Package 'fitPlotR'

December 19, 2025

Type Package

Title Plotting Probability Distributions

Version 0.1.0

Description Provides functions for plotting probability density functions, distribution functions, survival functions, hazard functions and computing distribution moments. The implementation is inspired by Delignette-Muller and Dutang (2015) <doi:10.18637/jss.v064.i04>.

License GPL-3

Encoding UTF-8

Imports grDevices, graphics, stats

RoxygenNote 7.3.3

NeedsCompilation no

Author Muhammad Osama [aut, cre]

Maintainer Muhammad Osama <muhammadosama0846@gmail.com>

Repository CRAN

Date/Publication 2025-12-19 20:50:02 UTC

Contents

dist_moments *Compute Moments of a Probability Distribution*

Description

Calculates mean, variance, skewness, kurtosis, median, and mode for a given probability distribution defined by PDF and CDF.

Usage

```
dist_moments(pdf, cdf, support = c(0, Inf), params = list())
```

Arguments

pdf	Function for the probability density function (PDF). Must accept x as the first argument and parameters as a named list.
cdf	Function for the cumulative distribution function (CDF). Must accept x as the first argument and parameters as a named list.
support	Numeric vector of length 2. The support (lower and upper limits) of the distribution.
params	Named list of parameters to pass to pdf and cdf.

Value

A data frame with Mean, Variance, Skewness, Kurtosis, Median, and Mode.

Examples

```
# Generalized Exponential Distribution
pdf_ge <- function(x, alpha, lambda) {
  alpha * lambda * (1 - exp(-lambda * x))^(alpha - 1) * exp(-lambda * x)
}
cdf_ge <- function(x, alpha, lambda) {
  (1 - exp(-lambda * x))^alpha
}
dist_moments(pdf_ge, cdf_ge, support = c(0, Inf), params = list(alpha = 2, lambda = 3))

# Exponentiated Weibull Distribution
pdf_expweibull <- function(x, a, b, c){
  a * b * c * exp(-(b*x)^c) * (b*x)^(c-1) * (1 - exp(-(b*x)^c))^(a-1)
}
cdf_expweibull <- function(x, a, b, c){
  (1 - exp(-(b*x)^c))^a
}
dist_moments(pdf_expweibull, cdf_expweibull, support = c(0, Inf),
             params = list(a = 1.0, b = 1.4, c = 2.3))
```

plot_cdf *Plot CDF for the Probability of a Distribution*

Description

#' Plots the CDF of a probability distribution. Supports multiple sets of parameters with shaded areas under the curves.

Usage

```
plot_cdf(
  cdf,
  param_list,
  xlim = c(0, 10),
  ylim = c(0, 1),
  n = 500,
  main = "Custom CDF Plot",
  xlab = "x",
  ylab = "CDF",
  colors = NULL,
  shade_colors = NULL,
  lwd = 3,
  lty = 2,
  grid = TRUE,
  grid_lty = 3,
  grid_col = "gray80",
  grid_lwd = 1
)
```

Arguments

cdf	Function that computes the CDF. Must accept x as the first argument.
param_list	A list of parameter lists. Each element is a named list of parameters for cdf.
xlim	Numeric vector of length 2, specifying the x-axis limits. Default is c(0, 10).
ylim	Numeric vector of length 2, specifying the y-axis limits. Default is c(0, 1).
n	Number of points to evaluate on the x-axis. Default is 500.
main	Title of the plot. Default is "Custom CDF Plot".
xlab	Label for the x-axis. Default is "x".
ylab	Label for the y-axis. Default is "CDF".
colors	Vector of colors for the lines. Default is rainbow.
shade_colors	Vector of colors for shading under curves. Default is semi-transparent version of colors.
lwd	Line width. Default is 3.
lty	Line type. Default is 2 (dashed).

grid	Logical, whether to draw a grid. Default is TRUE.
grid_lty	Line type for grid. Default is 3.
grid_col	Grid color. Default is "gray80".
grid_lwd	Grid line width. Default is 1.

Value

A CDF plot is displayed. The function invisibly returns NULL.

