# Package 'TidyDensity'

May 28, 2024

Title Functions for Tidy Analysis and Generation of Random Data
Version 1.5.0
<b>Description</b> To make it easy to generate random numbers based upon the underlying stats distribution functions. All data is returned in a tidy and structured format making working with the data simple and straight forward. Given that the data is returned in a tidy 'tibble' it lends itself to working with the rest of the 'tidyverse'.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.3.1
<pre>URL https://github.com/spsanderson/TidyDensity</pre>
BugReports https://github.com/spsanderson/TidyDensity/issues
<b>Depends</b> R (>= $4.1.0$ )
<b>Imports</b> magrittr, rlang (>= 0.4.11), dplyr, ggplot2, plotly, tidyr, purrr, actuar, methods, stats, patchwork, survival, nloptr, broom, tidyselect, data.table, stringr
Suggests rmarkdown, knitr, EnvStats
VignetteBuilder knitr
NeedsCompilation no
Author Steven Sanderson [aut, cre, cph] ( <a href="https://orcid.org/0009-0006-7661-8247">https://orcid.org/0009-0006-7661-8247</a> )
Maintainer Steven Sanderson <pre><spsanderson@gmail.com></spsanderson@gmail.com></pre>
Repository CRAN
<b>Date/Publication</b> 2024-05-28 13:50:03 UTC
R topics documented:
bootstrap_density_augment

pootstrap_q_augment	
pootstrap_q_vec	9
oootstrap_stat_plot	10
pootstrap_unnest_tbl	12
egmean	13
check_duplicate_rows	
chmean	
ci hi	
ci_lo	
ckurtosis	
cmean	
cmedian	
color_blind	
convert_to_ts	
esd	
eskewness	
evar	
dist_type_extractor	
quantile_normalize	
d_scale_color_colorblind	
d_scale_fill_colorblind	
idy_autoplot	
idy_bernoulli	
idy_beta	
idy_binomial	
idy_bootstrap	
idy_burr	
idy_cauchy	37
idy_chisquare	38
idy_combined_autoplot	40
idy_combine_distributions	
idy_distribution_comparison	43
idy_distribution_summary_tbl	45
idy_empirical	46
idy_exponential	47
idy_f	48
idy_four_autoplot	50
idy_gamma	51
idy_generalized_beta	
idy_generalized_pareto	
idy_geometric	
idy_hypergeometric	
idy_inverse_burr	
idy_inverse_exponential	
idy_inverse_gamma	
idy_inverse_gamma	
idy_inverse_pareto	
idy_inverse_pareto	
.14.4 111.16.16. M. T. M.	0/

tidy_kurtosis_vec
tidy_logistic
tidy_lognormal
tidy_mcmc_sampling
tidy_mixture_density
tidy_multi_dist_autoplot
tidy_multi_single_dist
tidy_negative_binomial
tidy_normal
tidy_paralogistic
tidy_pareto
tidy_pareto1
tidy_poisson
tidy_random_walk
<b>7</b> 1
tidy_range_statistic
tidy_triangular
•
tidy_weibull
tidy_zero_truncated_binomial
tidy_zero_truncated_geometric
tidy_zero_truncated_negative_binomial
tidy_zero_truncated_poisson
triangle_plot
util_bernoulli_param_estimate
util_bernoulli_stats_tbl
util_beta_aic
util_beta_param_estimate
util_beta_stats_tbl
util_binomial_aic
util_binomial_param_estimate
util_binomial_stats_tbl
util_burr_stats_tbl
util_cauchy_aic
util_cauchy_param_estimate
util_cauchy_stats_tbl
util_chisquare_param_estimate
util_chisquare_stats_tbl
util chisq aic
_ <u>_</u>
util_exponential_aic      util_exponential_param_estimate
util_exponential_stats_tbl   130     util f aic   131
uii i aic

util_f_param_estimate	. 132
$util\_f\_stats\_tbl  \dots $	. 133
util_gamma_aic	. 134
util_gamma_param_estimate	. 136
util_gamma_stats_tbl	
util_generalized_beta_aic	
util_generalized_beta_param_estimate	
util_generalized_beta_stats_tbl	
util_generalized_pareto_aic	
util_generalized_pareto_param_estimate	
util_generalized_pareto_stats_tbl	
util_geometric_aic	
util_geometric_param_estimate	
•	
util_geometric_stats_tbl	
util_hypergeometric_aic	
util_hypergeometric_param_estimate	
util_hypergeometric_stats_tbl	
util_inverse_burr_aic	
util_inverse_burr_param_estimate	
util_inverse_burr_stats_tbl	
util_inverse_pareto_aic	
util_inverse_pareto_param_estimate	
util_inverse_pareto_stats_tbl	
util_inverse_weibull_aic	. 162
util_inverse_weibull_param_estimate	. 163
util_inverse_weibull_stats_tbl	. 164
util_logistic_aic	. 165
util_logistic_param_estimate	. 167
util_logistic_stats_tbl	. 168
util_lognormal_aic	
util_lognormal_param_estimate	
util_lognormal_stats_tbl	
util_negative_binomial_aic	
util_negative_binomial_param_estimate	
util_negative_binomial_stats_tbl	
util_normal_aic	
util_normal_param_estimate	
util_normal_stats_tbl	
util_paralogistic_aic	
util_paralogistic_param_estimate	
util_paralogistic_stats_tbl	
util_pareto1_aic	
util_pareto1_param_estimate	
util_pareto1_stats_tbl	
util_pareto_aic	
util_pareto_param_estimate	
util_pareto_stats_tbl	
util_poisson_aic	. 192

	util_poisson_param_estimate	. 194
	util_poisson_stats_tbl	. 195
	util_triangular_aic	. 196
	util_triangular_param_estimate	. 198
	util_triangular_stats_tbl	. 199
	util_t_aic	. 200
	util_t_param_estimate	. 201
	util_t_stats_tbl	. 203
	util_uniform_aic	. 204
	util_uniform_param_estimate	. 205
	util_uniform_stats_tbl	. 206
	util_weibull_aic	. 207
	util_weibull_param_estimate	. 209
	util_weibull_stats_tbl	. 210
	util_zero_truncated_binomial_aic	. 211
	util_zero_truncated_binomial_param_estimate	. 213
	util_zero_truncated_binomial_stats_tbl	
	util_zero_truncated_geometric_aic	. 215
	util_zero_truncated_geometric_param_estimate	
	util_zero_truncated_geometric_stats_tbl	
	util_zero_truncated_negative_binomial_aic	. 219
	util_zero_truncated_negative_binomial_param_estimate	. 220
	util_zero_truncated_negative_binomial_stats_tbl	. 222
	util_zero_truncated_poisson_aic	
	util_zero_truncated_poisson_param_estimate	. 224
	util_zero_truncated_poisson_stats_tbl	
	• – –	
Index		228

bootstrap\_density\_augment

Bootstrap Density Tibble

# Description

Add density information to the output of tidy\_bootstrap(), and bootstrap\_unnest\_tbl().

# Usage

bootstrap\_density\_augment(.data)

# Arguments

. data  $\qquad \qquad \text{The data that is passed from the tidy\_bootstrap() or bootstrap\_unnest\_tbl()} \\ \text{functions.}$ 

6 bootstrap\_p\_augment

#### **Details**

This function takes as input the output of the tidy\_bootstrap() or bootstrap\_unnest\_tbl() and returns an augmented tibble that has the following columns added to it: x, y, dx, and dy.

It looks for an attribute that comes from using tidy\_bootstrap() or bootstrap\_unnest\_tbl() so it will not work unless the data comes from one of those functions.

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Bootstrap: bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_q_vec(), bootstrap_stat_plot(), bootstrap_unnest_tbl(), tidy_bootstrap()

Other Augment Function: bootstrap_p_augment(), bootstrap_q_augment()
```

## **Examples**

```
x <- mtcars$mpg
tidy_bootstrap(x) |>
bootstrap_density_augment()
tidy_bootstrap(x) |>
bootstrap_unnest_tbl() |>
bootstrap_density_augment()
```

bootstrap\_p\_augment Augment Bootstrap P

## **Description**

Takes a numeric vector and will return the ecdf probability.

## Usage

```
bootstrap_p_augment(.data, .value, .names = "auto")
```

#### **Arguments**

. data The data being passed that will be augmented by the function.

.value This is passed rlang::enquo() to capture the vectors you want to augment.

.names The default is "auto"

bootstrap\_p\_vec 7

#### **Details**

Takes a numeric vector and will return the ecdf probability of that vector. This function is intended to be used on its own in order to add columns to a tibble.

#### Value

A augmented tibble

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Augment Function: bootstrap_density_augment(), bootstrap_q_augment()
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_q_vec(), bootstrap_stat_plot(), bootstrap_unnest_tbl(), tidy_bootstrap()
```

## **Examples**

```
x <- mtcars$mpg
tidy_bootstrap(x) |>
bootstrap_unnest_tbl() |>
bootstrap_p_augment(y)
```

bootstrap\_p\_vec

Compute Bootstrap P of a Vector

#### **Description**

This function takes in a vector as it's input and will return the ecdf probability of a vector.

#### Usage

```
bootstrap_p_vec(.x)
```

## Arguments

. x

A numeric

#### **Details**

A function to return the ecdf probability of a vector.

#### Value

A vector

bootstrap\_q\_augment

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_q_augment(), bootstrap_q_vec(), bootstrap_stat_plot(), bootstrap_unnest_tbl(), tidy_bootstrap()

Other Vector Function: bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

#### **Examples**

```
x <- mtcars$mpg
bootstrap_p_vec(x)</pre>
```

bootstrap\_q\_augment

Augment Bootstrap Q

## **Description**

Takes a numeric vector and will return the quantile.

## Usage

```
bootstrap_q_augment(.data, .value, .names = "auto")
```

#### **Arguments**

.value This is passed rlang::enquo() to capture the vectors you want to augment.

.names The default is "auto"

#### **Details**

Takes a numeric vector and will return the quantile of that vector. This function is intended to be used on its own in order to add columns to a tibble.

#### Value

A augmented tibble

#### Author(s)

Steven P. Sanderson II, MPH

bootstrap\_q\_vec 9

#### See Also

```
Other Augment Function: bootstrap_density_augment(), bootstrap_p_augment()
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_vec(), bootstrap_stat_plot(), bootstrap_unnest_tbl(), tidy_bootstrap()
```

#### **Examples**

```
x <- mtcars$mpg
tidy_bootstrap(x) |>
bootstrap_unnest_tbl() |>
bootstrap_q_augment(y)
```

bootstrap\_q\_vec

Compute Bootstrap Q of a Vector

## **Description**

This function takes in a vector as it's input and will return the quantile of a vector.

#### Usage

```
bootstrap_q_vec(.x)
```

## **Arguments**

. x

A numeric

#### **Details**

A function to return the quantile of a vector.

#### Value

A vector

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_stat_plot(), bootstrap_unnest_tbl(), tidy_bootstrap()
Other Vector Function: bootstrap_p_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

10 bootstrap\_stat\_plot

#### **Examples**

```
x <- mtcars$mpg
bootstrap_q_vec(x)</pre>
```

bootstrap\_stat\_plot

Bootstrap Stat Plot

# Description

This function produces a plot of a cumulative statistic function applied to the bootstrap variable from tidy\_bootstrap() or after bootstrap\_unnest\_tbl() has been applied to it.

# Usage

```
bootstrap_stat_plot(
    .data,
    .value,
    .stat = "cmean",
    .show_groups = FALSE,
    .show_ci_labels = TRUE,
    .interactive = FALSE
)
```

# Arguments

.data	The data that comes from either tidy_bootstrap() or after bootstrap_unnest_tbl() is applied to it.
.value	The value column that the calculations are being applied to.
.stat	The cumulative statistic function being applied to the .value column. It must be quoted. The default is "cmean".
.show_groups	The default is FALSE, set to TRUE to get output of all simulations of the bootstrap data.
.show_ci_labels	
	The default is TRUE, this will show the last value of the upper and lower quantile.
.interactive	The default is FALSE, set to TRUE to get a plotly plot object back.

## **Details**

This function will take in data from either tidy\_bootstrap() directly or after apply bootstrap\_unnest\_tbl() to its output. There are several different cumulative functions that can be applied to the data. The accepted values are:

• "cmean" - Cumulative Mean

bootstrap\_stat\_plot 11

- "chmean" Cumulative Harmonic Mean
- "cgmean" Cumulative Geometric Mean
- "csum" = Cumulative Sum
- "cmedian" = Cumulative Median
- "cmax" = Cumulative Max
- "cmin" = Cumulative Min
- "cprod" = Cumulative Product
- "csd" = Cumulative Standard Deviation
- "cvar" = Cumulative Variance
- "cskewness" = Cumulative Skewness
- "ckurtosis" = Cumulative Kurtotsis

#### Value

A plot either ggplot2 or plotly.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Bootstrap_bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_q_vec(), bootstrap_unnest_tbl(), tidy_bootstrap()

Other Autoplot: tidy_autoplot(), tidy_combined_autoplot(), tidy_four_autoplot(), tidy_multi_dist_autoplot tidy_random_walk_autoplot()
```

```
x <- mtcars$mpg
tidy_bootstrap(x) |>
  bootstrap_stat_plot(y, "cmean")

tidy_bootstrap(x, .num_sims = 10) |>
  bootstrap_stat_plot(y,
    .stat = "chmean", .show_groups = TRUE,
    .show_ci_label = FALSE
)
```

12 bootstrap\_unnest\_tbl

#### **Description**

Unnest the data output from tidy\_bootstrap().

#### Usage

```
bootstrap_unnest_tbl(.data)
```

#### **Arguments**

.data

The data that is passed from the tidy\_bootstrap() function.

#### **Details**

This function takes as input the output of the tidy\_bootstrap() function and returns a two column tibble. The columns are sim\_number and y

It looks for an attribute that comes from using tidy\_bootstrap() so it will not work unless the data comes from that function.

#### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_qvec(), bootstrap_stat_plot(), tidy_bootstrap()
```

```
tb <- tidy_bootstrap(.x = mtcars$mpg)
bootstrap_unnest_tbl(tb)

bootstrap_unnest_tbl(tb) |>
   tidy_distribution_summary_tbl(sim_number)
```

cgmean 13

cgmean

Cumulative Geometric Mean

# Description

A function to return the cumulative geometric mean of a vector.

## Usage

```
cgmean(.x)
```

# Arguments

. X

A numeric vector

#### **Details**

A function to return the cumulative geometric mean of a vector. exp(cummean(log(.x)))

## Value

A numeric vector

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other\ Vector\ Function:\ bootstrap\_p\_vec(),\ bootstrap\_q\_vec(),\ chmean(),\ ckurtosis(),\ cmean(),\ cmedian(),\ csd(),\ cskewness(),\ cvar(),\ tidy\_kurtosis\_vec(),\ tidy\_scale\_zero\_one\_vec(),\ tidy\_skewness\_vec()
```

```
x <- mtcars$mpg
cgmean(x)</pre>
```

check\_duplicate\_rows

## **Description**

This function checks for duplicate rows in a data frame.

# Usage

```
check_duplicate_rows(.data)
```

### **Arguments**

.data

A data frame.

#### **Details**

This function checks for duplicate rows by comparing each row in the data frame to every other row. If a row is identical to another row, it is considered a duplicate.

#### Value

A logical vector indicating whether each row is a duplicate or not.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
duplicated, anyDuplicated
Other Utility: convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(),
util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(),
util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(),
util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic
util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(),
util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(),
util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(),
util_zero_truncated_poisson_aic()
```

```
data <- data.frame(
  x = c(1, 2, 3, 1),
  y = c(2, 3, 4, 2),
  z = c(3, 2, 5, 3)
)</pre>
```

chmean 15

```
check_duplicate_rows(data)
```

chmean

Cumulative Harmonic Mean

# Description

A function to return the cumulative harmonic mean of a vector.

# Usage

```
chmean(.x)
```

# Arguments

. X

A numeric vector

# **Details**

A function to return the cumulative harmonic mean of a vector. 1 / (cumsum(1 / .x))

# Value

A numeric vector

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

```
x <- mtcars$mpg
chmean(x)</pre>
```

ci\_hi

Confidence Interval Generic

# Description

Gets the upper 97.5% quantile of a numeric vector.

# Usage

```
ci_hi(.x, .na_rm = FALSE)
```

# Arguments

.x A vector of numeric values

.na\_rm A Boolean, defaults to FALSE. Passed to the quantile function.

## **Details**

Gets the upper 97.5% quantile of a numeric vector.

# Value

A numeric value.

# Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Statistic: ci_lo(), tidy_kurtosis_vec(), tidy_range_statistic(), tidy_skewness_vec(), tidy_stat_tbl()
```

```
x <- mtcars$mpg
ci_hi(x)</pre>
```

*ci\_lo* 17

ci\_lo

Confidence Interval Generic

# Description

Gets the lower 2.5% quantile of a numeric vector.

# Usage

```
ci_lo(.x, .na_rm = FALSE)
```

# Arguments

.x A vector of numeric values

.na\_rm A Boolean, defaults to FALSE. Passed to the quantile function.

## **Details**

Gets the lower 2.5% quantile of a numeric vector.

#### Value

A numeric value.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Statistic: ci_hi(), tidy_kurtosis_vec(), tidy_range_statistic(), tidy_skewness_vec(), tidy_stat_tbl()
```

```
x <- mtcars$mpg
ci_lo(x)</pre>
```

18 ckurtosis

ckurtosis

Cumulative Kurtosis

# Description

A function to return the cumulative kurtosis of a vector.

## Usage

```
ckurtosis(.x)
```

# Arguments

. X

A numeric vector

## **Details**

A function to return the cumulative kurtosis of a vector.

#### Value

A numeric vector

#### Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

```
x <- mtcars$mpg
ckurtosis(x)</pre>
```

cmean 19

cmean

Cumulative Mean

# Description

A function to return the cumulative mean of a vector.

# Usage

```
cmean(.x)
```

# Arguments

. x

A numeric vector

## **Details**

A function to return the cumulative mean of a vector. It uses dplyr::cummean() as the basis of the function.

#### Value

A numeric vector

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other\ Vector\ Function:\ bootstrap\_p\_vec(),\ bootstrap\_q\_vec(),\ cgmean(),\ chmean(),\ ckurtosis(),\ cmedian(),\ csd(),\ cskewness(),\ cvar(),\ tidy\_kurtosis\_vec(),\ tidy\_scale\_zero\_one\_vec(),\ tidy\_skewness\_vec()
```

```
x <- mtcars$mpg
cmean(x)</pre>
```

20 cmedian

cmedian

Cumulative Median

# Description

A function to return the cumulative median of a vector.

## Usage

```
cmedian(.x)
```

# Arguments

. X

A numeric vector

## **Details**

A function to return the cumulative median of a vector.

#### Value

A numeric vector

#### Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

```
x <- mtcars$mpg
cmedian(x)</pre>
```

color\_blind 21

-				
$\sim$	lor	hl	1.1	าฝ
CUJ	LUI	-	- 11	IU

Provide Colorblind Compliant Colors

#### Description

8 Hex RGB color definitions suitable for charts for colorblind people.

#### Usage

```
color_blind()
```

convert\_to\_ts

Convert Data to Time Series Format

#### **Description**

This function converts data in a data frame or tibble into a time series format. It is designed to work with data generated from tidy\_ distribution functions. The function can return time series data, pivot it into long format, or both.

#### Usage

```
convert_to_ts(.data, .return_ts = TRUE, .pivot_longer = FALSE)
```

## **Arguments**

. data A data frame or tibble to be converted into a time series format.

.return\_ts A logical value indicating whether to return the time series data. Default is

TRUE.

.pivot\_longer A logical value indicating whether to pivot the data into long format. Default is

FALSE.

#### **Details**

The function takes a data frame or tibble as input and processes it based on the specified options. It performs the following actions:

- 1. Checks if the input is a data frame or tibble; otherwise, it raises an error.
- 2. Checks if the data comes from a tidy\_ distribution function; otherwise, it raises an error.
- 3. Converts the data into a time series format, grouping it by "sim\_number" and transforming the "y" column into a time series.
- 4. Returns the result based on the chosen options:
  - If ret\_ts is set to TRUE, it returns the time series data.
  - If pivot\_longer is set to TRUE, it pivots the data into long format.
  - If both options are set to FALSE, it returns the data as a tibble.

22 csd

#### Value

The function returns the processed data based on the chosen options:

- If ret\_ts is set to TRUE, it returns time series data.
- If pivot\_longer is set to TRUE, it returns the data in long format.
- If both options are set to FALSE, it returns the data as a tibble.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

#### **Examples**

```
# Example 1: Convert data to time series format without returning time series data
x <- tidy_normal()
result <- convert_to_ts(x, FALSE)
head(result)

# Example 2: Convert data to time series format and pivot it into long format
x <- tidy_normal()
result <- convert_to_ts(x, FALSE, TRUE)
head(result)

# Example 3: Convert data to time series format and return the time series data
x <- tidy_normal()
result <- convert_to_ts(x)
head(result)</pre>
```

csd

Cumulative Standard Deviation

## **Description**

A function to return the cumulative standard deviation of a vector.

cskewness 23

# Usage

```
csd(.x)
```

## **Arguments**

. X

A numeric vector

#### **Details**

A function to return the cumulative standard deviation of a vector.

## Value

A numeric vector. Note: The first entry will always be NaN.

# Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

## **Examples**

```
x <- mtcars$mpg
csd(x)</pre>
```

cskewness

Cumulative Skewness

## **Description**

A function to return the cumulative skewness of a vector.

# Usage

```
cskewness(.x)
```

## **Arguments**

. X

A numeric vector

24 cvar

#### **Details**

A function to return the cumulative skewness of a vector.

#### Value

A numeric vector

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

## **Examples**

```
x <- mtcars$mpg
cskewness(x)</pre>
```

cvar

Cumulative Variance

#### **Description**

A function to return the cumulative variance of a vector.

## Usage

```
cvar(.x)
```

## **Arguments**

. x

A numeric vector

#### **Details**

A function to return the cumulative variance of a vector. exp(cummean(log(.x)))

#### Value

A numeric vector. Note: The first entry will always be NaN.

## Author(s)

Steven P. Sanderson II, MPH

dist\_type\_extractor 25

#### See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

#### **Examples**

```
x <- mtcars$mpg
cvar(x)</pre>
```

dist\_type\_extractor

Extract Distribution Type from Tidy Distribution Object

## Description

Get the distribution name in title case from the tidy\_distribution function.

## Usage

```
dist_type_extractor(.x)
```

## **Arguments**

. x

The attribute list passed from a tidy\_ distribution function.

## **Details**

This will extract the distribution type from a tidy\_ distribution function output using the attributes of that object. You must pass the attribute directly to the function. It is meant really to be used internally.

You should be passing if using manually the \$tibble\_type attribute.

#### Value

A character string

#### Author(s)

Steven P. Sanderson II,

```
tn <- tidy_normal()
atb <- attributes(tn)
dist_type_extractor(atb$tibble_type)</pre>
```

26 quantile\_normalize

quantile\_normalize

Perform quantile normalization on a numeric matrix/data.frame

# Description

This function will perform quantile normalization on two or more distributions of equal length. Quantile normalization is a technique used to make the distribution of values across different samples more similar. It ensures that the distributions of values for each sample have the same quantiles. This function takes a numeric matrix as input and returns a quantile-normalized matrix.

## Usage

```
quantile_normalize(.data, .return_tibble = FALSE)
```

#### **Arguments**

. data A numeric matrix where each column represents a sample.

.return\_tibble A logical value that determines if the output should be a tibble. Default is 'FALSE'.

#### Details

This function performs quantile normalization on a numeric matrix by following these steps:

- 1. Sort each column of the input matrix.
- 2. Calculate the mean of each row across the sorted columns.
- 3. Replace each column's sorted values with the row means.
- 4. Unsort the columns to their original order.

#### Value

A list object that has the following:

- 1. A numeric matrix that has been quantile normalized.
- 2. The row means of the quantile normalized matrix.
- 3. The sorted data
- 4. The ranked indices

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
rowMeans: Calculate row means.
    apply: Apply a function over the margins of an array.
    order: Order the elements of a vector.
   Other Utility: check_duplicate_rows(), convert_to_ts(), tidy_mcmc_sampling(), util_beta_aic(),
    util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(),
    util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(),
    util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_ai
    util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(),
    util_paralogistic_aic(),util_pareto1_aic(),util_pareto_aic(),util_poisson_aic(),
   util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomia
    util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(),
   util_zero_truncated_poisson_aic()
Examples
    # Create a sample numeric matrix
   data <- matrix(rnorm(20), ncol = 4)</pre>
    # Perform quantile normalization
```

```
sapply(function(x) quantile(x, probs = seq(0, 1, 1 / 4)))
quantile_normalize(
data.frame(rnorm(30),
           rnorm(30)),
           .return_tibble = TRUE)
```

normalized\_data <- quantile\_normalize(data)</pre>

as.data.frame(normalized\_data\$normalized\_data) |>

td\_scale\_color\_colorblind

normalized\_data

Provide Colorblind Compliant Colors

#### **Description**

**Provide Colorblind Compliant Colors** 

## Usage

```
td_scale_color_colorblind(..., theme = "td")
```

#### **Arguments**

Data passed to the function . . .

theme This defaults to td and that is the only allowed value 28 tidy\_autoplot

```
td_scale_fill_colorblind
```

Provide Colorblind Compliant Colors

## Description

**Provide Colorblind Compliant Colors** 

#### Usage

```
td_scale_fill_colorblind(..., theme = "td")
```

#### **Arguments**

... Data passed to the function

theme This defaults to td and that is the only allowed value

tidy\_autoplot

Automatic Plot of Density Data

## **Description**

This is an auto plotting function that will take in a tidy\_ distribution function and a few arguments, one being the plot type, which is a quoted string of one of the following:

- density
- quantile
- probablity
- qq
- mcmc

If the number of simulations exceeds 9 then the legend will not print. The plot subtitle is put together by the attributes of the table passed to the function.

# Usage

```
tidy_autoplot(
   .data,
   .plot_type = "density",
   .line_size = 0.5,
   .geom_point = FALSE,
   .point_size = 1,
   .geom_rug = FALSE,
   .geom_smooth = FALSE,
   .geom_jitter = FALSE,
   .interactive = FALSE
)
```

tidy\_autoplot 29

## **Arguments**

.data	The data passed in from a tidy_distribution function like tidy_normal()
.plot_type	This is a quoted string like 'density'
.line_size	The size param ggplot
.geom_point	A Boolean value of TREU/FALSE, FALSE is the default. TRUE will return a plot with ggplot2::ggeom_point()
.point_size	The point size param for ggplot
.geom_rug	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_rug()
.geom_smooth	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_smooth() The aes parameter of group is set to FALSE. This ensures a single smoothing band returned with SE also set to FALSE. Color is set to 'black' and linetype is 'dashed'.
.geom_jitter	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_jitter()
.interactive	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return an interactive plotly plot.

