# Package 'SkeweDF'

October 12, 2022

**Title** Optimization of Skewed Distributions with Birth-Death Processes

**Version** 0.1.0

**Description** Implementations of models which follow the Kolmogorov Birth-Death process framework and functions which utilize these Kolmogorov Birth-Death process models for analysis of skewed distribution functions.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** Rcpp (>= 1.0.5), dplyr (>= 1.0.2), parallel (>= 3.6.3), zipfR (>= 0.6-66), optimr (>= 2019-12.4), purrr (>= 0.3.4), matrixStats (>= 0.56.0), stringr (>= 1.4.0), methods (>= 3.6.3), grDevices (>= 3.6.3), graphics (>= 3.6.3), utils (>= 3.6.3)

**RoxygenNote** 7.1.1

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Author** Andre Grageda [aut, cre]

**Maintainer** Andre Grageda <gragedaa@upstate.edu>

**Repository** CRAN

**Date/Publication** 2021-01-16 09:40:05 UTC

## R topics documented:

---

`calculate_label_coords` *Label Coordinate Calculate Helper Function*

---

## Description

This function calculates coordinates for a plot given x and y bounds and location represented as percentage of plot area

## Usage

```
calculate_label_coords(
  x_lower_bound,
  x_upper_bound,
  y_lower_bound,
  y_upper_bound,
  x_buffer = 0.5,
  y_buffer = 0.5,
  log_scale_x = FALSE,
```

```
  log_scale_y = FALSE
)
```

## Arguments

| | |
|---|---|
| x_lower_bound | Numeric lowest value of x axis |
| x_upper_bound | Numeric highest value of x axis |
| y_lower_bound | Numeric lowest value of y axis |
| y_upper_bound | Numeric highest value of y axis |
| x_buffer | Numeric indicating location on x axis (0 - 1) |
| y_buffer | Numeric indicating location on y axis (0 - 1) |
| log_scale_x | Boolean indicating if x axis is log scale |
| log_scale_y | Boolean indicating if y axis is log scale |

---

# Exponential *Exponential Distribution Function*

---

## Description

This function generates a vector of n length of the Exponential distribution with parameters a and b.

## Usage

```
Exponential(n, a, b)
```

## Arguments

| | |
|---|---|
| n | Length of vector to be generated. |
| a | Parameter of the Exponential distribution function |
| b | Parameter of the Exponential distribution function |

## Examples

```
Exponential(100, 10000, 0.8)
```

## Generalized_Pareto *Generalized Pareto Distribution Function*

### Description

Returns vector of length k of Generalized Pareto given a parameters theta, c, b , and rho

### Usage

```
Generalized_Pareto(k, theta, c, b, rho)
```

### Arguments

| | |
|---|---|
| k | Length of vector to be generated |
| theta | Parameter of the Generalized Pareto function |
| c | Parameter of the Generalized Pareto function |
| b | Parameter of the Generalized Pareto function |
| rho | Parameter of the Generalized Pareto function |

## Generalized_Pareto_calc_P0_delta *Generalized Pareto Distribution Function P0 with defined delta*

### Description

Returns P0 of Generalized Pareto given a parameters theta, c, b , and rho

### Usage

```
Generalized_Pareto_calc_P0_delta(theta, c, b, rho, delta)
```

### Arguments

| | |
|---|---|
| theta | Parameter of the Generalized Pareto function |
| c | Parameter of the Generalized Pareto function |
| b | Parameter of the Generalized Pareto function |
| rho | Parameter of the Generalized Pareto function |
| delta | Value of difference between iterations in order to output a result. Decreasing this parameter will increase accuracy of P0. Delta > 0 |

## Generalized_Pareto_calc_P0_iter

*Generalized Pareto Distribution Function P0 with defined number of iterations*

### Description

Returns P0 of Generalized Pareto given a parameters theta, c, b , and rho

### Usage

```
Generalized_Pareto_calc_P0_iter(theta, c, b, rho, iter)
```

### Arguments

| | |
|---|---|
| theta | Parameter of the Generalized Pareto function |
| c | Parameter of the Generalized Pareto function |
| b | Parameter of the Generalized Pareto function |
| rho | Parameter of the Generalized Pareto function |
| iter | Number of iterations to be performed for summation calcuation. Increasing this parameter will increase accuracy of P0 |

## Generalized_Yule

*Generalized Yule Distribution Function*

### Description

This function generates a vector of n length of the Generalized Yule distribution with parameters rho and alpha.

