STAT 404 Final Project Plan

2024-11-07

Contents

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

Core Function Tests

Data Generation Tests

1. Test sim_binary_data:
    - Valid input produces correct output structure
    - Invalid probabilities throw error
    - Invalid sample sizes throw error
    - Output has correct dimensions and data types

Statistical Tests

1. Test calc_prop_diff:
    - Known input produces expected difference and SE
    - Handles edge cases (all 0s, all 1s)

Visualization Tests

1. Test visualization functions:
    - Return expected plot objects
    - Handle various input sizes
    - Produce valid graphical elements

Examples

We will explore the following parameter combinations:

Sample Sizes

- Small (n = 10)
- Medium (n = 30, 50)
- Large (n = 100)

Effect Sizes

- No effect (p1 = p2 = 0.5)
- Small effect (p1 = 0.52, p2 = 0.5)
- Medium effect (p1 = 0.7, p2 = 0.5)
- Large effect (p1 = 0.8, p2 = 0.4)

Number of Repetitions

- Coverage analysis: 10 to 1000 by 10s
- Permutation tests: 1000 reps
- Bootstrap: 1000 reps
- Power analysis: 100 reps per condition

Discussion

Plan for Collaboration

Core Functions Development

- Person 1: sim_binary_data, calc_prop_diff
- Person 2: repeated_sims, permutation_test
- Person 3: bootstrap_samples
- Person 4: visualization functions

Testing

- Each person writes tests for their functions
- Cross-review of tests

Documentation

- Person 1: Examples.Rmd
- Person 2: Function documentation
- Person 3: Testing documentation
- Person 4: Project coordination and integration

Expected Challenges

Technical Challenges

1. Implementing correct SE formulas
2. Creating effective visualizations
3. Managing computational efficiency for large simulations

Conceptual Challenges

1. Understanding bootstrap vs theoretical approaches
2. Interpreting permutation test results
3. Explaining CLT implications

Group Coordination

1. Maintaining consistent coding style
2. Integrating different components
3. Managing version control

Feasibility Assessment

Highly Feasible

- Core simulation functions
- Basic visualizations
- Testing framework
- Standard error calculations

Moderate Challenge

- Advanced visualizations
- Performance optimization
- Comprehensive documentation

Potential Issues

- Time constraints for advanced features
- Complex edge cases
- Integration testing

Clarifying Questions

1. Should the permutation test use standardized or raw differences?
2. For bootstrap intervals, should we implement percentile method as alternative?
3. Are there specific requirements for visualization style/formatting?
4. What level of optimization is expected for large sample sizes?
5. Should we include additional error handling beyond basic input validation?