

fitPlotR: An R package for Plotting Probability Distributions

Muhammad Osama
Department of Statistics
Quaid-i-Azam University

ABSTRACT

The **fitPlotR** package provides various set of functions for visualizing univariate probability distributions. The package contains an integrated set of functions to visualize probability density functions, cumulative distribution functions, survival functions, hazard functions. The package includes various functions to describe the basic characteristics of a probability distribution, such as mean, median, mode, dispersion, and various higher-order moments of a probability distribution, such as skewness and kurtosis. In addition to diagnostic plots, the package includes functions for plotting fitted probability density functions, fitted cumulative distribution functions, quantile-quantile (QQ) plot and probability-probability (PP) plot, along with comprehensive graphical data overview.

Keywords: R package; probability distributions; probability density function; cumulative distribution function; survival function; hazard function; moments; statistical graphics; data visualization.

1. Introduction

Probability distributions are the spine of statistical analysis. These distributions help researchers to describe mathematical functions that can represent uncertainty or variability in reliability engineering, survival analysis, and various other fields. These functions can be well explained using graphical illustrations, where key theoretical functions, like probability density functions (PDFs), cumulative functions (CDFs), survival functions (SFs), and hazard functions (HFs) can be compared effectively. Visualization of these functions allows researchers to interpret the behavior of different distributions.

While R provides extensive tools for statistical computation and graphics, existing packages typically require multiple steps or focus on fitting distributions to data rather than visualizing theoretical distributions visualize [1], `ggdensity` [2], `ggdist` [3], `ggridges` [4], `ggExtra` [5], `ggpubr` [6], `bayesplot` [7], `plotly` [8], `lattice` [9], `ggplot2` [10], `qqplotr` [11], `fanplot` [12], `vcd` [13], `EnvStat` [14], `GGally` [15], `MASS` [16], `cpd` [17], `gamlss` [18], `rriskDistributions` [19], `fitdistrplus` [20], `distributional` [21], `distr` [22], `DISTRIB` [23], `Newdistns` [24] and `extraDistr` [25].

To address this gap, we present **fitPlotR** an R package for visualizing theoretical probability distributions in an easy standardized way. **fitPlotR** allows you to create visually appealing and fully customizable representations of important functions associated with the probability distribution (i.e. probability density, cumulative, survival, and hazard functions). By allowing the user to specify the model parameters directly, **fitPlotR** will create customizable plots for several types of continuous probability distributions. By providing all tools in one easy-to-use package that does not require a user to have empirical data or fit distribution parameters through complex programming, students, educators and research practitioners can focus on understanding the mathematical nature of distributions, create customized visualizations and easily reproduce them through the use of this package.

- Received February 19, 2026, in final form February 2026.
- Muhammad Osama (corresponding author) is affiliated with Department of Statistics, Quaid-i-Azam University, Pakistan
muhammadosama0846@gmail.com

This paper provides a review of the principal features of **fitPlotR**. The package is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=fitPlotR>. The paper proceeds as follows: section 2 presents plots for different probability distributions, section 3 discusses the computation of distributional moments, section 4 presents fitted plots for different probability distributions, section 5 presents multiple fitted plots for different probability distributions, section 6 provides data overview and section 7 concludes the paper.

First, we need to install the package and then load using

```
install.packages("fitPlotR")
library(fitPlotR)
```

2. Plotting Probability Distribution

2.1 Probability Density Functions

The PDF describes the relative likelihood of a continuous random variable taking a specific value. Visualization of PDFs are essential for understanding the distribution shape and tail behavior. The PDFs in **fitPlotR** are visualized using the **plot_pdf()** function. This function requires the PDF of a distribution and a list of associated parameters, while additional graphical options can be used to customize the resulting plot. To illustrate the use of **plot_pdf()** function, consider the PDF of generalized exponential distribution and the PDF of exponentiated Weibull distribution. First, we plot the PDF of generalized exponential distribution. The PDF of generalized exponential distribution is given by

$$f(x, \alpha, \lambda) = \alpha \lambda \exp(-\lambda x) (1 - \exp(-\lambda x))^{\alpha-1} \quad x \geq 0, \alpha > 0, \lambda > 0$$

```
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")
```

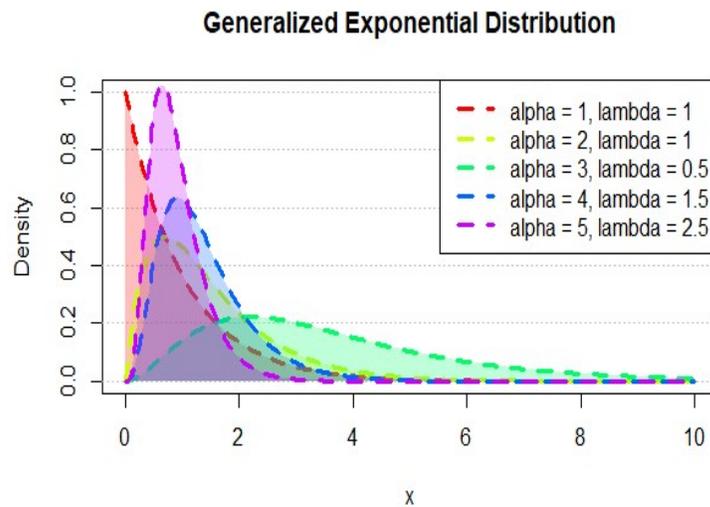


Figure 1: Plot for PDF of generalized exponential distribution

Now we plot PDF of exponentiated Weibull distribution. The PDF of exponentiated Weibull distribution is given by

$$f(x, a, b, c) = a b c (bx)^{c-1} \exp(-(bx)^c) (1 - \exp(-(bx)^c))^{a-1} \quad x \geq 0, a > 0, b > 0, c > 0$$

```
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))
```