### Usage

```
Generalized_Yule(n, rho, alpha)
```

### Arguments

| | |
|---|---|
| n | Length of vector to be generated. |
| rho | Parameter of the Generalized Yule distribution function |
| alpha | Parameter of the Generalized Yule distribution function: 0 <= alpha < 1 |

### Examples

```
Generalized_Yule(100, 3, 0.1)
```

## get_CI *Get Mean Confidence Interval Function*

### Description

This function generates a vector of confidence interval based on mean of data.

### Usage

```
get_CI(data, alpha)
```

### Arguments

| | |
|---|---|
| data | Data to get confidence interval from |
| alpha | Alpha for confidence interval calculation |

## get_median_CI *Get Median Confidence Interval Function*

### Description

This function generates a vector of ranked 95

### Usage

```
get_median_CI(data)
```

### Arguments

| | |
|---|---|
| data | Data to get confidence interval from |

## get_p0 *Psi Criterion for RGHD parameter ratios*

### Description

This function generates the Psi Criterion goodness of fit value given an empirical distribution for the 2m-RGHD function. Parameters r and q/r ratios are given, as well as desired weight of pmf and use of the weighted right-tail cumulative distribution function.

### Usage

```
get_p0(params, model_fn_name)
```

### Arguments

| | |
|---|---|
| params | Vector of parameter for the model function |
| model_fn_name | name of function as a character vector |

### Examples

```
params <- c(2, 3, 0.9)
get_p0(params, 'Kolmogorov Waring')
```

---

## global_fit_function *Global optimization of a given function given empirical data and parameter bounds*

---

### Description

This function generates a single set of optimized parameters and Psi Criterion for a given function within specified starting parameter bounds. This function uses a modified grid search method for optimization

### Usage

```
global_fit_function(
  param_bounds,
  data,
  model_fn_name,
  iter = 1,
  weighted_rt = FALSE,
  n_cores = 1,
  clust
)
```

### Arguments

| | |
|---|---|
| param_bounds | A list of sequences which indicate space where parameters should be generated and fit |
| data | Vector of observed values |
| model_fn_name | Character vector indicating name of function of theoretical model to be used. For example, for Generalized_Yule(n, rho, alpha), model_fn_name <- 'Generalied Yule' |
| iter | Integer indicating number of iterations to run grid search. Increasing iterations will increase decimal point precision of output parameters. |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| n_cores | Integer used to indicate number of cores to be used for this function if a socket cluster object is not defined. |

| | |
|---|---|
| clust | socket cluster object from 'parallel::makeCluster()'. This is used if you have already generated a socket cluster object and would like to run this functoin on it. If no object is defined, one will be made for this function call. |

---

`global_fit_RGHD_ratio` *Global optimization of the 2m-RGHD function given empirical data, r bounds, and q/r bounds.*

---

## Description

This function generates a single set of optimized parameters and Psi Criterion for a given function within specified starting parameter bounds. This function uses Limited Memory BFGS as it's gradient descent algorithm.

## Usage

```
global_fit_RGHD_ratio(
  param_bounds,
  data,
  iter,
  weighted_rt = FALSE,
  n_cores = 1,
  clust
)
```

## Arguments

| | |
|---|---|
| param_bounds | A list of sequences which indicate space where parameters should be generated and fit |
| data | Vector of observed values |
| iter | Integer indicating number of iterations to run grid search. Increasing iterations will increase decimal point precision of output parameters. |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| n_cores | Integer used to indicate number of cores to be used for this function if a socket cluster object is not defined. |
| clust | socket cluster object from 'parallel::makeCluster()'. This is used if you have already generated a socket cluster object and would like to run this functoin on it. If no object is defined, one will be made for this function call. |

## Kolmogorov_Waring *Kolmogorov Waring*

### Description

Calculates vector of n length of Kolmogorov distribution function given parameters

### Usage

```
Kolmogorov_Waring(n, a, b, theta)
```

### Arguments

| | |
|---|---|
| n | Length of vector to be generated |
| a | Parameter of the Kolmogorov Waring distribution function |
| b | Parameter of the Kolmogorov Waring distribution function |
| theta | Parameter of the Kolmogorov Waring distribution function |

## Kolmogorov_Waring_P0 *Kolmogorov Waring P0*

### Description

Calculates P0 of Kolmogorov Waring distribution function given parameters. Approximation is used if parameters meet a specific criteria.