#### **Details**

This function will spit out one of the following plots:

- density
- quantile
- probability
- qq
- mcmc

#### Value

A ggplot or a plotly plot.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

Other Autoplot: bootstrap\_stat\_plot(), tidy\_combined\_autoplot(), tidy\_four\_autoplot(), tidy\_multi\_dist\_autoplot(), tidy\_random\_walk\_autoplot()

```
tidy_normal(.num_sims = 5) |>
  tidy_autoplot()

tidy_normal(.num_sims = 20) |>
  tidy_autoplot(.plot_type = "qq")
```

30 tidy\_bernoulli

tidy\_bernoulli

Tidy Randomly Generated Bernoulli Distribution Tibble

## **Description**

This function will generate n random points from a Bernoulli distribution with a user provided, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q function of the distribution family.

#### Usage

```
tidy_bernoulli(.n = 50, .prob = 0.1, .num_sims = 1, .return_tibble = TRUE)
```

## **Arguments**

.n The number of randomly generated points you want.

.prob The probability of success/failure.

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## **Details**

This function uses the rbinom(), and its underlying p, d, and q functions. The *Bernoulli* distribution is a special case of the *Binomial* distribution with size = 1 hence this is why the binom functions are used and set to size = 1.

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

tidy\_beta 31

#### See Also

```
https://en.wikipedia.org/wiki/Bernoulli_distribution
```

```
Other Discrete Distribution: tidy_binomial(), tidy_hypergeometric(), tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), tidy_zero_truncated_poisson()
```

Other Bernoulli: util\_bernoulli\_param\_estimate(), util\_bernoulli\_stats\_tbl()

## **Examples**

```
tidy_bernoulli()
```

tidy\_beta

Tidy Randomly Generated Beta Distribution Tibble

## **Description**

This function will generate n random points from a beta distribution with a user provided, .shape1, .shape2, .ncp or non-centrality parameter, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

# Usage

```
tidy_beta(
    .n = 50,
    .shape1 = 1,
    .shape2 = 1,
    .ncp = 0,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

32 tidy\_beta

#### **Arguments**

.n	The number of randomly generated points you want.
.shape1	A non-negative parameter of the Beta distribution.
.shape2	A non-negative parameter of the Beta distribution.
.ncp	The non-centrality parameter of the Beta distribution.
.num_sims	The number of randomly generated simulations you want.
.return_tibble	A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying stats::rbeta(), and its underlying p, d, and q functions. For more information please see stats::rbeta()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://statisticsglobe.com/beta-distribution-in-r-dbeta-pbeta-qbeta-rbeta
https://en.wikipedia.org/wiki/Beta_distribution

Other Continuous Distribution: tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Beta: tidy_generalized_beta(), util_beta_param_estimate(), util_beta_stats_tbl()
```

```
tidy_beta()
```

tidy\_binomial 33

tidy\_binomial

Tidy Randomly Generated Binomial Distribution Tibble

#### **Description**

This function will generate n random points from a binomial distribution with a user provided, .size, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

#### Usage

```
tidy_binomial(
    .n = 50,
    .size = 0,
    .prob = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

#### **Arguments**

.n The number of randomly generated points you want.

. size Number of trials, zero or more.

.prob Probability of success on each trial.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## Details

This function uses the underlying stats::rbinom(), and its underlying p, d, and q functions. For more information please see stats::rbinom()

34 tidy\_bootstrap

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda366i.htm

Other Discrete Distribution: tidy_bernoulli(), tidy_hypergeometric(), tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), tidy_zero_truncated_poisson()

Other Binomial: tidy_negative_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_butil_binomial_param_estimate(), util_sinomial_stats_tbl(), util_negative_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_negative_binomial_stats_
```

#### **Examples**

```
tidy_binomial()
```

tidy\_bootstrap

Bootstrap Empirical Data

#### Description

Takes an input vector of numeric data and produces a bootstrapped nested tibble by simulation number.

#### Usage

```
tidy_bootstrap(
   .x,
   .num_sims = 2000,
   .proportion = 0.8,
   .distribution_type = "continuous"
)
```

#### **Arguments**

.x The vector of data being passed to the function. Must be a numeric vector.

.num\_sims The default is 2000, can be set to anything desired. A warning will pass to the console if the value is less than 2000.

.proportion How much of the original data do you want to pass through to the sampling function. The default is 0.80 (80%)

.distribution\_type

This can either be 'continuous' or 'discrete'

tidy\_burr 35

#### **Details**

This function will take in a numeric input vector and produce a tibble of bootstrapped values in a list. The table that is output will have two columns: sim\_number and bootstrap\_samples

The sim\_number corresponds to how many times you want the data to be resampled, and the bootstrap\_samples column contains a list of the boostrapped resampled data.

#### Value

A nested tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Bootstrap: bootstrap_density_augment(), bootstrap_p_augment(), bootstrap_p_vec(), bootstrap_q_augment(), bootstrap_qvec(), bootstrap_stat_plot(), bootstrap_unnest_tbl()
```

## **Examples**

```
x <- mtcars$mpg
tidy_bootstrap(x)</pre>
```

tidy\_burr

Tidy Randomly Generated Burr Distribution Tibble

## **Description**

This function will generate n random points from a Burr distribution with a user provided, .shape1, .shape2, .scale, .rate, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

36 tidy\_burr

#### Usage

```
tidy_burr(
    .n = 50,
    .shape1 = 1,
    .shape2 = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

#### **Arguments**

.n The number of randomly generated points you want.
.shape1 Must be strictly positive.
.shape2 Must be strictly positive.
.rate An alternative way to specify the .scale.
.scale Must be strictly positive.

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

#### **Details**

This function uses the underlying actuar::rburr(), and its underlying p, d, and q functions. For more information please see actuar::rburr()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II. MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

Other Burr: tidy\_inverse\_burr(), util\_burr\_param\_estimate(), util\_burr\_stats\_tbl()

```
Other Continuous Distribution: tidy_beta(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()
```

tidy\_cauchy 37

## **Examples**

```
tidy_burr()
```

tidy\_cauchy

Tidy Randomly Generated Cauchy Distribution Tibble

## **Description**

This function will generate n random points from a cauchy distribution with a user provided, .location, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_cauchy(
    .n = 50,
    .location = 0,
    .scale = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of randomly generated points you want.
.location The location parameter.
.scale The scale parameter, must be greater than or equal to 0.
.num\_sims The number of randomly generated simulations you want.
.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

38 tidy\_chisquare

### **Details**

This function uses the underlying stats::rcauchy(), and its underlying p, d, and q functions. For more information please see stats::rcauchy()

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3663.htm
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_pareto(), tidy_pareto(), tidy_pareto(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Cauchy: util_cauchy_param_estimate(), util_cauchy_stats_tbl()
```

## **Examples**

tidy\_cauchy()

tidy\_chisquare

Tidy Randomly Generated Chisquare (Non-Central) Distribution Tib-

## **Description**

This function will generate n random points from a chisquare distribution with a user provided, .df, .ncp, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

tidy\_chisquare 39

## Usage

```
tidy_chisquare(
    .n = 50,
    .df = 1,
    .ncp = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

. df Degrees of freedom (non-negative but can be non-integer)

. ncp Non-centrality parameter, must be non-negative.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

### **Details**

This function uses the underlying stats::rchisq(), and its underlying p, d, and q functions. For more information please see stats::rchisq()

## Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3666.htm
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_exponential(),
tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(),
tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(),
tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(),
tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(),
tidy_weibull(), tidy_zero_truncated_geometric()
Other Chisquare: util_chisquare_param_estimate(), util_chisquare_stats_tbl()
```

# **Examples**

```
tidy_chisquare()
```

```
tidy_combined_autoplot
```

Automatic Plot of Combined Multi Dist Data

## **Description**

This is an auto plotting function that will take in a tidy\_ distribution function and a few arguments, one being the plot type, which is a quoted string of one of the following:

- density
- quantile
- probablity
- qq
- mcmc

If the number of simulations exceeds 9 then the legend will not print. The plot subtitle is put together by the attributes of the table passed to the function.

## Usage

```
tidy_combined_autoplot(
    .data,
    .plot_type = "density",
    .line_size = 0.5,
    .geom_point = FALSE,
    .point_size = 1,
    .geom_rug = FALSE,
    .geom_smooth = FALSE,
    .geom_jitter = FALSE,
    .interactive = FALSE
)
```

# Arguments

.data	The data passed in from a the function tidy_multi_dist()
.plot_type	This is a quoted string like 'density'
.line_size	The size param ggplot
.geom_point	A Boolean value of TREU/FALSE, FALSE is the default. TRUE will return a plot with ggplot2::ggeom_point()
.point_size	The point size param for ggplot
.geom_rug	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of $ggplot2::geom_rug()$
.geom_smooth	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_smooth() The aes parameter of group is set to FALSE. This ensures a single smoothing band returned with SE also set to FALSE. Color is set to 'black' and linetype is 'dashed'.

```
.geom_jitter A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_jitter()

.interactive A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return an interactive plotly plot.
```

## **Details**

This function will spit out one of the following plots:

- density
- quantile
- probability
- qq
- mcmc

## Value

A ggplot or a plotly plot.

## Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Autoplot: bootstrap_stat_plot(), tidy_autoplot(), tidy_four_autoplot(), tidy_multi_dist_autoplot(), tidy_random_walk_autoplot()
```

# **Examples**

```
combined_tbl <- tidy_combine_distributions(
  tidy_normal(),
  tidy_gamma(),
  tidy_beta()
)

combined_tbl

combined_tbl |>
  tidy_combined_autoplot()

combined_tbl |>
  tidy_combined_autoplot(.plot_type = "qq")
```

```
tidy_combine_distributions
```

Combine Multiple Tidy Distributions of Different Types

## **Description**

This allows a user to specify any n number of tidy\_ distributions that can be combined into a single tibble. This is the preferred method for combining multiple distributions of different types, for example a Gaussian distribution and a Beta distribution.

This generates a single tibble with an added column of dist\_type that will give the distribution family name and its associated parameters.

# Usage

```
tidy_combine_distributions(...)
```

# Arguments

... The ... is where you can place your different distributions

## **Details**

Allows a user to generate a tibble of different tidy\_distributions

## Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Multiple Distribution: tidy_multi_single_dist()
```

# **Examples**

```
tn <- tidy_normal()
tb <- tidy_beta()
tc <- tidy_cauchy()

tidy_combine_distributions(tn, tb, tc)
## OR

tidy_combine_distributions(</pre>
```

```
tidy_normal(),
tidy_beta(),
tidy_cauchy(),
tidy_logistic()
```

tidy\_distribution\_comparison

Compare Empirical Data to Distributions

## Description

Compare some empirical data set against different distributions to help find the distribution that could be the best fit.

## Usage

```
tidy_distribution_comparison(
   .x,
   .distribution_type = "continuous",
   .round_to_place = 3
)
```

# **Arguments**

. x The data set being passed to the function

.distribution\_type

What kind of data is it, can be one of continuous or discrete

.round\_to\_place

How many decimal places should the parameter estimates be rounded off to for distibution construction. The default is 3

### **Details**

The purpose of this function is to take some data set provided and to try to find a distribution that may fit the best. A parameter of .distribution\_type must be set to either continuous or discrete in order for this the function to try the appropriate types of distributions.

The following distributions are used:

Continuous:

- tidy\_beta
- tidy\_cauchy
- tidy\_chisquare
- tidy\_exponential
- tidy\_gamma

- tidy\_logistic
- tidy\_lognormal
- tidy\_normal
- tidy\_pareto
- tidy\_uniform
- tidy\_weibull

## Discrete:

- tidy\_binomial
- tidy\_geometric
- tidy\_hypergeometric
- tidy\_poisson

The function itself returns a list output of tibbles. Here are the tibbles that are returned:

- · comparison\_tbl
- deviance\_tbl
- total\_deviance\_tbl
- aic\_tbl
- kolmogorov\_smirnov\_tbl
- multi\_metric\_tbl

The comparison\_tbl is a long tibble that lists the values of the density function against the given data.

The deviance\_tbl and the total\_deviance\_tbl just give the simple difference from the actual density to the estimated density for the given estimated distribution.

The aic\_tbl will provide the AIC for liklehood of the distribution.

The kolmogorov\_smirnov\_tbl for now provides a two.sided estimate of the ks.test of the estimated density against the empirical.

The multi\_metric\_tbl will summarise all of these metrics into a single tibble.

## Value

An invisible list object. A tibble is printed.

## Author(s)

Steven P. Sanderson II, MPH

## **Examples**

## **Description**

This function returns a summary statistics tibble. It will use the y column from the tidy\_ distribution function.

## Usage

```
tidy_distribution_summary_tbl(.data, ...)
```

## **Arguments**

```
.data The data that is going to be passed from a a tidy_ distribution function.... This is the grouping variable that gets passed to dplyr::group_by() and dplyr::select().
```

## **Details**

This function takes in a tidy\_distribution table and will return a tibble of the following information:

- sim\_number
- mean\_val
- median\_val
- std\_val
- min\_val
- max\_val
- skewness
- kurtosis
- range
- igr
- variance
- ci\_hi
- ci\_lo

The kurtosis and skewness come from the package healthyR.ai

46 tidy\_empirical

## Value

A summary stats tibble

## Author(s)

Steven P. Sanderson II, MPH

## **Examples**

```
library(dplyr)

tn <- tidy_normal(.num_sims = 5)
tb <- tidy_beta(.num_sims = 5)

tidy_distribution_summary_tbl(tn)
tidy_distribution_summary_tbl(tn, sim_number)

data_tbl <- tidy_combine_distributions(tn, tb)

tidy_distribution_summary_tbl(data_tbl)
tidy_distribution_summary_tbl(data_tbl, dist_type)</pre>
```

tidy\_empirical

Tidy Empirical

# Description

This function takes in a single argument of .x a vector and will return a tibble of information similar to the tidy\_ distribution functions. The y column is set equal to dy from the density function.

## Usage

```
tidy_empirical(.x, .num_sims = 1, .distribution_type = "continuous")
```

## **Arguments**

.x A vector of numbers

.num\_sims How many simulations should be run, defaults to 1.

.distribution\_type

A string of either "continuous" or "discrete". The function will default to "continuous"

## **Details**

This function takes in a single argument of .x a vector

tidy\_exponential 47

### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

## **Examples**

```
x <- mtcars$mpg
tidy_empirical(.x = x, .distribution_type = "continuous")
tidy_empirical(.x = x, .num_sims = 10, .distribution_type = "continuous")</pre>
```

tidy\_exponential

Tidy Randomly Generated Exponential Distribution Tibble

# **Description**

This function will generate n random points from a exponential distribution with a user provided, .rate, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_p$  and q data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

# Usage

```
tidy_exponential(.n = 50, .rate = 1, .num_sims = 1, .return_tibble = TRUE)
```

## **Arguments**

. n The number of randomly generated points you want.

.rate A vector of rates

. num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

48 tidy\_f

### **Details**

This function uses the underlying stats::rexp(), and its underlying p, d, and q functions. For more information please see stats::rexp()

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II. MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3667.htm
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()
```

Other Exponential: tidy\_inverse\_exponential(), util\_exponential\_param\_estimate(), util\_exponential\_stats\_

## **Examples**

tidy\_exponential()

tidy\_f

Tidy Randomly Generated F Distribution Tibble

## **Description**

This function will generate n random points from a rf distribution with a user provided, df1,df2, and ncp, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

tidy\_f

## Usage

```
tidy_f(
    .n = 50,
    .df1 = 1,
    .df2 = 1,
    .ncp = 0,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.
.df1 Degrees of freedom, Inf is allowed.
.df2 Degrees of freedom, Inf is allowed.
.ncp Non-centrality parameter.
.num\_sims The number of randomly generated simulations you want.
.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

This function uses the underlying stats::rf(), and its underlying p, d, and q functions. For more information please see stats::rf()

### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3665.htm
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(),
tidy_exponential(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(),
tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(),
tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(),
tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(),
tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()
Other F Distribution: util_f_param_estimate(), util_f_stats_tbl()
```

## **Examples**

```
tidy_f()
```

50 tidy\_four\_autoplot

tidy\_four\_autoplot

Automatic Plot of Density Data

## **Description**

This is an auto plotting function that will take in a tidy\_ distribution function and a few arguments, one being the plot type, which is a quoted string of one of the following:

- density
- quantile
- probablity
- qq
- mcmc

If the number of simulations exceeds 9 then the legend will not print. The plot subtitle is put together by the attributes of the table passed to the function.

# Usage

```
tidy_four_autoplot(
    .data,
    .line_size = 0.5,
    .geom_point = FALSE,
    .point_size = 1,
    .geom_rug = FALSE,
    .geom_smooth = FALSE,
    .geom_jitter = FALSE,
    .interactive = FALSE
)
```

## **Arguments**

.data	The data passed in from a tidy_distribution function like tidy_normal()
.line_size	The size param ggplot
.geom_point	A Boolean value of TREU/FALSE, FALSE is the default. TRUE will return a plot with ggplot2::ggeom_point()
.point_size	The point size param for ggplot
.geom_rug	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the
	<pre>use of ggplot2::geom_rug()</pre>
.geom_smooth	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the
	use of ggplot2::geom_smooth() The aes parameter of group is set to FALSE.
	This ensures a single smoothing band returned with SE also set to FALSE. Color
	is set to 'black' and linetype is 'dashed'.
.geom_jitter	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the
	<pre>use of ggplot2::geom_jitter()</pre>
.interactive	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return an
	interactive plotly plot.

tidy\_gamma 51

### **Details**

This function will spit out one of the following plots:

- density
- quantile
- probability
- qq
- mcmc

## Value

A ggplot or a plotly plot.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Autoplot: bootstrap_stat_plot(), tidy_autoplot(), tidy_combined_autoplot(), tidy_multi_dist_autoplot() tidy_random_walk_autoplot()
```

### **Examples**

```
tidy_normal(.num_sims = 5) |>
  tidy_four_autoplot()
```

tidy\_gamma

Tidy Randomly Generated Gamma Distribution Tibble

## Description

This function will generate n random points from a gamma distribution with a user provided, .shape, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

52 tidy\_gamma

## Usage

```
tidy_gamma(
    .n = 50,
    .shape = 1,
    .scale = 0.3,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

# **Arguments**

.n The number of randomly generated points you want.

. shape This is strictly 0 to infinity.

scale The standard deviation of the randomly generated data. This is strictly from 0

to infinity.

. num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

## **Details**

This function uses the underlying stats::rgamma(), and its underlying p, d, and q functions. For more information please see stats::rgamma()

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://www.statology.org/fit-gamma-distribution-to-dataset-in-r/
https://en.wikipedia.org/wiki/Gamma_distribution

Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Gamma: tidy_inverse_gamma(), util_gamma_param_estimate(), util_gamma_stats_tbl()
```

# Examples

```
tidy_gamma()
```

tidy\_generalized\_beta Tidy Randomly Generated Generalized Beta Distribution Tibble

## **Description**

This function will generate n random points from a generalized beta distribution with a user provided, .shape1, .shape2, .shape3, .rate, and/or .sclae, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_generalized_beta(
    .n = 50,
    .shape1 = 1,
    .shape2 = 1,
    .shape3 = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n	The number of randomly generated points you want.
.shape1	A non-negative parameter of the Beta distribution.
.shape2	A non-negative parameter of the Beta distribution.
.shape3	A non-negative parameter of the Beta distribution.
.rate	An alternative way to specify the .scale parameter.
.scale	Must be strictly positive.
.num_sims	The number of randomly generated simulations you want.
.return_tibble	A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

This function uses the underlying stats::rbeta(), and its underlying p, d, and q functions. For more information please see stats::rbeta()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://statisticsglobe.com/beta-distribution-in-r-dbeta-pbeta-qbeta-rbeta
https://en.wikipedia.org/wiki/Beta_distribution
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(),
tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_pareto(), tidy_geometric(),
tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(),
tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(),
tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(),
tidy_weibull(), tidy_zero_truncated_geometric()
Other Beta: tidy_beta(), util_beta_param_estimate(), util_beta_stats_tbl()
```

## **Examples**

tidy\_generalized\_beta()

tidy\_generalized\_pareto

Tidy Randomly Generated Generalized Pareto Distribution Tibble

## **Description**

This function will generate n random points from a generalized Pareto distribution with a user provided, .shape1, .shape2, .rate or .scale and number of #' random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.

- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_generalized_pareto(
    .n = 50,
    .shape1 = 1,
    .shape2 = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

. shape1 Must be positive.

. shape2 Must be positive.

.rate An alternative way to specify the .scale argument

. scale Must be positive.

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying actuar::rgenpareto(), and its underlying p, d, and q functions. For more information please see actuar::rgenpareto()

## Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

56 tidy\_geometric

### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html \\
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Pareto: tidy_inverse_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto_param_estimate(), util_pareto_stats_tbl()
```

## **Examples**

```
tidy_generalized_pareto()
```

tidy\_geometric

Tidy Randomly Generated Geometric Distribution Tibble

# Description

This function will generate n random points from a geometric distribution with a user provided, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_geometric(.n = 50, .prob = 1, .num_sims = 1, .return_tibble = TRUE)
```

tidy\_hypergeometric 57

## **Arguments**

.n The number of randomly generated points you want.
 .prob A probability of success in each trial 0 < prob <= 1.</li>
 .num\_sims The number of randomly generated simulations you want.
 .return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

This function uses the underlying stats::rgeom(), and its underlying p, d, and q functions. For more information please see stats::rgeom()

### Value

A tibble of randomly generated data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://en.wikipedia.org/wiki/Geometric_distribution
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Geometric: tidy_zero_truncated_geometric(), util_geometric_param_estimate(), util_geometric_stats_tbl()
```

## **Examples**

tidy\_geometric()

58 tidy\_hypergeometric

## **Description**

This function will generate n random points from a hypergeometric distribution with a user provided, m,nn, and k, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_hypergeometric(
    .n = 50,
    .m = 0,
    .nn = 0,
    .k = 0,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

.m The number of white balls in the urn.nn The number of black balls in the urn.k The number of balls drawn fro the urn.

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## **Details**

This function uses the underlying stats::rhyper(), and its underlying p, d, and q functions. For more information please see stats::rhyper()

## Value

A tibble of randomly generated data.

tidy\_inverse\_burr 59

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://en.wikipedia.org/wiki/Hypergeometric_distribution
```

```
Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), tidy_zero_truncated_poisson()
Other Hypergeometric: util_hypergeometric_param_estimate(), util_hypergeometric_stats_tbl()
```

## **Examples**

```
tidy_hypergeometric()
```

tidy\_inverse\_burr

Tidy Randomly Generated Inverse Burr Distribution Tibble

### **Description**

This function will generate n random points from an Inverse Burr distribution with a user provided, .shape1, .shape2, .scale, .rate, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_inverse_burr(
    .n = 50,
    .shape1 = 1,
    .shape2 = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

60 tidy\_inverse\_burr

## **Arguments**

. n The number of randomly generated points you want.

. shape1 Must be strictly positive.. shape2 Must be strictly positive.

.rate An alternative way to specify the .scale.

. scale Must be strictly positive.

.num\_sims The number of randomly generated simulations you want.

 $.\,return\_tibble\ A\ logical\ value\ indicating\ whether\ to\ return\ the\ result\ as\ a\ tibble.\ Default\ is$ 

TRUE.

#### Details

This function uses the underlying actuar::rinvburr(), and its underlying p, d, and q functions. For more information please see actuar::rinvburr()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
\verb|https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html| \\
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Burr: tidy_burr(), util_burr_param_estimate(), util_burr_stats_tbl()

Other Inverse Distribution: tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull()
```

## **Examples**

```
tidy_inverse_burr()
```

```
tidy_inverse_exponential
```

Tidy Randomly Generated Inverse Exponential Distribution Tibble

## **Description**

This function will generate n random points from an inverse exponential distribution with a user provided, .rate or .scale and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_inverse_exponential(
  .n = 50,
  .rate = 1,
  .scale = 1/.rate,
  .num_sims = 1,
  .return_tibble = TRUE
)
```

# **Arguments**

```
The number of randomly generated points you want.
.n
                  An alternative way to specify the .scale
.rate
.scale
                  Must be strictly positive.
```

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

This function uses the underlying actuar::rinvexp(), and its underlying p, d, and q functions. For more information please see actuar::rinvexp()

62 tidy\_inverse\_gamma

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Exponential: tidy_exponential(), util_exponential_param_estimate(), util_exponential_stats_tbl()

Other Inverse Distribution: tidy_inverse_burr(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull()
```

## **Examples**

tidy\_inverse\_exponential()

tidy\_inverse\_gamma

Tidy Randomly Generated Inverse Gamma Distribution Tibble

## **Description**

This function will generate n random points from an inverse gamma distribution with a user provided, .shape, .rate, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

tidy\_inverse\_gamma 63

## Usage

```
tidy_inverse_gamma(
    .n = 50,
    .shape = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

. shape Must be strictly positive.

.rate An alternative way to specify the .scale

. scale Must be strictly positive.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## **Details**

This function uses the underlying actuar::rinvgamma(), and its underlying p, d, and q functions. For more information please see actuar::rinvgamma()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_pareto(), tidy_pareto(), tidy_pareto(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Gamma: tidy_gamma(), util_gamma_param_estimate(), util_gamma_stats_tbl()

Other Inverse Distribution: tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull()
```

64 tidy\_inverse\_normal

## **Examples**

```
tidy_inverse_gamma()
```

tidy\_inverse\_normal

Tidy Randomly Generated Inverse Gaussian Distribution Tibble

## **Description**

This function will generate n random points from an Inverse Gaussian distribution with a user provided, .mean, .shape, .dispersionThe function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_inverse_normal(
    .n = 50,
    .mean = 1,
    .shape = 1,
    .dispersion = 1/.shape,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of randomly generated points you want.

.mean Must be strictly positive..shape Must be strictly positive.

. dispersion An alternative way to specify the . shape.

. num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

tidy\_inverse\_pareto 65

### **Details**

This function uses the underlying actuar::rinvgauss(). For more information please see rinvgauss()

## Value

A tibble of randomly generated data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Gaussian: tidy_normal(), util_normal_param_estimate(), util_normal_stats_tbl()

Other Inverse Distribution: tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_pareto(), tidy_inverse_weibull()
```

## **Examples**

tidy\_inverse\_normal()

tidy\_inverse\_pareto

Tidy Randomly Generated Inverse Pareto Distribution Tibble

## **Description**

This function will generate n random points from an inverse pareto distribution with a user provided, .shape, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

66 tidy\_inverse\_pareto

## Usage

```
tidy_inverse_pareto(
    .n = 50,
    .shape = 1,
    .scale = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of randomly generated points you want.

. shape Must be positive.. scale Must be positive.

. num\_sims The number of randomly generated simulations you want.

 $. \, return\_tibble \ \ A \ logical \ value \ indicating \ whether \ to \ return \ the \ result \ as \ a \ tibble. \ \, Default \ is$ 

TRUE.

### **Details**

This function uses the underlying actuar::rinvpareto(), and its underlying p, d, and q functions. For more information please see actuar::rinvpareto()

## Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Pareto: tidy_generalized_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto_param_estimate(), util_pareto_stats_tbl()

Other Inverse Distribution: tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_weibull()
```

tidy\_inverse\_weibull 67

## **Examples**

```
tidy_inverse_pareto()
```

tidy\_inverse\_weibull Tidy Randomly Generated Inverse Weibull Distribution Tibble

# Description

This function will generate n random points from a weibull distribution with a user provided, .shape, .scale, .rate, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_inverse_weibull(
    .n = 50,
    .shape = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n	The number of randomly generated points you want.
. shape	Must be strictly positive.
.rate	An alternative way to specify the .scale.
.scale	Must be strictly positive.
.num_sims	The number of randomly generated simulations you want.
.return_tibble	A logical value indicating whether to return the result as a tibble. Default is TRUE.

