RAND / UCLA Method (RAM) Simulations Report

Joshua J. Cook, M.S., ACRP-PM, CCRC, Achraf Cohen, Ph.D.

2024-02-11

Table of contents

# Setup  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(boot)  
library(MASS)

Attaching package: 'MASS'  
  
The following object is masked from 'package:dplyr':  
  
 select

library(DescTools)  
library(reshape2)

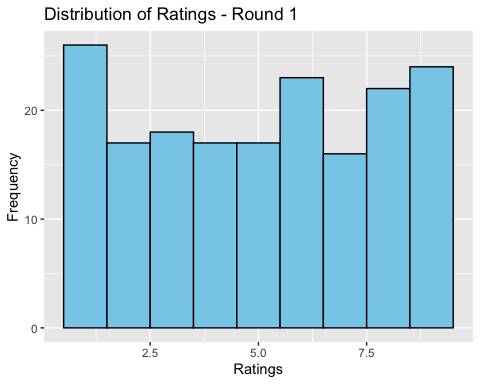
Attaching package: 'reshape2'  
  
The following object is masked from 'package:tidyr':  
  
 smiths

library(ggridges)

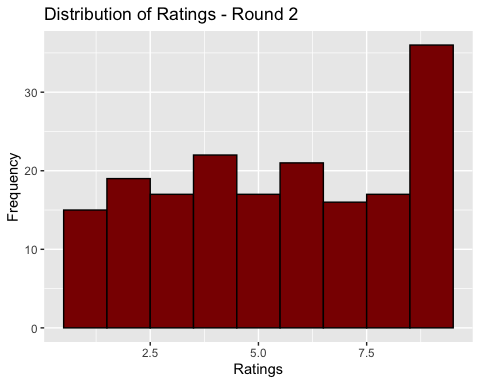
# Sample Size

## n = 9

# Simulation parameters  
num\_experts <- 9   
num\_questions <- 20   
  
# Function to simulate expert ratings 1-9  
simulate\_ratings <- function(num\_experts, num\_questions) {  
 matrix(sample(1:9, num\_experts \* num\_questions, replace = TRUE),   
 nrow = num\_experts, ncol = num\_questions)  
}  
  
# Simulate Round 1 ratings  
ratings\_round1 <- simulate\_ratings(num\_experts, num\_questions)  
  
# Function to adjust ratings after example panel discussion  
adjust\_ratings <- function(ratings) {  
 apply(ratings, 2, function(x) {  
 if (runif(1) > 0.5) { # Random chance to adjust  
 ifelse(seq\_along(x) <= ncol(ratings) / 2, pmin(x + 1, 9), pmax(x - 1, 1))  
 } else {  
 x # No adjustment  
 }  
 })  
}  
  
# Simulate Round 2 ratings after panel meeting adjustments  
ratings\_round2 <- adjust\_ratings(ratings\_round1)  
  
# Function to ensure ratings are within the 1-9 scale  
validate\_ratings <- function(ratings) {  
 pmin(pmax(ratings, 1), 9)  
}  
  
# Ensure ratings are within the 1-9 scale after adjustments  
ratings\_round2 <- validate\_ratings(ratings\_round2)  
  
# Function to calculate appropriateness scores with categorization based on RAND/UCLA handbook  
calculate\_appropriateness <- function(ratings) {  
 # Convert to matrix if not already  
 if (!is.matrix(ratings)) {  
 ratings <- matrix(ratings, nrow = length(ratings), byrow = TRUE)  
 }  
   
 # Calculate mean and categorize  
 scores <- rowMeans(ratings, na.rm = TRUE)  
 categories <- ifelse(scores > 6, "Appropriate", ifelse(scores < 4, "Inappropriate", "Uncertain"))  
   
 return(list(scores = scores, categories = categories))  
}  
  
# Calculate appropriateness scores for both rounds  
results\_round1 <- calculate\_appropriateness(ratings\_round1)  
results\_round2 <- calculate\_appropriateness(ratings\_round2)  
  
