

Algorithm for User-Interface

Algorithm 1 Generate Data

Require: `datagen.type`, `n`, `dist`, `two_samples`, `priors`, `x_weights`, `y_weights`

Ensure: A list containing `x` and `y`

```
1: if datagen.type == 1 then                                ▷ Synthetic Data Generation
2:   if dist == "mixture" then
3:     Normalize x_weights and y_weights
4:     for  $i = 1$  to  $n$  do
5:       Generate x[i] based on priors and x_weights
6:       if two_samples == TRUE then
7:         Generate y[i] based on priors and y_weights
8:       end if
9:     end for
10:  else
11:    Generate x from dist
12:    if two_samples == TRUE then
13:      Generate y from dist
14:    end if
15:  end if
16: else
17:   if datagen.type == 2 then                                ▷ Load Data from CSV
18:     Prompt user to select a CSV file
19:     if two_samples == TRUE then
20:       Extract two columns as x and y
21:     else
22:       Extract the first column as x
23:     end if
24:   end if
25: end if
26: return A list containing x and y
```

Algorithm 2 Calculate Power

Require: `alpha`, `N`, `twosamples`, `dist`, `sample_size`, `test`

Ensure: A vector of power values

```
1: Initialize an empty vector powr_t
2: for n in sample_size do
3:   Initialize an empty vector pval_t
4:   for  $i = 1$  to N do
5:     Generate data using Algorithm 1
6:     Compute the p-value using test
7:   end for
8:   Compute power as the proportion of p-values  $< \alpha$ 
9:   Store the power in powr_t
10: end for
11: return powr_t
```

Algorithm 3 Calculate Type I Error

Require: `alpha`, `N`, `twosamples`, `dist`, `sample_size`, `test`

Ensure: A vector of Type I error rates

```
1: Initialize an empty vector error
2: for n in sample_size do
3:   Initialize an empty vector pval_t
4:   for  $i = 1$  to N do
5:     Generate data using Algorithm 1
6:     Compute the p-value using test
7:   end for
8:   Compute Type I error as the proportion of p-values  $< \alpha$ 
9:   Store the error in error
10: end for
11: return error
```

Algorithm 4 Bootstrap Two-Sample Test

Require: `x`, `y`

Ensure: Bootstrap test results

```
1: Generate data using Algorithm 1 with datagen.type = 2 and two_samples = TRUE
2: Perform a bootstrap two-sample test on x and y
3: return Test results
```