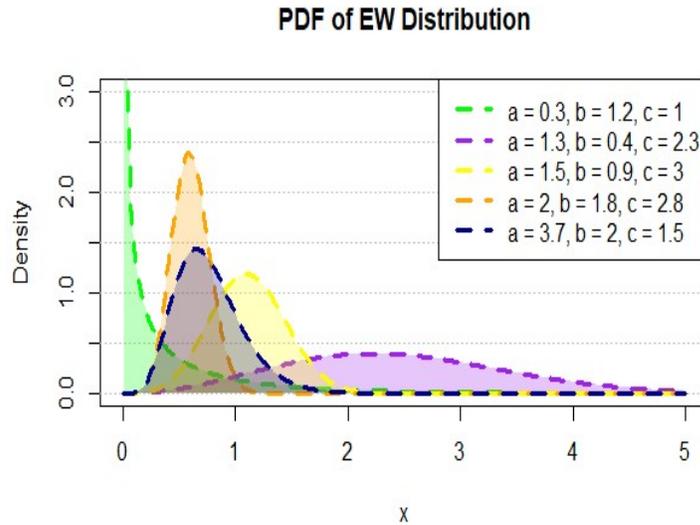


Figure 2: Plot for PDF of exponentiated Weibull distribution

2.2 Cumulative Distribution Function

The CDF represents the probability that a random variable is less than or equal to a given value.

$$F(x) = P(X \leq x)$$

The CDFs in **fitPlotR** are visualized using the **plot_cdf()** function. This function requires the CDF of a distribution and a list of associated parameters, other graphical parameters allows further customization of the resulting plot. To demonstrate the use of **plot_cdf()** function, consider the CDF of generalized exponential distribution and the CDF of exponentiated Weibull distribution.

First we plot the CDF of generalized exponential distribution. The CDF of generalized exponential distribution is given by

$$F(x, \alpha, \lambda) = (1 - \exp(-\lambda x))^\alpha \quad \alpha > 0, \lambda > 0$$

```
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")
```

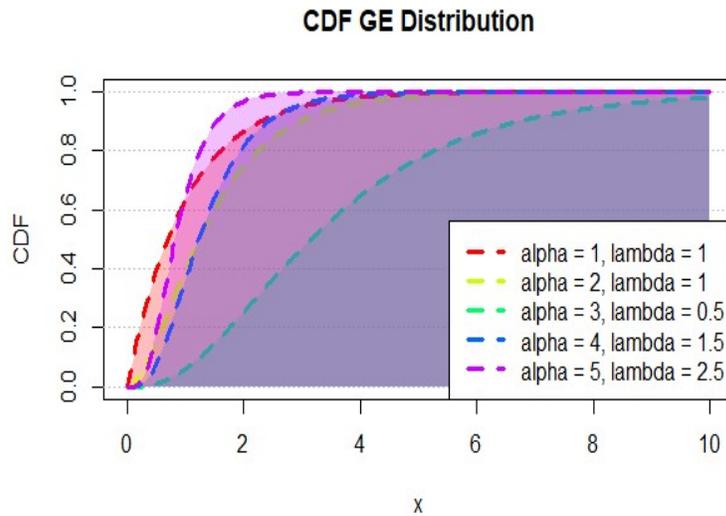


Figure 3: Plot for CDF of generalized exponential distribution

Now we plot the CDF of exponentiated Weibull distribution. The CDF of exponentiated Weibull distribution is given by

$$F(x, a, b, c) = (1 - \exp(-(bx)^c))^a \quad a > 0, b > 0, c > 0$$

```

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))

```

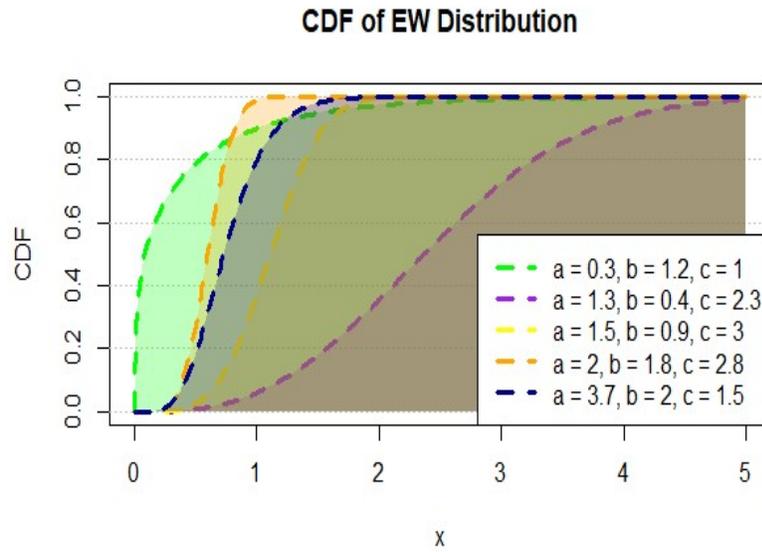


Figure 4: Plot for CDF of exponentiated Weibull distribution

2.3 Survival Function

The survival function, also known as the reliability function it is defined as the complement of the cumulative CDF.

