Coefficient	0.466	0.431	0.010	0.014
Standard Error	0.062	0.057	0.017	0.017
P Value	0.000	0.000	0.561	0.412
Effect Size	0.358	0.333	0.018	0.025
Reviewer Intercept Std. Dev.	0.507	0.501	0.286	0.288
PI Intercept Std. Dev.	0.883	0.769	0.100	0.097
SRG Intercept Std. Dev.	0.343	0.304	0.084	0.087
Residual Std. Dev.	1.300	1.296	0.565	0.563

Race coefficient estimates, their effect sizes, and variance components estimates from four hierarchical linear models for preliminary overall impact scores fit on n = 7471 reviews of 2566 applications (matched subset). Model 1 controls for structural covariates; Model 2 controls for structural and matching covariates; Model 3 controls for structural covariates and criterion scores; Model 4 controls for structural, matching covariates, and criterion scores. Coefficient estimates for control variables are not shown.

Table S9, racial disparity models: random subset analysis, public-use data.

	Structural	Structural + Applicant/Application-Specific	Structural + Criteria	Structural + Applicant/Application-Specific + Criteria
Race Coefficient	0.700	0.497	0.031	0.026
Standard Error	0.064	0.060	0.017	0.017
P Value	0.000	0.000	0.071	0.143
Effect Size	0.533	0.382	0.054	0.045
Reviewer Intercept Std. Dev.	0.490	0.509	0.274	0.275

	Structural	Structural + Applicant/Application-Specific	Structural + Criteria	Structural + Applicant/Application-Specific + Criteria
PI Intercept Std. Dev.	0.936	0.803	0.093	0.090
SRG Intercept Std. Dev.	0.306	0.289	0.084	0.085
Residual Std. Dev.	1.312	1.302	0.567	0.565

Race coefficient estimates, their effect sizes, and variance components estimates from four hierarchical linear models for preliminary overall impact scores fit on n = 8595 reviews of 3045 applications (random subset). Model 1 controls for structural covariates; Model 2 controls for structural and matching covariates; Model 3 controls for structural covariates and criterion scores; Model 4 controls for structural, matching covariates, and criterion scores. Coefficient estimates for control variables are not shown.

Commensuration Bias Models

Next, we generate the results of the commensuration analysis as reported in Table S10 and Figures S3 and S4 of the Reproducibility section of the supplement. These results are comparable to the full-data results in Table 6 of the main paper, discussed in the section titled “Commensuration Model for Preliminary Overall Impact Scores” and elaborated on in the supplement’s “Commensuration Practices” section. “NA” in tables refers to random effect standard deviation estimates not having standard errors or p-values.

```
# Commensuration model includes structural variables, application-/applicant-specific covariate
s, criterion scores, PI race, and PI race-criterion score interactions
commensuration <- paste(structural_app_app_crit,
                        ' + (',
                        paste(criteria, collapse = ' + '), ') * PI_RACE')

# Model fitting
commensuration_matched <- lmer(as.formula(commensuration),
                               data = d_matched,
                               REML = T,
                               control = optimizer)

commensuration_random <- lmer(as.formula(commensuration),
                              data = d_random,
                              REML = T,
                              control = optimizer)
```

Table S10, commensuration bias model: matched subset analysis, public-use data.

	Estimate	Std. Error	P-value
SIGNIFICANCE_INIT	0.263	0.008	0.000

	Estimate	Std. Error	P-value
INVESTIGATOR_INIT	0.060	0.011	0.000
INNOVATION_INIT	0.132	0.008	0.000
APPROACH_INIT	0.604	0.007	0.000
ENVIRONMENT_INIT	0.019	0.011	0.090
PI Race = Black	-0.031	0.047	0.508
SIGNIFICANCE_INIT * PI Race = Black	-0.035	0.013	0.008
INVESTIGATOR_INIT * PI Race = Black	0.017	0.017	0.337
INNOVATION_INIT * PI Race = Black	-0.021	0.014	0.125
APPROACH_INIT * PI Race = Black	0.045	0.012	0.000
ENVIRONMENT_INIT * PI Race = Black	-0.009	0.018	0.630
Reviewer Intercept Std. Dev.	0.288	NA	NA
PI Intercept Std. Dev.	0.092	NA	NA
SRG Intercept Std. Dev.	0.088	NA	NA
Residual Std. Dev.	0.562	NA	NA

