Learning to Estimate Distribution Parameters in R with fitdistr()

Authored by

Mohammed looti

November 15, 2025

Introduction to Parameter Estimation Using R's fitdistr() Function

Characterizing the underlying probability distribution of observed data stands as a cornerstone of rigorous statistical modeling and predictive analysis. This crucial process allows researchers to move beyond simple descriptive statistics, enabling generalization from a limited sample to the broader population and thereby facilitating robust forecasting and inference. Within the highly versatile R programming environment, the primary utility for performing this task is the **fitdistr()** function. This robust function is an integral component of the essential MASS package (Modern Applied Statistics with S) and provides a sophisticated, reliable mechanism for accurately estimating distribution parameters by maximizing the underlying likelihood function.

When approaching a new dataset, analysts frequently hypothesize that the data generation mechanism adheres to a known theoretical probabilistic form, such as the Normal Distribution, the Exponential Distribution, or the Gamma Distribution. Parameter estimation is the specialized statistical endeavor aimed at identifying the precise numerical values (e.g., the mean and standard deviation) that best define this theoretical shape. The goal is to ensure the chosen distribution curve aligns as closely as possible with the empirical data observed. The **fitdistr()** function elegantly automates this complex optimization process, delivering precise maximum likelihood parameter estimates complemented by their corresponding standard errors, which are vital components for subsequent statistical inference and hypothesis testing.

This comprehensive guide is designed to thoroughly explore the operational mechanics of **fitdistr()**, illuminate the statistical principles--particularly Maximum Likelihood Estimation--that underpin its operation, and furnish a detailed, step-by-step practical example. Through the demonstration of fitting a theoretical model to simulated data in R, readers will gain mastery over this function. Proficiency with **fitdistr()** is fundamentally critical for anyone engaged in advanced statistical modeling where accurately defining and understanding the data's underlying probability distribution is paramount to sound research outcomes.

Understanding the Role of Maximum Likelihood Estimation (MLE)

The fundamental statistical methodology that powers the calculations performed by the **fitdistr()** function is the principle of Maximum Likelihood Estimation (MLE). MLE is recognized globally as a highly effective and statistically sound method for deriving estimates for the parameters of nearly any statistical model. The core objective of MLE is remarkably straightforward: to determine the set of parameter values for a specified distribution that simultaneously maximizes the likelihood function, thereby identifying the values that render the observed sample data most probable.

To properly conceptualize this process, consider the challenge of fitting a standard bell curve, or a Normal Distribution, to a collection of data points. Hypothetically, an infinite number of unique bell curves exist, each uniquely defined by a distinct combination of mean and standard deviation

values. MLE employs a systematic search and optimization procedure across this vast space of possibilities. It diligently seeks the specific mean and standard deviation pairing where the Probability Density Function (PDF) assigns the highest collective probability to the actual data points that were empirically observed in the sample. In essence, MLE identifies the population parameters that make the sample "most likely" to have been generated under the assumed distributional model.

Because MLE relies on the sophisticated, analytical maximization of the likelihood function, the resulting parameter estimates typically exhibit desirable statistical properties, most notably consistency and asymptotic efficiency, especially as the sample size increases. While the **fitdistr()** function efficiently manages the intensive numerical and mathematical optimization processes internally, grasping that it operates via Maximum Likelihood Estimation is essential. This understanding is particularly important for accurately interpreting results when working with non-standard, truncated, or highly complex distributions where careful consideration of the likelihood concept is required for valid inference.

Deconstructing the fitdistr() Syntax and Core Arguments

The design of the **fitdistr()** function in R emphasizes ease of use and accessibility, typically requiring only two primary arguments for performing standard distribution fitting tasks. A thorough comprehension of these core arguments is indispensable for users aiming to successfully and flexibly apply the function across a wide spectrum of datasets and diverse distribution types encountered in applied statistics.

The basic syntax utilized by the function is defined as follows:

fitdistr(x, densefun, ...)

Here is a detailed breakdown of the required arguments that govern the function's operation:

x: This argument mandates the input of a numeric vector. This vector must contain the observed data values for which the analyst intends to estimate the underlying distribution parameters. It represents the complete, raw sample data gathered during the experimental phase or observational study.
densefun: This critical argument specifies the theoretical probability distribution model to which the observed data should be mathematically fitted. It must be input as a character string (e.g., `"normal"` or `"gamma"`) that precisely corresponds to the name of the distribution's density function as it is implemented within base R (though without the leading 'd').

The ellipsis (**...**) structure provides a gateway for passing optional, additional parameters directly into the optimization routine used by **fitdistr()**. The most common use of this feature is providing

informed starting values for the parameters being estimated. For many frequently used distributions, **fitdistr()** possesses internal algorithms capable of automatically calculating statistically reasonable initial estimates. However, when tackling more complex or exotic distributions, or in scenarios where the MLE algorithm encounters convergence difficulties, supplying robust, informed starting values can dramatically enhance both the accuracy and the computational speed of the estimation process.