$$S(x) = 1 - F(x)$$

where $F(x)$ is the CDF of a random variable x . The SFs in **fitPlotR** are visualized using the **plot_sf()** function. This function requires the SF of a distribution and a list of associated parameters, additional graphical parameters allows further customization of the resulting plot. To illustrate the use of **plot_sf()** function, consider the SF of generalized exponential distribution and the SF of exponentiated Weibull distribution. First we plot the SF of generalized exponential distribution. The SF of generalized exponential distribution is given by

$$S(x, \alpha, \lambda) = 1 - (1 - \exp(-\lambda x))^\alpha \quad \alpha > 0, \lambda > 0$$

```
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")
```

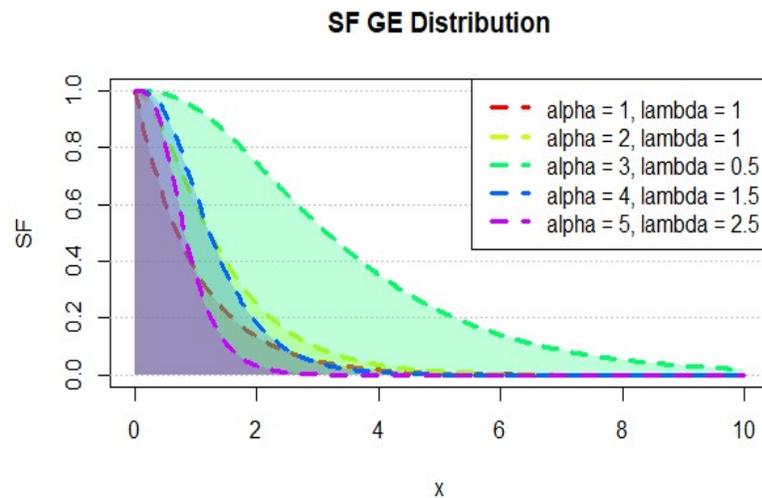


Figure 5: Plot for SF of generalized exponential distribution

Now we plot the SF of exponentiated Weibull distribution. The SF of exponentiated Weibull distribution is given by

$$F(x, a, b, c) = 1 - (1 - \exp(-(bx)^c))^a \quad a > 0, b > 0, c > 0$$

```
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))
```

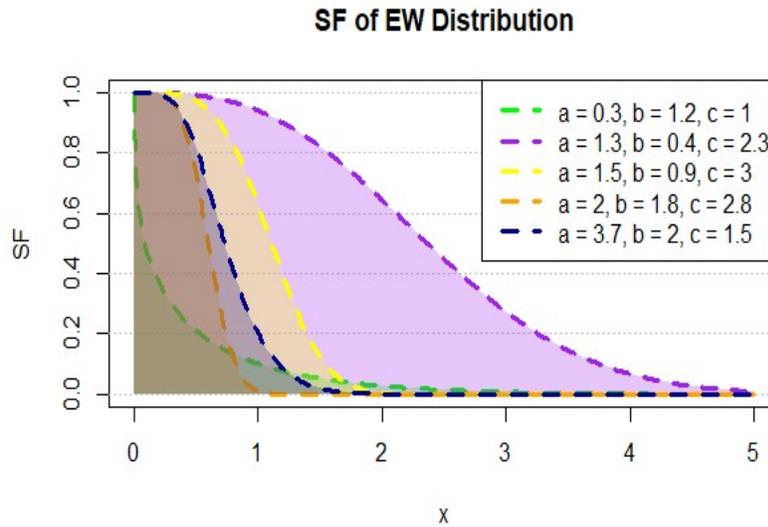


Figure 6: Plot for SF of exponentiated Weibull distribution

2.4 Hazard Function

If $f(x)$ is the PDF and $S(x)$ is the SF, then the hazard function is defined as