### Usage

```
Kolmogorov_Waring_P0(a, b, theta)
```

### Arguments

| | |
|---|---|
| a | Parameter of the Kolmogorov Waring distribution function |
| b | Parameter of the Kolmogorov Waring distribution function |
| theta | Parameter of the Kolmogorov Waring distribution function |

## Kolmogorov_Waring_P0_calc *Kolmogorov Waring P0 calculation*

### Description

Calculates P0 of Kolmogorov Waring distribution function given parameters

### Usage

```
Kolmogorov_Waring_P0_calc(a, b, theta)
```

### Arguments

| | |
|---|---|
| a | Parameter of the Kolmogorov Waring distribution function |
| b | Parameter of the Kolmogorov Waring distribution function |
| theta | Parameter of the Kolmogorov Waring distribution function |

## local_fit_function *Local optimization of a given function given empirical data and parameter bounds*

### Description

This function generates a table of optimized parameters and Psi Criterion for a given function within specified starting parameter bounds. This function uses Limited Memory BFGS as it's gradient descent algorithm.

### Usage

```
local_fit_function(
  param_bounds,
  data,
  model_fn_name,
  weighted_rt = FALSE,
  par_chunk = 100,
  par_chunk_size = 10,
  n_cores = 1,
  clust,
  left_trunc = 1,
  right_trunc = left_trunc + length(data) - 1
)
```

### Arguments

| | |
|---|---|
| param_bounds | A list of sequences which indicate space where parameters should be generated and fit |
| data | Vector of observed values |
| model_fn_name | Character vector indicating name of function of theoretical model to be used. For example, for Generalized_Yule(n, rho, alpha), model_fn_name <- 'Generalied Yule' |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| par_chunk | Integer used to indicate number of optimization chunks to be run. Total number of rows in the output table = par_chunk * par_chunk_size |
| par_chunk_size | Integer used to indicate number of starting parameters to be generated and optimized in a given chunk. Total number of rows in the output table = par_chunk * par_chunk_size |
| n_cores | Integer used to indicate number of cores to be used for this function if a socket cluster object is not defined. |
| clust | socket cluster object from 'parallel::makeCluster()'. This is used if you have already generated a socket cluster object and would like to run this functoin on it. If no object is defined, one will be made for this function call. |
| left_trunc | Int used to determine starting index of model to use for optimization |
| right_trunc | Int used to determine ending index of model to use for optimization |

---

## local_fit_RGHD_ratio *Local optimization of the 2m-RGHD function given empirical data, r bounds, and q/r bounds.*

---

### Description

This function generates a table of optimized parameters and Psi Criterion for a given function within specified starting parameter bounds. This function uses Limited Memory BFGS as it's gradient descent algorithm.

### Usage

```
local_fit_RGHD_ratio(
  param_bounds,
  data,
  weighted_rt = FALSE,
  par_chunk = 100,
  par_chunk_size = 10,
  n_cores = 1,
  clust,
  left_trunc = 1,
  right_trunc = left_trunc + length(data) - 1
)
```

### Arguments

| | |
|---|---|
| param_bounds | A list of sequences which indicate space where parameters should be generated and fit |
| data | Vector of observed values |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| par_chunk | Integer used to indicate number of optimization chunks to be run. Total number of rows in the output table = par_chunk * par_chunk_size |
| par_chunk_size | Integer used to indicate number of starting parameters to be generated and optimized in a given chunk. Total number of rows in the output table = par_chunk * par_chunk_size |
| n_cores | Integer used to indicate number of cores to be used for this function if a socket cluster object is not defined. |
| clust | socket cluster object from 'parallel::makeCluster()'. This is used if you have already generated a socket cluster object and would like to run this functoin on it. If no object is defined, one will be made for this function call. |
| left_trunc | Int used to determine starting index of model to use for optimization |
| right_trunc | Int used to determine ending index of model to use for optimization |

