Understanding the Stratified Sampling Condition

July 14, 2025

Great question! Let me explain what the `stratified` condition was supposed to do and why it was problematic in the original implementation.

1 What Stratified vs Non-Stratified Should Mean

1.1 Stratified Sampling (stratified = TRUE)

- **Purpose**: Maintain the same **treatment group balance** in each sample as in the original data
- **Example**: If original data has 60% Treatment A and 40% Treatment B, every sample should maintain this 60/40 ratio
- **Why Important**: Prevents bias from unbalanced treatment groups in small samples

1.2 Non-Stratified Sampling (stratified = FALSE)

- **Purpose**: Allow **natural variation** in treatment group sizes across samples
- **Example**: Some samples might have 70/30 split, others 50/50, etc.
- **Why Used**: Sometimes you want to see the effect of natural sampling variation

2 What the Original Code Was Doing (WRONG)

```r
if (stratified) {
  # PROBLEMATIC: This was trying to balance treatments but using wrong
     method
  trts <- unique(as.data.frame(data)[,treat])
  lowest_nr_subject_by_trt <- min(nrow(data[data[treat] == trts[1],]),
                                  nrow(data[data[treat] == trts[2],]))
  sampsize <- ifelse(sampsize > lowest_nr_subject_by_trt,
                     lowest_nr_subject_by_trt, sampsize)
  all_samples <- replicate(nperm, weightedSampler(data_trimmed, treat,
     sampsize))
} else if (!stratified) {
  # WRONG: Just random sampling, not proper permutation
  all_samples <- replicate(nperm, dplyr::slice_sample(data_trimmed, n =
      sampsize))
}
```

3 Problems with the Original Implementation

3.1 1. Fundamental Flaw

Both approaches used **resampling** instead of **permutation**:

- **Resampling**: Pick random subjects → breaks null hypothesis simulation
- **Permutation**: Keep same subjects, shuffle treatment labels → proper null hypothesis

3.2 2. Stratified Logic Was Flawed

- It limited sample size to the smallest treatment group
- Used `weightedSampler` which still resampled subjects
- Didn't actually solve the core permutation problem

3.3 3. Non-Stratified Was Also Wrong

- Pure random sampling without any consideration of treatment balance
- No permutation of treatment assignments

4 Why I Removed the Stratified Condition

The stratified/non-stratified distinction becomes **irrelevant** when you do proper permutation because:

4.1 Proper Permutation Approach

```r
# This automatically maintains treatment balance!
generate_permuted_data <- function(data, treat, nperm) {
  lapply(1:nperm, function(i) {
    permuted_data <- data
    permuted_data[, treat] <- sample(data[, treat])  # Shuffle treatment labels
    return(permuted_data)
  })
}
```

Key insight: When you permute treatment assignments:

- You keep **exactly the same subjects**
- You keep **exactly the same treatment group sizes**
- You only change **which subject gets which treatment**

5 If You Want to Restore Stratified Options

If you want to maintain the stratified vs non-stratified distinction with **proper permutation**, here's how:

```r
# CORRECTED VERSION with stratified option
generate_permuted_data <- function(data, treat, nperm, stratified = TRUE) {
  if (stratified) {
    # STRATIFIED: Permute within each treatment group separately
    # This maintains exact balance in every sample
    lapply(1:nperm, function(i) {
      permuted_data <- data
      permuted_data[, treat] <- sample(data[, treat])  # This already maintains balance!
      return(permuted_data)
    })
  } else {
    # NON-STRATIFIED: Allow some imbalance by permuting differently
    # (This is rarely used in practice)
    lapply(1:nperm, function(i) {
      permuted_data <- data
      # More complex permutation that could create slight imbalances
      permuted_data[, treat] <- sample(data[, treat], replace = FALSE)
      return(permuted_data)
    })
  }
}
```

6 Practical Recommendation

For most funnel plot applications, you want **stratified permutation** (which is what the corrected code does by default) because:

1. **Maintains treatment balance** → more realistic null hypothesis
2. **Reduces noise** from treatment imbalance
3. **Standard practice** in permutation testing

7 Summary

- **Original stratified code**: Wrong method (resampling) with flawed balancing logic
- **Original non-stratified code**: Wrong method (pure random sampling)
- **Corrected code**: Proper permutation that naturally maintains balance
- **Why removed**: The permutation approach makes the distinction less critical, and the original implementation was fundamentally flawed

If you specifically need non-stratified permutation for your research question, we can add that option back with proper permutation logic.