68 tidy\_kurtosis\_vec

### **Details**

This function uses the underlying actuar::rinvweibull(), and its underlying p, d, and q functions. For more information please see actuar::rinvweibull()

#### Value

A tibble of randomly generated data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Weibull: tidy_weibull(), util_weibull_param_estimate(), util_weibull_stats_tbl()

Other Inverse Distribution: tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto()
```

## **Examples**

```
tidy_inverse_weibull()
```

tidy\_kurtosis\_vec

Compute Kurtosis of a Vector

## **Description**

This function takes in a vector as it's input and will return the kurtosis of that vector. The length of this vector must be at least four numbers. The kurtosis explains the sharpness of the peak of a distribution of data.

```
((1/n) * sum(x - mu)^4) / ((()1/n) * sum(x - mu)^2)^2
```

## Usage

```
tidy_kurtosis_vec(.x)
```

## **Arguments**

. x A numeric vector of length four or more.

tidy\_logistic 69

### **Details**

A function to return the kurtosis of a vector.

### Value

The kurtosis of a vector

### Author(s)

Steven P. Sanderson II. MPH

#### See Also

```
https://en.wikipedia.org/wiki/Kurtosis
```

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
Other Statistic: ci_hi(), ci_lo(), tidy_range_statistic(), tidy_skewness_vec(), tidy_stat_tbl()
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_scale_zero_one_vec(), tidy_skewness_vec()
```

## **Examples**

```
tidy_kurtosis_vec(rnorm(100, 3, 2))
```

tidy\_logistic

Tidy Randomly Generated Logistic Distribution Tibble

## **Description**

This function will generate n random points from a logistic distribution with a user provided, .location, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

70 tidy\_logistic

## Usage

```
tidy_logistic(
    .n = 50,
    .location = 0,
    .scale = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of randomly generated points you want.

. location The location parameter . scale The scale parameter

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

### **Details**

This function uses the underlying stats::rlogis(), and its underlying p, d, and q functions. For more information please see stats::rlogis()

## Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://en.wikipedia.org/wiki/Logistic_distribution
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Logistic: tidy_paralogistic(), util_logistic_param_estimate(), util_logistic_stats_tbl()
```

## **Examples**

```
tidy_logistic()
```

tidy\_lognormal 71

tidy\_lognormal

Tidy Randomly Generated Lognormal Distribution Tibble

## **Description**

This function will generate n random points from a lognormal distribution with a user provided, .meanlog, .sdlog, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_lognormal(
    .n = 50,
    .meanlog = 0,
    .sdlog = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

n The number of randomly generated points you want.
 meanlog Mean of the distribution on the log scale with default 0
 sdlog Standard deviation of the distribution on the log scale with default 1

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

This function uses the underlying stats::rlnorm(), and its underlying p, d, and q functions. For more information please see stats::rlnorm()

### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3669.htm

Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(),
tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(),
tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(),
tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(),
tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_triangular(),
tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()
```

Other Lognormal: util\_lognormal\_param\_estimate(), util\_lognormal\_stats\_tbl()

# **Examples**

```
tidy_lognormal()
```

tidy\_mcmc\_sampling Tidy MCMC Sampling

## Description

This function performs Markov Chain Monte Carlo (MCMC) sampling on the input data and returns tidy data and a plot representing the results.

## Usage

```
tidy_mcmc_sampling(.x, .fns = "mean", .cum_fns = "cmean", .num_sims = 2000)
```

## **Arguments**

. x The data vector for MCMC sampling.

. fns The function(s) to apply to each MCMC sample. Default is "mean".

.cum\_fns The function(s) to apply to the cumulative MCMC samples. Default is "cmean".

.num\_sims The number of simulations. Default is 2000.

tidy\_mixture\_density 73

#### **Details**

Perform MCMC sampling and return tidy data and a plot.

The function takes a data vector as input and performs MCMC sampling with the specified number of simulations. It applies user-defined functions to each MCMC sample and to the cumulative MCMC samples. The resulting data is formatted in a tidy format, suitable for further analysis. Additionally, a plot is generated to visualize the MCMC samples and cumulative statistics.

#### Value

A list containing tidy data and a plot.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto1_aic(), util_pareto1_aic(), util_pareto1_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

#### **Examples**

```
# Generate MCMC samples
set.seed(123)
data <- rnorm(100)
result <- tidy_mcmc_sampling(data, "median", "cmedian", 500)
result</pre>
```

```
tidy_mixture_density Tidy Mixture Data
```

## Description

Create mixture model data and resulting density and line plots.

## Usage

```
tidy_mixture_density(...)
```

#### **Arguments**

The random data you want to pass. Example rnorm(50,0,1) or something like tidy\_normal(.mean = 5, .sd = 1)

### **Details**

This function allows you to make mixture model data. It allows you to produce density data and plots for data that is not strictly of one family or of one single type of distribution with a given set of parameters.

For example this function will allow you to mix say tidy\_normal(.mean = 0, .sd = 1) and tidy\_normal(.mean = 5, .sd = 1) or you can mix and match distributions.

The output is a list object with three components.

- 1. Data
- input\_data (The random data passed)
- dist\_tbl (A tibble of the passed random data)
- density\_tbl (A tibble of the x and y data from stats::density())
- 1. Plots
- line\_plot Plots the dist\_tbl
- dens\_plot Plots the density\_tbl
- 1. Input Functions
- input\_fns A list of the functions and their parameters passed to the function itself

### Value

A list object

## Author(s)

Steven P. Sanderson II, MPH

```
output <- tidy_mixture_density(rnorm(100, 0, 1), tidy_normal(.mean = 5, .sd = 1))
output$data
output$plots
output$input_fns</pre>
```

```
tidy_multi_dist_autoplot
```

Automatic Plot of Multi Dist Data

## **Description**

This is an auto plotting function that will take in a tidy\_ distribution function and a few arguments, one being the plot type, which is a quoted string of one of the following:

- density
- quantile
- probablity
- qq
- mcmc

If the number of simulations exceeds 9 then the legend will not print. The plot subtitle is put together by the attributes of the table passed to the function.

## Usage

```
tidy_multi_dist_autoplot(
    .data,
    .plot_type = "density",
    .line_size = 0.5,
    .geom_point = FALSE,
    .point_size = 1,
    .geom_rug = FALSE,
    .geom_smooth = FALSE,
    .geom_jitter = FALSE,
    .interactive = FALSE
)
```

### **Arguments**

.data	The data passed in from a the function tidy_multi_dist()
.plot_type	This is a quoted string like 'density'
.line_size	The size param ggplot
.geom_point	A Boolean value of TREU/FALSE, FALSE is the default. TRUE will return a plot with ggplot2::ggeom_point()
.point_size	The point size param for ggplot
.geom_rug	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of $ggplot2::geom_rug()$
.geom_smooth	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_smooth() The aes parameter of group is set to FALSE. This ensures a single smoothing band returned with SE also set to FALSE. Color is set to 'black' and linetype is 'dashed'.

.geom\_jitter A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom\_jitter()

.interactive A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return an interactive plotly plot.

#### **Details**

This function will spit out one of the following plots:

- density
- quantile
- probability
- qq
- mcmc

### Value

A ggplot or a plotly plot.

# Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Autoplot: bootstrap_stat_plot(), tidy_autoplot(), tidy_combined_autoplot(), tidy_four_autoplot(), tidy_random_walk_autoplot()
```

tidy\_multi\_single\_dist

```
tidy_multi_single_dist
```

Generate Multiple Tidy Distributions of a single type

77

## **Description**

Generate multiple distributions of data from the same tidy\_ distribution function.

### Usage

```
tidy_multi_single_dist(.tidy_dist = NULL, .param_list = list())
```

## **Arguments**

```
.tidy_dist The type of tidy_ distribution that you want to run. You can only choose one.
.param_list This must be a list() object of the parameters that you want to pass through to
```

the TidyDensity tidy\_ distribution function.

### **Details**

Generate multiple distributions of data from the same tidy\_ distribution function. This allows you to simulate multiple distributions of the same family in order to view how shapes change with parameter changes. You can then visualize the differences however you choose.

#### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

Other Multiple Distribution: tidy\_combine\_distributions()

tidy\_negative\_binomial

Tidy Randomly Generated Negative Binomial Distribution Tibble

# Description

This function will generate n random points from a negative binomial distribution with a user provided, .size, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_negative_binomial(
    .n = 50,
    .size = 1,
    .prob = 0.1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

#### **Arguments**

.n	The number of	of randomly	generated 1	points you want.

. size target for number of successful trials, or dispersion parameter (the shape param-

eter of the gamma mixing distribution). Must be strictly positive, need not be

integer.

. prob Probability of success on each trial where 0 < .prob <= 1.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

#### **Details**

This function uses the underlying stats::rnbinom(), and its underlying p, d, and q functions. For more information please see stats::rnbinom()

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

```
Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_hypergeometric(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), tidy_zero_truncated_poisson()
```

```
Other Binomial: tidy_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial() util_binomial_param_estimate(), util_binomial_stats_tbl(), util_negative_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
```

```
tidy_negative_binomial()
```

80 tidy\_normal

tidy\_normal

Tidy Randomly Generated Gaussian Distribution Tibble

### Description

This function will generate n random points from a Gaussian distribution with a user provided, .mean, .sd - standard deviation and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the dnorm, pnorm and qnorm data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

#### **Usage**

```
tidy_normal(.n = 50, .mean = 0, .sd = 1, .num_sims = 1, .return_tibble = TRUE)
```

### Arguments

.n The number of randomly generated points you want.

. mean The mean of the randomly generated data.

. sd The standard deviation of the randomly generated data.
. num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## **Details**

This function uses the underlying stats::rnorm(), stats::pnorm(), and stats::qnorm() functions to generate data from the given parameters. For more information please see stats::rnorm()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

tidy\_paralogistic 81

#### See Also

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Gaussian: tidy_inverse_normal(), util_normal_param_estimate(), util_normal_stats_tbl()
```

#### **Examples**

```
tidy_normal()
```

tidy\_paralogistic

Tidy Randomly Generated Paralogistic Distribution Tibble

### **Description**

This function will generate n random points from a paralogistic distribution with a user provided, .shape, .rate, .scale and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_paralogistic(
    .n = 50,
    .shape = 1,
    .rate = 1,
    .scale = 1/.rate,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

82 tidy\_paralogistic

### **Arguments**

	.n	The number	of randomly	generated	points you want.
--	----	------------	-------------	-----------	------------------

. shape Must be strictly positive.

.rate An alternative way to specify the .scale

. scale Must be strictly positive.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

#### **Details**

This function uses the underlying actuar::rparalogis(), and its underlying p, d, and q functions. For more information please see actuar::rparalogis()

#### Value

A tibble of randomly generated data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://en.wikipedia.org/wiki/Logistic_distribution
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Logistic: tidy_logistic(), util_logistic_param_estimate(), util_logistic_stats_tbl()
```

```
tidy_paralogistic()
```

tidy\_pareto 83

tidy\_pareto

Tidy Randomly Generated Pareto Distribution Tibble

### **Description**

This function will generate n random points from a pareto distribution with a user provided, . shape, . scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- $\bullet$  q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_pareto(
    .n = 50,
    .shape = 10,
    .scale = 0.1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

. shape Must be positive.. scale Must be positive.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## Details

This function uses the underlying actuar::rpareto(), and its underlying p, d, and q functions. For more information please see actuar::rpareto()

84 tidy\_pareto1

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto_param_estimate(), util_pareto_stats_tbl()
```

## **Examples**

tidy\_pareto()

tidy\_pareto1

Tidy Randomly Generated Pareto Single Parameter Distribution Tibble

## **Description**

This function will generate n random points from a single parameter pareto distribution with a user provided, .shape, .min, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

tidy\_pareto1 85

### Usage

```
tidy_pareto1(
    .n = 50,
    .shape = 1,
    .min = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## Arguments

. n The number of randomly generated points you want.

. shape Must be positive.

.min The lower bound of the support of the distribution.

. num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying actuar::rpareto1(), and its underlying p, d, and q functions. For more information please see actuar::rpareto1()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(),
tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(),
tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(),
tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(),
tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_t(), tidy_triangular(),
tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()
Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto(), util_pareto1_aic(),
util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto_param_estimate(),
util_pareto_stats_tbl()
```

```
tidy_pareto1()
```

86 tidy\_poisson

tidy\_poisson

Tidy Randomly Generated Poisson Distribution Tibble

### **Description**

This function will generate n random points from a Poisson distribution with a user provided, .lambda, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

#### Usage

```
tidy_poisson(.n = 50, .lambda = 1, .num_sims = 1, .return_tibble = TRUE)
```

### **Arguments**

.n The number of randomly generated points you want.

. lambda A vector of non-negative means.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying stats::rpois(), and its underlying p, d, and q functions. For more information please see stats::rpois()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

tidy\_random\_walk 87

#### See Also

```
https://r-coder.com/poisson-distribution-r/
https://en.wikipedia.org/wiki/Poisson_distribution
Other Poisson: tidy_zero_truncated_poisson(), util_poisson_param_estimate(), util_poisson_stats_tbl(),
util_zero_truncated_poisson_param_estimate(), util_zero_truncated_poisson_stats_tbl()
Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_hypergeometric(),
tidy_negative_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(),
tidy_zero_truncated_poisson()
```

### **Examples**

```
tidy_poisson()
```

tidy\_random\_walk

Tidy Random Walk

### **Description**

Takes in the data from a tidy\_ distribution function and applies a random walk calculation of either cum\_prod or cum\_sum to y.

### Usage

```
tidy_random_walk(
   .data,
   .initial_value = 0,
   .sample = FALSE,
   .replace = FALSE,
   .value_type = "cum_prod"
)
```

## **Arguments**

.initial\_value The default is 0, this can be set to whatever you want.

. sample This is a boolean value TRUE/FALSE. The default is FALSE. If set to TRUE

then the y value from the tidy\_ distribution function is sampled.

replace This is a boolean value TRUE/FALSE. The default is FALSE. If set to TRUE

AND . sample is set to TRUE then the replace parameter of the sample function

will be set to TRUE.

.value\_type This can take one of three different values for now. These are the following:

• "cum\_prod" - This will take the cumprod of y

• "cum\_sum" - This will take the cumsum of y

#### **Details**

Monte Carlo simulations were first formally designed in the 1940's while developing nuclear weapons, and since have been heavily used in various fields to use randomness solve problems that are potentially deterministic in nature. In finance, Monte Carlo simulations can be a useful tool to give a sense of how assets with certain characteristics might behave in the future. While there are more complex and sophisticated financial forecasting methods such as ARIMA (Auto-Regressive Integrated Moving Average) and GARCH (Generalised Auto-Regressive Conditional Heteroskedasticity) which attempt to model not only the randomness but underlying macro factors such as seasonality and volatility clustering, Monte Carlo random walks work surprisingly well in illustrating market volatility as long as the results are not taken too seriously.

#### Value

An ungrouped tibble.

### Author(s)

Steven P. Sanderson II, MPH

### **Examples**

```
tidy_normal(.sd = .1, .num_sims = 25) %>%
  tidy_random_walk()
```

```
tidy_random_walk_autoplot
```

Automatic Plot of Random Walk Data

# Description

This is an auto-plotting function that will take in a tidy\_ distribution function and a few arguments with regard to the output of the visualization.

If the number of simulations exceeds 9 then the legend will not print. The plot subtitle is put together by the attributes of the table passed to the function.

### Usage

```
tidy_random_walk_autoplot(
    .data,
    .line_size = 0.5,
    .geom_rug = FALSE,
    .geom_smooth = FALSE,
    .interactive = FALSE
)
```

### **Arguments**

.data	The data passed in from a tidy_distribution function like tidy_normal()
.line_size	The size param ggplot
.geom_rug	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_rug()
.geom_smooth	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return the use of ggplot2::geom_smooth() The aes parameter of group is set to FALSE. This ensures a single smoothing band returned with SE also set to FALSE. Color is set to 'black' and linetype is 'dashed'.
.interactive	A Boolean value of TRUE/FALSE, FALSE is the default. TRUE will return an interactive plotly plot.

## **Details**

This function will produce a simple random walk plot from a tidy\_ distribution function.

# Value

A ggplot or a plotly plot.

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other\ Autoplot:\ bootstrap\_stat\_plot(),\ tidy\_autoplot(),\ tidy\_combined\_autoplot(),\ tidy\_four\_autoplot(),\ tidy\_multi\_dist\_autoplot()
```

```
tidy_normal(.sd = .1, .num_sims = 5) |>
  tidy_random_walk(.value_type = "cum_sum") |>
  tidy_random_walk_autoplot()

tidy_normal(.sd = .1, .num_sims = 20) |>
  tidy_random_walk(.value_type = "cum_sum", .sample = TRUE, .replace = TRUE) |>
  tidy_random_walk_autoplot()
```

90 tidy\_range\_statistic

# Description

Takes in a numeric vector and returns back the range of that vector

## Usage

```
tidy_range_statistic(.x)
```

## **Arguments**

.x A numeric vector

#### **Details**

Takes in a numeric vector and returns the range of that vector using the diff and range functions.

# Value

A single number, the range statistic

### Author(s)

Steven P. Sandeson II, MPH

## See Also

```
Other Statistic: ci_hi(), ci_lo(), tidy_kurtosis_vec(), tidy_skewness_vec(), tidy_stat_tbl()
```

```
tidy_range_statistic(seq(1:10))
```

```
tidy_scale_zero_one_vec
```

Vector Function Scale to Zero and One

### **Description**

Takes a numeric vector and will return a vector that has been scaled from [0,1]

## Usage

```
tidy_scale_zero_one_vec(.x)
```

#### **Arguments**

. x

A numeric vector to be scaled from [0,1] inclusive.

#### **Details**

Takes a numeric vector and will return a vector that has been scaled from [0,1] The input vector must be numeric. The computation is fairly straightforward. This may be helpful when trying to compare the distributions of data where a distribution like beta which requires data to be between 0 and 1

$$y[h] = (x - min(x))/(max(x) - min(x))$$

## Value

A numeric vector

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(), cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_skewness_vec()
```

```
vec_1 <- rnorm(100, 2, 1)
vec_2 <- tidy_scale_zero_one_vec(vec_1)

dens_1 <- density(vec_1)
dens_2 <- density(vec_2)
max_x <- max(dens_1$x, dens_2$x)
max_y <- max(dens_1$y, dens_2$y)
plot(dens_1,</pre>
```

92 tidy\_skewness\_vec

```
asp = max_y / max_x, main = "Density vec_1 (Red) and vec_2 (Blue)",
col = "red", xlab = "", ylab = "Density of Vec 1 and Vec 2"
)
lines(dens_2, col = "blue")
```

tidy\_skewness\_vec

Compute Skewness of a Vector

## **Description**

This function takes in a vector as it's input and will return the skewness of that vector. The length of this vector must be at least four numbers. The skewness explains the 'tailedness' of the distribution of data.

```
((1/n) * sum(x - mu)^3) / ((()1/n) * sum(x - mu)^2)^(3/2)
```

### Usage

```
tidy_skewness_vec(.x)
```

### **Arguments**

A numeric vector of length four or more.

#### **Details**

A function to return the skewness of a vector.

## Value

The skewness of a vector

#### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://en.wikipedia.org/wiki/Skewness
```

```
Other Statistic: ci_hi(), ci_lo(), tidy_kurtosis_vec(), tidy_range_statistic(), tidy_stat_tbl()
Other Vector Function: bootstrap_p_vec(), bootstrap_q_vec(), cgmean(), chmean(), ckurtosis(),
cmean(), cmedian(), csd(), cskewness(), cvar(), tidy_kurtosis_vec(), tidy_scale_zero_one_vec()
```

```
tidy_skewness_vec(rnorm(100, 3, 2))
```

tidy\_stat\_tbl 93

tidy\_stat\_tbl

Tidy Stats of Tidy Distribution

## **Description**

A function to return the stat function values of a given tidy\_ distribution output.

### Usage

```
tidy_stat_tbl(
   .data,
   .x = y,
   .fns,
   .return_type = "vector",
   .use_data_table = FALSE,
   ...
)
```

### **Arguments**

. data The input data coming from a tidy\_ distribution function.

.x The default is y but can be one of the other columns from the input data.

. fns  $\qquad \qquad \text{The default is IQR, but this can be any stat function like quantile or median}$ 

cic.

.return\_type The default is "vector" which returns an sapply object.

.use\_data\_table

The default is FALSE, TRUE will use data.table under the hood and still return a tibble. If this argument is set to TRUE then the .return\_type parameter will be ignored.

Addition function arguments to be supplied to the parameters of . fns

#### **Details**

A function to return the value(s) of a given tidy\_ distribution function output and chosen column from it. This function will only work with tidy\_ distribution functions.

There are currently three different output types for this function. These are:

- "vector" which gives an sapply() output
- "list" which gives an lapply() output, and
- "tibble" which returns a tibble in long format.

Currently you can pass any stat function that performs an operation on a vector input. This means you can pass things like IQR, quantile and their associated arguments in the ... portion of the function.

94 tidy\_t

This function also by default will rename the value column of the tibble to the name of the function. This function will also give the column name of sim\_number for the tibble output with the corresponding simulation numbers as the values.

For the sapply and lapply outputs the column names will also give the simulation number information by making column names like sim\_number\_1 etc.

There is an option of .use\_data\_table which can greatly enhance the speed of the calculations performed if used while still returning a tibble. The calculations are performed after turning the input data into a data.table object, performing the necessary calculation and then converting back to a tibble object.

#### Value

A return of object of either sapply lapply or tibble based upon user input.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Statistic: ci_hi(), ci_lo(), tidy_kurtosis_vec(), tidy_range_statistic(), tidy_skewness_vec()
```

### **Examples**

```
tn <- tidy_normal(.num_sims = 3)

p <- c(0.025, 0.25, 0.5, 0.75, 0.95)

tidy_stat_tbl(tn, y, quantile, "vector", probs = p, na.rm = TRUE)
tidy_stat_tbl(tn, y, quantile, "list", probs = p)
tidy_stat_tbl(tn, y, quantile, "tibble", probs = p)
tidy_stat_tbl(tn, y, quantile, .use_data_table = TRUE, probs = p, na.rm = TRUE)</pre>
```

tidy\_t

Tidy Randomly Generated T Distribution Tibble

### **Description**

This function will generate n random points from a rt distribution with a user provided, df, ncp, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

• sim\_number The current simulation number.

tidy\_t

- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

#### Usage

```
tidy_t(.n = 50, .df = 1, .ncp = 0, .num_sims = 1, .return_tibble = TRUE)
```

#### **Arguments**

. n The number of randomly generated points you want.

. df Degrees of freedom, Inf is allowed.

. ncp Non-centrality parameter.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

#### **Details**

This function uses the underlying stats::rt(), and its underlying p, d, and q functions. For more information please see stats::rt()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3664.htm

Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_triangular(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other T Distribution: util_t_stats_tbl()
```

```
tidy_t()
```

96 tidy\_triangular

tidy\_triangular

Generate Tidy Data from Triangular Distribution

## **Description**

This function generates tidy data from the triangular distribution.

## Usage

```
tidy_triangular(
    .n = 50,
    .min = 0,
    .max = 1,
    .mode = 1/2,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of x values for each simulation.
.min The minimum value of the triangular distribution.
.max The maximum value of the triangular distribution.
.mode The mode (peak) value of the triangular distribution.
.num\_sims The number of simulations to perform.
.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

### **Details**

The function takes parameters for the triangular distribution (minimum, maximum, mode), the number of x values (n), the number of simulations (num\_sims), and an option to return the result as a tibble (return\_tibble). It performs various checks on the input parameters to ensure validity. The result is a data frame or tibble with tidy data for further analysis.

## Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

tidy\_uniform 97

#### See Also

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_uniform(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Triangular: util_triangular_param_estimate(), util_triangular_stats_tbl()
```

## **Examples**

```
tidy_triangular(.return_tibble = TRUE)
```

tidy\_uniform

Tidy Randomly Generated Uniform Distribution Tibble

#### **Description**

This function will generate n random points from a uniform distribution with a user provided, .min and .max values, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the  $d_-$ ,  $p_-$  and  $q_-$  data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

#### Usage

```
tidy_uniform(.n = 50, .min = 0, .max = 1, .num_sims = 1, .return_tibble = TRUE)
```

# Arguments

.n The number of randomly generated points you want.

.min A lower limit of the distribution..max An upper limit of the distribution

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

98 tidy\_weibull

#### **Details**

This function uses the underlying stats::runif(), and its underlying p, d, and q functions. For more information please see stats::runif()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3662.htm
```

```
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_weibull(), tidy_zero_truncated_geometric()

Other Uniform: util_uniform_param_estimate(), util_uniform_stats_tbl()
```

## **Examples**

tidy\_uniform()

tidy\_weibull

Tidy Randomly Generated Weibull Distribution Tibble

### **Description**

This function will generate n random points from a weibull distribution with a user provided, .shape, .scale, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

tidy\_weibull 99

### Usage

```
tidy_weibull(
    .n = 50,
    .shape = 1,
    .scale = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

.n The number of randomly generated points you want.

. shape Shape parameter defaults to 0.. scale Scale parameter defaults to 1.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is

TRUE.