---

## Lorentzian *Lorentzian Distribution Function*

---

### Description

This function generates a vector of n length of the Lorentzian distribution

### Usage

```
Lorentzian(n, gamma, x0, c)
```

### Arguments

| | |
|---|---|
| n | Length of vector to be generated. |
| gamma | Parameter of the Lorenzian distribution function |
| x0 | Parameter of the Lorenzian distribution function indicating center of function |
| c | Parameter of the Lorenzian distribution function indicating center of function |

### Examples

```
Lorentzian_calc(5, 5.5, 6, 2)
```

## Lorentzian_calc *Lorentzian Distribution Function calculation*

### Description

This function calculates value of Lorentzian function at x

### Usage

```
Lorentzian_calc(x, gamma, x0, c)
```

### Arguments

| | |
|---|---|
| x | Index of function |
| gamma | Parameter of the Lorenzian distribution function |
| x0 | Parameter of the Lorenzian distribution function indicating center of function |
| c | Parameter of the Lorenzian distribution function indicating center of function |

### Examples

```
Lorentzian_calc(5, 5.5, 6, 2)
```

## parameter_post_processing *Parameter Optimization Helper Function*

### Description

This function adds in additional columns to the optimized parameter output dataframe

### Usage

```
parameter_post_processing(parameter_df, model_fn_name, data)
```

### Arguments

| | |
|---|---|
| parameter_df | Output dataframe of optimized parameters using local algorithm |
| model_fn_name | Character vector used to indicate name of model function used for optimization |
| data | Vector of observed values |

`plot_model` *Plot Model Helper Function*

## Description

This function generates various plots of empirical data and models

## Usage

```
plot_model(
  title,
  model_fn_name,
  data,
  parameter_df,
  n_parameters,
  plot_folder_name,
  xlab,
  left_trunc = 1
)
```

## Arguments

| | |
|---|---|
| `title` | Character vector indicating title of the empirical dataset, this will be present on every plot, this also determines the name of the folder where plots will be |
| `model_fn_name` | Character vector used to indicate name of model function used for optimization |
| `data` | Vector of observed values |
| `parameter_df` | Data frame of optimized parameters and other model function values (p0, Psi, etc) |
| `n_parameters` | Int of number of parameters used in model funciton |
| `plot_folder_name` | Character vector indicating folder or directory name to be used when outputting plot images |
| `xlab` | Character vector indicating x axis label of plots, indicates what the random variable is |
| `left_trunc` | Int indicating starting index of model function used for optimization |

psi_criterion *Psi Criterion*

## Description

This function generates the Psi Criterion goodness of fit value given an empirical distribution, theoretical modeled distribution, and number of parameters in the theoretical distribution.

## Usage

```
psi_criterion(data, model, n_parameters)
```

## Arguments

| | |
|---|---|
| data | Vector of observed values |
| model | Vector of theoretical values to be compared |
| n_parameters | Number of parameters of function used to generate model |

## Examples

```
obs_data <- c(100,75,20,1)
model_data <- Kolmogorov_Waring(length(obs_data), 2, 3, 0.9)
psi <- psi_criterion(obs_data, model_data, 3)
```

psi_criterion_function
*Psi Criterion given a function*

## Description

This function generates the Psi Criterion goodness of fit value given an empirical distribution. The function and parameters are given, as well as desired weight of pmf and use of the weighted right-tail cumulative distribution function.