Preliminary criterion, race, commensuration (race-criterion interaction) coefficients, and variance components estimates for preliminary overall impact scores on n = 7471 reviews of 2566 applications (matched subset). Control variables (coefficient estimates are not shown) are the matching variables.

Table S10, commensuration bias model: random subset analysis, public-use data.

	Estimate	Std. Error	P-value
SIGNIFICANCE_INIT	0.259	0.008	0.000
INVESTIGATOR_INIT	0.099	0.011	0.000
INNOVATION_INIT	0.143	0.008	0.000
APPROACH_INIT	0.618	0.007	0.000
ENVIRONMENT_INIT	0.002	0.012	0.841
PI Race = Black	0.125	0.044	0.004
SIGNIFICANCE_INIT * PI Race = Black	-0.021	0.013	0.097

	Estimate	Std. Error	P-value
INVESTIGATOR_INIT * PI Race = Black	-0.030	0.017	0.073
INNOVATION_INIT * PI Race = Black	-0.039	0.013	0.003
APPROACH_INIT * PI Race = Black	0.031	0.011	0.006
ENVIRONMENT_INIT * PI Race = Black	0.010	0.018	0.584
Reviewer Intercept Std. Dev.	0.274	NA	NA
PI Intercept Std. Dev.	0.089	NA	NA
SRG Intercept Std. Dev.	0.085	NA	NA
Residual Std. Dev.	0.565	NA	NA

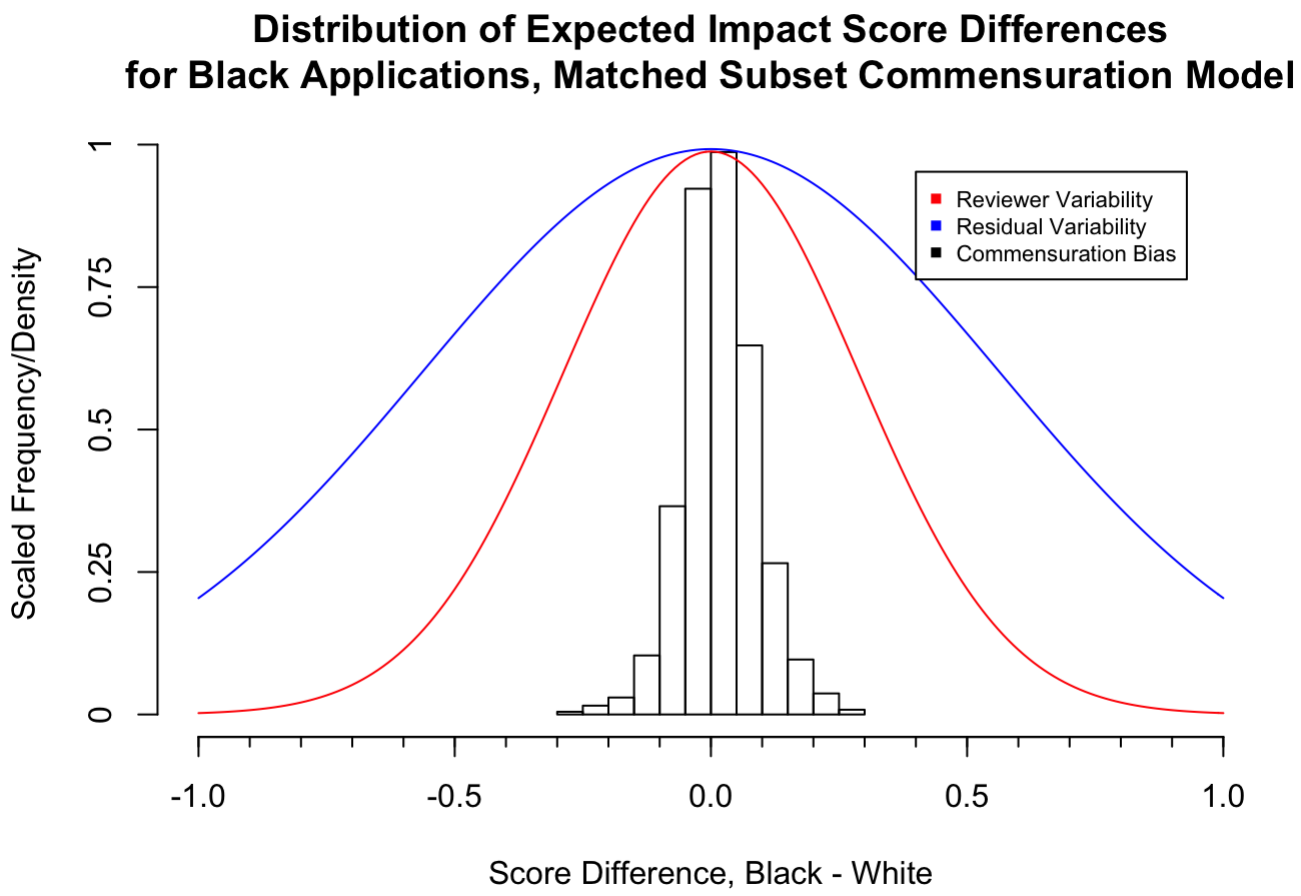
Preliminary criterion, race, commensuration (race-criterion interaction) coefficients, and variance components estimates for preliminary overall impact scores on n = 8595 reviews of 3045 applications (random subset). Control variables (coefficient estimates are not shown) are the matching variables.