Examples

```
# Example 1: Generalized Exponential Distribution
ge_cdf <- function(x, alpha, lambda) {
  (1 - exp(-lambda * x))^alpha
}
param_values <- list(
  list(alpha = 1, lambda = 1),
  list(alpha = 2, lambda = 1),
  list(alpha = 3, lambda = 0.5),
  list(alpha = 4, lambda = 1.5),
  list(alpha = 5, lambda = 2.5)
)
plot_cdf(cdf = ge_cdf, param_list = param_values, main = "CDF GE Distribution")

# Example 2: Exponentiated Weibull Distribution
cdf_expweibull <- function(x, a, b, c){
  (1 - exp(-(b*x)^c))^a
}
param_values <- list(
  list(a = 0.3, b = 1.2, c = 1.0),
  list(a = 1.3, b = 0.4, c = 2.3),
  list(a = 1.5, b = 0.9, c = 3.0),
  list(a = 2.0, b = 1.8, c = 2.8),
  list(a = 3.7, b = 2.0, c = 1.5)
)
colors <- c("green", "purple", "yellow", "orange", "darkblue")
plot_cdf(cdf = cdf_expweibull, param_list = param_values,
         main = "CDF of EW Distribution", colors = colors, xlim = c(0, 5))
```

---

plot_data *Plot Data Overview*

Description

Generates a 2x2 plot layout showing: Histogram, Boxplot, Kernel Density, and TTT Plot.

Usage

```
plot_data(
  x,
  col = "steelblue",
  border = "black",
  transparency = 0.5,
  lwd = 2,
  breaks = "Sturges"
)
```

Arguments

x	Numeric vector of data.
col	Color for plots (default "steelblue").
border	Border color for histograms/boxplots (default "black").
transparency	Transparency for filled areas (0-1, default 0.5).
lwd	Line width for plots (default 2).
breaks	Histogram breaks (default "Sturges").

Value

NULL (plots are drawn)

Examples

```
set.seed(123)
mydata <- rexp(100, 1)
plot_data(mydata, col = "darkblue", transparency = 0.35, lwd = 2)
```

---

plot_fitted *Plot Fitted Distribution with Diagnostic Plots*

Description

This function produces a 2x2 panel of plots for a fitted distribution: 1. Fitted PDF over histogram of data 2. Fitted CDF vs empirical CDF 3. QQ-plot (Theoretical vs Sample Quantiles) 4. PP-plot (Fitted CDF vs Empirical CDF)

Usage

```
plot_fitted(
  data,
  pdf_fun,
  cdf_fun,
  par,
  q_fun = NULL,
```

```
  xlim = NULL,
  ylim_pdf = NULL,
  ylim_cdf = NULL,
  lwd = 3,
  lty = 2,
  col_pdf = "yellow",
  col_cdf = "red",
  col_qq = "purple",
  col_pp = "darkgreen"
)
```

Arguments

data	Numeric vector of observed data.
pdf_fun	Function to compute the PDF; must take (par, x).
cdf_fun	Function to compute the CDF; must take (par, x).
par	Numeric vector of fitted parameters.
q_fun	Optional quantile function; must take (par, p). Default = NULL.
xlim	Numeric vector of length 2 for x-axis limits (default = range of data).
ylim_pdf	Numeric vector for y-axis limits of the PDF plot.
ylim_cdf	Numeric vector for y-axis limits of the CDF plot.
lwd	Line width for curves (default = 3).
lty	Line type for curves (default = 2).
col_pdf	Color for the PDF curve (default = "yellow").
col_cdf	Color for the CDF curve (default = "red").
col_qq	Color for QQ-plot points (default = "purple").
col_pp	Color for PP-plot points (default = "darkgreen").

Value

NULL (plots are generated as a side effect)

---

plot_hf *Plot Hazard Function of a Probability Distribution*

Description

#' Plots the PDF of a probability distribution. Supports multiple sets of parameters with shaded areas under the curves.