$$h(x) = \frac{f(x)}{S(x)}$$

The HFs in **fitPlotR** are visualized using the **plot_hf()** function. This function requires the HF of a distribution and a list of associated parameters, while additional graphical options can be used to customize the resulting plot. To demonstrate the use of **plot_hf()** function, consider the HF of generalized exponential distribution and the HF of exponentiated Weibull distribution. Now we plot the HF of 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")
```

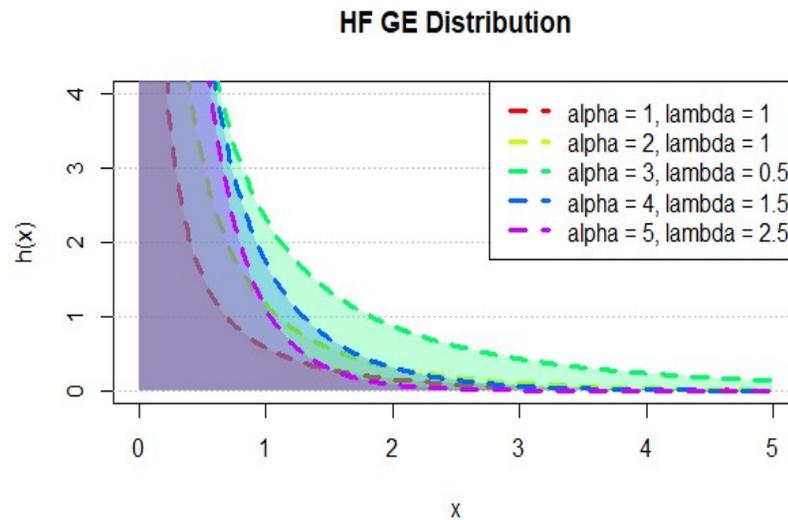


Figure 7: Plot for HF of generalized exponential distribution

Now we plot the HF of 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, xlim = c(0, 4))
```

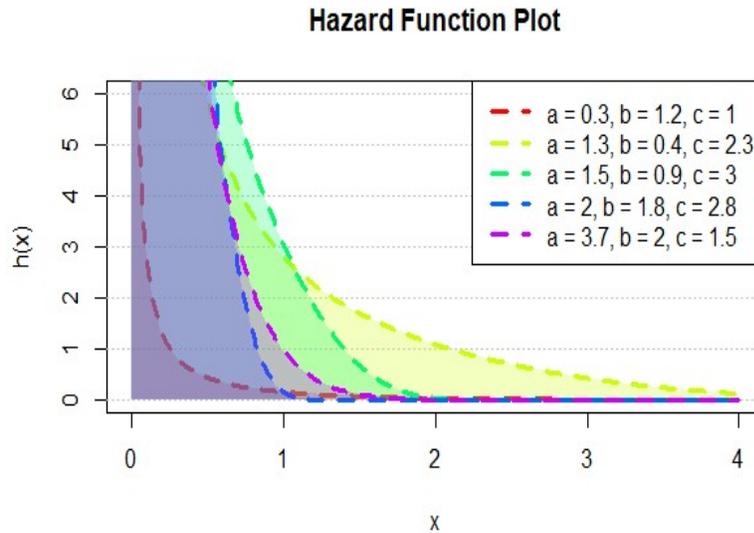


Figure 8: Plot for HF of exponentiated Weibull distribution

3. Moments of Probability Distributions

Probability distributions can summarize their defining characteristics with the moments of the distribution. The mean of the distribution represents its first moment; the second central moment (or variance) represents the average squared deviation of values from the mean. The higher-order moments of a distribution such as the skewness and kurtosis describe how far the distribution is from symmetry and how peaked the distribution is, respectively. The median and mode of the distribution are also important central measures. These moments can be calculated for any probability distribution by defining PDF and CDF. The **fitPlotR** package provides the **dist_moments()** function to compute these moments directly. To demonstrate the use of **dist_moments()** function, consider the exponential distribution and beta distribution. First, we find moments for exponential distribution. The PDF and CDF of exponential distribution are given as

$$f(x, \alpha) = \alpha \exp(-\alpha x) \quad x \geq 0, \alpha > 0$$

and the CDF

$$F(x, \alpha) = 1 - \exp(-\alpha x)$$

```
pdf_exp <- function(x, alpha) {
  alpha*exp(-alpha*x)
}
cdf_exp <- function(x, alpha) {
  1-exp(-alpha*x)
}
dist_moments(pdf_exp, cdf_exp, support = c(0, Inf), params = list(alpha = 2))

## Mean Variance Skewness Kurtosis Median Mode
## 1 0.5 0.25 2 9 0.3465733 0
```

Now find the moments for beta distribution.

```
pdf_beta <- function(x, a, b){
  dbeta(x, shape1 = a, shape2 = b)
}
cdf_beta <- function(x, a, b){
  pbeta(x, shape1 = a, shape2 = b)
}
dist_moments(pdf_beta, cdf_beta, support = c(0, 1),
  params = list(a = 1.0, b = 1.4))

##           Mean   Variance  Skewness Kurtosis   Median Mode
## 1 0.4166667 0.07148696 0.2833459 1.987017 0.3904977 0
```

3.1 Plotting Fitted Distribution with Diagnostic Plots

To plot fitted distribution with diagnostics plots such as fitted PDF overlaid on the histogram of the data, fitted CDF against the empirical CDF, quantile-quantile (QQ) plot and probability-probability (PP) plot. The **fitPlotR** package provide the function **plot_fitted()** which automatically generates all these plots by defining the PDF, CDF and quantile function of a distribution. To illustrate the use of **plot_fitted()** function we consider Weibull distribution and gamma distribution. First, we plot for Weibull distribution. The PDF of Weibull distribution is given by