Practical Demonstration: Fitting a Normal Distribution in R

To effectively illustrate the practical utility and deployment of the **fitdistr()** function, we will now execute a detailed, hands-on example focusing on the simulation and subsequent fitting of a Normal Distribution. This pedagogical approach begins with the generation of a synthetic dataset designed to accurately mimic typical real-world observations that are expected to follow a Gaussian pattern.

For this demonstration, we utilize the base R function **rnorm()** to generate a vector comprising 200 distinct observations. Critically, these observations are intentionally sampled from a Normal Distribution defined by a known population mean (mu) of 10 and a population standard deviation ($sigma$) of 3. We ensure that this entire example is perfectly reproducible for any user following along by invoking the **set.seed()** function before data generation.

 # Set a seed to ensure reproducibility of the simulated data
set.seed(1)

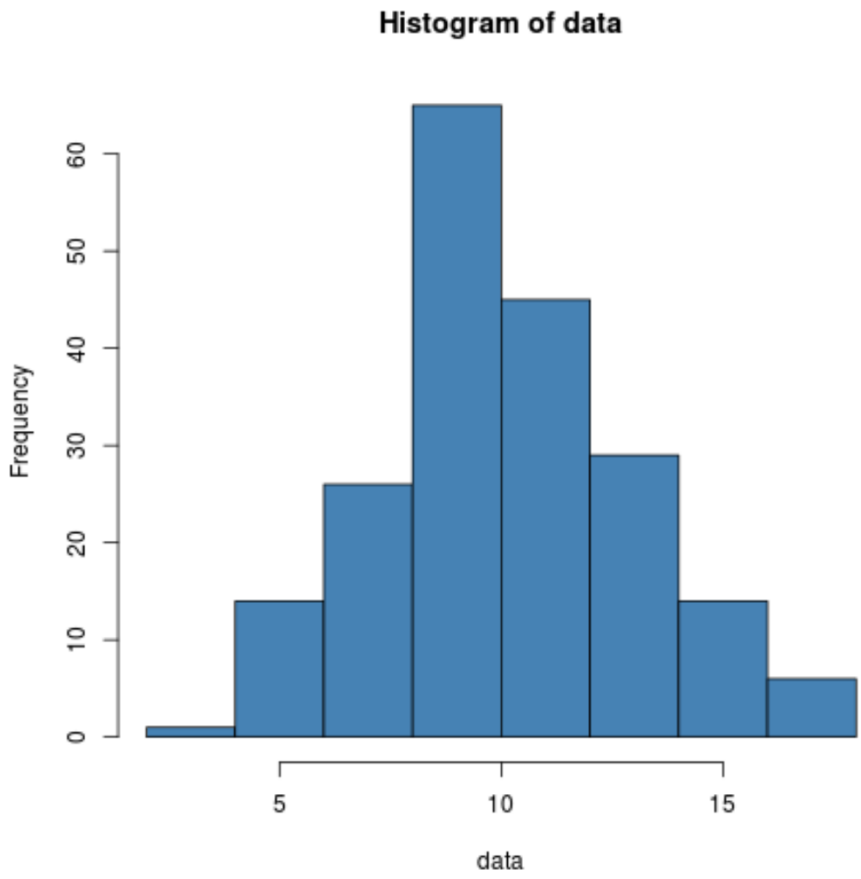
# Generate a sample of 200 observations from a Normal Distribution (mean=10, sd=3)
data <- rnorm(200, mean=10, sd=3)

# Display the first six observations of the generated sample data
head(data)

8.120639 10.550930 7.493114 14.785842 10.988523 7.538595

Prior to initiating any formal parameter estimation, standard statistical protocol mandates visualizing the data's empirical distribution. We achieve this using the base R graphics function **hist()** to construct a histogram, which provides immediate visual confirmation regarding the overall shape and symmetry of the dataset, confirming its suitability for a Normal Distribution model.

hist(data, col='steelblue')



As visually confirmed by the histogram, the data clearly exhibits the classical symmetrical bell shape, providing strong empirical evidence that a Normal Distribution model is indeed the most appropriate fit for this specific dataset. With this visual confirmation secured, we proceed to the core task: employing the **fitdistr()** function to statistically estimate the two critical parameters (the mean and the standard deviation) that uniquely define this derived probability curve.

Analyzing the Output and Interpreting Estimated Parameters

With the preparatory steps finalized, the next stage involves loading the required MASS package into the R session environment. Subsequently, we invoke the **fitdistr()** function, explicitly instructing it by providing the generated data vector and specifying the target distribution name as `"normal"`. The function then autonomously executes the complex iterative procedure central to Maximum Likelihood Estimation (MLE).

library(MASS)

# estimate parameters of the distribution
fitdistr(data, "normal")

mean sd
10.1066189 2.7803148
(0.1965979) (0.1390157)

The resulting output generated by **fitdistr()** is structured to provide two distinct lines of information for every parameter estimated. The first line clearly presents the actual estimated parameter value, which has been mathematically derived through the maximum likelihood procedure. For this specific example, the resulting estimated mean ($hat{mu}$) is calculated as **10.1066189**, and the estimated standard deviation ($hat{sigma}$) is determined to be **2.7803148**. These values constitute the optimal parameters defining the Normal Distribution curve that best fits the 200 observed data points.