#### **Details**

This function uses the underlying stats::rweibull(), and its underlying p, d, and q functions. For more information please see stats::rweibull()

### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://www.itl.nist.gov/div898/handbook/eda/section3/eda3669.htm

Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(), tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(), tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(), tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(), tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(), tidy_t(), tidy_triangular(), tidy_uniform(), tidy_zero_truncated_geometric()

Other Weibull: tidy_inverse_weibull(), util_weibull_param_estimate(), util_weibull_stats_tbl()
```

```
tidy_weibull()
```

```
tidy_zero_truncated_binomial
```

Tidy Randomly Generated Binomial Distribution Tibble

## **Description**

This function will generate n random points from a zero truncated binomial distribution with a user provided, .size, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

## Usage

```
tidy_zero_truncated_binomial(
   .n = 50,
   .size = 1,
   .prob = 1,
   .num_sims = 1,
   .return_tibble = TRUE
)
```

TRUE.

### **Arguments**

```
    .n The number of randomly generated points you want.
    .size Number of trials, zero or more.
    .prob Probability of success on each trial 0 <= prob <= 1.</li>
    .num_sims The number of randomly generated simulations you want.
    .return_tibble A logical value indicating whether to return the result as a tibble. Default is
```

## Details

This function uses the underlying actuar::rztbinom(), and its underlying p, d, and q functions. For more information please see actuar::rztbinom()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_hypergeometric(),
tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_negative_binomial(),
tidy_zero_truncated_poisson()
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_negative_binomial(),
util_binomial_param_estimate(), util_binomial_stats_tbl(), util_negative_binomial_param_estimate(),
util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(),
util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
Other Zero Truncated Distribution: tidy_zero_truncated_geometric(), tidy_zero_truncated_poisson(),
util_zero_truncated_binomial_param_estimate()
```

### **Examples**

```
tidy_zero_truncated_binomial()
```

```
tidy_zero_truncated_geometric
```

Tidy Randomly Generated Zero Truncated Geometric Distribution Tibble

### **Description**

This function will generate n random points from a zero truncated Geometric distribution with a user provided, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_zero_truncated_geometric(
    .n = 50,
    .prob = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

#### **Arguments**

.n The number of randomly generated points you want.
 .prob A probability of success in each trial 0 < prob <= 1.</li>
 .num\_sims The number of randomly generated simulations you want.
 .return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying actuar::rztgeom(), and its underlying p, d, and q functions. For more information please see actuar::rztgeom()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
Other Geometric: tidy_geometric(), util_geometric_param_estimate(), util_geometric_stats_tbl()
Other Continuous Distribution: tidy_beta(), tidy_burr(), tidy_cauchy(), tidy_chisquare(),
tidy_exponential(), tidy_f(), tidy_gamma(), tidy_generalized_beta(), tidy_generalized_pareto(),
tidy_geometric(), tidy_inverse_burr(), tidy_inverse_exponential(), tidy_inverse_gamma(),
tidy_inverse_normal(), tidy_inverse_pareto(), tidy_inverse_weibull(), tidy_logistic(),
tidy_lognormal(), tidy_normal(), tidy_paralogistic(), tidy_pareto(), tidy_pareto1(),
tidy_t(), tidy_triangular(), tidy_uniform(), tidy_weibull()
Other Zero Truncated Distribution: tidy_zero_truncated_binomial(), tidy_zero_truncated_poisson(),
util_zero_truncated_binomial_param_estimate()
```

```
tidy_zero_truncated_geometric()
```

```
tidy_zero_truncated_negative_binomial
```

Tidy Randomly Generated Binomial Distribution Tibble

#### **Description**

This function will generate n random points from a zero truncated binomial distribution with a user provided, .size, .prob, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_zero_truncated_negative_binomial(
    .n = 50,
    .size = 0,
    .prob = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

### **Arguments**

```
    .n The number of randomly generated points you want.
    .size Number of trials, zero or more.
    .prob Probability of success on each trial 0 <= prob <= 1.</li>
```

.num\_sims The number of randomly generated simulations you want.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

## Details

This function uses the underlying actuar::rztnbinom(), and its underlying p, d, and q functions. For more information please see actuar::rztnbinom()

#### Value

A tibble of randomly generated data.

#### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html
Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_hypergeometric(),
tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_poisson()
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(),
util_binomial_param_estimate(), util_binomial_stats_tbl(), util_negative_binomial_param_estimate(),
util_zero_truncated_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
```

Other Zero Truncated Negative Distribution: util\_zero\_truncated\_negative\_binomial\_param\_estimate()

### **Examples**

tidy\_zero\_truncated\_negative\_binomial()

tidy\_zero\_truncated\_poisson

Tidy Randomly Generated Zero Truncated Poisson Distribution Tibble

## **Description**

This function will generate n random points from a Zero Truncated Poisson distribution with a user provided, .lambda, and number of random simulations to be produced. The function returns a tibble with the simulation number column the x column which corresponds to the n randomly generated points, the d\_, p\_ and q\_ data points as well.

The data is returned un-grouped.

The columns that are output are:

- sim\_number The current simulation number.
- x The current value of n for the current simulation.
- y The randomly generated data point.
- dx The x value from the stats::density() function.
- dy The y value from the stats::density() function.
- p The values from the resulting p\_ function of the distribution family.
- q The values from the resulting q\_ function of the distribution family.

### Usage

```
tidy_zero_truncated_poisson(
    .n = 50,
    .lambda = 1,
    .num_sims = 1,
    .return_tibble = TRUE
)
```

## **Arguments**

.n The number of randomly generated points you want.

. lambda A vector of non-negative means.

.return\_tibble A logical value indicating whether to return the result as a tibble. Default is TRUE.

#### **Details**

This function uses the underlying actuar::rztpois(), and its underlying p, d, and q functions. For more information please see actuar::rztpois()

#### Value

A tibble of randomly generated data.

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
https://openacttexts.github.io/Loss-Data-Analytics/ChapSummaryDistributions.html

Other Poisson: tidy_poisson(), util_poisson_param_estimate(), util_poisson_stats_tbl(),
 util_zero_truncated_poisson_param_estimate(), util_zero_truncated_poisson_stats_tbl()

Other Zero Truncated Distribution: tidy_zero_truncated_binomial(), tidy_zero_truncated_geometric(),
 util_zero_truncated_binomial_param_estimate()

Other Discrete Distribution: tidy_bernoulli(), tidy_binomial(), tidy_hypergeometric(),
 tidy_negative_binomial(), tidy_poisson(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_
```

```
tidy_zero_truncated_poisson()
```

106 triangle\_plot

triangle\_plot

Triangle Distribution PDF Plot

# Description

This function generates a probability density function (PDF) plot for the triangular distribution.

# Usage

```
triangle_plot(.data, .interactive = FALSE)
```

# Arguments

.data Tidy data from the tidy\_triangular function.

. interactive A logical value indicating whether to return an interactive plot using plotly. De-

fault is FALSE.

### **Details**

The function checks if the input data is a data frame or tibble, and if it comes from the tidy\_triangular function. It then extracts necessary attributes for the plot and creates a PDF plot using ggplot2. The plot includes data points and segments to represent the triangular distribution.

### Value

The function returns a ggplot2 object representing the probability density function plot for the triangular distribution.

## Author(s)

Steven P. Sanderson II, MPH

```
# Example: Generating a PDF plot for the triangular distribution
data <- tidy_triangular(.n = 50, .min = 0, .max = 1, .mode = 1/2, .num_sims = 1,
.return_tibble = TRUE)
triangle_plot(data)</pre>
```

```
util_bernoulli_param_estimate
```

Estimate Bernoulli Parameters

### **Description**

This function will attempt to estimate the Bernoulli prob parameter given some vector of values .x. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated Bernoulli data.

## Usage

```
util_bernoulli_param_estimate(.x, .auto_gen_empirical = TRUE)
```

#### **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector . x is a numeric vector. It will attempt to estimate the prob parameter of a Bernoulli distribution.

## Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_geometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_lognormal_param_estimate(), util_normal_param_estimate(), util_param_estimate(), util_param_estimate(
```

```
util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate() util_zero_truncated_geometric_param_estimate(), util_zero_truncated_negative_binomial_param_estimate util_zero_truncated_poisson_param_estimate()

Other Bernoulli: tidy_bernoulli(), util_bernoulli_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

tb <- tidy_bernoulli(.prob = .1) |> pull(y)
output <- util_bernoulli_param_estimate(tb)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_bernoulli_stats_tbl
```

Distribution Statistics

### **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_bernoulli_stats_tbl(.data)
```

### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

util\_beta\_aic 109

#### See Also

```
Other Distribution Statistics: util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_statutil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_bernoulli() |>
  util_bernoulli_stats_tbl() |>
  glimpse()
```

util\_beta\_aic

Calculate Akaike Information Criterion (AIC) for Beta Distribution

## Description

This function estimates the parameters of a beta distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

### Usage

```
util_beta_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data to be fitted to a beta distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a beta distribution fitted to the provided data.

Initial parameter estimates: The choice of initial values can impact the convergence of the optimization.

Optimization method: You might explore different optimization methods within optim for potentially better performance. Data transformation: Depending on your data, you may need to apply transformations (e.g., scaling to [0,1] interval) before fitting the beta distribution.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted beta distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic() util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto1_aic(), util_pareto1_aic(), util_pareto1_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rbeta(30, 1, 1)
util_beta_aic(x)</pre>
```

util\_beta\_param\_estimate

Estimate Beta Parameters

## **Description**

This function will automatically scale the data from 0 to 1 if it is not already. This means you can pass a vector like mtcars\$mpg and not worry about it.

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated beta data.

Three different methods of shape parameters are supplied:

- Bayes
- NIST mme
- EnvStats mme, see EnvStats::ebeta()

### Usage

```
util_beta_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### **Arguments**

.x The vector of data to be passed to the function. Must be numeric, and all values must be  $0 \le x \le 1$ 

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the beta shape1 and shape2 parameters given some vector of values.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_gamealized_param_estimate(), util_gamea_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_param_estimate(), util_logistic_param_estimate(), util_logistic_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_param_estimate(), ut
```

# Examples

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_beta_param_estimate(x)</pre>
```

112 util\_beta\_stats\_tbl

```
output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

tb <- rbeta(50, 2.5, 1.4)
util_beta_param_estimate(tb)$parameter_tbl</pre>
```

## **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_beta_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

```
Other Beta: tidy_beta(), tidy_generalized_beta(), util_beta_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl(), util_f_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_par
```

util\_binomial\_aic 113

```
util_poisson_stats_tbl(), util_t_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(),
util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_sta
util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_beta() |>
  util_beta_stats_tbl() |>
  glimpse()
```

util\_binomial\_aic

Calculate Akaike Information Criterion (AIC) for Binomial Distribution

### **Description**

This function estimates the size and probability parameters of a binomial distribution from the provided data and then calculates the AIC value based on the fitted distribution.

### Usage

```
util_binomial_aic(.x)
```

### **Arguments**

. x A numeric vector containing the data to be fitted to a binomial distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a binomial distribution fitted to the provided data.

This function fits a binomial distribution to the provided data. It estimates the size and probability parameters of the binomial distribution from the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the size and probability parameters of the binomial distribution.

Optimization method: Since the parameters are directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted binomial distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic() util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

#### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rbinom(30, size = 10, prob = 0.2)
util_binomial_aic(x)</pre>
```

util\_binomial\_param\_estimate

Estimate Binomial Parameters

## **Description**

This function will check to see if some given vector .x is either a numeric vector or a factor vector with at least two levels then it will cause an error and the function will abort. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated binomial data.

## Usage

```
util_binomial_param_estimate(.x, .size = NULL, .auto_gen_empirical = TRUE)
```

### **Arguments**

. x The vector of data to be passed to the function. Must be numeric, and all values must be  $0 \le x \le 1$ 

. size Number of trials, zero or more.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the binomial p\_hat and size parameters given some vector of values

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(),
util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(),
util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(),
util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(),
util_geometric_param_estimate(),util_hypergeometric_param_estimate(),util_inverse_burr_param_estima
util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate()
util_lognormal_param_estimate(), util_negative_binomial_param_estimate(), util_normal_param_estimate(
util_paralogistic_param_estimate(), util_pareto1_param_estimate(), util_pareto_param_estimate(),
util_poisson_param_estimate(), util_t_param_estimate(), util_triangular_param_estimate(),
util_uniform_param_estimate(),util_weibull_param_estimate(),util_zero_truncated_binomial_param_esti
util_zero_truncated_geometric_param_estimate(), util_zero_truncated_negative_binomial_param_estimate
util_zero_truncated_poisson_param_estimate()
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(),
tidy_zero_truncated_negative_binomial(),util_binomial_stats_tbl(),util_negative_binomial_param_esti
util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(),
util_zero_truncated_negative_binomial_param_estimate(),util_zero_truncated_negative_binomial_stats_
```

# **Examples**

```
library(dplyr)
library(ggplot2)

tb <- rbinom(50, 1, .1)
output <- util_binomial_param_estimate(tb)

output$parameter_tbl

output$combined_data_tbl |>
   tidy_combined_autoplot()
```

```
util_binomial_stats_tbl
```

Distribution Statistics

#### Description

Returns distribution statistics in a tibble.

#### **Usage**

```
util_binomial_stats_tbl(.data)
```

#### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

```
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), util_binomial_param_estimate(), util_negative_binomial_param_util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_util_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stautil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_binomial() |>
  util_binomial_stats_tbl() |>
  glimpse()
```

util\_burr\_param\_estimate

Estimate Burr Parameters

### **Description**

This function will attempt to estimate the Burr prob parameter given some vector of values .x. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated Burr data.

# Usage

```
util_burr_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector .x is a numeric vector. It will attempt to estimate the prob parameter of a Burr distribution.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

118 util\_burr\_stats\_tbl

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate() util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate() util_zero_truncated_binomial_param_estimate() util_zero_truncated_poisson_param_estimate()

Other Burr: tidy_burr(), tidy_inverse_burr(), util_burr_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

tb <- tidy_burr(.shape1 = 1, .shape2 = 2, .rate = .3) |> pull(y)
output <- util_burr_param_estimate(tb)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

### **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_burr_stats_tbl(.data)
```

#### **Arguments**

.data

The data being passed from a tidy\_distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

util\_cauchy\_aic 119

#### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_burr() |>
  util_burr_stats_tbl() |>
  glimpse()
```

util\_cauchy\_aic

Calculate Akaike Information Criterion (AIC) for Cauchy Distribution

# Description

This function estimates the parameters of a Cauchy distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### Usage

```
util_cauchy_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a Cauchy distribution.

120 util\_cauchy\_aic

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Cauchy distribution fitted to the provided data.

This function fits a Cauchy distribution to the provided data using maximum likelihood estimation. It first estimates the initial parameters of the Cauchy distribution using the method of moments. Then, it optimizes the negative log-likelihood function using the provided data and the initial parameter estimates. Finally, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates for the initial location and scale parameters of the Cauchy distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted Cauchy distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rcauchy(30)
util_cauchy_aic(x)</pre>
```

util\_cauchy\_param\_estimate

Estimate Cauchy Parameters

### **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated cauchy data.

#### Usage

```
util_cauchy_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the cauchy location and scale parameters given some vector of values.

## Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_baram_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_param_estimate(), util_par
```

122 util\_cauchy\_stats\_tbl

```
util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate()
util_zero_truncated_geometric_param_estimate(), util_zero_truncated_negative_binomial_param_estimate
util_zero_truncated_poisson_param_estimate()
Other Cauchy: tidy_cauchy(), util_cauchy_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- tidy_cauchy(.location = 0, .scale = 1)$y
output <- util_cauchy_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

### **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_cauchy_stats_tbl(.data)
```

# Arguments

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Cauchy: tidy_cauchy(), util_cauchy_param_estimate()

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl(), util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_statutil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

#### **Examples**

```
library(dplyr)

tidy_cauchy() |>
  util_cauchy_stats_tbl() |>
  glimpse()
```

```
util_chisquare_param_estimate
```

Estimate Chisquare Parameters

# Description

This function will attempt to estimate the Chisquare prob parameter given some vector of values .x. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated Chisquare data.

#### Usage

```
util_chisquare_param_estimate(.x, .auto_gen_empirical = TRUE)
```

# **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector .x is a numeric vector. It will attempt to estimate the prob parameter of a Chisquare distribution. The function first performs tidyeval on the input data to ensure it's a numeric vector. It then checks if there are at least two data points, as this is a requirement for parameter estimation.

The estimation of the chi-square distribution parameters is performed using maximum likelihood estimation (MLE) implemented with the bbmle package. The negative log-likelihood function is minimized to obtain the estimates for the degrees of freedom (doff) and the non-centrality parameter (ncp). Initial values for the optimization are set based on the sample variance and mean, but these can be adjusted if necessary.

If the estimation fails or encounters an error, the function returns NA for both doff and ncp.

Finally, the function returns a tibble containing the following information:

**dist\_type** The type of distribution, which is "Chisquare" in this case.

**samp\_size** The sample size, i.e., the number of data points in the input vector.

**min** The minimum value of the data points.

max The maximum value of the data points.

**mean** The mean of the data points.

degrees\_of\_freedom The estimated degrees of freedom (doff) for the chi-square distribution.

**ncp** The estimated non-centrality parameter (ncp) for the chi-square distribution.

Additionally, if the argument .auto\_gen\_empirical is set to TRUE (which is the default behavior), the function also returns a combined tibble containing both empirical and chi-square distribution data, obtained by calling tidy\_empirical and tidy\_chisquare, respectively.

## Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_geometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Chisquare: tidy_chisquare(), util_chisquare_stats_tbl()
```

util\_chisquare\_stats\_tbl 125

## **Examples**

```
library(dplyr)
library(ggplot2)

tc <- tidy_chisquare(.n = 500, .df = 6, .ncp = 1) |> pull(y)
output <- util_chisquare_param_estimate(tc)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_chisquare_stats_tbl
```

Distribution Statistics

# Description

Returns distribution statistics in a tibble.

# Usage

```
util_chisquare_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

126 util\_chisq\_aic

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_exponential_stats_tbl(), util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stats_util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_chisquare() |>
  util_chisquare_stats_tbl() |>
  glimpse()
```

util\_chisq\_aic

Calculate Akaike Information Criterion (AIC) for Chi-Square Distribution

## Description

This function estimates the parameters of a chi-square distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### Usage

```
util_chisq_aic(.x)
```

#### **Arguments**

.x A numeric vector containing the data to be fitted to a chi-square distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a chi-square distribution fitted to the provided data.

### Value

The AIC value calculated based on the fitted chi-square distribution to the provided data.

util\_exponential\_aic 127

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rchisq(30, df = 3)
util_chisq_aic(x)</pre>
```

```
util_exponential_aic Calculate Akaike Information Criterion (AIC) for Exponential Distri-
bution
```

#### **Description**

This function estimates the rate parameter of an exponential distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### Usage

```
util_exponential_aic(.x)
```

## Arguments

.x A numeric vector containing the data to be fitted to an exponential distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for an exponential distribution fitted to the provided data.

This function fits an exponential distribution to the provided data using maximum likelihood estimation. It estimates the rate parameter of the exponential distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the reciprocal of the mean of the data as the initial estimate for the rate parameter.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted exponential distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rexp(30)
util_exponential_aic(x)</pre>
```

```
util_exponential_param_estimate

*Estimate Exponential Parameters*
```

## **Description**

This function will attempt to estimate the exponential rate parameter given some vector of values. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated exponential data.

### Usage

```
util_exponential_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function. Must be numeric.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector . x is a numeric vector.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate())
Other Exponential: tidy_exponential(), tidy_inverse_exponential(), util_exponential_stats_tbl()
```

# Examples

```
library(dplyr)
library(ggplot2)

te <- tidy_exponential(.rate = .1) |> pull(y)
output <- util_exponential_param_estimate(te)
output$parameter_tbl</pre>
```

```
output$combined_data_tbl |>
  tidy_combined_autoplot()
```

```
util_exponential_stats_tbl
```

Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

#### Usage

```
util_exponential_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_logistic_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_statutil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

util\_f\_aic 131

### **Examples**

```
library(dplyr)

tidy_exponential() |>
  util_exponential_stats_tbl() |>
  glimpse()
```

util\_f\_aic

Calculate Akaike Information Criterion (AIC) for F Distribution

## Description

This function estimates the parameters of a F distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_f_aic(.x)
```

#### **Arguments**

. x

A numeric vector containing the data to be fitted to an F distribution.

## **Details**

This function calculates the Akaike Information Criterion (AIC) for an F distribution fitted to the provided data.

This function fits an F distribution to the input data using maximum likelihood estimation and then computes the Akaike Information Criterion (AIC) based on the fitted distribution.

#### Value

The AIC value calculated based on the fitted F distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

```
rf for generating F-distributed data, optim for optimization.
```

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(),
```

132 util\_f\_param\_estimate

```
util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(),
util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomia
util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(),
util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Generate F-distributed data
set.seed(123)
x <- rf(100, df1 = 5, df2 = 10, ncp = 1)
# Calculate AIC for the generated data
util_f_aic(x)</pre>
```

### **Description**

Estimate F Distribution Parameters

### Usage

```
util_f_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function, where the data comes from the rf() function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the F distribution parameters given some vector of values produced by rf(). The estimation method is from the NIST Engineering Statistics Handbook.

## Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

util\_f\_stats\_tbl

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_geometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_negative_binomial_param_estimate(), util_normal_param_estimate(), util_paralogistic_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate())
Other F Distribution: tidy_f(), util_f_stats_tbl()
```

### **Examples**

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- rf(100, df1 = 5, df2 = 10, ncp = 1)
output <- util_f_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
   tidy_combined_autoplot()
```

util\_f\_stats\_tbl

Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

#### **Usage**

```
util_f_stats_tbl(.data)
```

## Arguments

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

134 util\_gamma\_aic

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Distribution: tidy_f(), util_f_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stautil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_f() |>
  util_f_stats_tbl() |>
  glimpse()
```

util\_gamma\_aic

Calculate Akaike Information Criterion (AIC) for Gamma Distribution

### **Description**

This function estimates the shape and scale parameters of a gamma distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

### Usage

```
util_gamma_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a gamma distribution.

util\_gamma\_aic 135

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a gamma distribution fitted to the provided data.

This function fits a gamma distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the gamma distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the gamma distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted gamma distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic() util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rgamma(30, shape = 1)
util_gamma_aic(x)</pre>
```

```
util_gamma_param_estimate
```

Estimate Gamma Parameters

### **Description**

This function will attempt to estimate the gamma shape and scale parameters given some vector of values. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated gamma data.

## Usage

```
util_gamma_param_estimate(.x, .auto_gen_empirical = TRUE)
```

#### **Arguments**

.x The vector of data to be passed to the function. Must be numeric.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector . x is a numeric vector.

### Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_param_esti
```

util\_gamma\_stats\_tbl 137

```
util_zero_truncated_geometric_param_estimate(), util_zero_truncated_negative_binomial_param_estimate
util_zero_truncated_poisson_param_estimate()
Other Gamma: tidy_gamma(), tidy_inverse_gamma(), util_gamma_stats_tbl()
```

# **Examples**

```
library(dplyr)
library(ggplot2)

tg <- tidy_gamma(.shape = 1, .scale = .3) |> pull(y)
output <- util_gamma_param_estimate(tg)

output$parameter_tbl

output$combined_data_tbl |>
   tidy_combined_autoplot()
```

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_gamma_stats_tbl(.data)
```

# Arguments

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_geometric_statutil_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

#### **Examples**

```
library(dplyr)

tidy_gamma() |>
  util_gamma_stats_tbl() |>
  glimpse()
```

```
util_generalized_beta_aic
```

Calculate Akaike Information Criterion (AIC) for Generalized Beta Distribution

## Description

This function estimates the shape1, shape2, shape3, and rate parameters of a generalized Beta distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

### Usage

```
util_generalized_beta_aic(.x)
```

#### **Arguments**

.x A numeric vector containing the data to be fitted to a generalized Beta distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a generalized Beta distribution fitted to the provided data.

This function fits a generalized Beta distribution to the provided data using maximum likelihood estimation. It estimates the shape1, shape2, shape3, and rate parameters of the generalized Beta distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses reasonable initial estimates for the shape1, shape2, shape3, and rate parameters of the generalized Beta distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted generalized Beta distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_taic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
util_generalized_beta_param_estimate
```

Estimate Generalized Beta Parameters

### **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated generalized Beta data.

#### **Usage**

```
util_generalized_beta_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the generalized Beta shape1, shape2, shape3, and rate parameters given some vector of values.

#### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_pareto_param_estimate(), util_geometric_param_estimate() util_hypergeometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_e util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate() util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(),
```

util\_weibull\_param\_estimate(), util\_zero\_truncated\_binomial\_param\_estimate(), util\_zero\_truncated\_geoutil\_zero\_truncated\_negative\_binomial\_param\_estimate(), util\_zero\_truncated\_poisson\_param\_estimate()
Other Generalized Beta: util\_generalized\_beta\_stats\_tbl()

### **Examples**

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- tidy_generalized_beta(100, .shape1 = 2, .shape2 = 3, .shape3 = 4, .rate = 5)[["y"]]
output <- util_generalized_beta_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl %>%
    tidy_combined_autoplot()
```

```
util_generalized_beta_stats_tbl 
 Distribution Statistics
```

## **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_generalized_beta_stats_tbl(.data)
```

### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and return the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Generalized Beta: util_generalized_beta_param_estimate()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_pareto_stats_tbl(), util_geometric_stats_util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto2_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)
set.seed(123)
tidy_generalized_beta() |>
  util_generalized_beta_stats_tbl() |>
  glimpse()
```

```
util_generalized_pareto_aic
```

Calculate Akaike Information Criterion (AIC) for Generalized Pareto Distribution

# **Description**

This function estimates the shape1, shape2, and rate parameters of a generalized Pareto distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### Usage

```
util_generalized_pareto_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data to be fitted to a generalized Pareto distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a generalized Pareto distribution fitted to the provided data.

This function fits a generalized Pareto distribution to the provided data using maximum likelihood estimation. It estimates the shape1, shape2, and rate parameters of the generalized Pareto distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape1, shape2, and rate parameters of the generalized Pareto distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted generalized Pareto distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- actuar::rgenpareto(100, shape1 = 1, shape2 = 2, scale = 3)
util_generalized_pareto_aic(x)</pre>
```

```
util_generalized_pareto_param_estimate
```

Estimate Generalized Pareto Parameters

### **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated generalized Pareto data.

#### **Usage**

```
util_generalized_pareto_param_estimate(.x, .auto_gen_empirical = TRUE)
```

# **Arguments**

The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the generalized Pareto shape1, shape2, and rate parameters given some vector of values.

#### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_geometric_param_estimate(), util_hypergeometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_e util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_param_estimate() util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(),
```

```
util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Generalized Pareto: util_generalized_pareto_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- tidy_generalized_pareto(100, .shape1 = 1, .shape2 = 2, .scale = 3)[["y"]]
output <- util_generalized_pareto_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl %>%
    tidy_combined_autoplot()
```

```
util_generalized_pareto_stats_tbl

Distribution Statistics
```

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_generalized_pareto_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and return the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

146 util\_geometric\_aic

#### See Also

```
Other Generalized Pareto: util_generalized_pareto_param_estimate()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_geometric_stats_tbl util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto2_stats_tbl(), util_pareto3_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_generalized_pareto() |>
  util_generalized_pareto_stats_tbl() |>
  glimpse()
```

util\_geometric\_aic

Calculate Akaike Information Criterion (AIC) for Geometric Distribution

## Description

This function estimates the probability parameter of a geometric distribution from the provided data and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_geometric_aic(.x)
```

## Arguments

.x A numeric vector containing the data to be fitted to a geometric distribution.

## Details

This function calculates the Akaike Information Criterion (AIC) for a geometric distribution fitted to the provided data.

This function fits a geometric distribution to the provided data. It estimates the probability parameter of the geometric distribution from the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimate as a starting point for the probability parameter of the geometric distribution.

Optimization method: Since the parameter is directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted geometric distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aicutil_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rgeom(100, prob = 0.2)
util_geometric_aic(x)</pre>
```

```
util_geometric_param_estimate

Estimate Geometric Parameters
```

# Description

This function will attempt to estimate the geometric prob parameter given some vector of values .x. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated geometric data.