Usage

```
plot_hf(
  pdf,
  cdf,
  param_list,
  xlim = c(0, 10),
  ylim = c(0, 6),
  n = 500,
  main = "Hazard Function Plot",
  xlab = "x",
  ylab = "h(x)",
  colors = NULL,
  shade_colors = NULL,
  lwd = 3,
  lty = 2,
  grid = TRUE,
  grid_lty = 3,
  grid_col = "gray80",
  grid_lwd = 1
)
```

Arguments

pdf	Function that computes the PDF. Must accept x as the first argument.
cdf	Function that computes the CDF. Must accept x as the first argument.
param_list	A list of parameter lists. Each element is a named list of parameters for pdf and cdf.
xlim	Numeric vector of length 2, specifying the x-axis limits. Default is c(0, 10).
ylim	Numeric vector of length 2, specifying the y-axis limits. Default is c(0, 6).
n	Number of points to evaluate on the x-axis. Default is 500.
main	Title of the plot. Default is "Hazard Function Plot".
xlab	Label for the x-axis. Default is "x".
ylab	Label for the y-axis. Default is "h(x)".
colors	Vector of colors for the lines. Default is rainbow.
shade_colors	Vector of colors for shading under curves. Default is semi-transparent version of colors.
lwd	Line width. Default is 3.
lty	Line type. Default is 2 (dashed).
grid	Logical, whether to draw a grid. Default is TRUE.
grid_lty	Line type for grid. Default is 3.
grid_col	Grid color. Default is "gray80".
grid_lwd	Grid line width. Default is 1.

Value

A hazard function plot is displayed. The function invisibly returns NULL.

Examples

```
# Example 1: Generalized Exponential Distribution
pdf_ge <- function(x, alpha, lambda) {
  alpha * lambda * exp(-lambda*x) * (1 - exp(-lambda*x))^(alpha - 1)
}
cdf_ge <- function(x, alpha, lambda) {
  1 - (1 - exp(-lambda*x))^alpha
}
param_values <- list(
  list(alpha = 1, lambda = 1),
  list(alpha = 2, lambda = 1),
  list(alpha = 3, lambda = 0.5),
  list(alpha = 4, lambda = 1.5),
  list(alpha = 5, lambda = 2.5)
)
plot_hf(pdf_ge, cdf_ge, param_values, xlim=c(0,5), ylim=c(0,4), main="HF GE Distribution")

# Example 2: Exponentiated Weibull Distribution
pdf_expweibull <- function(x, a, b, c){
  a * b * c * exp(-(b*x)^c) * (b*x)^(c-1) * (1 - exp(-(b*x)^c))^(a-1)
}
cdf_expweibull <- function(x, a, b, c){
  1 - (1 - exp(-(b*x)^c))^a
}
param_values <- list(
  list(a = 0.3, b = 1.2, c = 1.0),
  list(a = 1.3, b = 0.4, c = 2.3),
  list(a = 1.5, b = 0.9, c = 3.0),
  list(a = 2.0, b = 1.8, c = 2.8),
  list(a = 3.7, b = 2.0, c = 1.5)
)
plot_hf(pdf_expweibull, cdf_expweibull, param_values)
```

---

plot_multi_fitted *Plot Multiple Fitted Distributions*

Description

Creates 2x2 plots for multiple fitted distributions: Fitted PDFs, Fitted CDFs vs Empirical CDF, QQ-Plots, and PP-Plots.

Usage

```
plot_multi_fitted(
  data,
  pdf_list,
  cdf_list,
  qf_list,
  params_list,
  dist_names = NULL,
  col_list = NULL,
  lty_list = NULL,
  lwd_list = NULL,
  main_pdf = "Fitted PDFs",
  main_cdf = "Fitted CDFs",
  main_qq = "QQ Plots",
  main_pp = "PP Plots",
  xlab = "x"
)
```

Arguments

data	Numeric vector of observed data.
pdf_list	List of PDF functions. Each function should take x and par.
cdf_list	List of CDF functions. Each function should take x and par.
qf_list	List of quantile functions (inverse CDF). Each function should take p and par.
params_list	List of parameter vectors corresponding to each distribution.
dist_names	Optional vector of distribution names.
col_list	Optional vector of colors for each distribution.
lty_list	Optional vector of line types for each distribution.
lwd_list	Optional vector of line widths for each distribution.
main_pdf	Title for PDF plot.
main_cdf	Title for CDF plot.
main_qq	Title for QQ plot.
main_pp	Title for PP plot.
xlab	Label for x-axis.