$$f(x, a, b) = abx^{a-1} \exp(-bx^a) \quad x \geq 0, a > 0, b > 0$$

the CDF of Weibull distribution is given as

$$F(x, a, b) = 1 - \exp(-bx^a)$$

and the quantile function respectively

$$Q(p) = \left(-\frac{1}{b} \ln(1-p) \right)^{1/a}, \quad 0 < p < 1.$$

```
pdf_weibull <- function(par, x){
  a <- par[1]
  b <- par[1]
  a * b * x^(a-1) * exp(- b*x^(a))
}
cdf_weibull <- function(par, x){
  a <- par[1]
  b <- par[1]
  1 - exp(-b * x^a)
}
qf_weibull <- function(par, p){
  a <- par[1]
  b <- par[1]
  (-log(1 - p) / b)^(1 / a)
}
set.seed(123)
```

```

data <- rweibull(30, 2, 1) # generate data from weibull distribution
a <- 1.823181 # MLE for a
b <- 1.291451 # MLE for b
par <- c(a, b)
plot_fitted(
  data      = data,
  pdf_fun   = pdf_weibull,
  cdf_fun   = cdf_weibull,
  q_fun     = qf_weibull,
  par       = par
)

```

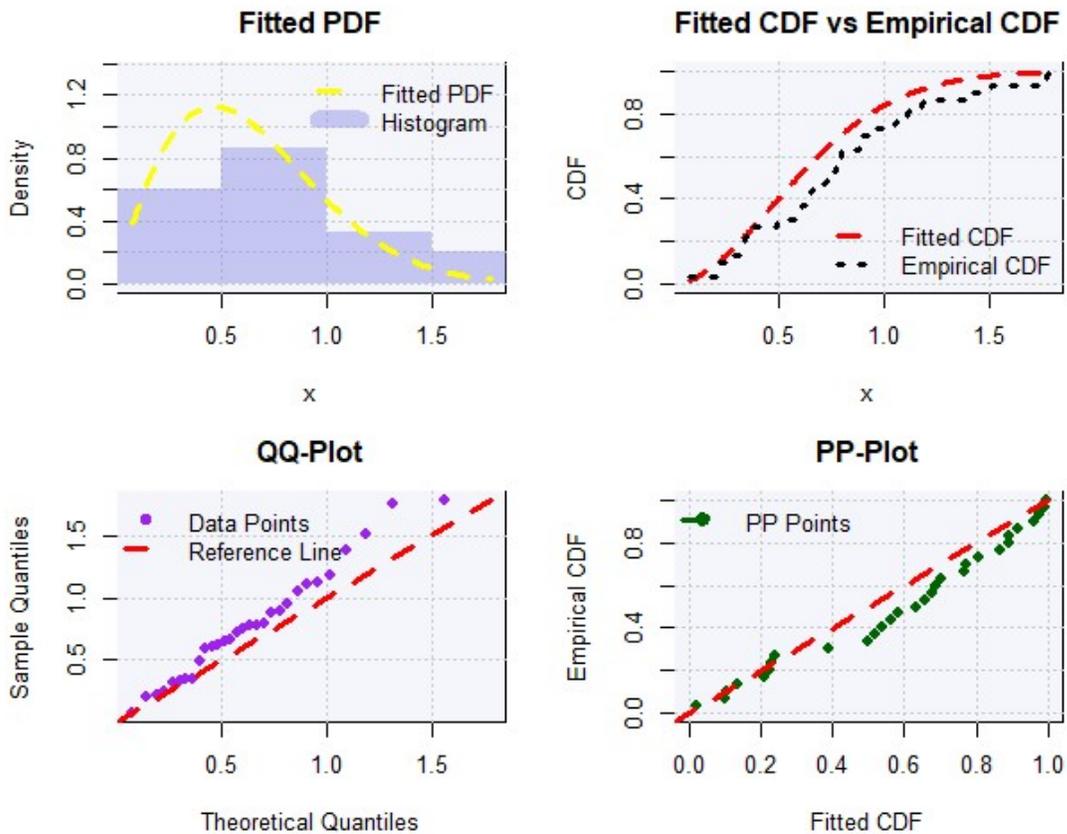


Figure 9: Fitted Plots for Weibull distribution.

Now for gamma distribution.

```

pdf_gamma <- function(par, x){
  a <- par[1]
  b <- par[2]
  dgamma(x, shape = a, scale = b)
}
cdf_gamma <- function(par, x){
  a <- par[1]
  b <- par[2]
  pgamma(x, shape = a, scale = b)
}

```

```

}
qf_gamma <- function(par, p){
  a <- par[1]
  b <- par[2]
  qgamma(p, shape = a, scale = b)
}
set.seed(123)
data <- rgamma(30, shape = 2, scale = 1) # generate data from gamma distribut
ion
a <- 2.0 # MLE for a
b <- 1.0 # MLE for b
par <- c(a, b)
plot_fitted(
  data      = data,
  pdf_fun   = pdf_gamma,
  cdf_fun   = cdf_gamma,
  q_fun     = qf_gamma,
  par      = par
)