The second line of the output, enclosed within parentheses, conveys the estimated **standard error** corresponding to the parameter presented immediately above it. The standard error is a critically important measure of the precision associated with the parameter estimate; statistically, smaller standard error values directly translate to higher statistical confidence in the estimated parameter value. For instance, the standard error for the estimated mean is 0.1965979. These standard errors are not merely descriptive; they serve as the foundation for constructing formal confidence intervals around the parameters, thereby establishing a plausible range of values for the true population parameter based on the characteristics observed within the sample data.

By directly comparing these calculated estimates against the parameters used in our initial data simulation (true population mean=10, true standard deviation=3), we can observe that the values derived by **fitdistr()** are exceptionally close to the true population characteristics. This congruence powerfully validates the effectiveness and statistical rigor of the Maximum Likelihood Estimation technique, even when operating with a moderately sized sample of 200 observations. Consequently, we can confidently assert that the vector of observed values follows a Normal Distribution rigorously defined by these estimated parameters.

Comprehensive List of Supported Distributions in fitdistr()

A core advantage and significant strength of the fitdistr() function resides in its remarkable versatility regarding distribution type. The critical **densefun** argument is engineered to accept a comprehensive array of built-in statistical distributions, effectively catering to the analytical needs of both continuous and discrete data types commonly encountered across various scientific disciplines.

When specifying the **densefun** argument, users must strictly adhere to the naming convention utilized by the R programming environment's base density functions. This involves providing the distribution name as a character string that corresponds to the distribution's density function (e.g.,

`dnorm` for Normal, `dgamma` for Gamma), but crucially, the input string must omit the leading 'd' prefix (i.e., use `"normal"` or `"gamma"`).

The function robustly supports Maximum Likelihood Estimation for the parameters of the following distribution names:

beta: Primarily utilized for modeling probabilities, proportions, or rates, where the variable is naturally bounded between 0 and 1.
cauchy: A distribution characterized by heavy tails, often employed in theoretical physics and situations where outliers are common.
chi-squared: A foundational distribution frequently used in various forms of hypothesis testing and in the construction of confidence intervals related to variance.
exponential: Used predominantly to model the time elapsed until a specific event occurs, especially relevant in reliability engineering and queuing theory.
gamma: A highly flexible and versatile distribution suitable for modeling non-negative, continuous data that often exhibits skewness.
geometric: A discrete probability distribution focused on modeling the number of Bernoulli trials required to achieve the first successful outcome.
lognormal: Ideally suited for data whose natural logarithm follows a Normal Distribution, commonly observed in areas such as financial returns and size distributions.
logistic: Utilized in modeling cumulative growth curves and serving as the foundational probability distribution for logistic regression models.
negative binomial: A discrete distribution modeling the total number of successes observed before a pre-specified number of failures has occurred.
normal: The ubiquitous Gaussian, or bell-shaped, curve, which finds widespread applicability across nearly all scientific and statistical fields.
Poisson: A discrete distribution used for modeling the number of independent events occurring within a fixed, specified interval of time or space.
t: Student's t-distribution, indispensable for statistical inference, particularly in scenarios involving small sample sizes or when the population variance is unknown.
Weibull: Extensively employed in survival analysis, materials strength testing, and reliability engineering due to its flexibility in modeling various failure rates.

This extensive and varied list ensures that **fitdistr()** remains a comprehensively robust and adaptable tool, capable of accurately characterizing the distributional properties of almost any empirical dataset encountered by practitioners working in applied statistics.

Conclusion: Leveraging fitdistr() for Advanced Modeling

The **fitdistr()** function, integrated within the essential MASS package, provides statisticians and

data analysts with an indispensable utility for accurate distribution fitting. By rigorously implementing Maximum Likelihood Estimation (MLE), the function efficiently generates precise estimates for core distribution parameters, such as the mean and standard deviation. This capability is paramount for establishing robust statistical models, conducting valid hypothesis testing, and performing reliable predictive analysis that is grounded in the data's underlying probability model. The function's defining characteristic is its ability to seamlessly handle an expansive spectrum of distributions, ranging from the fundamental Normal Distribution to more complex models like the Weibull or Gamma distributions, thereby firmly cementing its position as a core and non-negotiable component of the statistical workflow in the R environment.

For readers seeking to deepen their expertise in related statistical tasks, advanced data processing, and visualization techniques executed within R, the following resources are provided for additional comprehensive guidance and technical instruction:

Additional Resources

The following tutorials explain how to perform other common statistical tasks and visualizations in R:

How to Plot a Normal Distribution in R

How to Perform a Shapiro-Wilk Test for Normality in R