# STAT 404 Final Project Plan

2024-11-19

## Design

### Core Functions Pseudo-code

#### Data Generation and Analysis Functions

```
Function: sim_binary_data
Inputs: p1, p2 (probabilities), n1, n2 (sample sizes)
Output: dataframe with columns (group, response)
Process:
1. Validate inputs
   - Check p1, p2 between 0 and 1
   - Check n1, n2 are positive integers
2. Generate responses
   - Create group1 data: n1 bernoulli trials with p1
   - Create group2 data: n2 bernoulli trials with p2
3. Combine and return data
   - Create dataframe with group labels and responses

Function: calc_prop_diff
Inputs: data (dataframe with group, response)
Output: list with difference and standard error
Process:
1. Calculate proportions for each group
2. Calculate sample sizes for each group
3. Calculate difference in proportions
4. Calculate SE using formula: sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2)
5. Return list(diff = difference, se = standard_error)

Function: repeated_sims
Inputs: p1, p2, n1, n2, reps (number of repetitions)
Output: dataframe of simulation results
Process:
1. Initialize storage for results
2. For rep in 1:reps
   - Generate new dataset using sim_binary_data
   - Calculate statistics using calc_prop_diff
   - Store results
3. Return results dataframe
```

### Statistical Test Functions

```
Function: permutation_test
Inputs: data, reps
Output: list with null distribution and observed statistic
Process:
1. Calculate observed test statistic
2. For rep in 1:reps
   - Randomly permute group labels
   - Calculate test statistic
   - Store result
3. Return list(null_dist, obs_stat)

Function: bootstrap_samples
Inputs: data, reps
Output: vector of bootstrap statistics
Process:
1. For rep in 1:reps
   - Sample data with replacement
   - Calculate difference in proportions
   - Store result
2. Return vector of results
```

## Visualization Functions Pseudo-code

```
Function: plot_sampling_dist
Inputs: n_values, p1, p2, reps
Output: grid of plots
Process:
1. For each n in n_values
   - Run repeated simulations
   - Calculate standardized differences
2. Create two plots
   - Raw differences plot
   - Standardized differences vs N(0,1)
3. Arrange plots side by side

Function: plot_confidence_coverage
Inputs: p1, p2, n1, n2, alpha, max_reps
Output: line plot
Process:
1. Create sequence of repetition numbers
2. For each rep number
   - Run simulations
   - Calculate confidence intervals
   - Calculate coverage proportion
3. Plot coverage vs repetitions

Function: plot_permutation_test
Inputs: data, reps
Output: histogram/density plot
```

```
Process:
1. Run permutation test
2. Create plot showing
   - Null distribution
   - Observed statistic
   - Theoretical normal curve

Function: plot_bootstrap_comparison
Inputs: data, reps, conf_level
Output: grid of comparison plots
Process:
1. Calculate theoretical and bootstrap statistics
2. Create comparison plots
   - Standard error comparison
   - Confidence interval comparison
   - Distribution comparison
3. Arrange plots in grid
```

# Tests

Our testing strategy will focus on three main areas. For data generation, we will verify that sim_binary_data produces correctly structured output with valid inputs and appropriately handles invalid inputs. For statistical calculations, we will test calc_prop_diff against known values and edge cases to ensure accuracy. Finally, for visualization functions, we will verify that they produce valid plot objects and handle various input scenarios appropriately.

# Examples

For our exploration of the statistical properties, we plan to investigate a range of scenarios. We will examine four different sample sizes: small (n = 10), medium (n = 30, 50), and large (n = 100). These will be combined with four effect sizes ranging from no effect (p1 = p2 = 0.5) to large effects (p1 = 0.8, p2 = 0.4). For reproducibility and accuracy, we will use 1000 repetitions for our permutation tests and bootstrap analyses, while coverage analysis will explore sequences from 10 to 1000 by increments of 10. Power analysis will be conducted with 100 repetitions per condition to balance computational efficiency with accuracy.

# Discussion

## Plan for Collaboration

Rahif will focus on the foundational data generation and analysis functions (sim_binary_data and calc_prop_diff), while Abhiram will handle the more complex statistical implementations including repeated_sims and permutation_test. Ronnie will develop the bootstrap functionality and visualization components. For testing and documentation, each team member will write tests for their assigned functions, with cross-review procedures in place to ensure quality. Documentation responsibilities have been distributed with Rahif handling Examples.Rmd and project coordination, Abhiram focusing on function documentation, and Ronnie managing testing documentation and integration.

## Expected Challenges

On the technical side, ensuring correct implementation of standard error formulas and creating effective visualizations that clearly communicate results will require careful attention. The conceptual challenges include developing a deep understanding of bootstrap methods versus theoretical approaches and accurately interpreting permutation test results. Additionally, group coordination presents its own challenges, particularly in maintaining consistent coding style across team members and effectively managing version control for collaborative development.

## Feasibility Assessment

Based on our current understanding and team capabilities, we believe the core components of the project are highly feasible, including the basic simulation functions, standard error calculations, and testing framework. More challenging aspects that will require additional effort include advanced visualizations and performance optimization for large sample sizes. Time constraints may impact our ability to implement some advanced features, and complex edge cases will need consideration. Despite these challenges, we are confident in our ability to deliver a well-functioning implementation of the requirements.

## Clarifying Questions

No clarifying questions at this time