### **Usage**

```
util_geometric_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector .x is a numeric vector. It will attempt to estimate the prob parameter of a geometric distribution.

#### Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_paratosimate(), util_paratosimate(), util_paratosimate(), util_paratosimate(), util_paratosimate(), util_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geomutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Geometric: tidy_geometric(), tidy_zero_truncated_geometric(), util_geometric_stats_tbl()
```

```
library(dplyr)
library(ggplot2)

tg <- tidy_geometric(.prob = .1) |> pull(y)
output <- util_geometric_param_estimate(tg)
output$parameter_tbl</pre>
```

```
util_geometric_stats_tbl
```

```
output$combined_data_tbl |>
  tidy_combined_autoplot()
```

```
util_geometric_stats_tbl
```

Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

#### Usage

```
util_geometric_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Geometric: tidy_geometric(), tidy_zero_truncated_geometric(), util_geometric_param_estimate()

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto2_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_biutil_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_geometric() |>
  util_geometric_stats_tbl() |>
  glimpse()
```

util\_hypergeometric\_aic

Calculate Akaike Information Criterion (AIC) for Hypergeometric Distribution

# **Description**

This function estimates the parameters m, n, and k of a hypergeometric distribution from the provided data and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_hypergeometric_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data to be fitted to a hypergeometric distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a hypergeometric distribution fitted to the provided data.

This function fits a hypergeometric distribution to the provided data. It estimates the parameters m, n, and k of the hypergeometric distribution from the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function does not estimate parameters; they are directly calculated from the data.

Optimization method: Since the parameters are directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted hypergeometric distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

#### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rhyper(100, m = 10, n = 10, k = 5)
util_hypergeometric_aic(x)</pre>
```

```
util_hypergeometric_param_estimate

Estimate Hypergeometric Parameters
```

## Description

This function will attempt to estimate the geometric prob parameter given some vector of values .x. Estimate m, the number of white balls in the urn, or m+n, the total number of balls in the urn, for a hypergeometric distribution.

#### Usage

```
util_hypergeometric_param_estimate(
    .x,
    .m = NULL,
    .total = NULL,
    .k,
    .auto_gen_empirical = TRUE
)
```

### **Arguments**

. m

lls out of a sample
annot have missing,

Non-negative integer indicating the number of white balls in the urn. You must supply .m or .total, but not both. You cannot have missing values.

. total A positive integer indicating the total number of balls in the urn (i.e., m+n). You must supply .m or . total, but not both. You cannot have missing values.

.k A positive integer indicating the number of balls drawn without replacement from the urn. You cannot have missing values.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will see if the given vector .x is a numeric integer. It will attempt to estimate the prob parameter of a geometric distribution. Missing (NA), undefined (NaN), and infinite (Inf, -Inf) values are not allowed. Let .x be an observation from a hypergeometric distribution with parameters .m = M, .n = N, and .k = K. In R nomenclature, .x represents the number of white balls drawn out of a sample of .k balls drawn without replacement from an urn containing .m white balls and .n black balls. The total number of balls in the urn is thus .m + .n. Denote the total number of balls by T = .m + .n

#### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_parato_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_param_estimate(), util_inverse_param_estimate(), util_inverse_param_estimate(), util_normal_param_estimate(), util_paralogistic_param_estimate(), util_paratol_param_estimate(), util_paratol_param_estimate(), util_param_estimate(), util_taram_estimate(), util_
```

# **Examples**

```
library(dplyr)
library(ggplot2)

th <- rhyper(10, 20, 30, 5)
output <- util_hypergeometric_param_estimate(th, .total = 50, .k = 5)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_hypergeometric_stats_tbl
```

Distribution Statistics

# Description

Returns distribution statistics in a tibble.

# Usage

```
util_hypergeometric_stats_tbl(.data)
```

# Arguments

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

154 util\_inverse\_burr\_aic

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto2_stats_tbl(), util_pareto3_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_biutil_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_hypergeometric() |>
  util_hypergeometric_stats_tbl() |>
  glimpse()
```

 ${\tt util\_inverse\_burr\_aic} \quad {\it Calculate\ Akaike\ Information\ Criterion\ (AIC)\ for\ Inverse\ Burr\ Distribution}$ 

## **Description**

This function estimates the shape1, shape2, and rate parameters of an inverse Burr distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_inverse_burr_aic(.x)
```

# Arguments

.x A numeric vector containing the data to be fitted to an inverse Burr distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for an inverse Burr distribution fitted to the provided data.

This function fits an inverse Burr distribution to the provided data using maximum likelihood estimation. It estimates the shape1, shape2, and rate parameters of the inverse Burr distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape1, shape2, and rate parameters of the inverse Burr distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted inverse Burr distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- tidy_inverse_burr(100, .shape1 = 2, .shape2 = 3, .scale = 1)[["y"]]
util_inverse_burr_aic(x)</pre>
```

```
util_inverse_burr_param_estimate
```

Estimate Inverse Burr Parameters

## Description

This function will attempt to estimate the inverse Burr shape1, shape2, and rate parameters given some vector of values .x. The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated inverse Burr data.

#### **Usage**

```
util_inverse_burr_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will see if the given vector .x is a numeric vector. It will attempt to estimate the shape1, shape2, and rate parameters of an inverse Burr distribution.

#### Value

A tibble/list

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_beta_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_paralogistic_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Inverse Burr: util_inverse_burr_stats_tbl()
```

```
library(dplyr)
library(ggplot2)

set.seed(123)
tb <- tidy_burr(.shape1 = 1, .shape2 = 2, .rate = .3) |> pull(y)
output <- util_inverse_burr_param_estimate(tb)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_inverse_burr_stats_tbl
```

## **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_inverse_burr_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

## **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Inverse Burr: util_inverse_burr_param_estimate()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_paralogistic_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)
set.seed(123)
tidy_inverse_burr() |>
  util_inverse_burr_stats_tbl() |>
  glimpse()
```

util\_inverse\_pareto\_aic

Calculate Akaike Information Criterion (AIC) for Inverse Pareto Distribution

# Description

This function estimates the shape and scale parameters of an inverse Pareto distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

# Usage

```
util_inverse_pareto_aic(.x)
```

## **Arguments**

. x A numeric vector containing the data to be fitted to an inverse Pareto distribution.

## Details

This function calculates the Akaike Information Criterion (AIC) for an inverse Pareto distribution fitted to the provided data.

This function fits an inverse Pareto distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the inverse Pareto distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the inverse Pareto distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted inverse Pareto distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_poisson_aic(), util_taic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- tidy_inverse_pareto(.n = 100, .shape = 2, .scale = 1)[["y"]]
util_inverse_pareto_aic(x)</pre>
```

```
\verb"util_inverse_pareto_param_estimate"
```

Estimate Inverse Pareto Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated inverse Pareto data.

### Usage

```
util_inverse_pareto_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function.

```
.auto_gen_empirical
```

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the inverse Pareto shape and scale parameters given some vector of values.

#### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_binomial_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()

Other Inverse Pareto: util_inverse_pareto_stats_tbl()
```

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- tidy_inverse_pareto(.n = 100, .shape = 2, .scale = 1)[["y"]]
output <- util_inverse_pareto_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl %>%
    tidy_combined_autoplot()
```

```
util_inverse_pareto_stats_tbl

Distribution Statistics
```

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_inverse_pareto_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Inverse Pareto: util_inverse_pareto_param_estimate()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto1_stats_tbl(), util_pareto1_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

```
library(dplyr)

tidy_inverse_pareto() |>
  util_inverse_pareto_stats_tbl() |>
  glimpse()
```

```
util_inverse_weibull_aic
```

Calculate Akaike Information Criterion (AIC) for Inverse Weibull Distribution

# **Description**

This function estimates the shape and scale parameters of an inverse Weibull distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_inverse_weibull_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data to be fitted to an inverse Weibull distribu-

### Details

This function calculates the Akaike Information Criterion (AIC) for an inverse Weibull distribution fitted to the provided data.

This function fits an inverse Weibull distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the inverse Weibull distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the inverse Weibull distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

## Value

The AIC value calculated based on the fitted inverse Weibull distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- tidy_inverse_weibull(.n = 100, .shape = 2, .scale = 1)[["y"]]
util_inverse_weibull_aic(x)</pre>
```

```
\verb"util_inverse_weibull_param_estimate"
```

Estimate Inverse Weibull Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated inverse Weibull data.

## Usage

```
util_inverse_weibull_param_estimate(.x, .auto_gen_empirical = TRUE)
```

#### **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the inverse Weibull shape and rate parameters given some vector of values.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Inverse Weibull: util_inverse_weibull_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- tidy_inverse_weibull(100, .shape = 2, .scale = 1)[["y"]]
output <- util_inverse_weibull_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl %>%
    tidy_combined_autoplot()
```

```
util_inverse_weibull_stats_tbl

Distribution Statistics
```

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_inverse_weibull_stats_tbl(.data)
```

util\_logistic\_aic 165

#### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Inverse Weibull: util_inverse_weibull_param_estimate()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)
set.seed(123)
tidy_inverse_weibull() |>
  util_inverse_weibull_stats_tbl() |>
  glimpse()
```

util\_logistic\_aic

Calculate Akaike Information Criterion (AIC) for Logistic Distribution

## **Description**

This function estimates the location and scale parameters of a logistic distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

166 util\_logistic\_aic

#### Usage

```
util_logistic_aic(.x)
```

#### **Arguments**

.x A numeric vector containing the data to be fitted to a logistic distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a logistic distribution fitted to the provided data.

This function fits a logistic distribution to the provided data using maximum likelihood estimation. It estimates the location and scale parameters of the logistic distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the location and scale parameters of the logistic distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted logistic distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rlogis(30)
util_logistic_aic(x)</pre>
```

util\_logistic\_param\_estimate

Estimate Logistic Parameters

# **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated logistic data.

Three different methods of shape parameters are supplied:

- MLE
- MME
- MMUE

## Usage

```
util_logistic_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

## **Details**

This function will attempt to estimate the logistic location and scale parameters given some vector of values.

## Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_baram_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Logistic: tidy_logistic(), tidy_paralogistic(), util_logistic_stats_tbl()
```

### **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_logistic_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

t <- rlogis(50, 2.5, 1.4)
util_logistic_param_estimate(t)$parameter_tbl</pre>
```

```
util_logistic_stats_tbl

Distribution Statistics
```

## Description

Returns distribution statistics in a tibble.

#### Usage

```
util_logistic_stats_tbl(.data)
```

## Arguments

. data The data being passed from a tidy\_distribution function.

util\_lognormal\_aic 169

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Logistic: tidy_logistic(), tidy_paralogistic(), util_logistic_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

#### **Examples**

```
library(dplyr)

tidy_logistic() |>
  util_logistic_stats_tbl() |>
  glimpse()
```

util\_lognormal\_aic

Calculate Akaike Information Criterion (AIC) for Log-Normal Distribution

# Description

This function estimates the meanlog and sdlog parameters of a log-normal distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_lognormal_aic(.x)
```

170 util\_lognormal\_aic

### **Arguments**

.x A numeric vector containing the data to be fitted to a log-normal distribution.

## **Details**

This function calculates the Akaike Information Criterion (AIC) for a log-normal distribution fitted to the provided data.

This function fits a log-normal distribution to the provided data using maximum likelihood estimation. It estimates the meanlog and sdlog parameters of the log-normal distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the meanlog and sdlog parameters of the log-normal distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted log-normal distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rlnorm(100, meanlog = 0, sdlog = 1)
util_lognormal_aic(x)</pre>
```

```
util_lognormal_param_estimate
```

Estimate Lognormal Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated lognormal data.

Three different methods of shape parameters are supplied:

```
mme, see EnvStats::elnorm()mle, see EnvStats::elnorm()
```

## Usage

```
util_lognormal_param_estimate(.x, .auto_gen_empirical = TRUE)
```

#### **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the lognormal meanlog and log sd parameters given some vector of values.

## Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_invers
```

```
util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimutil_negative_binomial_param_estimate(), util_normal_param_estimate(), util_paralogistic_param_estimate(), util_pareto1_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Lognormal: tidy_lognormal(), util_lognormal_stats_tbl()
```

### **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_lognormal_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

tb <- tidy_lognormal(.meanlog = 2, .sdlog = 1) |> pull(y)
util_lognormal_param_estimate(tb)$parameter_tbl
```

```
util_lognormal_stats_tbl

Distribution Statistics
```

# Description

Returns distribution statistics in a tibble.

## Usage

```
util_lognormal_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto1_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_biutil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_lognormal() |>
  util_lognormal_stats_tbl() |>
  glimpse()
```

```
util_negative_binomial_aic
```

Calculate Akaike Information Criterion (AIC) for Negative Binomial Distribution

## **Description**

This function estimates the parameters size (r) and probability (prob) of a negative binomial distribution from the provided data and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_negative_binomial_aic(.x)
```

## Arguments

.x A numeric vector containing the data to be fitted to a negative binomial distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a negative binomial distribution fitted to the provided data.

This function fits a negative binomial distribution to the provided data. It estimates the parameters size (r) and probability (prob) of the negative binomial distribution from the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimate as a starting point for the size (r) parameter of the negative binomial distribution, and the probability (prob) is estimated based on the mean and variance of the data.

Optimization method: Since the parameters are directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted negative binomial distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
data <- rnbinom(n = 100, size = 5, mu = 10)
util_negative_binomial_aic(data)</pre>
```

```
util_negative_binomial_param_estimate

Estimate Negative Binomial Parameters
```

# **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated negative binomial data.

Three different methods of shape parameters are supplied:

- MLE/MME
- MMUE
- MLE via optim function.

## Usage

```
util_negative_binomial_param_estimate(
   .x,
   .size = 1,
   .auto_gen_empirical = TRUE
)
```

## **Arguments**

.x The vector of data to be passed to the function.

. size The size parameter, the default is 1.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the negative binomial size and prob parameters given some vector of values.

## Value

A tibble/list

# Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_estimate(), util_gamma_param_estimate(), util_hypergeometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()

Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial_stats_tbl(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- as.integer(mtcars$mpg)
output <- util_negative_binomial_param_estimate(x, .size = 1)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

t <- rnbinom(50, 1, .1)
util_negative_binomial_param_estimate(t, .size = 1)$parameter_tbl</pre>
```

```
util_negative_binomial_stats_tbl

Distribution Statistics
```

### **Description**

Returns distribution statistics in a tibble.

# Usage

```
util_negative_binomial_stats_tbl(.data)
```

## **Arguments**

. data The data being passed from a tidy\_ distribution function.

util\_normal\_aic 177

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

#### Value

A tibble

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_paralogistic_stats_tbl(), util_parato1_stats_tbl(), util_pareto2_stats_tbl(), util_parato3_stats_tbl(), util_pareto3_stats_tbl(), util_parato3_stats_tbl(), util_parato3_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_biutil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_biutil_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_negative_binomial() |>
  util_negative_binomial_stats_tbl() |>
  glimpse()
```

util\_normal\_aic

Calculate Akaike Information Criterion (AIC) for Normal Distribution

## Description

This function estimates the parameters of a normal distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_normal_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a normal distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a normal distribution fitted to the provided data.

#### Value

The AIC value calculated based on the fitted normal distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_paralogistic_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
data <- rnorm(30)
util_normal_aic(data)</pre>
```

```
util_normal_param_estimate
```

Estimate Normal Gaussian Parameters

## Description

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated normal data.

Three different methods of shape parameters are supplied:

- MLE/MME
- MVUE

#### **Usage**

```
util_normal_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the normal gaussian mean and standard deviation parameters given some vector of values.

#### Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_beta_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_logistic_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_tparam_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Gaussian: tidy_inverse_normal(), tidy_normal(), util_normal_stats_tbl()
```

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_normal_param_estimate(x)
output$parameter_tbl</pre>
```

util\_normal\_stats\_tbl

```
output$combined_data_tbl |>
   tidy_combined_autoplot()

t <- rnorm(50, 0, 1)
util_normal_param_estimate(t)$parameter_tbl</pre>
```

util\_normal\_stats\_tbl Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

### Usage

```
util_normal_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Gaussian: tidy_inverse_normal(), tidy_normal(), util_normal_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_biutil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

util\_paralogistic\_aic 181

## **Examples**

```
library(dplyr)

tidy_normal() |>
  util_normal_stats_tbl() |>
  glimpse()
```

util\_paralogistic\_aic Calculate Akaike Information Criterion (AIC) for Paralogistic Distribution

## **Description**

This function estimates the shape and rate parameters of a paralogistic distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_paralogistic_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a paralogistic distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a paralogistic distribution fitted to the provided data.

This function fits a paralogistic distribution to the provided data using maximum likelihood estimation. It estimates the shape and rate parameters of the paralogistic distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and rate parameters of the paralogistic distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted paralogistic distribution to the provided data.

# Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_pareto_aic(), util_poisson_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()

Other Paralogistic: util_paralogistic_param_estimate(), util_paralogistic_stats_tbl()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- tidy_paralogistic(30, .shape = 2, .rate = 1)[["y"]]
util_paralogistic_aic(x)</pre>
```

```
util_paralogistic_param_estimate
```

Estimate Paralogistic Parameters

## Description

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated paralogistic data.

The method of parameter estimation is:

• MLE

## Usage

```
util_paralogistic_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function.

```
.auto\_gen\_empirical
```

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the paralogistic shape and rate parameters given some vector of values.

#### Value

A tibble/list

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_hypergeometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_to_param_estimate(), util_to_param_estimate(), util_to_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Paralogistic: util_paralogistic_aic(), util_paralogistic_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_paralogistic_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

t <- tidy_paralogistic(50, 2.5, 1.4)[["y"]]
util_paralogistic_param_estimate(t)$parameter_tbl</pre>
```

```
util_paralogistic_stats_tbl
```

Distribution Statistics for Paralogistic Distribution

# Description

Returns distribution statistics in a tibble.

184 util\_pareto1\_aic

### **Usage**

```
util_paralogistic_stats_tbl(.data)
```

### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

## See Also

```
Other Paralogistic: util_paralogistic_aic(), util_paralogistic_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_poisson_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_butil_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_negative_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

# **Examples**

```
library(dplyr)
set.seed(123)
tidy_paralogistic(.n = 50, .shape = 5, .rate = 6) |>
  util_paralogistic_stats_tbl() |>
  glimpse()
```

util\_pareto1\_aic

Calculate Akaike Information Criterion (AIC) for Pareto Distribution

## **Description**

This function estimates the shape and scale parameters of a Pareto distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

util\_pareto1\_aic 185

## Usage

```
util_pareto1_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a Pareto distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Pareto distribution fitted to the provided data.

This function fits a Pareto distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the Pareto distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the Pareto distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

## Value

The AIC value calculated based on the fitted Pareto distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto_aic(), util_poisson_aic(), util_t_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()

Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto_param_estimate(), util_pareto5_stats_tbl()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- tidy_pareto1()$y
util_pareto1_aic(x)</pre>
```

```
util_pareto1_param_estimate
```

Estimate Pareto Parameters

### **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated Pareto data.

Two different methods of shape parameters are supplied:

- LSE
- MLE

## Usage

```
util_pareto1_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function.

```
.auto_gen_empirical
```

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the Pareto shape and scale parameters given some vector of values.

## Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_binomial_param_estimate(), util_logistic_param_estimate(), util_logistic_param_estimate(), util_param_estimate(), util_pa
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars[["mpg"]]
output <- util_pareto1_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

set.seed(123)
t <- tidy_pareto1(.n = 100, .shape = 1.5, .min = 1)[["y"]]
util_pareto1_param_estimate(t)$parameter_tbl</pre>
```

```
util_pareto1_stats_tbl

Distribution Statistics for Pareto1 Distribution
```

# **Description**

Returns distribution statistics in a tibble.

# Usage

```
util_pareto1_stats_tbl(.data)
```

## **Arguments**

. data The data being passed from a tidy\_ distribution function.

188 util\_pareto\_aic

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

#### See Also

```
Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto_param_estimate(), util_pareto_stats_tbl

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl

util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_pareto1() |>
  util_pareto1_stats_tbl() |>
  glimpse()
```

util\_pareto\_aic

Calculate Akaike Information Criterion (AIC) for Pareto Distribution

## **Description**

This function estimates the shape and scale parameters of a Pareto distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### **Usage**

```
util_pareto_aic(.x)
```

## **Arguments**

.x A numeric vector containing the data to be fitted to a Pareto distribution.

util\_pareto\_aic 189

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Pareto distribution fitted to the provided data.

This function fits a Pareto distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the Pareto distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the Pareto distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted Pareto distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_poisson_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

# **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- TidyDensity::tidy_pareto()$y
util_pareto_aic(x)</pre>
```

```
util_pareto_param_estimate
```

Estimate Pareto Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated pareto data.

Two different methods of shape parameters are supplied:

- LSE
- MLE

### Usage

```
util_pareto_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function..auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

## **Details**

This function will attempt to estimate the pareto shape and scale parameters given some vector of values.

### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_invers
```

util\_pareto\_stats\_tbl 191

```
util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate() util_lognormal_param_estimate(), util_normal_param_estimate() util_paralogistic_param_estimate(), util_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()

Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto1_stats_tbl()
```

# **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_pareto_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

t <- tidy_pareto(50, 1, 1) |> pull(y)
util_pareto_param_estimate(t)$parameter_tbl
```

```
util_pareto_stats_tbl Distribution Statistics
```

## Description

Returns distribution statistics in a tibble.

## Usage

```
util_pareto_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

192 util\_poisson\_aic

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Pareto: tidy_generalized_pareto(), tidy_inverse_pareto(), tidy_pareto(), tidy_pareto1(), util_pareto1_aic(), util_pareto1_param_estimate(), util_pareto1_stats_tbl(), util_pareto1_param_estimate().

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto1_stats_tbl(), util_pareto1_stats_tbl(), util_poisson_stats_tbl(), util_tstats_tbl(), util_tstats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_pareto() |>
  util_pareto_stats_tbl() |>
  glimpse()
```

util\_poisson\_aic

Calculate Akaike Information Criterion (AIC) for Poisson Distribution

## **Description**

This function estimates the lambda parameter of a Poisson distribution from the provided data and then calculates the AIC value based on the fitted distribution.

### **Usage**

```
util_poisson_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a Poisson distribution.

util\_poisson\_aic 193

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Poisson distribution fitted to the provided data.

This function fits a Poisson distribution to the provided data. It estimates the lambda parameter of the Poisson distribution from the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimate as a starting point for the lambda parameter of the Poisson distribution.

Optimization method: Since the parameter is directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted Poisson distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_pareto1_aic(), util_pareto1_aic(), util_pareto_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_poisson_aic()
```

# **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rpois(100, lambda = 2)
util_poisson_aic(x)</pre>
```

```
util_poisson_param_estimate
```

Estimate Poisson Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated poisson data.

## Usage

```
util_poisson_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the pareto lambda parameter given some vector of values.

# Value

A tibble/list

# Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_t_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
```

util\_poisson\_stats\_tbl 195

```
Other Poisson: tidy_poisson(), tidy_zero_truncated_poisson(), util_poisson_stats_tbl(), util_zero_truncated_poisson_param_estimate(), util_zero_truncated_poisson_stats_tbl()
```

# **Examples**

```
library(dplyr)
library(ggplot2)

x <- as.integer(mtcars$mpg)
output <- util_poisson_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

t <- rpois(50, 5)
util_poisson_param_estimate(t)$parameter_tbl</pre>
```

```
util_poisson_stats_tbl
```

Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_poisson_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

# Author(s)

Steven P. Sanderson II, MPH

196 util\_triangular\_aic

### See Also

```
Other Poisson: tidy_poisson(), tidy_zero_truncated_poisson(), util_poisson_param_estimate(), util_zero_truncated_poisson_param_estimate(), util_zero_truncated_poisson_stats_tbl()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_paralogistic_stats_tbl(), util_paralogistic_stats_tbl(), util_parato_stats_tbl(), util_parato_stats_tbl(), util_parato_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_poisson() |>
  util_poisson_stats_tbl() |>
  glimpse()
```

# **Description**

This function estimates the parameters of a triangular distribution (min, max, and mode) from the provided data and calculates the AIC value based on the fitted distribution.

### **Usage**

```
util_triangular_aic(.x)
```

## Arguments

.x A numeric vector containing the data to be fitted to a triangular distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a triangular distribution fitted to the provided data.

The function operates in several steps:

util\_triangular\_aic 197

1. **Parameter Estimation**: The function extracts the minimum, maximum, and mode values from the data via the TidyDensity::util\_triangular\_param\_estimate function. It returns these initial parameters as the starting point for optimization.

- 2. **Negative Log-Likelihood Calculation**: A custom function calculates the negative log-likelihood using the EnvStats::dtri function to obtain density values for each data point. The densities are logged manually to simulate the behavior of a log parameter.
- 3. **Parameter Validation**: During optimization, the function checks that the constraints min <= mode <= max are met, and returns an infinite loss if not.
- 4. **Optimization**: The optimization process utilizes the "SANN" (Simulated Annealing) method to minimize the negative log-likelihood and find optimal parameter values.
- 5. **AIC Calculation**: The Akaike Information Criterion (AIC) is calculated using the optimized negative log-likelihood and the total number of parameters (3).