# Visualize the distribution of ratings for Round 1  
ggplot(melt(ratings\_round1), aes(value)) +   
 geom\_histogram(binwidth = 1, fill = "skyblue", color = "black") +   
 ggtitle("Distribution of Ratings - Round 1") +   
 xlab("Ratings") + ylab("Frequency")

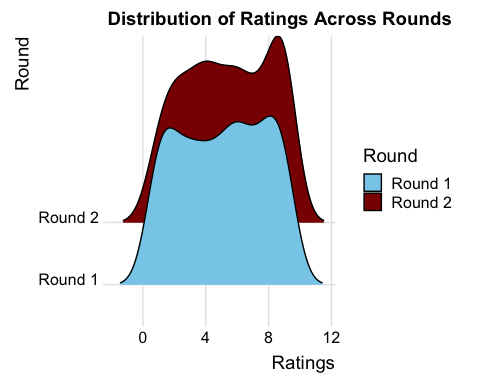


# Visualize the distribution of ratings for Round 2  
ggplot(melt(ratings\_round2), aes(value)) +   
 geom\_histogram(binwidth = 1, fill = "darkred", color = "black") +   
 ggtitle("Distribution of Ratings - Round 2") +   
 xlab("Ratings") + ylab("Frequency")



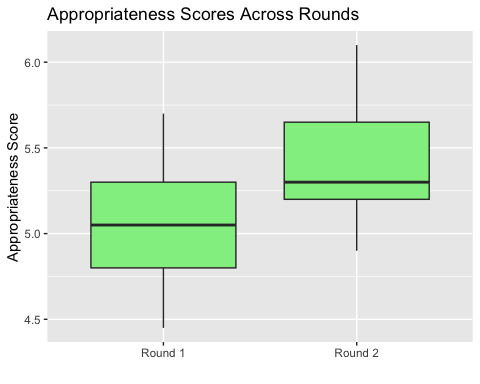
# Melt the ratings matrices and label round identifier  
ratings\_round1\_melted <- melt(ratings\_round1)  
ratings\_round1\_melted$Round <- 'Round 1'  
  
ratings\_round2\_melted <- melt(ratings\_round2)  
ratings\_round2\_melted$Round <- 'Round 2'  
  
# Combine the melted data frames  
ratings\_combined <- rbind(ratings\_round1\_melted, ratings\_round2\_melted)  
  
ggplot(ratings\_combined, aes(x = value, y = Round, fill = Round)) +  
 geom\_density\_ridges\_gradient(scale = 3, rel\_min\_height = 0.01) +  
 scale\_fill\_manual(values = c("Round 1" = "skyblue", "Round 2" = "darkred")) +  
 ggtitle("Distribution of Ratings Across Rounds") +  
 xlab("Ratings") +  
 ylab("Round") +  
 theme\_ridges()

Picking joint bandwidth of 0.86



# Create a vector of scores for Round 1, with NA padding if necessary  
scores\_round1 <- results\_round1$scores  
  
if(length(scores\_round1) < num\_questions) {  
 scores\_round1 <- c(scores\_round1, rep(NA, num\_questions - length(scores\_round1)))  
}  
  
# Create a vector of scores for Round 2, with NA padding if necessary  
scores\_round2 <- results\_round2$scores  
  
if(length(scores\_round2) < num\_questions) {  
 scores\_round2 <- c(scores\_round2, rep(NA, num\_questions - length(scores\_round2)))  
}  
  
# Create the data frame for visualization  
df <- data.frame(  
 Round = rep(c("Round 1", "Round 2"), each = num\_questions),  
 Score = c(scores\_round1, scores\_round2)  
)  
  
ggplot(df, aes(x = Round, y = Score, group = Round)) +  
 geom\_boxplot(fill = "lightgreen") +  
 ggtitle("Appropriateness Scores Across Rounds") +  
 ylab("Appropriateness Score") + xlab("")

Warning: Removed 22 rows containing non-finite values (`stat\_boxplot()`).



print(results\_round1$scores)