Value

NULL (plots are generated as a side effect)

Examples

```
# Example Multiple Distributions
set.seed(1)
data <- rexp(200, 1.1)
```

```
# Exponential
pdf_exp <- function(x, par) par[1] * exp(-par[1] * x)
cdf_exp <- function(x, par) 1 - exp(-par[1] * x)
qf_exp  <- function(p, par) -log(1 - p) / par[1]

# Generalized Exponential
pdf_gexp <- function(x, par) {
  a <- par[1]; l <- par[2]
  a * l * exp(-l*x) * (1-exp(-l*x))^(a-1)
}
cdf_gexp <- function(x, par) {
  a <- par[1]; l <- par[2]
  (1-exp(-l*x))^a
}
qf_gexp <- function(p, par) {
  a <- par[1]; l <- par[2]
  -log(1 - p^(1/a)) / l
}

# Weibull
pdf_weibull <- function(x, par) {
  k <- par[1]; l <- par[2]
  (k/l) * (x/l)^(k-1) * exp(-(x/l)^k)
}
cdf_weibull <- function(x, par) {
  k <- par[1]; l <- par[2]
  1 - exp(-(x/l)^k)
}
qf_weibull <- function(p, par) {
  k <- par[1]; l <- par[2]
  l * (-log(1 - p))^(1/k)
}

# Normal
pdf_norm <- function(x, par) dnorm(x, par[1], par[2])
cdf_norm <- function(x, par) pnorm(x, par[1], par[2])
qf_norm <- function(p, par) qnorm(p, par[1], par[2])

data <- rexp(200, 1)
# Call the plot function
plot_multi_fitted(
  data = data,
  pdf_list = list(pdf_exp, pdf_gexp, pdf_weibull, pdf_norm),
  cdf_list = list(cdf_exp, cdf_gexp, cdf_weibull, cdf_norm),
  qf_list  = list(qf_exp, qf_gexp, qf_weibull, qf_norm),
  params_list = list(
    c(1.1),
    c(2, 1.3),
    c(1.5, 2),
    c(0, 1)
  ),
  dist_names = c("Exp", "GExp", "Weibull", "Normal"),
  col_list = c("blue", "red", "darkgreen", "purple"),
```

```
  lty_list = c(1, 2, 3, 4),
  lwd_list = c(3, 3, 3, 3)
)
```

`plot_pdf` *Plot for the PDF of a Probability distribution.*

Description

Plots the PDF of a probability distribution. Supports multiple sets of parameters with shaded areas under the curves.

Usage

```
plot_pdf(
  pdf,
  param_list,
  xlim = c(0, 10),
  ylim = c(0, 6),
  n = 500,
  main = "PDF Plot",
  xlab = "x",
  ylab = "Density",
  colors = NULL,
  shade_colors = NULL,
  lwd = 3,
  lty = 2,
  grid = TRUE,
  grid_lty = 3,
  grid_col = "gray80",
  grid_lwd = 1
)
```

Arguments

`pdf`	Function that computes the PDF. Must accept x as the first argument.
`param_list`	A list of parameter lists. Each element is a named list of parameters for `pdf`.
`xlim`	Numeric vector of length 2, specifying the x-axis limits. Default is c(0, 10).
`ylim`	Numeric vector of length 2, specifying the y-axis limits. Default is c(0, 6).
`n`	Number of points to evaluate on the x-axis. Default is 500.
`main`	Title of the plot. Default is "PDF Plot".
`xlab`	Label for the x-axis. Default is "x".
`ylab`	Label for the y-axis. Default is "Density".
`colors`	Vector of colors for the lines. Default is `rainbow`.

shade_colors	Vector of colors for shading under curves. Default is semi-transparent version of colors.
lwd	Line width. Default is 3.
lty	Line type. Default is 2 (dashed).
grid	Logical, whether to draw a grid. Default is TRUE.
grid_lty	Line type for grid. Default is 3.
grid_col	Grid color. Default is "gray80".
grid_lwd	Grid line width. Default is 1.

Value

A PDF plot is displayed. The function invisibly returns NULL.