### Value

The AIC value calculated based on the fitted triangular distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_pareto_aic(), util_inverse_pareto_aic(),
```

# **Examples**

```
# Example: Calculate AIC for a sample dataset
set.seed(123)
data <- tidy_triangular(.min = 0, .max = 1, .mode = 1/2)$y
util_triangular_aic(data)</pre>
```

```
util_triangular_param_estimate
```

Estimate Triangular Parameters

## **Description**

This function will attempt to estimate the triangular min, mode, and max parameters given some vector of values.

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated beta data.

## Usage

```
util_triangular_param_estimate(.x, .auto_gen_empirical = TRUE)
```

# **Arguments**

.x The vector of data to be passed to the function. Must be numeric, and all values must be  $0 \le x \le 1$ 

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the triangular min, mode, and max parameters given some vector of values.

## Value

A tibble/list

# Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_param_estimate(), util_generalized_param_estimate(), util_inverse_binr_param_estimate(), util_inverse_binr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_logistic_pa
```

```
util_lognormal_param_estimate(), util_negative_binomial_param_estimate(), util_normal_param_estimate() util_paralogistic_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Util_triangular: tidy_triangular(), util_triangular_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- mtcars$mpg
output <- util_triangular_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

params <- tidy_triangular()$y |>
    util_triangular_param_estimate()
params$parameter_tbl
```

```
util_triangular_stats_tbl

Distribution Statistics
```

# Description

Returns distribution statistics in a tibble.

## Usage

```
util_triangular_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ distribution function.

### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

200 util\_t\_aic

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

# **Examples**

```
library(dplyr)

tidy_triangular() |>
  util_triangular_stats_tbl() |>
  glimpse()
```

util\_t\_aic

Calculate Akaike Information Criterion (AIC) for t Distribution

## **Description**

This function estimates the parameters of a t distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

### **Usage**

```
util_t_aic(.x)
```

# Arguments

.x A numeric vector containing the data to be fitted to a t distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a t distribution fitted to the provided data.

This function fits a t distribution to the input data using maximum likelihood estimation and then computes the Akaike Information Criterion (AIC) based on the fitted distribution.

util\_t\_param\_estimate 201

### Value

The AIC value calculated based on the fitted t distribution to the provided data.

# Author(s)

Steven P. Sanderson II, MPH

### See Also

rt for generating t-distributed data, optim for optimization.

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_romal_aic(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Generate t-distributed data
set.seed(123)
x <- rt(100, df = 5, ncp = 0.5)
# Calculate AIC for the generated data
util_t_aic(x)</pre>
```

### **Description**

Estimate t Distribution Parameters

## Usage

```
util_t_param_estimate(.x, .auto_gen_empirical = TRUE)
```

## **Arguments**

.x The vector of data to be passed to the function, where the data comes from the rt() function.

202 util\_t\_param\_estimate

```
.auto_gen_empirical
```

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

### **Details**

This function will attempt to estimate the t distribution parameters given some vector of values produced by rt(). The estimation method uses both method of moments and maximum likelihood estimation.

### Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_gamma_estimate(), util_gamma_estimate(), util_generalized_beta_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_lognormal_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

set.seed(123)
x <- rt(100, df = 10, ncp = 0.5)
output <- util_t_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

util\_t\_stats\_tbl 203

util\_t\_stats\_tbl

Distribution Statistics

# **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_t_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other T Distribution: tidy_t()
```

```
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

204 util\_uniform\_aic

## **Examples**

```
library(dplyr)

tidy_t() |>
  util_t_stats_tbl() |>
  glimpse()
```

util\_uniform\_aic

Calculate Akaike Information Criterion (AIC) for Uniform Distribu-

# **Description**

This function estimates the min and max parameters of a uniform distribution from the provided data and then calculates the AIC value based on the fitted distribution.

### Usage

```
util_uniform_aic(.x)
```

## **Arguments**

.x A numeric vector containing the data to be fitted to a uniform distribution.

# **Details**

This function calculates the Akaike Information Criterion (AIC) for a uniform distribution fitted to the provided data.

This function fits a uniform distribution to the provided data. It estimates the min and max parameters of the uniform distribution from the range of the data. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the minimum and maximum values of the data as starting points for the min and max parameters of the uniform distribution.

Optimization method: Since the parameters are directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

## Value

The AIC value calculated based on the fitted uniform distribution to the provided data.

## Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto1_aic(), util_pareto_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- runif(30)
util_uniform_aic(x)</pre>
```

```
util_uniform_param_estimate
```

Estimate Uniform Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated uniform data.

# Usage

```
util_uniform_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### **Arguments**

. x The vector of data to be passed to the function.

```
.auto_gen_empirical
```

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

## **Details**

This function will attempt to estimate the uniform min and max parameters given some vector of values.

### Value

A tibble/list

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_est
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- tidy_uniform(.min = 1, .max = 3)$y
output <- util_uniform_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_uniform_stats_tbl
```

Distribution Statistics

## **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_uniform_stats_tbl(.data)
```

util\_weibull\_aic 207

### **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Uniform: tidy_uniform(), util_uniform_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_veibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

## **Examples**

```
library(dplyr)

tidy_uniform() |>
  util_uniform_stats_tbl() |>
  glimpse()
```

util\_weibull\_aic

Calculate Akaike Information Criterion (AIC) for Weibull Distribution

## **Description**

This function estimates the shape and scale parameters of a Weibull distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

208 util\_weibull\_aic

### Usage

```
util_weibull_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data to be fitted to a Weibull distribution.

#### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Weibull distribution fitted to the provided data.

This function fits a Weibull distribution to the provided data using maximum likelihood estimation. It estimates the shape and scale parameters of the Weibull distribution using maximum likelihood estimation. Then, it calculates the AIC value based on the fitted distribution.

Initial parameter estimates: The function uses the method of moments estimates as starting points for the shape and scale parameters of the Weibull distribution.

Optimization method: The function uses the optim function for optimization. You might explore different optimization methods within optim for potentially better performance.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

#### Value

The AIC value calculated based on the fitted Weibull distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto1_aic(), util_pareto_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

### **Examples**

```
# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rweibull(100, shape = 2, scale = 1)
util_weibull_aic(x)</pre>
```

```
util_weibull_param_estimate
```

Estimate Weibull Parameters

### Description

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated weibull data.

### Usage

```
util_weibull_param_estimate(.x, .auto_gen_empirical = TRUE)
```

# **Arguments**

. x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

# Details

This function will attempt to estimate the weibull shape and scale parameters given some vector of values.

### Value

A tibble/list

## Author(s)

Steven P. Sanderson II, MPH

# See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_param_estimate(), util_param_estim
```

210 util\_weibull\_stats\_tbl

```
util_uniform_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_geoutil_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()

Other Weibull: tidy_inverse_weibull(), tidy_weibull(), util_weibull_stats_tbl()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- tidy_weibull(.shape = 1, .scale = 2)$y
output <- util_weibull_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl %>%
    tidy_combined_autoplot()
```

```
util_weibull_stats_tbl
```

Distribution Statistics

# **Description**

Returns distribution statistics in a tibble.

## Usage

```
util_weibull_stats_tbl(.data)
```

# Arguments

.data

The data being passed from a tidy\_ distribution function.

### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

# Value

A tibble

## Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Weibull: tidy_inverse_weibull(), tidy_weibull(), util_weibull_param_estimate()

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_verse_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stats_tbl(), util_zero_truncated_poisson_stats_tbl(), util_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)

tidy_weibull() |>
  util_weibull_stats_tbl() |>
  glimpse()
```

```
util_zero_truncated_binomial_aic
```

Calculate Akaike Information Criterion (AIC) for Zero-Truncated Binomial Distribution

# **Description**

This function estimates the parameters (size and prob) of a ZTB distribution from the provided data using maximum likelihood estimation (via the optim() function), and then calculates the AIC value based on the fitted distribution.

## Usage

```
util_zero_truncated_binomial_aic(.x)
```

### **Arguments**

.x A numeric vector containing the data (non-zero counts) to be fitted to a ZTB distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a zero-truncated binomial (ZTB) distribution fitted to the provided data.

**Initial parameter estimates:** The choice of initial values for size and prob can impact the convergence of the optimization. Consider using prior knowledge or method of moments estimates to obtain reasonable starting values.

**Optimization method:** The default optimization method used is "L-BFGS-B," which allows for box constraints to keep the parameters within valid bounds. You might explore other optimization methods available in optim() for potentially better performance or different constraint requirements.

**Data requirements:** The input data .x should consist of non-zero counts, as the ZTB distribution does not include zero values. Additionally, the values in .x should be less than or equal to the estimated size parameter.

**Goodness-of-fit:** While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen ZTB model using visualization (e.g., probability plots, histograms) and other statistical tests (e.g., chi-square goodness-of-fit test) to ensure it adequately describes the data.

#### Value

The AIC value calculated based on the fitted ZTB distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_generalized_pareto_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_poisson_aic()
```

## **Examples**

```
# Example data
set.seed(123)
x <- tidy_zero_truncated_binomial(30, .size = 10, .prob = 0.4)[["y"]]
# Calculate AIC
util_zero_truncated_binomial_aic(x)</pre>
```

util\_zero\_truncated\_binomial\_param\_estimate

Estimate Zero Truncated Binomial Parameters

## **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated binomial data.

One method of estimating the parameters is done via:

• MLE via optim function.

### Usage

```
util_zero_truncated_binomial_param_estimate(.x, .auto_gen_empirical = TRUE)
```

### Arguments

.x The vector of data to be passed to the function..auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

## **Details**

This function will attempt to estimate the zero truncated binomial size and prob parameters given some vector of values.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_binomial_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_normal_param_estimate()
```

```
util_paralogistic_param_estimate(), util_pareto1_param_estimate(), util_pareto_param_estimate(),
util_poisson_param_estimate(), util_t_param_estimate(), util_triangular_param_estimate(),
util_uniform_param_estimate(), util_weibull_param_estimate(), util_zero_truncated_geometric_param_est
util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate()
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(),
tidy_zero_truncated_negative_binomial(), util_binomial_param_estimate(), util_binomial_stats_tbl(),
util_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
Other Zero Truncated Distribution: tidy_zero_truncated_binomial(), tidy_zero_truncated_geometric(),
tidy_zero_truncated_poisson()
```

## **Examples**

```
library(dplyr)
library(ggplot2)

x <- as.integer(mtcars$mpg)
output <- util_zero_truncated_binomial_param_estimate(x)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

set.seed(123)
t <- tidy_zero_truncated_binomial(100, 10, .1)[["y"]]
util_zero_truncated_binomial_param_estimate(t)$parameter_tbl</pre>
```

```
util_zero_truncated_binomial_stats_tbl
```

Distribution Statistics for Zero Truncated Binomial Distribution

# Description

Returns distribution statistics in a tibble for Zero Truncated Binomial distribution.

### Usage

```
util_zero_truncated_binomial_stats_tbl(.data)
```

# **Arguments**

.data

The data being passed from a tidy\_ distribution function.

#### **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

### Value

A tibble

#### See Also

```
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), util_binomial_param_estimate(), util_binomial_stats_tbl(), util_negative_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_negative_binomial_param_estimate(), util_zero_truncated_negative_binomial_stats_
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_lognormal_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_geometric_stats_tbl(), util_zero_truncated_negative_binoutil_zero_truncated_poisson_stats_tbl()
```

### **Examples**

```
library(dplyr)
set.seed(123)
tidy_zero_truncated_binomial(.size = 10, .prob = 0.1) |>
  util_zero_truncated_binomial_stats_tbl() |>
  glimpse()
```

```
util_zero_truncated_geometric_aic
```

Calculate Akaike Information Criterion (AIC) for Zero-Truncated Geometric Distribution

## **Description**

This function estimates the probability parameter of a Zero-Truncated Geometric distribution from the provided data and calculates the AIC value based on the fitted distribution.

### Usage

```
util_zero_truncated_geometric_aic(.x)
```

#### **Arguments**

. x A numeric vector containing the data to be fitted to a Zero-Truncated Geometric distribution.

### **Details**

This function calculates the Akaike Information Criterion (AIC) for a Zero-Truncated Geometric distribution fitted to the provided data.

This function fits a Zero-Truncated Geometric distribution to the provided data. It estimates the probability parameter using the method of moments and calculates the AIC value.

Optimization method: Since the parameter is directly calculated from the data, no optimization is needed.

Goodness-of-fit: While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen model using visualization and other statistical tests.

### Value

The AIC value calculated based on the fitted Zero-Truncated Geometric distribution to the provided data.

### Author(s)

Steven P. Sanderson II, MPH

### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto_aic(), util_pareto_aic(), util_reacto(), util_triangular_aic(), util_uniform_aic(), util_weibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_negative_binomial_aic(), util_zero_truncated_logistic_aic(), util
```

## **Examples**

```
library(actuar)

# Example: Calculate AIC for a sample dataset
set.seed(123)
x <- rztgeom(100, prob = 0.2)
util_zero_truncated_geometric_aic(x)</pre>
```

util\_zero\_truncated\_geometric\_param\_estimate

Estimate Zero-Truncated Geometric Parameters

#### **Description**

This function will estimate the prob parameter for a Zero-Truncated Geometric distribution from a given vector .x. The function returns a list with a parameter table, and if .auto\_gen\_empirical is set to TRUE, the empirical data is combined with the estimated distribution data.

# Usage

```
util_zero_truncated_geometric_param_estimate(.x, .auto_gen_empirical = TRUE)
```

#### **Arguments**

.x The vector of data to be passed to the function. Must contain non-negative integers and should have no zeros.

.auto\_gen\_empirical

Boolean value (default TRUE) that, when set to TRUE, will generate tidy\_empirical() output for .x and combine it with the estimated distribution data.

#### **Details**

This function will attempt to estimate the prob parameter of the Zero-Truncated Geometric distribution using given vector .x as input data. If the parameter .auto\_gen\_empirical is set to TRUE, the empirical data in .x will be run through the tidy\_empirical() function and combined with the estimated zero-truncated geometric data.

# Value

A tibble/list

# See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_gamma_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_generalized_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_burr_param_estimate(), util_logistic_param_estimate(), util_lognormal_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_poisson_param_estimate())
Other Zero-Truncated Geometric: util_zero_truncated_geometric_stats_tbl()
```

# **Examples**

```
library(actuar)
library(dplyr)
library(ggplot2)
library(actuar)
```

```
set.seed(123)
ztg <- rztgeom(100, prob = 0.2)
output <- util_zero_truncated_geometric_param_estimate(ztg)
output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

```
util_zero_truncated_geometric_stats_tbl
```

Distribution Statistics for Zero-Truncated Geometric

#### **Description**

Returns distribution statistics for Zero-Truncated Geometric distribution in a tibble.

#### Usage

```
util_zero_truncated_geometric_stats_tbl(.data)
```

## **Arguments**

.data

The data being passed from a tidy\_ztgeom distribution function.

#### **Details**

This function takes in a tibble generated by a tidy\_ztgeom distribution function and returns the relevant statistics for a Zero-Truncated Geometric distribution. It requires data to be passed from a tidy\_ztgeom distribution function.

# Value

A tibble

#### See Also

```
Other Zero-Truncated Geometric: util_zero_truncated_geometric_param_estimate()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_normal_stats_tbl(), util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_negative_binomutil_zero_truncated_poisson_stats_tbl()
```

# **Examples**

```
library(dplyr)
set.seed(123)
tidy_zero_truncated_geometric(.prob = 0.1) |>
  util_zero_truncated_geometric_stats_tbl() |>
  glimpse()
```

```
util_zero_truncated_negative_binomial_aic
```

Calculate Akaike Information Criterion (AIC) for Zero-Truncated Negative Binomial Distribution

# **Description**

This function estimates the parameters (size and prob) of a ZTNB distribution from the provided data using maximum likelihood estimation (via the optim() function), and then calculates the AIC value based on the fitted distribution.

# Usage

```
util_zero_truncated_negative_binomial_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data (non-zero counts) to be fitted to a ZTNB distribution.

## **Details**

This function calculates the Akaike Information Criterion (AIC) for a zero-truncated negative binomial (ZTNB) distribution fitted to the provided data.

**Initial parameter estimates:** The choice of initial values for size and prob can impact the convergence of the optimization. Consider using prior knowledge or method of moments estimates to obtain reasonable starting values.

**Optimization method:** The default optimization method used is "Nelder-Mead". You might explore other optimization methods available in optim() for potentially better performance or different constraint requirements.

**Data requirements:** The input data .x should consist of non-zero counts, as the ZTNB distribution does not include zero values.

**Goodness-of-fit:** While AIC is a useful metric for model comparison, it's recommended to also assess the goodness-of-fit of the chosen ZTNB model using visualization (e.g., probability plots, histograms) and other statistical tests (e.g., chi-square goodness-of-fit test) to ensure it adequately describes the data.

#### Value

The AIC value calculated based on the fitted ZTNB distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_hypergeometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto1_aic(), util_pareto1_aic(), util_veibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_poisson_index.
```

## **Examples**

```
library(actuar)

# Example data
set.seed(123)
x <- rztnbinom(30, size = 2, prob = 0.4)

# Calculate AIC
util_zero_truncated_negative_binomial_aic(x)</pre>
```

```
util_zero_truncated_negative_binomial_param_estimate

Estimate Zero Truncated Negative Binomial Parameters
```

#### **Description**

The function will return a list output by default, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated negative binomial data.

One method of estimating the parameters is done via:

• MLE via optim function.

#### Usage

```
util_zero_truncated_negative_binomial_param_estimate(
   .x,
   .auto_gen_empirical = TRUE
)
```

## **Arguments**

.x The vector of data to be passed to the function.

.auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function will attempt to estimate the zero truncated negative binomial size and prob parameters given some vector of values.

#### Value

A tibble/list

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(),
util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(),
util_chisquare_param_estimate(), util_exponential_param_estimate(), util_f_param_estimate(),
util_gamma_param_estimate(),util_generalized_beta_param_estimate(),util_generalized_pareto_param_es
util_geometric_param_estimate(),util_hypergeometric_param_estimate(),util_inverse_burr_param_estima
util_inverse_pareto_param_estimate(), util_inverse_weibull_param_estimate(), util_logistic_param_esti
util_lognormal_param_estimate(), util_negative_binomial_param_estimate(), util_normal_param_estimate(
util_paralogistic_param_estimate(),util_pareto1_param_estimate(),util_pareto_param_estimate(),
util_poisson_param_estimate(), util_t_param_estimate(), util_triangular_param_estimate(),
util_uniform_param_estimate(),util_weibull_param_estimate(),util_zero_truncated_binomial_param_esti
util_zero_truncated_geometric_param_estimate(),util_zero_truncated_poisson_param_estimate()
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(),
tidy_zero_truncated_negative_binomial(), util_binomial_param_estimate(), util_binomial_stats_tbl(),
util_negative_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate(),
util_zero_truncated_binomial_stats_tbl(),util_zero_truncated_negative_binomial_stats_tbl()
Other Zero Truncated Negative Distribution: tidy_zero_truncated_negative_binomial()
```

## **Examples**

```
library(dplyr)
library(ggplot2)
library(actuar)

x <- as.integer(mtcars$mpg)
output <- util_zero_truncated_negative_binomial_param_estimate(x)</pre>
```

```
output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()

set.seed(123)
t <- rztnbinom(100, 10, .1)
util_zero_truncated_negative_binomial_param_estimate(t)$parameter_tbl</pre>
```

```
util\_zero\_truncated\_negative\_binomial\_stats\_tbl\\ Distribution\ Statistics\ for\ Zero-Truncated\ Negative\ Binomial
```

# Description

Computes distribution statistics for a zero-truncated negative binomial distribution.

# Usage

```
util_zero_truncated_negative_binomial_stats_tbl(.data)
```

## **Arguments**

.data

The data from a zero-truncated negative binomial distribution.

#### **Details**

This function computes statistics for a zero-truncated negative binomial distribution.

# Value

A tibble with distribution statistics.

# Author(s)

Steven P. Sanderson II, MPH

## See Also

```
Other Binomial: tidy_binomial(), tidy_negative_binomial(), tidy_zero_truncated_binomial(), tidy_zero_truncated_negative_binomial(), util_binomial_param_estimate(), util_binomial_stats_tbl(), util_negative_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_binomial_param_estimate()
Other Negative Binomial: util_negative_binomial_stats_tbl()
Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto.
```

```
util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(),
util_inverse_pareto_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(),
util_lognormal_stats_tbl(), util_negative_binomial_stats_tbl(), util_normal_stats_tbl(),
util_paralogistic_stats_tbl(), util_pareto1_stats_tbl(), util_pareto_stats_tbl(),
util_poisson_stats_tbl(), util_t_stats_tbl(), util_triangular_stats_tbl(), util_uniform_stats_tbl(),
util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_sta
util_zero_truncated_poisson_stats_tbl()
```

# **Examples**

```
library(dplyr)

tidy_zero_truncated_negative_binomial(.size = 1, .prob = 0.1) |>
  util_zero_truncated_negative_binomial_stats_tbl() |>
  glimpse()
```

```
util_zero_truncated_poisson_aic
```

Calculate Akaike Information Criterion (AIC) for zero-truncated poisson Distribution

# Description

This function estimates the parameters of a zero-truncated poisson distribution from the provided data using maximum likelihood estimation, and then calculates the AIC value based on the fitted distribution.

#### **Usage**

```
util_zero_truncated_poisson_aic(.x)
```

# **Arguments**

.x A numeric vector containing the data to be fitted to a zero-truncated poisson distribution.

#### Details

This function calculates the Akaike Information Criterion (AIC) for a zero-truncated poisson distribution fitted to the provided data.

# Value

The AIC value calculated based on the fitted zero-truncated poisson distribution to the provided data.

#### Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Utility: check_duplicate_rows(), convert_to_ts(), quantile_normalize(), tidy_mcmc_sampling(), util_beta_aic(), util_binomial_aic(), util_cauchy_aic(), util_chisq_aic(), util_exponential_aic(), util_f_aic(), util_gamma_aic(), util_generalized_beta_aic(), util_generalized_pareto_aic(), util_geometric_aic(), util_inverse_burr_aic(), util_inverse_pareto_aic(), util_inverse_weibull_aic(), util_logistic_aic(), util_lognormal_aic(), util_negative_binomial_aic(), util_normal_aic(), util_paralogistic_aic(), util_pareto1_aic(), util_pareto_aic(), util_pareto1_aic(), util_pareto1_aic(), util_veibull_aic(), util_zero_truncated_binomial_aic(), util_zero_truncated_geometric_aic(), util_zero_truncated_negative_aic(), util_zero_truncated_negative_aic
```

## **Examples**

```
library(actuar)

# Example 1: Calculate AIC for a sample dataset
set.seed(123)
x <- rztpois(30, lambda = 3)
util_zero_truncated_poisson_aic(x)</pre>
```

```
util_zero_truncated_poisson_param_estimate

Estimate Zero Truncated Poisson Parameters
```

# **Description**

This function will attempt to estimate the Zero Truncated Poisson lambda parameter given some vector of values .x. The function will return a tibble output, and if the parameter .auto\_gen\_empirical is set to TRUE then the empirical data given to the parameter .x will be run through the tidy\_empirical() function and combined with the estimated Zero Truncated Poisson data.

# Usage

```
util_zero_truncated_poisson_param_estimate(.x, .auto_gen_empirical = TRUE)
```

# **Arguments**

.x The vector of data to be passed to the function. Must be non-negative integers. .auto\_gen\_empirical

This is a boolean value of TRUE/FALSE with default set to TRUE. This will automatically create the tidy\_empirical() output for the .x parameter and use the tidy\_combine\_distributions(). The user can then plot out the data using \$combined\_data\_tbl from the function output.

#### **Details**

This function estimates the parameter lambda of a Zero-Truncated Poisson distribution based on a vector of non-negative integer values .x. The Zero-Truncated Poisson distribution is a discrete probability distribution that models the number of events occurring in a fixed interval of time, given that at least one event has occurred.

The estimation is performed by minimizing the negative log-likelihood of the observed data .x under the Zero-Truncated Poisson model. The negative log-likelihood function used for optimization is defined as:

$$-\sum_{i=1}^{n} \log(P(X_i = x_i \mid X_i > 0, \lambda))$$

where  $\$  ( $\$  X\_i \) are the observed values in .x and lambda is the parameter of the Zero-Truncated Poisson distribution.

The optimization process uses the optim function to find the value of lambda that minimizes this negative log-likelihood. The chosen optimization method is Brent's method (method = "Brent") within a specified interval  $[0, \max(.x)]$ .

If .auto\_gen\_empirical is set to TRUE, the function will generate empirical data statistics using tidy\_empirical() for the input data .x and then combine this empirical data with the estimated Zero-Truncated Poisson distribution using tidy\_combine\_distributions(). This combined data can be accessed via the \$combined\_data\_tbl element of the function output.

The function returns a tibble containing the estimated parameter lambda along with other summary statistics of the input data (sample size, minimum, maximum).