## Usage

```
psi_criterion_function(
  params,
  data,
  model_fn,
  pmf_weight = 0,
  weighted_rt = FALSE,
  left_trunc = 1,
  right_trunc = left_trunc + length(data) - 1
)
```

### Arguments

| | |
|---|---|
| params | Vector of parameters for model_fn, not including n. For example, for Generalized_Yule(n, rho, alpha), params will be c(rho, alpha) |
| data | Vector of observed values |
| model_fn | Function of theoretical model to be used. For example, for Generalized_Yule(n, rho, alpha), model_fn <- Generalied_Yule |
| pmf_weight | Numeric of weight given to probability mass function for generation of Psi Criterion. For example, if pmf_weight <- 0.5, 50 percent of the Psi Criterion value will be attributed to the probability mass function while the other 50 percent will be attributed to the right-tail cumulative distribution function. |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| left_trunc | Int used to determine starting index of model to use for optimization |
| right_trunc | Int used to determine ending index of model to use for optimization |

### Examples

```
obs_data <- c(100,75,20,1)
parameters <- c(1,2,0.8)
psi <- psi_criterion_function(parameters, obs_data, Kolmogorov_Waring)
```

---

## psi_criterion_RGHD_ratio

*Psi Criterion for RGHD parameter ratios*

---

### Description

This function generates the Psi Criterion goodness of fit value given an empirical distribution for the 2m-RGHD function. Parameters r and q/r ratios are given, as well as desired weight of pmf and use of the weighted right-tail cumulative distribution function.

### Usage

```
psi_criterion_RGHD_ratio(
  params,
  data,
  m,
  pmf_weight = 0,
  weighted_rt = FALSE,
  left_trunc = 1,
  right_trunc = left_trunc + length(data) - 1
)
```

### Arguments

| | |
|---|---|
| params | Vector of parameters for model_fn, not including n. For example, for 2m-RGHD (m=2), params <- c(3, 5, 0.3, 1.5). In this case r1 = 3, r2 = 5, q1/r1 = 0.3, and q2/r2 = 1.5 |
| data | Vector of observed values |
| m | m parameter for 2m-RGHD function |
| pmf_weight | Numeric of weight given to probability mass function for generation of Psi Criterion. For example, if pmf_weight <- 0.5, 50 percent of the Psi Criterion value will be attributed to the probability mass function while the other 50 percent will be attributed to the right-tail cumulative distribution function. |
| weighted_rt | Boolean used to determine if the weighted right-tail cumulative distribution function should be used or not. |
| left_trunc | Int used to determine starting index of model to use for optimization |
| right_trunc | Int used to determine ending index of model to use for optimization |

### Examples

```
obs_data <- c(100,75,20,1)
parameters <- c(3, 5, 0.3, 1.5)
psi <- psi_criterion_RGHD_ratio(parameters, obs_data, 2)
```

---

## RGHD *2m-RGHD Distribution Function*

---

### Description

Returns doubly truncated vector of 2m-RGHD function values where input is 1-J

### Usage

```
RGHD(J, m, r, q, P0_iter = 100L, P0_included = FALSE)
```

### Arguments

| | |
|---|---|
| J | Length of vector to be generated |
| m | Parameter of the 2m-RGHD function, this defines number of r and q parameters of the function |
| r | R vector containing r parameters from 1:m |
| q | R vector containing q parameters from 1:m |
| P0_iter | Integer indicating number of iterations to use for calculation of P0, increasing this parameter will increase accuracy of P0 |
| P0_included | Boolean used to include P0 in vector or not |

## RGHD_P0 *2m-RGHD Distribution Function P0*

### Description

Calculates P0 given a set of parameters

### Usage

```
RGHD_P0(m, r, q)
```

### Arguments

| | |
|---|---|
| m | Parameter of the 2m-RGHD function, this defines number of r and q parameters of the function |
| r | R vector containing r parameters from 1:m |
| q | R vector containing q parameters from 1:m |

## RGHD_P0_calc *2m-RGHD Distribution Function P0 calculation*

### Description

Calculates P0 given a set of parameters

### Usage

```
RGHD_P0_calc(sigma_upper, m, r, q)
```

### Arguments

| | |
|---|---|
| sigma_upper | Int which determine number of iterations for calculation to go through, this is needed to approximate sigma infinity |
| m | Parameter of the 2m-RGHD function, this defines number of r and q parameters of the function |
| r | R vector containing r parameters from 1:m |
| q | R vector containing q parameters from 1:m |

## right_tail_cdf *Right-Tail Cumulative Distribution Function*

### Description

This function generates a vector of the right-tail cumulative distribution function of a given vector of values.