Examples

```
# Example 1 with Generalized Exponential Distribution
ge_pdf <- function(x, alpha, lambda) {
alpha * lambda * exp(-lambda * x) * (1 - exp(-lambda * x))^(alpha - 1)
}
param_values <- list(list(alpha = 1, lambda = 1),
                     list(alpha = 2, lambda = 1),
                     list(alpha = 3, lambda = 0.5),
                     list(alpha = 4, lambda = 1.5),
                     list(alpha = 5, lambda = 2.5))
plot_pdf(pdf = ge_pdf, param_list = param_values, ylim = c(0, 1),
        main = "Generalized Exponential Distribution")

# Example 2 with Exponentiated Weibull Distribution
pdf_expweibull <- function(x, a, b, c){
a * b * c * exp(-(b*x)^c) *
 (b*x)^(c-1) * (1 - exp(-(b*x)^c))^(a-1)
}
param_values <- list(list(a = 0.3, b = 1.2, c = 1.0),
                     list(a = 1.3, b = 0.4, c = 2.3),
                     list(a = 1.5, b = 0.9, c = 3.0),
                     list(a = 2.0, b = 1.8, c = 2.8),
                     list(a = 3.7, b = 2.0, c = 1.5))
colors <- c("green", "purple", "yellow", "orange", "darkblue")
plot_pdf(pdf = pdf_expweibull, param_list = param_values,
        main = "PDF of EW Distribution",
        colors = colors, xlim = c(0, 5), ylim = c(0, 3))
```

plot_sf *Plot for the Survival Function of a Probability Distribution*

Description

#' Plots the SF of a probability distribution. Supports multiple sets of parameters with shaded areas under the curves.

Usage

```
plot_sf(
  sf,
  param_list,
  xlim = c(0, 10),
  ylim = c(0, 1),
  n = 500,
  main = "SF Plot",
  xlab = "x",
  ylab = "SF",
  colors = NULL,
  shade_colors = NULL,
  lwd = 3,
  lty = 2,
  grid = TRUE,
  grid_lty = 3,
  grid_col = "gray80",
  grid_lwd = 1
)
```

Arguments

sf	Function that computes the SF. Must accept x as the first argument.
param_list	A list of parameter lists. Each element is a named list of parameters for sf.
xlim	Numeric vector of length 2, specifying the x-axis limits. Default is c(0, 10).
ylim	Numeric vector of length 2, specifying the y-axis limits. Default is c(0, 1).
n	Number of points to evaluate on the x-axis. Default is 500.
main	Title of the plot. Default is "SF Plot".
xlab	Label for the x-axis. Default is "x".
ylab	Label for the y-axis. Default is "SF".
colors	Vector of colors for the lines. Default is rainbow.
shade_colors	Vector of colors for shading under curves. Default is semi-transparent version of colors.
lwd	Line width. Default is 3.
lty	Line type. Default is 2 (dashed).

grid	Logical, whether to draw a grid. Default is TRUE.
grid_lty	Line type for grid. Default is 3.
grid_col	Grid color. Default is "gray80".
grid_lwd	Grid line width. Default is 1.

Value

A SF plot is displayed. The function invisibly returns NULL.

Examples

```
# Example 1: Generalized Exponential Distribution
ge_sf <- function(x, alpha, lambda) {
  1 - (1 - exp(-lambda * x))^alpha
}
param_values <- list(
  list(alpha = 1, lambda = 1),
  list(alpha = 2, lambda = 1),
  list(alpha = 3, lambda = 0.5),
  list(alpha = 4, lambda = 1.5),
  list(alpha = 5, lambda = 2.5)
)
plot_sf(sf = ge_sf, param_list = param_values, main = "SF GE Distribution")

# Example 2: Exponentiated Weibull Distribution
sf_expweibull <- function(x, a, b, c) {
  1 - (1 - exp(-(b*x)^c))^a
}
param_values <- list(
  list(a = 0.3, b = 1.2, c = 1.0),
  list(a = 1.3, b = 0.4, c = 2.3),
  list(a = 1.5, b = 0.9, c = 3.0),
  list(a = 2.0, b = 1.8, c = 2.8),
  list(a = 3.7, b = 2.0, c = 1.5)
)
colors <- c("green", "purple", "yellow", "orange", "darkblue")
plot_sf(sf = sf_expweibull, param_list = param_values,
        main = "SF of EW Distribution", colors = colors, xlim = c(0, 5))
```

Index