#### Value

A tibble/list

# Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Parameter Estimation: util_bernoulli_param_estimate(), util_beta_param_estimate(), util_binomial_param_estimate(), util_burr_param_estimate(), util_cauchy_param_estimate(), util_chisquare_param_estimate(), util_param_estimate(), util_f_param_estimate(), util_gamma_param_estimate(), util_generalized_beta_param_estimate(), util_generalized_pareto_param_estimate(), util_geometric_param_estimate(), util_hypergeometric_param_estimate(), util_inverse_burr_param_estimate(), util_inverse_pareto_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_triangular_param_estimate(), util_uniform_param_estimate(), util_param_estimate(), util_zero_truncated_binomial_param_estimate(), util_zero_truncated_param_estimate(), util_poisson_param_estimate(), util_param_estimate(), util_pa
```

# **Examples**

```
library(dplyr)
library(ggplot2)

tc <- tidy_zero_truncated_poisson() |> pull(y)
output <- util_zero_truncated_poisson_param_estimate(tc)

output$parameter_tbl

output$combined_data_tbl |>
    tidy_combined_autoplot()
```

# Description

Returns distribution statistics in a tibble.

# Usage

```
util_zero_truncated_poisson_stats_tbl(.data)
```

# Arguments

.data

The data being passed from a tidy\_ distribution function.

# **Details**

This function will take in a tibble and returns the statistics of the given type of tidy\_ distribution. It is required that data be passed from a tidy\_ distribution function.

## Value

A tibble

# Author(s)

Steven P. Sanderson II, MPH

#### See Also

```
Other Poisson: tidy_poisson(), tidy_zero_truncated_poisson(), util_poisson_param_estimate(), util_poisson_stats_tbl(), util_zero_truncated_poisson_param_estimate()

Other Distribution Statistics: util_bernoulli_stats_tbl(), util_beta_stats_tbl(), util_binomial_stats_tbl(), util_burr_stats_tbl(), util_cauchy_stats_tbl(), util_chisquare_stats_tbl(), util_exponential_stats_tbl util_f_stats_tbl(), util_gamma_stats_tbl(), util_generalized_beta_stats_tbl(), util_generalized_pareto_util_geometric_stats_tbl(), util_hypergeometric_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_burr_stats_tbl(), util_inverse_weibull_stats_tbl(), util_logistic_stats_tbl(), util_lognormal_stats_tbl(), util_paratol_stats_tbl(), util_paratol_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_pareto_stats_tbl(), util_uniform_stats_tbl(), util_weibull_stats_tbl(), util_zero_truncated_binomial_stats_tbl(), util_zero_truncated_geometric_stautil_zero_truncated_negative_binomial_stats_tbl()
```

# Examples

```
library(dplyr)

tidy_zero_truncated_poisson() |>
  util_zero_truncated_poisson_stats_tbl() |>
  glimpse()
```

# **Index**

```
util_zero_truncated_negative_binomial_stats_tbl,
* Augment Function
    bootstrap_density_augment, 5
                                                * Bootstrap
    bootstrap_p_augment, 6
                                                    bootstrap_density_augment, 5
    bootstrap_q_augment, 8
* Autoplot
                                                    bootstrap_p_augment, 6
    bootstrap_stat_plot, 10
                                                    bootstrap_p_vec, 7
    tidy_autoplot, 28
                                                    bootstrap_q_augment, 8
    tidy_combined_autoplot, 40
                                                    bootstrap_q_vec, 9
    tidy_four_autoplot, 50
                                                    bootstrap_stat_plot, 10
                                                    bootstrap_unnest_tbl, 12
    tidy_multi_dist_autoplot, 75
                                                    tidy_bootstrap, 34
    tidy_random_walk_autoplot, 88
                                                * Burr
* Bernoulli
    tidy_bernoulli, 30
                                                    tidy_burr, 35
                                                    tidy_inverse_burr, 59
    util_bernoulli_param_estimate, 107
                                                    util_burr_param_estimate, 117
    util_bernoulli_stats_tbl, 108
* Beta
                                                    util_burr_stats_tbl, 118
                                                * Cauchy
    tidy_beta, 31
                                                    tidy_cauchy, 37
    tidy_generalized_beta, 53
                                                    util_cauchy_param_estimate, 121
    util_beta_param_estimate, 110
                                                    util_cauchy_stats_tbl, 122
    util_beta_stats_tbl, 112
                                                * Chisquare
* Binomaial
                                                    tidy_chisquare, 38
    util_negative_binomial_stats_tbl,
                                                    util_chisquare_param_estimate, 123
        176
* Binomial
                                                    util_chisquare_stats_tbl, 125
                                                * Continuous Distribution
    tidy_binomial, 33
                                                    tidy_beta, 31
    tidy_negative_binomial, 78
                                                    tidy_burr, 35
    tidy_zero_truncated_binomial, 100
    tidy_zero_truncated_negative_binomial,
                                                    tidy_cauchy, 37
                                                    tidy_chisquare, 38
        103
                                                    tidy_exponential, 47
    util_binomial_param_estimate, 114
                                                    tidy_f, 48
    util_binomial_stats_tbl, 116
                                                    tidy_gamma, 51
    util_negative_binomial_param_estimate,
                                                    tidy_generalized_beta, 53
    \verb|util_zero_truncated_binomial_param_estimate|, | \verb|tidy_generalized_pareto|, 54| \\
                                                    tidy_geometric, 56
                                                    tidy_inverse_burr, 59
    util_zero_truncated_binomial_stats_tbl,
                                                    tidy_inverse_exponential, 61
    util_zero_truncated_negative_binomial_param_etidynaineyerse_gamma, 62
        220
                                                    tidy_inverse_normal, 64
```

tidy_inverse_pareto, 65	util_paralogistic_stats_tbl, 183
tidy_inverse_weibull,67	util_pareto1_stats_tbl, 187
tidy_logistic,69	util_pareto_stats_tbl, 191
tidy_lognormal, 71	util_poisson_stats_tbl, 195
tidy_normal, 80	util_t_stats_tbl, 203
tidy_paralogistic,81	util_triangular_stats_tbl, 199
tidy_pareto, 83	util_uniform_stats_tbl, 206
tidy_pareto1,84	util_weibull_stats_tbl, 210
tidy_t, 94	util_zero_truncated_binomial_stats_tbl,
tidy_triangular,96	214
tidy_uniform, 97	<pre>util_zero_truncated_geometric_stats_tbl,</pre>
tidy_weibull,98	218
tidy_zero_truncated_geometric, 101	util_zero_truncated_negative_binomial_stats_tbl,
* Discrete Distribution	222
tidy_bernoulli,30	<pre>util_zero_truncated_poisson_stats_tbl,</pre>
tidy_binomial, 33	226
tidy_hypergeometric, 57	* Empirical
tidy_negative_binomial, 78	tidy_distribution_comparison, 43
tidy_poisson, 86	* Exponential
tidy_zero_truncated_binomial, 100	tidy_exponential, 47
tidy_zero_truncated_negative_binomial,	tidy_inverse_exponential, 61
103	util_exponential_param_estimate,
tidy_zero_truncated_poisson, 104	128
* Distribution Statistics	util_exponential_stats_tbl, 130
util_bernoulli_stats_tbl, 108	* F Distribution
util_beta_stats_tbl, 112	tidy_f, 48
util_binomial_stats_tbl, 116	util_f_param_estimate, 132
util_burr_stats_tbl, 118	util_f_stats_tbl, 133
util_cauchy_stats_tbl, 122	* Gamma
util_chisquare_stats_tbl, 125	tidy_gamma,51
util_exponential_stats_tbl, 130	tidy_inverse_gamma, 62
util_f_stats_tbl, 133	util_gamma_param_estimate, 136
util_gamma_stats_tbl, 137	util_gamma_stats_tbl, 137
util_generalized_beta_stats_tbl,	* Gaussian
141	tidy_inverse_normal, 64
<pre>util_generalized_pareto_stats_tbl,</pre>	tidy_normal, 80
145	util_normal_param_estimate, 178
util_geometric_stats_tbl, 149	util_normal_stats_tbl, 180
util_hypergeometric_stats_tbl, 153	* Generalized Beta
util_inverse_burr_stats_tbl, 157	util_generalized_beta_param_estimate,
util_inverse_pareto_stats_tbl, 160	140
util_inverse_weibull_stats_tbl,	util_generalized_beta_stats_tbl,
164	141
util_logistic_stats_tbl, 168	* Generalized Pareto
util_lognormal_stats_tbl, 172	util_generalized_pareto_param_estimate,
util_negative_binomial_stats_tbl,	144
176	util_generalized_pareto_stats_tbl,
util_normal_stats_tbl, 180	145

* Geometric	176
tidy_geometric, 56	<pre>util_zero_truncated_negative_binomial_stats_tbl,</pre>
$tidy\_zero\_truncated\_geometric, 101$	222
util_geometric_param_estimate, 147	* Negative Distribution
util_geometric_stats_tbl, 149	tidy_negative_binomial,78
* Helper	* Paralogistic
<pre>dist_type_extractor, 25</pre>	util_paralogistic_aic, 181
* Hypergeometric	util_paralogistic_param_estimate,
tidy_hypergeometric,57	182
<pre>util_hypergeometric_param_estimate,</pre>	util_paralogistic_stats_tbl, 183
151	* Parameter Estimation
<pre>util_hypergeometric_stats_tbl, 153</pre>	util_bernoulli_param_estimate, 107
* Inverse Burr	util_beta_param_estimate, 110
util_inverse_burr_param_estimate,	util_binomial_param_estimate, 114
155	util_burr_param_estimate, 117
util_inverse_burr_stats_tbl, 157	util_cauchy_param_estimate, 121
* Inverse Distribution	util_chisquare_param_estimate, 123
tidy_inverse_burr,59	util_exponential_param_estimate,
tidy_inverse_exponential, 61	128
tidy_inverse_gamma,62	util_f_param_estimate, 132
tidy_inverse_normal, 64	util_gamma_param_estimate, 136
tidy_inverse_pareto, 65	util_generalized_beta_param_estimate,
tidy_inverse_weibull,67	140
* Inverse Pareto	<pre>util_generalized_pareto_param_estimate,</pre>
util_inverse_pareto_param_estimate,	144
159	util_geometric_param_estimate, 147
util_inverse_pareto_stats_tbl, 160	util_hypergeometric_param_estimate,
* Inverse Weibull	151
util_inverse_weibull_param_estimate,	util_inverse_burr_param_estimate,
163	155
util_inverse_weibull_stats_tbl,	util_inverse_pareto_param_estimate,
164	159
* Logistic	util_inverse_weibull_param_estimate,
tidy_logistic, 69	163
tidy_paralogistic, 81	util_logistic_param_estimate, 167
util_logistic_param_estimate, 167	util_lognormal_param_estimate, 171
util_logistic_stats_tbl, 168	util_negative_binomial_param_estimate,
* Lognormal	175
tidy_lognormal, 71	util_normal_param_estimate, 178
util_lognormal_param_estimate, 171	util_paralogistic_param_estimate, 182
util_lognormal_stats_tbl, 172 * Mixture Data	util_pareto1_param_estimate, 186
tidy_mixture_density, 73	util_pareto_param_estimate, 190
* Multiple Distribution	
* Multiple Distribution tidy_combine_distributions, 42	util_poisson_param_estimate, 194
tidy_combine_distributions, 42 tidy_multi_single_dist, 77	util_t_param_estimate, 201
* Negative Binomial	util_triangular_param_estimate, 198
util negative hinomial stats th	util uniform param estimate 205

util_weibull_param_estimate,209	tidy_uniform, 97
util_zero_truncated_binomial_param_estimate,	util_uniform_param_estimate, 205
213	util_uniform_stats_tbl, 206
util_zero_truncated_geometric_param_estima <b>t</b>	ețility
216	<pre>check_duplicate_rows, 14</pre>
util_zero_truncated_negative_binomial_param_	_e <b>sdrivarte_</b> to_ts, 21
220	quantile_normalize, 26
util_zero_truncated_poisson_param_estimate,	tidy_mcmc_sampling,72
224	util_beta_aic, 109
* Pareto	util_binomial_aic, 113
tidy_generalized_pareto,54	util_cauchy_aic, 119
tidy_inverse_pareto,65	util_chisq_aic, 126
tidy_pareto, 83	util_exponential_aic, 127
tidy_pareto1,84	util_f_aic, 131
util_pareto1_aic, 184	util_gamma_aic, 134
util_pareto1_param_estimate, 186	util_generalized_beta_aic, 138
util_pareto1_stats_tbl, 187	util_generalized_pareto_aic, 142
util_pareto_param_estimate, 190	util_geometric_aic, 146
util_pareto_stats_tbl, 191	util_hypergeometric_aic, 150
* Poisson	util_inverse_burr_aic, 154
tidy_poisson, $86$	util_inverse_pareto_aic, 158
tidy_zero_truncated_poisson, 104	util_inverse_weibull_aic, 162
util_poisson_param_estimate, 194	util_logistic_aic, 165
util_poisson_stats_tbl, 195	util_lognormal_aic, 169
util_zero_truncated_poisson_param_estimate,	util_negative_binomial_aic, 173
224	util_normal_aic, 177
util_zero_truncated_poisson_stats_tbl,	util_paralogistic_aic, 181
226	util_pareto1_aic, 184
* Statistic	util_pareto_aic, 188
ci_hi, 16	util_poisson_aic, 192
ci_lo, 17	$util_t_aic, 200$
tidy_kurtosis_vec,68	util_triangular_aic, 196
tidy_range_statistic, 90	util_uniform_aic, 204
tidy_skewness_vec,92	util_weibull_aic, 207
tidy_stat_tbl, 93	util_zero_truncated_binomial_aic,
* Summary Statistics	211
tidy_distribution_summary_tbl,45	util_zero_truncated_geometric_aic,
* T Distribution	215
tidy_t,94	util_zero_truncated_negative_binomial_aic
util_t_stats_tbl, 203	219
* Table Data	util_zero_truncated_poisson_aic,
tidy_distribution_summary_tbl,45	223
e	ector Function
tidy_triangular,96	bootstrap_p_vec, 7
util_triangular_param_estimate,	bootstrap_q_vec, 9
198	cgmean, 13
util_triangular_stats_tbl, 199	chmean, 15
* Uniform	ckurtosis, 18

cmean, 19	<pre>actuar::rztnbinom(), 103</pre>
cmedian, 20	actuar::rztpois(), 105
csd, 22	anyDuplicated, <i>14</i>
cskewness, 23	apply, 27
cvar, 24	
tidy_kurtosis_vec,68	bootstrap_density_augment, 5, 7-9, 11, 12,
tidy_scale_zero_one_vec,91	35
tidy_skewness_vec,92	bootstrap_p_augment, 6, 6, 8, 9, 11, 12, 35
* Visualization	bootstrap_p_vec, 6, 7, 7, 9, 11–13, 15,
triangle_plot, 106	18–20, 23–25, 35, 69, 91, 92
* Weibull	bootstrap_q_augment, 6-8, 8, 9, 11, 12, 35
tidy_inverse_weibull,67	bootstrap_q_vec, 6-9, 9, 11-13, 15, 18-20,
tidy_weibull, 98	23–25, 35, 69, 91, 92
$\verb util_weibull_param_estimate , 209 $	bootstrap_stat_plot, 6-9, 10, 12, 29, 35,
util_weibull_stats_tbl, 210	41, 51, 76, 89
* Zero Truncated Distribution	bootstrap_unnest_tbl, 6-9, 11, 12, 35
tidy_zero_truncated_binomial, 100	
tidy_zero_truncated_geometric, 101	cgmean, 8, 9, 13, 15, 18–20, 23–25, 69, 91, 92
tidy_zero_truncated_poisson, 104	check_duplicate_rows, 14, 22, 27, 73, 110,
util_zero_truncated_binomial_param_estim	ate, 114, 120, 127, 128, 131, 135, 139,
213	143, 147, 151, 155, 159, 163, 166,
* Zero Truncated Negative Distribution	170, 174, 178, 182, 185, 189, 193,
<pre>tidy_zero_truncated_negative_binomial,</pre>	197, 201, 205, 208, 212, 216, 220,
103	224
util_zero_truncated_negative_binomial_pa	
220	ci_hi, 16, 17, 69, 90, 92, 94
* Zero Truncated	ci_lo, 16, 17, 69, 90, 92, 94
<pre>util_zero_truncated_poisson_stats_tbl,</pre>	ckurtosis, 8, 9, 13, 15, 18, 19, 20, 23–25, 69, 91, 92
* Zero-Truncated Geometric	cmean, 8, 9, 13, 15, 18, 19, 20, 23–25, 69, 91,
util_zero_truncated_geometric_param_esti	mate, 92
216	cmedian, 8, 9, 13, 15, 18, 19, 20, 23–25, 69,
<pre>util_zero_truncated_geometric_stats_tbl,</pre>	91, 92
218	color_blind, 21
* t Distribution	convert_to_ts, 14, 21, 27, 73, 110, 114, 120,
util_t_param_estimate, 201	127, 128, 131, 135, 139, 143, 147,
	151, 155, 159, 163, 166, 170, 174,
actuar::rburr(), 36	178, 182, 185, 189, 193, 197, 201,
actuar::rgenpareto(), 55	205, 208, 212, 216, 220, 224
actuar::rinvburr(),60	csd, 8, 9, 13, 15, 18–20, 22, 24, 25, 69, 91, 92
actuar::rinvexp(), 61	cskewness, 8, 9, 13, 15, 18–20, 23, 23, 25, 69,
actuar::rinvgamma(), 63	91, 92
actuar::rinvpareto(), 66	cvar, 8, 9, 13, 15, 18–20, 23, 24, 24, 69, 91, 92
actuar::rinvweibull(), 68	
actuar::rparalogis(), 82	dist_type_extractor, 25
actuar::rpareto(), 83	dplyr::cummean(), 19
actuar::rpareto1(),85	dplyr::group_by(),45
actuar::rztbinom(), 100	dplyr::select(),45
( 100	
actuar::rztgeom(), 102	duplicated, <i>14</i>

EnvStats::ebeta(), 110 EnvStats::elnorm(), 171	tidy_binomial, 31, 33, 59, 79, 87, 101, 104, 105, 115, 116, 176, 214, 215, 221,
	222
optim, 131, 175, 201, 213, 220 order, 27	tidy_bootstrap, 6-9, 11, 12, 34 tidy_burr, 32, 35, 38, 39, 48, 49, 52, 54, 56, 57, 60, 62, 63, 65, 66, 68, 70, 72, 81,
quantile_normalize, 14, 22, 26, 73, 110, 114, 120, 127, 128, 131, 135, 139, 143, 147, 151, 155, 159, 163, 166, 170, 174, 178, 182, 185, 189, 193, 197, 201, 205, 208, 212, 216, 220, 224	82, 84, 85, 95, 97-99, 102, 118, 119 tidy_cauchy, 32, 36, 37, 39, 48, 49, 52, 54, 56, 57, 60, 62, 63, 65, 66, 68, 70, 72, 81, 82, 84, 85, 95, 97-99, 102, 122, 123 tidy_chisquare, 32, 36, 38, 38, 48, 49, 52,
rf, <i>131</i>	54, 56, 57, 60, 62, 63, 65, 66, 68, 70,
rinvgauss(), 65	72, 81, 82, 84, 85, 95, 97–99, 102,
rlang::enquo(), 6, 8	124, 126
rowMeans, 27	tidy_combine_distributions, 42, 77
rt, 201	tidy_combined_autoplot, 11, 29, 40, 51, 76, 89
stats::density(), 30, 31, 33, 35, 37, 38, 47, 48, 51, 53, 55, 56, 58, 59, 61, 62, 64, 65, 67, 69, 71, 78, 80, 81, 83, 84, 86, 95, 97, 98, 100, 101, 103, 104	tidy_distribution_comparison, 43 tidy_distribution_summary_tbl, 45 tidy_empirical, 46 tidy_exponential, 32, 36, 38, 39, 47, 49, 52,
stats::rbeta(), 32, 54	54, 56, 57, 60, 62, 63, 65, 66, 68, 70
stats::rbinom(), 33	72, 81, 82, 84, 85, 95, 97–99, 102,
stats::rcauchy(), 38	129, 130
stats::rchisq(), 39	tidy_f, 32, 36, 38, 39, 48, 48, 52, 54, 56, 57,
stats::rexp(), 48	60, 62, 63, 65, 66, 68, 70, 72, 81, 82,
stats::rf(),49	84, 85, 95, 97–99, 102, 133, 134
stats::rgamma(), 52	tidy_four_autoplot, 11, 29, 41, 50, 76, 89
stats::rgeom(), 57	tidy_gamma, 32, 36, 38, 39, 48, 49, 51, 54, 56,
stats::rhyper(), 58	57, 60, 62, 63, 65, 66, 68, 70, 72, 81,
stats::rlnorm(), 71	82, 84, 85, 95, 97–99, 102, 137, 138
stats::rlogis(), 70	tidy_generalized_beta, 32, 36, 38, 39, 48,
stats::rnbinom(), 79	49, 52, 53, 56, 57, 60, 62, 63, 65, 66,
stats::rnorm(), $80$	68, 70, 72, 81, 82, 84, 85, 95, 97–99
stats::rpois(), 86	102, 111, 112
stats::rt(), 95	tidy_generalized_pareto, 32, 36, 38, 39,
stats::runif(), 98	48, 49, 52, 54, 54, 57, 60, 62, 63, 65,
stats::rweibull(),99	66, 68, 70, 72, 81, 82, 84, 85, 95, 97–99, 102, 185, 187, 188, 191, 192
td_scale_color_colorblind, 27	tidy_geometric, 32, 36, 38, 39, 48, 49, 52,
td_scale_fill_colorblind, 28	54, 56, 56, 60, 62, 63, 65, 66, 68, 70,
tidy_autoplot, 11, 28, 41, 51, 76, 89	72, 81, 82, 84, 85, 95, 97–99, 102,
tidy_bernoulli, 30, 34, 59, 79, 87, 101, 104,	148, 149
105, 108, 109	tidy_hypergeometric, 31, 34, 57, 79, 87,
tidy_beta, 31, 36, 38, 39, 48, 49, 52, 54, 56,	101, 104, 105, 152, 154
57, 60, 62, 63, 65, 66, 68, 70, 72, 81,	tidy_inverse_burr, 32, 36, 38, 39, 48, 49,
82, 84, 85, 95, 97–99, 102, 111, 112	52, 54, 56, 57, 59, 62, 63, 65, 66, 68

```
70, 72, 81, 82, 84, 85, 95, 97–99,
          102, 118, 119
tidy_inverse_exponential, 32, 36, 38, 39,
         48, 49, 52, 54, 56, 57, 60, 61, 63, 65,
         66, 68, 70, 72, 81, 82, 84, 85, 95,
          97–99, 102, 129, 130
tidy_inverse_gamma, 32, 36, 38, 39, 48, 49,
         52, 54, 56, 57, 60, 62, 62, 65, 66, 68,
          70, 72, 81, 82, 84, 85, 95, 97–99,
          102, 137, 138
tidy_inverse_normal, 32, 36, 38, 39, 48, 49,
         52, 54, 56, 57, 60, 62, 63, 64, 66, 68,
          70, 72, 81, 82, 84, 85, 95, 97–99,
          102, 179, 180
tidy_inverse_pareto, 32, 36, 38, 39, 48, 49,
         52, 54, 56, 57, 60, 62, 63, 65, 65, 68,
          70, 72, 81, 82, 84, 85, 95, 97–99,
          102, 185, 187, 188, 191, 192
tidy_inverse_weibull, 32, 36, 38, 39, 48,
         49, 52, 54, 56, 57, 60, 62, 63, 65, 66,
         67, 70, 72, 81, 82, 84, 85, 95, 97–99,
         102, 210, 211
tidy_kurtosis_vec, 8, 9, 13, 15-20, 23-25,
          68, 90-92, 94
tidy_logistic, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 69, 72,
         81, 82, 84, 85, 95, 97–99, 102, 168,
tidy_lognormal, 32, 36, 38, 39, 48, 49, 52,
         54, 56, 57, 60, 62, 63, 65, 66, 68, 70,
         71, 81, 82, 84, 85, 95, 97–99, 102,
         172, 173
tidy_mcmc_sampling, 14, 22, 27, 72, 110,
         114, 120, 127, 128, 131, 135, 139,
         143, 147, 151, 155, 159, 163, 166,
         170, 174, 178, 182, 185, 189, 193,
          197, 201, 205, 208, 212, 216, 220,
         224
tidy_mixture_density, 73
tidy_multi_dist_autoplot, 11, 29, 41, 51,
         75, 89
tidy_multi_single_dist, 42,77
tidy_negative_binomial, 31, 34, 59, 78, 87,
          101, 104, 105, 115, 116, 176, 214,
          215, 221, 222
tidy_normal, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 70, 72,
         80, 82, 84, 85, 95, 97–99, 102, 179,
```