```

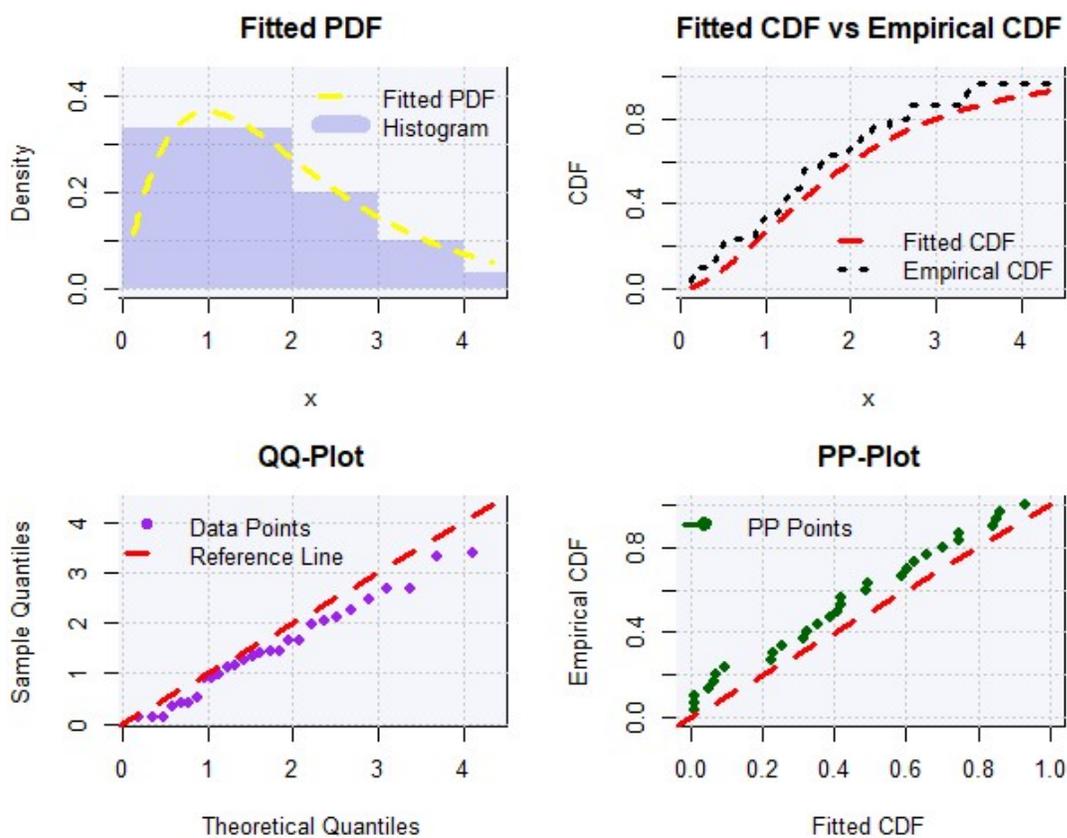


Figure 10: Fitted Plots for Gamma distribution.

3.2 Plotting Multiple Fitted Distribution with Diagnostic Plots

To plot multiple fitted distribution with diagnostics plots such as fitted PDFs overlaid on the histogram of the observed data, fitted CDFs against the empirical CDF, quantile-quantile (QQ) plot and probability-probability (PP) plot. The **fitPlotR** package provides the function **plot_multi_fitted()** which automatically generates all these plots by defining PDF, CDF and quantile function of a distribution. To demonstrate the use of **plot_multi_fitted()** consider different probability distributions such as exponential distribution, generalized exponential distribution, Weibull distribution and Normal distribution.

```
# 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])
set.seed(123)
data <- rexp(50, 1) # generate data from exponential distribution
# 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),
# Estimated Parameters
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)
)