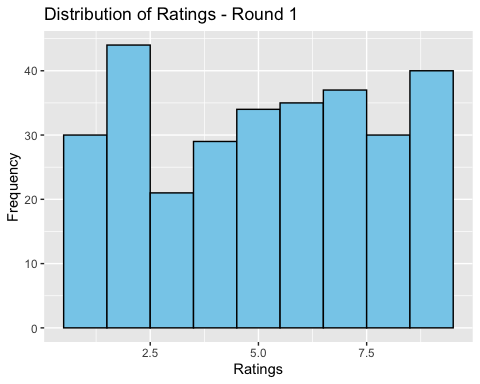
[1] 4.70 5.30 4.45 5.35 5.15 5.70 5.05 4.95 4.80

print(results\_round2$scores)

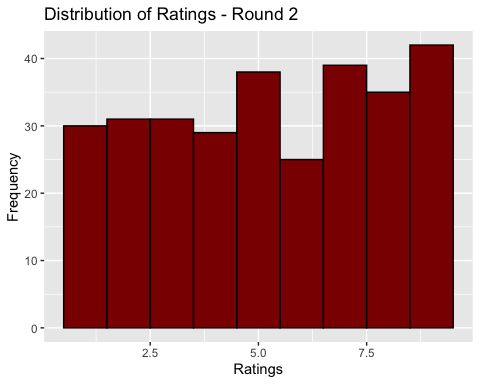
[1] 5.10 5.70 4.90 5.65 5.55 6.10 5.30 5.25 5.20

## n = 15

# Simulation parameters  
num\_experts <- 15 # increased from 9 based on handbook  
num\_questions <- 20   
  
# Function to simulate expert ratings  
simulate\_ratings <- function(num\_experts, num\_questions) {  
 matrix(sample(1:9, num\_experts \* num\_questions, replace = TRUE),   
 nrow = num\_experts, ncol = num\_questions)  
}  
  
# Simulate Round 1 ratings  
ratings\_round1 <- simulate\_ratings(num\_experts, num\_questions)  
  
# Function to adjust ratings based on panel discussion  
adjust\_ratings <- function(ratings) {  
 apply(ratings, 2, function(x) {  
 if (runif(1) > 0.5) { # Random chance to adjust  
 ifelse(seq\_along(x) <= ncol(ratings) / 2, pmin(x + 1, 9), pmax(x - 1, 1))  
 } else {  
 x # No adjustment  
 }  
 })  
}  
  
# Simulate Round 2 ratings after panel meeting adjustments  
ratings\_round2 <- adjust\_ratings(ratings\_round1)  
  
# Function to ensure ratings are within the 1-9 scale  
validate\_ratings <- function(ratings) {  
 pmin(pmax(ratings, 1), 9)  
}  
  
# Ensure ratings are within the 1-9 scale after adjustments  
ratings\_round2 <- validate\_ratings(ratings\_round2)  
  
# Function to calculate appropriateness scores with categorization  
calculate\_appropriateness <- function(ratings) {  
 # Convert to matrix if not already  
 if (!is.matrix(ratings)) {  
 ratings <- matrix(ratings, nrow = length(ratings), byrow = TRUE)  
 }  
   
 # Calculate mean and categorize  
 scores <- rowMeans(ratings, na.rm = TRUE)  
 categories <- ifelse(scores > 6, "Appropriate", ifelse(scores < 4, "Inappropriate", "Uncertain"))  
   
 return(list(scores = scores, categories = categories))  
}  
  
# Calculate appropriateness scores for both rounds  
results\_round1 <- calculate\_appropriateness(ratings\_round1)  
results\_round2 <- calculate\_appropriateness(ratings\_round2)  
  
# Visualize the distribution of ratings for Round 1  
ggplot(melt(ratings\_round1), aes(value)) +   
 geom\_histogram(binwidth = 1, fill = "skyblue", color = "black") +   
 ggtitle("Distribution of Ratings - Round 1") +   
 xlab("Ratings") + ylab("Frequency")