### Usage

```
right_tail_cdf(x)
```

### Arguments

| | |
|---|---|
| x | Length of vector to be generated. |

### Examples

```
x <- c(1,2,3,4,5)
right_tail_cdf(x)
```

## skeweDF_auto *SkeweDF Auto Helper Function*

### Description

This function will automatically optimize parameters for an empirical dataset given a model function and generate plots and tables

### Usage

```
skeweDF_auto(
  title = "Dataset",
  data,
  xlab = "Random Variable",
  param_bounds,
  model_fn_name,
  left_trunc = 1,
  right_trunc = left_trunc + length(data) - 1,
  n_cores = 1
)
```

### Arguments

| | |
|---|---|
| title | Character vector indicating title of the empirical dataset, this will be present on every plot, this also determines the name of the folder where plots will be |
| data | Vector of observed values |
| xlab | Character vector indicating x axis label of plots, indicates what the random variable is |
| param_bounds | A list of sequences which indicate space where parameters should be generated and fit |
| model_fn_name | Character vector used to indicate name of model function used for optimization |
| left_trunc | Int used to determine starting index of model to use for optimization |
| right_trunc | Int used to determine ending index of model to use for optimization |
| n_cores | Integer used to indicate number of cores to be used for this function if a socket cluster object is not defined. |

---

## weighted_left_tail_cdf

*Weighted Left-Tail Cumulative Distribution Function*

---

### Description

This function generates a vector of the weighted left-tail cumulative distribution function of a given vector of values. The weight of of each variable is determined by its position in the vector. For example, with a vector of length 5, element 1 will have weight 5/(5+4+3+2+1). Element 1 will have weight 5/(5+4+3+2+1)

### Usage

```
weighted_left_tail_cdf(x)
```

### Arguments

| | |
|---|---|
| x | Length of vector to be generated. |

### Examples

```
x <- c(1,2,3,4,5)
weighted_left_tail_cdf(x)
```

## weighted_right_tail_cdf *Weighted Right-Tail Cumulative Distribution Function*

### Description

This function generates a vector of the weighted right-tail cumulative distribution function of a given vector of values. The weight of of each variable is determined by its position in the vector. For example, with a vector of length 5, element 5 will have weight 5/(5+4+3+2+1). Element 1 will have weight 1/(5+4+3+2+1)

### Usage

```
weighted_right_tail_cdf(x)
```

### Arguments

| | |
|---|---|
| x | Length of vector to be generated. |

### Examples

```
x <- c(1,2,3,4,5)
weighted_right_tail_cdf(x)
```

## write_input_table *Write Input Table Helper Function*

### Description

This function generates table of input data

### Usage

```
write_input_table(folder_name, data)
```

### Arguments

| | |
|---|---|
| folder_name | Character vector indicating folder or directory name to be used when outputting table |
| data | Vector of observed values |

## write_parameter_table *Write Parameter Table Helper Function*

### Description

This function generates table of optimized parameters

### Usage

```
write_parameter_table(parameter_df, folder_name, model_fn_name, RGHD_m = 0)
```

### Arguments

| | |
|---|---|
| parameter_df | Data frame of optimized parameters and other model function values (p0, Psi, etc) |
| folder_name | Character vector indicating folder or directory name to be used when outputting table |
| model_fn_name | Character vector used to indicate name of model function used for optimization |
| RGHD_m | Int indicating m value of 2m-RGHD function if applicable |

## write_summary_table *Write Summary Table Helper Function*

### Description

This function generates summary statistics table of optimized parameters

### Usage

```
write_summary_table(parameter_df, folder_name, model_fn_name, RGHD_m = 0)
```

### Arguments

| | |
|---|---|
| parameter_df | Data frame of optimized parameters and other model function values (p0, Psi, etc) |
| folder_name | Character vector indicating folder or directory name to be used when outputting table |
| model_fn_name | Character vector used to indicate name of model function used for optimization |
| RGHD_m | Int indicating m value of 2m-RGHD function if applicable |

`Yule` *Yule Distribution Function*

## Description

This function generates a vector of n length of the Yule distribution with parameter rho.

## Usage

```
Yule(n, rho)
```

## Arguments

| | |
|---|---|
| `n` | Length of vector to be generated. |
| `rho` | Parameter of the Yule distribution function |

## Examples

```
Yule(100, 3)
```

# Index