```

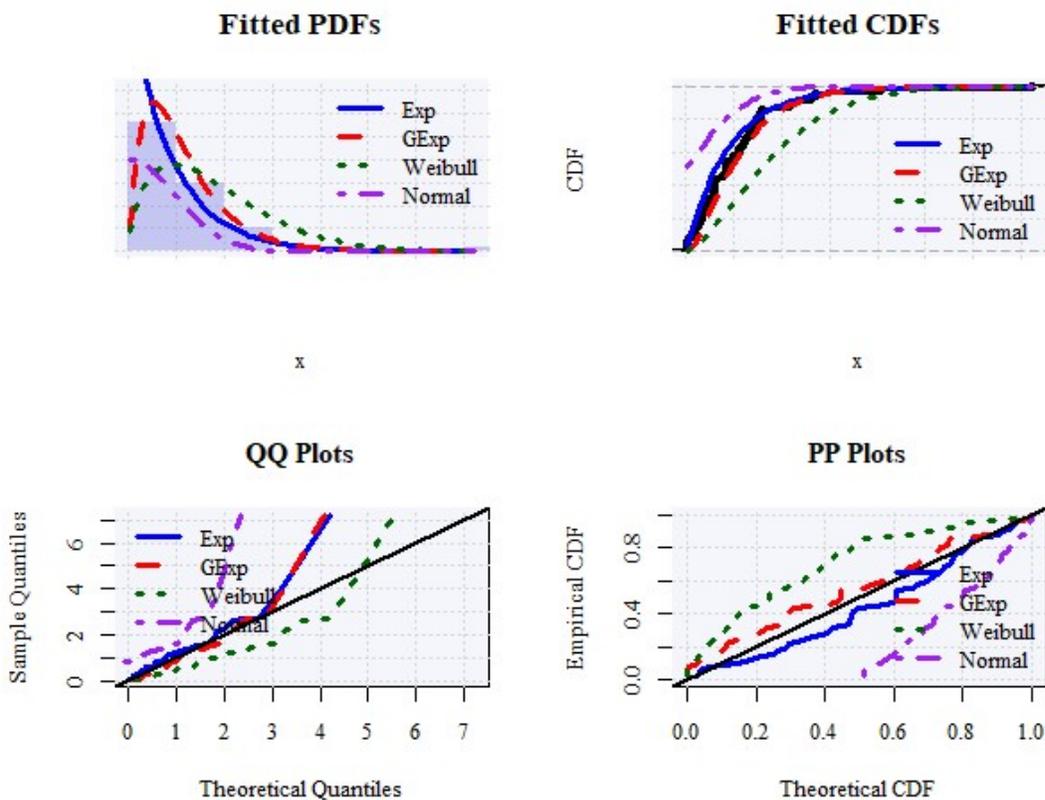


Figure 11: Fitted Multiple Plots.

3.3 Plotting Data Overview

To plot histogram, boxplot, kernel density and Total Time on Test (TTT) plot of the observed data **fitPlotR** package provides function **plot_data** which automatically generates all these plots. To demonstrate the use of **plot_data** function we generate random data from exponential distribution.

```
set.seed(1234)
data <- rexp(30, 1)
plot_data(data)
```

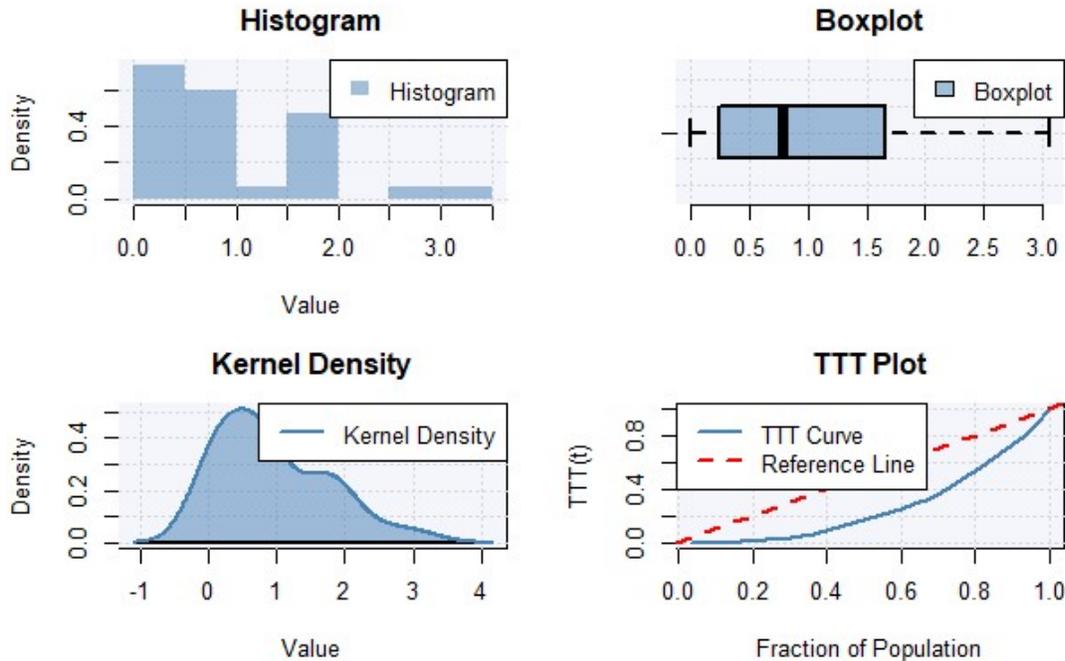


Figure 12: Data Plots.

4. Conclusion

This paper presents fitPlotR an R package for visualizing probability distributions. Mainly the package enables visualization of continuous probability distributions by providing tools to plot probability density functions (PDFs), cumulative distribution functions (CDFs), survival function (SFs) and hazard functions (HFs) in a consistent and customizable manner. In addition to theoretical visualization, fitPlotR offers comprehensive diagnostic tools for fitted distributions, including fitted PDF(s), fitted versus empirical CDF(s), quantile–quantile (QQ) plot(s), and probability–probability (PP) plot(s) through a single function call. The inclusion of functions for computing key distributional moments further enhances its utility for statistical analysis. Moreover, fitPlotR provides an integrated data overview facility that generates histogram, boxplot, kernel density plot, and Total Time on Test (TTT) plot.

By keeping programming simple and focusing on reproducibility and clarity, the package is specially fitting for students, teachers, and researchers in the fields of statistics, reliability engineering, and survival analysis. The package provides functions that are extremely concise and use a highly standardized input structure at the same time. This makes fine, publication-quality graphics possible without requiring lengthy programming. Given this simplicity, the barrier to its use by beginners and students learning probability distributions and their properties is lowered, while the work flow is transparent and reproducible, making analyses easy to replicate and verify. Educators will find the package useful in effective classroom demonstrations and visual explanations of theoretical concepts. Rapid exploration, comparison, and communication of distributional behavior and model adequacy facilitate applied statistical studies for researchers.