```
180
tidy_paralogistic, 32, 36, 38, 39, 48, 49,
         52, 54, 56, 57, 60, 62, 63, 65, 66, 68,
         70, 72, 81, 81, 84, 85, 95, 97–99,
         102, 168, 169
tidy_pareto, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 70, 72,
         81, 82, 83, 85, 95, 97–99, 102, 185,
         187, 188, 191, 192
tidy_pareto1, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 70, 72,
         81, 82, 84, 84, 95, 97–99, 102, 185,
         187, 188, 191, 192
tidy_poisson, 31, 34, 59, 79, 86, 101, 104,
         105, 195, 196, 225, 227
tidy_random_walk, 87
tidy_random_walk_autoplot, 11, 29, 41, 51,
         76,88
tidy_range_statistic, 16, 17, 69, 90, 92, 94
tidy_scale_zero_one_vec, 8, 9, 13, 15,
         18-20, 23-25, 69, 91, 92
tidy_skewness_vec, 8, 9, 13, 15-20, 23-25,
         69, 90, 91, 92, 94
tidy_stat_tbl, 16, 17, 69, 90, 92, 93
tidy_t, 32, 36, 38, 39, 48, 49, 52, 54, 56, 57,
         60, 62, 63, 65, 66, 68, 70, 72, 81, 82,
         84, 85, 94, 97–99, 102, 203
tidy_triangular, 32, 36, 38, 39, 48, 49, 52,
         54, 56, 57, 60, 62, 63, 65, 66, 68, 70,
         72, 81, 82, 84, 85, 95, 96, 98, 99,
         102, 199, 200
tidy_uniform, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 70, 72,
         81, 82, 84, 85, 95, 97, 97, 99, 102,
         206, 207
tidy_weibull, 32, 36, 38, 39, 48, 49, 52, 54,
         56, 57, 60, 62, 63, 65, 66, 68, 70, 72,
         81, 82, 84, 85, 95, 97, 98, 98, 102,
         210, 211
tidy_zero_truncated_binomial, 31, 34, 59,
         79, 87, 100, 102, 104, 105, 115, 116,
         176, 214, 215, 221, 222
tidy_zero_truncated_geometric, 32, 36,
         38, 39, 48, 49, 52, 54, 56, 57, 60, 62,
         63, 65, 66, 68, 70, 72, 81, 82, 84, 85,
         95, 97–99, 101, 101, 105, 148, 149,
```

214

tidy\_zero\_truncated\_negative\_binomial,

31, 34, 59, 79, 87, 101, 103, 105,	109, 112, 115, 116, 119, 123, 126,
115, 116, 176, 214, 215, 221, 222	130, 134, 138, 142, 146, 149, 154,
tidy_zero_truncated_poisson, 31, 34, 59,	157, 161, 165, 169, 173, 176, 177,
79, 87, 101, 102, 104, 104, 195, 196,	180, 184, 188, 192, 196, 200, 203,
214, 225, 227	207, 211, 214, 215, 218, 221, 222,
triangle_plot, 106	227
- '	util_burr_param_estimate, 36, 60, 107,
util_bernoulli_param_estimate, 31, 107,	111, 115, 117, 119, 121, 124, 129,
109, 111, 115, 118, 121, 124, 129,	133, 136, 140, 144, 148, 152, 156,
133, 136, 140, 144, 148, 152, 156,	160, 164, 168, 171, 176, 179, 183,
160, 164, 168, 171, 176, 179, 183,	187, 190, 194, 198, 202, 206, 209,
187, 190, 194, 198, 202, 206, 209,	213, 217, 221, 225
213, 217, 221, 225	util_burr_stats_tbl, 36, 60, 109, 112, 116,
util_bernoulli_stats_tbl, 31, 108, 108,	118, 118, 123, 126, 130, 134, 138,
112, 116, 119, 123, 126, 130, 134,	142, 146, 149, 154, 157, 161, 165,
138, 142, 146, 149, 154, 157, 161,	169, 173, 177, 180, 184, 188, 192,
165, 169, 173, 177, 180, 184, 188,	196, 200, 203, 207, 211, 215, 218,
192, 196, 200, 203, 207, 211, 215,	222, 227
218, 222, 227	
util_beta_aic, 14, 22, 27, 73, 109, 114, 120,	util_cauchy_aic, 14, 22, 27, 73, 110, 114,
127, 128, 131, 135, 139, 143, 147,	119, 127, 128, 131, 135, 139, 143,
151, 155, 159, 163, 166, 170, 174,	147, 151, 155, 159, 163, 166, 170,
178, 182, 185, 189, 193, 197, 201,	174, 178, 182, 185, 189, 193, 197,
205, 208, 212, 216, 220, 224	201, 205, 208, 212, 216, 220, 224
util_beta_param_estimate, 32, 54, 107,	util_cauchy_param_estimate, 38, 107, 111,
110, 112, 115, 118, 121, 124, 129,	115, 118, 121, 123, 124, 129, 133,
133, 136, 140, 144, 148, 152, 156,	136, 140, 144, 148, 152, 156, 160,
160, 164, 168, 171, 176, 179, 183,	164, 168, 171, 176, 179, 183, 187,
187, 190, 194, 198, 202, 206, 209,	190, 194, 198, 202, 206, 209, 213,
213, 217, 221, 225	217, 221, 225
util_beta_stats_tbl, 32, 54, 109, 111, 112,	util_cauchy_stats_tbl, 38, 109, 112, 116,
116, 119, 123, 126, 130, 134, 138,	119, 122, 122, 126, 130, 134, 138,
142, 146, 149, 154, 157, 161, 165,	142, 146, 149, 154, 157, 161, 165,
169, 173, 177, 180, 184, 188, 192,	169, 173, 177, 180, 184, 188, 192,
196, 200, 203, 207, 211, 215, 218,	196, 200, 203, 207, 211, 215, 218,
222, 227	222, 227
util_binomial_aic, 14, 22, 27, 73, 110, 113,	util_chisq_aic, 14, 22, 27, 73, 110, 114,
120, 127, 128, 131, 135, 139, 143,	<i>120</i> , 126, <i>128</i> , <i>131</i> , <i>135</i> , <i>139</i> , <i>143</i> ,
147, 151, 155, 159, 163, 166, 170,	147, 151, 155, 159, 163, 166, 170,
174, 178, 182, 185, 189, 193, 197,	174, 178, 182, 185, 189, 193, 197,
201, 205, 208, 212, 216, 220, 224	201, 205, 208, 212, 216, 220, 224
util_binomial_param_estimate, 34, 79,	util_chisquare_param_estimate, 39, 107,
101, 104, 107, 111, 114, 116, 118,	111, 115, 118, 121, 123, 126, 129,
121, 124, 129, 133, 136, 140, 144,	133, 136, 140, 144, 148, 152, 156,
148, 152, 156, 160, 164, 168, 171,	160, 164, 168, 171, 176, 179, 183,
176, 179, 183, 187, 190, 194, 198,	187, 190, 194, 198, 202, 206, 209,
202, 206, 209, 213–215, 217, 221,	213, 217, 221, 225
222, 225	util_chisquare_stats_tbl, 39, 109, 112,
util_binomial_stats_tbl, 34, 79, 101, 104,	116, 119, 123, 124, 125, 130, 134,

138, 142, 146, 149, 154, 157, 161,	187, 190, 194, 198, 202, 206, 209,
165, 169, 173, 177, 180, 184, 188,	213, 217, 221, 225
192, 196, 200, 203, 207, 211, 215,	util_gamma_stats_tbl, 52, 63, 109, 112,
218, 222, 227	116, 119, 123, 126, 130, 134, 137,
util_exponential_aic, <i>14</i> , <i>22</i> , <i>27</i> , <i>73</i> , <i>110</i> ,	137, 142, 146, 149, 154, 157, 161,
114, 120, 127, 127, 131, 135, 139,	165, 169, 173, 177, 180, 184, 188,
143, 147, 151, 155, 159, 163, 166,	192, 196, 200, 203, 207, 211, 215,
170, 174, 178, 182, 185, 189, 193,	218, 222, 227
197, 201, 205, 208, 212, 216, 220,	util_generalized_beta_aic, 14, 22, 27, 73,
224	110, 114, 120, 127, 128, 131, 135,
util_exponential_param_estimate, 48, 62,	138, 143, 147, 151, 155, 159, 163,
107, 111, 115, 118, 121, 124, 128,	166, 170, 174, 178, 182, 185, 189,
130, 133, 136, 140, 144, 148, 152,	193, 197, 201, 205, 208, 212, 216,
156, 160, 164, 168, 171, 176, 179,	220, 224
183, 187, 190, 194, 198, 202, 206,	<pre>util_generalized_beta_param_estimate,</pre>
209, 213, 217, 221, 225	107, 111, 115, 118, 121, 124, 129,
util_exponential_stats_tbl, 48, 62, 109,	133, 136, 140, 142, 144, 148, 152,
<i>112</i> , <i>116</i> , <i>119</i> , <i>123</i> , <i>126</i> , <i>129</i> , 130,	156, 160, 164, 168, 171, 176, 179,
134, 138, 142, 146, 149, 154, 157,	183, 187, 190, 194, 198, 202, 206,
161, 165, 169, 173, 177, 180, 184,	209, 213, 217, 221, 225
188, 192, 196, 200, 203, 207, 211,	util_generalized_beta_stats_tbl, 109,
215, 218, 222, 227	112, 116, 119, 123, 126, 130, 134,
util_f_aic, 14, 22, 27, 73, 110, 114, 120,	<i>138</i> , <i>141</i> , 141, <i>146</i> , <i>149</i> , <i>154</i> , <i>157</i> ,
<i>127, 128,</i> 131 <i>, 135, 139, 143, 147</i> ,	161, 165, 169, 173, 177, 180, 184,
151, 155, 159, 163, 166, 170, 174,	188, 192, 196, 200, 203, 207, 211,
178, 182, 185, 189, 193, 197, 201,	215, 218, 222, 227
205, 208, 212, 216, 220, 224	util_generalized_pareto_aic, 14, 22, 27,
util_f_param_estimate, 49, 107, 111, 115,	73, 110, 114, 120, 127, 128, 131,
118, 121, 124, 129, 132, 134, 136,	<i>135</i> , <i>139</i> , 142, <i>147</i> , <i>151</i> , <i>155</i> , <i>159</i> ,
140, 144, 148, 152, 156, 160, 164,	163, 166, 170, 174, 178, 182, 185,
168, 171, 176, 179, 183, 187, 190,	189, 193, 197, 201, 205, 208, 212,
194, 198, 202, 206, 209, 213, 217,	216, 220, 224
221, 225	util_generalized_pareto_param_estimate,
util_f_stats_tbl, 49, 109, 112, 116, 119,	107, 111, 115, 118, 121, 124, 129,
123, 126, 130, 133, 133, 138, 142,	133, 136, 140, 144, 146, 148, 152,
146, 149, 154, 157, 161, 165, 169,	156, 160, 164, 168, 171, 176, 179,
173, 177, 180, 184, 188, 192, 196,	183, 187, 190, 194, 198, 202, 206,
200, 203, 207, 211, 215, 218, 222,	209, 213, 217, 221, 225
227	util_generalized_pareto_stats_tbl, 109,
util_gamma_aic, <i>14</i> , <i>22</i> , <i>27</i> , <i>73</i> , <i>110</i> , <i>114</i> ,	112, 116, 119, 123, 126, 130, 134,
<i>120, 127, 128, 131,</i> 134, <i>139, 143,</i>	138, 142, 145, 145, 149, 154, 157,
147, 151, 155, 159, 163, 166, 170,	161, 165, 169, 173, 177, 180, 184,
174, 178, 182, 185, 189, 193, 197,	188, 192, 196, 200, 203, 207, 211,
201, 205, 208, 212, 216, 220, 224	215, 218, 222, 227
util_gamma_param_estimate, 52, 63, 107,	util_geometric_aic, 14, 22, 27, 73, 110,
111, 115, 118, 121, 124, 129, 133,	114, 120, 127, 128, 131, 135, 139,
136, 138, 140, 144, 148, 152, 156,	<i>143</i> , 146, <i>151</i> , <i>155</i> , <i>159</i> , <i>163</i> , <i>166</i> ,
160, 164, 168, 171, 176, 179, 183,	170, 174, 178, 182, 185, 189, 193,

197, 201, 205, 208, 212, 216, 220,	192, 196, 200, 203, 207, 211, 215,
224	218, 223, 227
util_geometric_param_estimate, 57, 102,	util_inverse_pareto_aic, 14, 22, 27, 73,
107, 111, 115, 118, 121, 124, 129,	110, 114, 120, 127, 128, 131, 135,
133, 136, 140, 144, 147, 149, 152,	139, 143, 147, 151, 155, 158, 163,
156, 160, 164, 168, 171, 176, 179,	166, 170, 174, 178, 182, 185, 189,
183, 187, 190, 194, 198, 202, 206,	193, 197, 201, 205, 208, 212, 216,
209, 213, 217, 221, 225	220, 224
util_geometric_stats_tbl, 57, 102, 109,	util_inverse_pareto_param_estimate,
112, 116, 119, 123, 126, 130, 134,	107, 111, 115, 118, 121, 124, 129,
138, 142, 146, 148, 149, 154, 157,	133, 136, 140, 144, 148, 152, 156,
161, 165, 169, 173, 177, 180, 184,	159, 161, 164, 168, 172, 176, 179,
188, 192, 196, 200, 203, 207, 211,	183, 187, 191, 194, 198, 202, 206,
215, 218, 223, 227	209, 213, 217, 221, 225
util_hypergeometric_aic, <i>14</i> , <i>22</i> , <i>27</i> , <i>73</i> ,	util_inverse_pareto_stats_tbl, 109, 112
110, 114, 120, 127, 128, 131, 135,	116, 119, 123, 126, 130, 134, 138,
139, 143, 147, 150, 155, 159, 163,	142, 146, 149, 154, 157, 160, 160,
166, 170, 174, 178, 182, 185, 189,	165, 169, 173, 177, 180, 184, 188,
193, 197, 201, 205, 208, 212, 216,	192, 196, 200, 203, 207, 211, 215,
220, 224	218, 223, 227
util_hypergeometric_param_estimate, 59,	util_inverse_weibull_aic, 14, 22, 27, 73,
107, 111, 115, 118, 121, 124, 129,	110, 114, 120, 127, 128, 131, 135,
133, 136, 140, 144, 148, 151, 154,	139, 143, 147, 151, 155, 159, 162,
156, 160, 164, 168, 171, 176, 179,	166, 170, 174, 178, 182, 185, 189,
183, 187, 190, 194, 198, 202, 206,	193, 197, 201, 205, 208, 212, 216,
209, 213, 217, 221, 225	220, 224
util_hypergeometric_stats_tbl, 59, 109,	util_inverse_weibull_param_estimate,
112, 116, 119, 123, 126, 130, 134,	107, 111, 115, 118, 121, 124, 129,
138, 142, 146, 149, 152, 153, 157,	133, 136, 140, 144, 148, 152, 156,
161, 165, 169, 173, 177, 180, 184,	160, 163, 165, 168, 172, 176, 179,
188, 192, 196, 200, 203, 207, 211,	183, 187, 191, 194, 198, 202, 206,
215, 218, 223, 227	209, 213, 217, 221, 225
util_inverse_burr_aic, 14, 22, 27, 73, 110,	util_inverse_weibull_stats_tbl, 109,
114, 120, 127, 128, 131, 135, 139,	112, 116, 119, 123, 126, 130, 134,
143, 147, 151, 154, 159, 163, 166,	138, 142, 146, 149, 154, 157, 161,
170, 174, 178, 182, 185, 189, 193,	164, 164, 169, 173, 177, 180, 184,
197, 201, 205, 208, 212, 216, 220,	188, 192, 196, 200, 203, 207, 211,
224	215, 218, 223, 227
util_inverse_burr_param_estimate, 107,	util_logistic_aic, 14, 22, 27, 73, 110, 114
111, 115, 118, 121, 124, 129, 133,	120, 127, 128, 131, 135, 139, 143,
136, 140, 144, 148, 152, 155, 157,	120, 127, 128, 131, 133, 139, 143, 147, 151, 155, 159, 163, 165, 170,
160, 164, 168, 171, 176, 179, 183,	174, 178, 182, 185, 189, 193, 197,
187, 190, 194, 198, 202, 206, 209,	201, 205, 208, 212, 216, 220, 224
213, 217, 221, 225	
	util_logistic_param_estimate, 70, 82,
util_inverse_burr_stats_tbl, 109, 112,	107, 111, 115, 118, 121, 124, 129,
116, 119, 123, 126, 130, 134, 138,	133, 136, 140, 144, 148, 152, 156,
142, 146, 149, 154, 156, 157, 161,	160, 164, 167, 169, 172, 176, 179,
165, 169, 173, 177, 180, 184, 188,	183, 187, 191, 194, 198, 202, 206,

209, 213, 217, 221, 225	201, 205, 208, 212, 216, 220, 224
util_logistic_stats_tbl, 70, 82, 109, 112,	util_normal_param_estimate, 65, 81, 107,
116, 119, 123, 126, 130, 134, 138,	111, 115, 118, 121, 124, 129, 133,
142, 146, 149, 154, 157, 161, 165,	136, 140, 144, 148, 152, 156, 160,
168, 168, 173, 177, 180, 184, 188,	164, 168, 172, 176, 178, 180, 183,
192, 196, 200, 203, 207, 211, 215,	187, 191, 194, 199, 202, 206, 209,
218, 223, 227	213, 217, 221, 225
util_lognormal_aic, <i>14</i> , <i>22</i> , <i>27</i> , <i>73</i> , <i>110</i> ,	util_normal_stats_tbl, 65, 81, 109, 112,
114, 120, 127, 128, 131, 135, 139,	116, 119, 123, 126, 130, 134, 138,
143, 147, 151, 155, 159, 163, 166,	142, 146, 149, 154, 157, 161, 165,
169, 174, 178, 182, 185, 189, 193,	<i>169, 173, 177, 179,</i> 180 <i>, 184, 188</i> ,
197, 201, 205, 208, 212, 216, 220,	192, 196, 200, 203, 207, 211, 215,
224	218, 223, 227
util_lognormal_param_estimate, 72, 107,	util_paralogistic_aic, 14, 22, 27, 73, 110,
111, 115, 118, 121, 124, 129, 133,	114, 120, 127, 128, 132, 135, 139,
136, 140, 144, 148, 152, 156, 160,	143, 147, 151, 155, 159, 163, 166,
164, 168, 171, 173, 176, 179, 183,	170, 174, 178, 181, 183–185, 189,
187, 191, 194, 199, 202, 206, 209,	193, 197, 201, 205, 208, 212, 216,
213, 217, 221, 225	220, 224
util_lognormal_stats_tbl, 72, 109, 112,	util_paralogistic_param_estimate, 107,
116, 119, 123, 126, 130, 134, 138,	111, 115, 118, 121, 124, 129, 133,
142, 146, 149, 154, 157, 161, 165,	136, 140, 144, 148, 152, 156, 160,
169, 172, 172, 177, 180, 184, 188,	164, 168, 172, 176, 179, 182, 182,
192, 196, 200, 203, 207, 211, 215,	184, 187, 191, 194, 199, 202, 206,
218, 223, 227	209, 214, 217, 221, 225
util_negative_binomial_aic, <i>14</i> , 22, 27,	util_paralogistic_stats_tbl, 109, 112,
73, 110, 114, 120, 127, 128, 131,	116, 119, 123, 126, 130, 134, 138,
135, 139, 143, 147, 151, 155, 159,	142, 146, 149, 154, 157, 161, 165,
163, 166, 170, 173, 178, 182, 185,	169, 173, 177, 180, 182, 183, 183,
189, 193, 197, 201, 205, 208, 212,	188, 192, 196, 200, 203, 207, 211,
216, 220, 224	215, 218, 223, 227
util_negative_binomial_param_estimate,	util_pareto1_aic, 14, 22, 27, 56, 66, 73, 84,
34, 79, 101, 104, 107, 111, 115, 116,	85, 110, 114, 120, 127, 128, 132,
118, 121, 124, 129, 133, 136, 140,	135, 139, 143, 147, 151, 155, 159,
144, 148, 152, 156, 160, 164, 168,	163, 166, 170, 174, 178, 182, 184,
172, 175, 179, 183, 187, 191, 194,	187–189, 191–193, 197, 201, 205,
199, 202, 206, 209, 213–215, 217,	208, 212, 216, 220, 224
221, 222, 225	util_pareto1_param_estimate, 56, 66, 84,
util_negative_binomial_stats_tbl, 109,	85, 107, 111, 115, 118, 121, 124,
112, 116, 119, 123, 126, 130, 134,	129, 133, 136, 140, 144, 148, 152,
138, 142, 146, 149, 154, 157, 161,	156, 160, 164, 168, 172, 176, 179,
165, 169, 173, 176, 180, 184, 188,	183, 185, 186, 188, 191, 192, 194,
192, 196, 200, 203, 207, 211, 215,	199, 202, 206, 209, 214, 217, 221,
218, 222, 223, 227	225
util_normal_aic, 14, 22, 27, 73, 110, 114, 120, 127, 128, 131, 135, 139, 143,	util_pareto1_stats_tbl, 56, 66, 84, 85, 109, 112, 116, 119, 123, 126, 130,
120, 127, 128, 131, 133, 139, 143, 147, 151, 155, 159, 163, 166, 170,	134, 138, 142, 146, 149, 154, 157,
174, 171, 133, 139, 163, 166, 170, 174, 177, 182, 185, 189, 193, 197,	154, 158, 142, 140, 149, 154, 157, 161, 165, 169, 173, 177, 180, 184,
174, 177, 102, 103, 109, 193, 197,	101, 103, 109, 1/3, 1//, 100, 184,

185, 187, 187, 191, 192, 196, 200,	123, 126, 130, 134, 138, 142, 146,
203, 207, 211, 215, 218, 223, 227	149, 154, 157, 161, 165, 169, 173,
util_pareto_aic, 14, 22, 27, 73, 110, 114,	177, 180, 184, 188, 192, 196, 200,
120, 127, 128, 132, 135, 139, 143,	203, 207, 211, 215, 218, 223, 227
147, 151, 155, 159, 163, 166, 170,	util_triangular_aic, 14, 22, 27, 73, 110,
174, 178, 182, 185, 188, 193, 197,	114, 120, 127, 128, 132, 135, 139,
201, 205, 208, 212, 216, 220, 224	143, 147, 151, 155, 159, 163, 166,
util_pareto_param_estimate, 56, 66, 84,	170, 174, 178, 182, 185, 189, 193,
85, 107, 111, 115, 118, 121, 124,	196, 201, 205, 208, 212, 216, 220,
129, 133, 136, 140, 144, 148, 152,	224
156, 160, 164, 168, 172, 176, 179,	util_triangular_param_estimate, 97, 107,
	•
183, 185, 187, 188, 190, 192, 194,	111, 115, 118, 121, 124, 129, 133,
199, 202, 206, 209, 214, 217, 221,	136, 140, 144, 148, 152, 156, 160,
225	164, 168, 172, 176, 179, 183, 187,
util_pareto_stats_tbl, 56, 66, 84, 85, 109,	191, 194, 198, 200, 202, 206, 209,
112, 116, 119, 123, 126, 130, 134,	214, 217, 221, 225
138, 142, 146, 149, 154, 157, 161,	util_triangular_stats_tbl, 97, 109, 113,
165, 169, 173, 177, 180, 184, 185,	116, 119, 123, 126, 130, 134, 138,
187, 188, 191, 191, 196, 200, 203,	142, 146, 149, 154, 157, 161, 165,
207, 211, 215, 218, 223, 227	169, 173, 177, 180, 184, 188, 192,
util_poisson_aic, 14, 22, 27, 73, 110, 114,	196, 199, 199, 203, 207, 211, 215,
120, 127, 128, 132, 135, 139, 143,	218, 223, 227
147, 151, 155, 159, 163, 166, 170,	util_uniform_aic, 14, 22, 27, 73, 110, 114,
174, 178, 182, 185, 189, 192, 197,	120, 127, 128, 132, 135, 139, 143,
201, 205, 208, 212, 216, 220, 224	147, 151, 155, 159, 163, 166, 170,
util_poisson_param_estimate, 87, 105,	174, 178, 182, 185, 189, 193, 197,
107, 111, 115, 118, 121, 124, 129,	201, 204, 208, 212, 216, 220, 224
133, 136, 140, 144, 148, 152, 156,	util_uniform_param_estimate, 98, 108,
160, 164, 168, 172, 176, 179, 183,	111, 115, 118, 122, 124, 129, 133,
187, 191, 194, 196, 199, 202, 206,	136, 140, 144, 148, 152, 156, 160,
209, 214, 217, 221, 225, 227	164, 168, 172, 176, 179, 183, 187,
util_poisson_stats_tbl, 87, 105, 109, 113,	191, 194, 199, 202, 205, 207, 210,
116, 119, 123, 126, 130, 134, 138,	214, 217, 221, 225
142, 146, 149, 154, 157, 161, 165,	util_uniform_stats_tbl, 98, 109, 113, 116,
169, 173, 177, 180, 184, 188, 192,	119, 123, 126, 130, 134, 138, 142,
195, 195, 200, 203, 207, 211, 215,	146, 149, 154, 157, 161, 165, 169,
218, 223, 225, 227	173, 177, 180, 184, 188, 192, 196,
util_t_aic, 14, 22, 27, 73, 110, 114, 120,	200, 203, 206, 206, 211, 215, 218,
127, 128, 132, 135, 139, 143, 147,	223, 227
151, 155, 159, 163, 166, 170, 174,	util_weibull_aic, 14, 22, 27, 73, 110, 114,
178, 182, 185, 189, 193, 197, 200,	120, 127, 128, 132, 135, 139, 143,
205, 208, 212, 216, 220, 224	147, 151, 155, 159, 163, 166, 170,
util_t_param_estimate, 107, 111, 115, 118,	174, 178, 182, 185, 189, 193, 197,
121, 124, 129, 133, 136, 140, 144,	201, 205, 207, 212, 216, 220, 224
148, 152, 156, 160, 164, 168, 172,	util_weibull_param_estimate, 68, 99, 108,
176, 179, 183, 187, 191, 194, 199,	111, 115, 118, 122, 124, 129, 133,
201, 206, 209, 214, 217, 221, 225	111, 113, 116, 122, 124, 129, 133, 136, 141, 145, 148, 152, 156, 160,
util t stats tbl. 95, 109, 113, 116, 119.	150, 141, 145, 146, 152, 150, 160, 164. 168. 172. 176, 179, 183. 187.
$u_{L11}_{L}_{L} = u_{L3}_{L3}_{L3}_{L3}_{L3}_{L3}_{L3}_{L3}$	104, 100, 1/2, 1/0, 1/7, 103, 10/.

```
191, 194, 199, 202, 206, 209, 211,
                                                               128, 132, 135, 139, 143, 147, 151,
         214, 217, 221, 225
                                                               155, 159, 163, 166, 170, 174, 178,
                                                               182, 185, 189, 193, 197, 201, 205,
util_weibull_stats_tbl, 68, 99, 109, 113,
                                                               208, 212, 216, 219, 224
         116, 119, 123, 126, 130, 134, 138,
                                                     util_zero_truncated_negative_binomial_param_estimate,
         142, 146, 149, 154, 157, 161, 165,
                                                               34, 79, 101, 104, 108, 111, 115, 116,
         169, 173, 177, 180, 184, 188, 192,
                                                               118, 122, 124, 129, 133, 137, 141,
         196, 200, 203, 207, 210, 210, 215,
                                                               145, 148, 152, 156, 160, 164, 168,
         218, 223, 227
                                                               172, 176, 179, 183, 187, 191, 194,
util_zero_truncated_binomial_aic, 14,
                                                               199, 202, 206, 210, 214, 215, 217,
         22, 27, 73, 110, 114, 120, 127, 128,
                                                               220, 222, 225
         132, 135, 139, 143, 147, 151, 155,
                                                     util_zero_truncated_negative_binomial_stats_tbl,
         159, 163, 166, 170, 174, 178, 182,
                                                               34, 79, 101, 104, 109, 113, 115, 116,
         185, 189, 193, 197, 201, 205, 208,
                                                               119, 123, 126, 130, 134, 138, 142,
         211, 216, 220, 224
                                                               146, 149, 154, 157, 161, 165, 169,
util_zero_truncated_binomial_param_estimate,
                                                               173, 176, 177, 180, 184, 188, 192,
         34, 79, 101, 102, 104, 105, 108, 111,
                                                               196, 200, 203, 207, 211, 214, 215,
         115, 116, 118, 122, 124, 129, 133,
                                                               218, 221, 222, 227
         136, 141, 145, 148, 152, 156, 160,
                                                     util_zero_truncated_poisson_aic, 14, 22,
         164, 168, 172, 176, 179, 183, 187,
                                                               27, 73, 110, 114, 120, 127, 128, 132,
         191, 194, 199, 202, 206, 210, 213,
                                                               135, 139, 143, 147, 151, 155, 159,
         215, 217, 221, 222, 225
                                                               163, 166, 170, 174, 178, 182, 185,
util_zero_truncated_binomial_stats_tbl,
                                                               189, 193, 197, 201, 205, 208, 212,
         34, 79, 101, 104, 109, 113, 115, 116,
                                                               216, 220, 223
         119, 123, 126, 130, 134, 138, 142,
                                                     util_zero_truncated_poisson_param_estimate,
         146, 149, 154, 157, 161, 165, 169,
                                                               87, 105, 108, 111, 115, 118, 122,
         173, 176, 177, 180, 184, 188, 192,
                                                               124, 129, 133, 137, 141, 145, 148,
         196, 200, 203, 207, 211, 214, 214,
                                                               152, 156, 160, 164, 168, 172, 176,
         218, 221–223, 227
                                                               179, 183, 187, 191, 194–196, 199,
util_zero_truncated_geometric_aic, 14,
                                                               202, 206, 210, 214, 217, 221, 224,
         22, 27, 73, 110, 114, 120, 127, 128,
                                                               227
         132, 135, 139, 143, 147, 151, 155,
                                                     \verb"util_zero_truncated_poisson_stats_tbl",
         159, 163, 166, 170, 174, 178, 182,
                                                               87, 105, 109, 113, 116, 119, 123,
         185, 189, 193, 197, 201, 205, 208,
                                                               126, 130, 134, 138, 142, 146, 149,
         212, 215, 220, 224
                                                               154, 157, 161, 165, 169, 173, 177,
util_zero_truncated_geometric_param_estimate,
                                                               180, 184, 188, 192, 195, 196, 200,
         108, 111, 115, 118, 122, 124, 129,
                                                               203, 207, 211, 215, 218, 223, 225,
         133, 137, 141, 145, 148, 152, 156,
                                                               226
         160, 164, 168, 172, 176, 179, 183,
         187, 191, 194, 199, 202, 206, 210,
         214, 216, 218, 221, 225
util_zero_truncated_geometric_stats_tbl,
         109, 113, 116, 119, 123, 126, 130,
         134, 138, 142, 146, 149, 154, 157,
         161, 165, 169, 173, 177, 180, 184,
         188, 192, 196, 200, 203, 207, 211,
         215, 217, 218, 223, 227
util_zero_truncated_negative_binomial_aic,
```

14, 22, 27, 73, 110, 114, 120, 127,