Commensuration Practices Analysis

The following code computes the expected differences, under a given model, in preliminary overall impact scores between black and white applicants who are assumed to have identical values on all application- and applicant-specific covariates except race. Figures S3 and S4, reproduced below, display histograms of these expected differences under the matched and random subset analyses for the public-use data. The expected differences are computed as E[Black-White], so that a positive expected difference is a worse expected score for the black applicant.

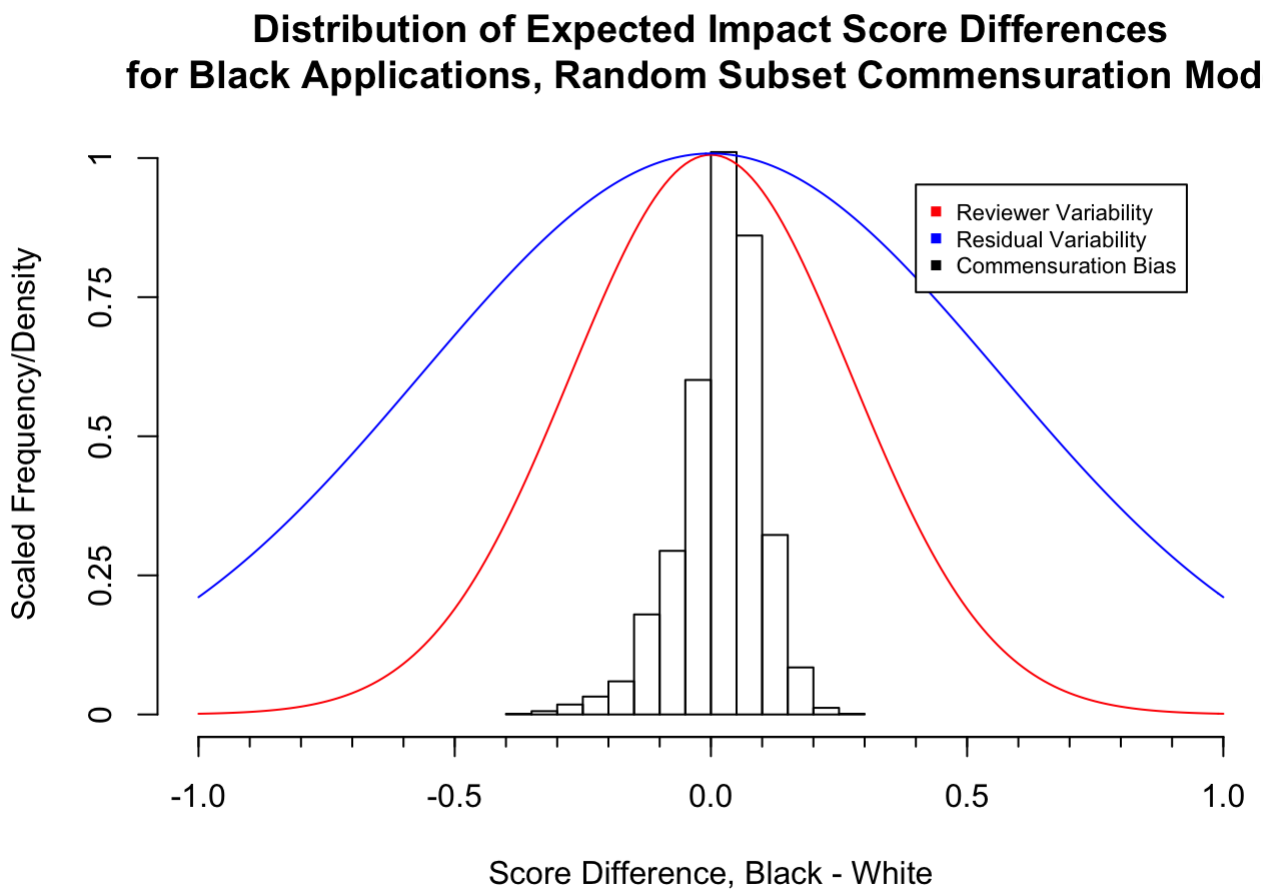
```
# coefficient estimates
criteria_coef_matched <- summary(commensuration_matched)$coefficients[(p_matched-9):(p_matched-5
),1]
interaction_coef_matched <- summary(commensuration_matched)$coefficients[(p_matched-4):p_matche
d,1]
race_coef_matched <- summary(commensuration_matched)$coefficients[2]

criteria_coef_random <- summary(commensuration_random)$coefficients[(p_random-9):(p_random-5),1]
interaction_coef_random <- summary(commensuration_random)$coefficients[(p_random-4):p_random,1]
race_coef_random <- summary(commensuration_random)$coefficients[2]

# Functions that compute the expected difference in preliminary overall impact score, black-whit
e, for given criterion scores for a white and a black applicant
matched_score <- function(c_white, c_black) {
  dif <- c_black%*%(criteria_coef_matched + interaction_coef_matched) -
        c_white%*%criteria_coef_matched + race_coef_matched
}

random_score <- function(c_white, c_black) {
  dif <- c_black%*%(criteria_coef_random + interaction_coef_random) -
        c_white%*%criteria_coef_random + race_coef_random
}

# For each black application, compute the expected score difference between that application and
an identical application from a white applicant
black_scores_matched <- apply(long[long$GROUP_ID %in% c('Matched Black', 'All Black'),criteria],
                              1, function(x) matched_score(x,x))
black_scores_random <- apply(long[long$GROUP_ID %in% c('Matched Black', 'All Black'),criteria],
                             1, function(x) random_score(x,x))
```

Figure S3, matched subset analysis, public-use data.



Distribution of estimated expected preliminary overall impact score differences due to commensuration (histogram) an distributions of reviewer intercepts (red line) and model residuals (blue line), under the matched subset commensuration model.

Figure S4, random subset analysis, public-use data.



Distribution of estimated expected preliminary overall impact score differences due to commensuration (histogram) an distributions of reviewer intercepts (red line) and model residuals (blue line), under the random subset commensuration model.