References

- [1]. Abel, G, J. (2026). *fanplot: Visualisation of Sequential Probability Distributions Using Fan Charts*. R package version 4.0.1 URL <https://CRAN.R-project.org/package=fanplot>
- [2]. Almeida, A., Loy, A., and Hofmann, H. (2025). *qqplotr: Quantile-Quantile Plot Extensions for ggplot2*. R package version 0.0.7 URL <https://CRAN.R-project.org/package=qqplotr>
- [3]. Akman, R, et.al. (2023). *new.dist: Alternative Continuous and Discrete Distributions*. R package version 0.1.1 URL <https://CRAN.R-project.org/package=new.dist>
- [4]. Attali, D., and Baker, C. (2025). *ggExtra: Add Marginal Histograms to ggplot2, and More ggplot2 Enhancements*. R package version 0.11.0 URL <https://CRAN.R-project.org/package=ggExtra>
- [5]. Balamuta, J. (2023). *visualize: Graph Probability Distributions with User Supplied Parameters and Statistics*. R package version 4.5.0, URL <https://CRAN.R-project.org/package=visualize>
- [6]. Belgorodski, N, et.al. (2017). *rriskDistributions: Fitting Distributions to Given Data or Known Quantiles*. R package version 2.1.2 URL <https://CRAN.R-project.org/package=rriskDistributions>
- [7]. Camphouse, F, et.al. (2025). *distr: Object Oriented Implementation of Distributions*. R package version 2.9.7 URL <https://CRAN.R-project.org/package=distr>
- [8]. Delignette-Muller, M-L, et.al. (2026). *fitdistrplus: Help to Fit of a Parametric Distribution to Non-Censored or Censored Data*. R package version 1.2-6 URL <https://CRAN.R-project.org/package=fitdistrplus>
- [9]. Gabry, J, et.al. (2025). *bayesplot: Plotting for Bayesian Models*. R package version 1.15.0 URL <https://CRAN.R-project.org/package=bayesplot>
- [10]. Kay, M., and Wiernik, B, M. (2025). *ggdist: Visualizations of Distributions and Uncertainty*. R package version 3.3.3 URL <https://CRAN.R-project.org/package=ggdist>
- [11]. Kassambara, A. (2025). *ggpubr: ggplot2 Based Publication Ready Plots*. R package version 0.6.2 URL <https://CRAN.R-project.org/package=ggpubr> ,
- [12]. Meyer, D, er.al. (2024). *vcd: Visualizing Categorical Data*. R package version 1.4-13 URL <https://CRAN.R-project.org/package=vcd>
- [13]. Millard, S, P., and Kowarik A. (2025). *EnvStats: Package for Environmental Statistics, Including US EPA Guidance*. R package version 3.1.0 URL <https://CRAN.R-project.org/package=EnvStats>
- [14]. Nadarajah, S., and Rocha, R, et.al. (2026). *Newdistns: Computes Pdf, Cdf, Quantile and Random Numbers, Measures of Inference for 19 General Families of Distributions*. R package version 2.1 URL <https://CRAN.R-project.org/package=Newdistns>
- [15]. O'Hara-Wild, M, et.al. (2026). *distributional: Vectorised Probability Distributions*. R package version 0.6.0 URL <https://CRAN.R-project.org/package=distributional>
- [16]. Otto, J., and Kahle, D. (2023). *ggdensity: Interpretable Bivariate Density Visualization with ggplot2*. R package version 1.0.0 URL <https://CRAN.R-project.org/package=ggdensity>
- [17]. Parchami, A. (2016). *DISTRIB: Four Essential Functions for Statistical Distributions Analysis: A New Functional Approach*. R package version 1.0 URL <https://CRAN.R-project.org/package=DISTRIB>
- [18]. Ripley, B, et.al. (2025). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-65 URL <https://CRAN.R-project.org/package=MASS>
- [19]. Sarkar, D, et.al. (2025). *lattice: Trellis Graphics for R*. R package version 0.22-7 URL <https://CRAN.R-project.org/package=lattice>

- [20]. Schloerke, B, et.al. (2025). *GGally: Extension to ggplot2*. R package version 2.4.0 URL <https://CRAN.R-project.org/package=GGally>
- [21]. Sievert, C, et.al. (2026). *plotly: Create Interactive Web Graphics via plotly.js*. R package version 4.12.0 URL <https://CRAN.R-project.org/package=plotly>
- [22]. Stasinopoulos, M, et.al. (2025). *gamlss: Generalized Additive Models for Location Scale and Shape*. R package version 5.5-0 URL <https://CRAN.R-project.org/package=gamlss>
- [23]. Vilchez-Lopez, S, et.al. (2024). *cpd: Complex Pearson Distributions*. R package version 0.3.3 URL <https://CRAN.R-project.org/package=cpd>
- [24]. Wilke, CO. (2025). *ggridges: Ridgeline Plots in ggplot2*. R package version 0.5.7 URL <https://CRAN.R-project.org/package=ggridges>
- [25]. Wickham, H, et.al. (2025). *ggplot2: Create Elegant Data Visualizations Using the Grammar of Graphics*. R package version 4.0.1 URL <https://CRAN.R-project.org/package=ggplot2>
- [26]. Wolodzko, T., and Klinke, S. (2026). *extraDistr: Additional Univariate and Multivariate Distributions*. R package version 1.10.0.2 URL <https://CRAN.R-project.org/package=extraDistr>