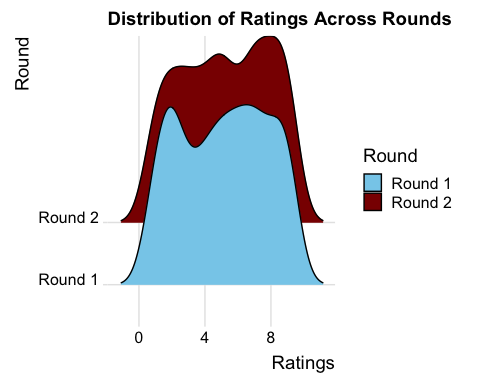


# Visualize the distribution of ratings for Round 2  
ggplot(melt(ratings\_round2), aes(value)) +   
 geom\_histogram(binwidth = 1, fill = "darkred", color = "black") +   
 ggtitle("Distribution of Ratings - Round 2") +   
 xlab("Ratings") + ylab("Frequency")



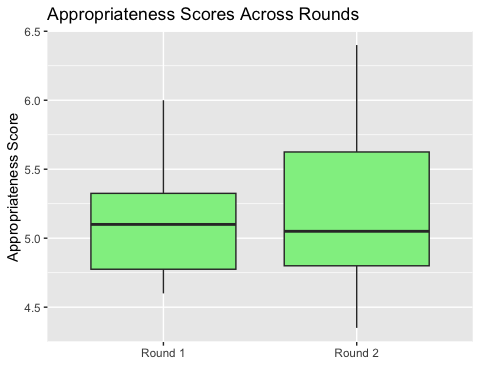
# Melt the ratings matrices and add a round identifier  
ratings\_round1\_melted <- melt(ratings\_round1)  
ratings\_round1\_melted$Round <- 'Round 1'  
  
ratings\_round2\_melted <- melt(ratings\_round2)  
ratings\_round2\_melted$Round <- 'Round 2'  
  
# Combine the melted data frames  
ratings\_combined <- rbind(ratings\_round1\_melted, ratings\_round2\_melted)  
  
ggplot(ratings\_combined, aes(x = value, y = Round, fill = Round)) +  
 geom\_density\_ridges\_gradient(scale = 3, rel\_min\_height = 0.01) +  
 scale\_fill\_manual(values = c("Round 1" = "skyblue", "Round 2" = "darkred")) +  
 ggtitle("Distribution of Ratings Across Rounds") +  
 xlab("Ratings") +  
 ylab("Round") +  
 theme\_ridges()

Picking joint bandwidth of 0.757



# Create a vector of scores for Round 1, with NA padding if necessary  
scores\_round1 <- results\_round1$scores  
  
if(length(scores\_round1) < num\_questions) {  
 scores\_round1 <- c(scores\_round1, rep(NA, num\_questions - length(scores\_round1)))  
}  
  
# Create a vector of scores for Round 2, with NA padding if necessary  
scores\_round2 <- results\_round2$scores  
  
if(length(scores\_round2) < num\_questions) {  
 scores\_round2 <- c(scores\_round2, rep(NA, num\_questions - length(scores\_round2)))  
}  
  
df <- data.frame(  
 Round = rep(c("Round 1", "Round 2"), each = num\_questions),  
 Score = c(scores\_round1, scores\_round2)  
)  
  
ggplot(df, aes(x = Round, y = Score, group = Round)) +  
 geom\_boxplot(fill = "lightgreen") +  
 ggtitle("Appropriateness Scores Across Rounds") +  
 ylab("Appropriateness Score") + xlab("")

Warning: Removed 10 rows containing non-finite values (`stat\_boxplot()`).



print(results\_round1$scores)

[1] 5.25 5.80 4.70 5.10 5.30 4.85 6.00 4.60 5.55 4.60 4.85 5.05 5.10 4.70 5.35

print(results\_round2$scores)

[1] 5.60 6.25 5.05 5.50 5.65 5.25 6.40 5.00 5.95 4.95 4.45 4.60 4.70 4.35 4.90