

Use the Gamma Distribution in R (With Examples)

Authored by
Mohammed loot

November 3, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Use the Gamma Distribution in R (With Examples)*.
PSYCHOLOGICAL STATISTICS. Retrieved from
<https://statistics.arabpsychology.com/?p=8973>

In the expansive field of [statistics](#), the [gamma distribution](#) stands out as an exceptionally versatile continuous probability distribution. It is routinely employed to accurately model positive, right-skewed data across numerous disciplines, offering a robust framework for phenomena such as waiting times in queueing systems, cumulative damage in reliability engineering, or predicting rainfall totals and insurance claim amounts. Understanding its applications and mathematical properties is crucial for advanced statistical modeling.

The statistical programming language [R](#) provides a complete and intuitive set of functions dedicated to manipulating this distribution. These built-in tools empower analysts and researchers to perform critical tasks, including calculating precise probabilities, determining specific thresholds (quantiles), and simulating [random variables](#) that adhere to the gamma model. This guide delves into the core R functions and provides practical, hands-on examples for effective implementation.

Mastering the Core Functions for the Gamma Distribution in R

To harness the full power of the **gamma distribution** in R, users must become familiar with the standard four prefix functions: `d`, `p`, `q`, and `r`. This naming convention is consistently applied across virtually all continuous and discrete distributions within R, making mastery of the gamma functions transferable to others like the Normal or Exponential distributions.

Each function requires the specification of the distribution's defining parameters--typically the **shape** (often denoted as k or α) and the **rate** (often θ or β , or alternatively, the **scale** parameter, where **scale** = $1/\text{rate}$). These two parameters dictate the exact form, location, and spread of the resulting distribution curve, allowing for incredible flexibility in modeling diverse datasets.

Here is a detailed breakdown of the fundamental functions available in [R](#) for working with the **gamma distribution**:

`dgamma(x, shape, rate)` - This function calculates the value of the [Probability Density Function](#) (PDF) at a given point `x`. The PDF value itself does not represent a probability but rather the relative likelihood of observing that value, which is essential for plotting the distribution's characteristic shape.

`pgamma(q, shape, rate)` - This calculates the value of the [Cumulative Distribution Function](#) (CDF) for a specific quantile `q`. The result is the exact probability that a random observation drawn from the distribution will be less than or equal to `q`, allowing for direct probability calculation.

`qgamma(p, shape, rate)` - As the inverse of `pgamma()`, this function calculates the [quantile](#) value associated with a specified cumulative probability `p`. It answers the question: "What value marks the boundary for the lowest `p` percent of the distribution?"

`rgamma(n, shape, rate)` - This function is vital for simulation studies, generating `n` pseudo-random values that perfectly follow the specified distribution parameters. These generated samples are

indispensable for Monte Carlo methods and hypothesis testing.

Example 1: Visualizing Probability Density using `dgamma()`

The `dgamma()` function is the primary tool for understanding the visual shape of the **gamma distribution**. By calculating the height of the probability curve (the density) across a range of values, we can accurately plot the Probability Density Function (PDF).

The resulting PDF plot clearly illustrates where observations are most concentrated and how the likelihood of observing values diminishes as they move away from the mode. In the example below, we define a sequence of x-values ranging from 0 to 2 and calculate the corresponding gamma density values, using a shape parameter of 5 and the default rate parameter of 1. A fine increment (`by=0.01`) ensures a smooth, continuous curve, offering a high-fidelity visualization of the theoretical distribution.

Define x-values from 0 to 2, using small increments for a smooth curve

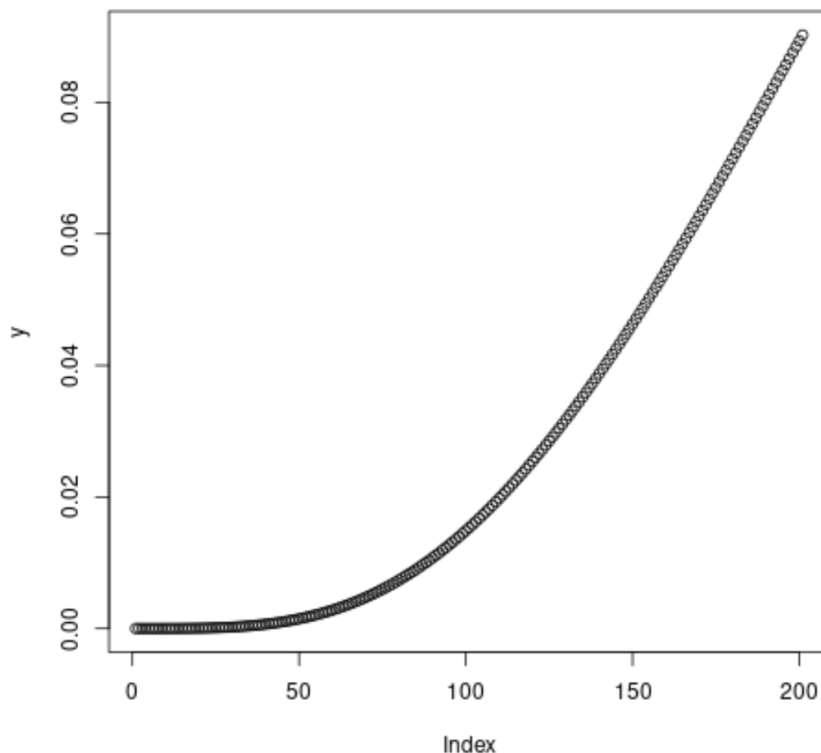
```
x <- seq(0, 2, by=0.01)
```

Calculate gamma density (y-values) for each x-value (shape=5, rate=1 default)

```
y <- dgamma(x, shape=5)
```

Create density plot to visualize the distribution's shape

```
plot(y)
```



Example 2: Calculating Cumulative Probability with pgamma()

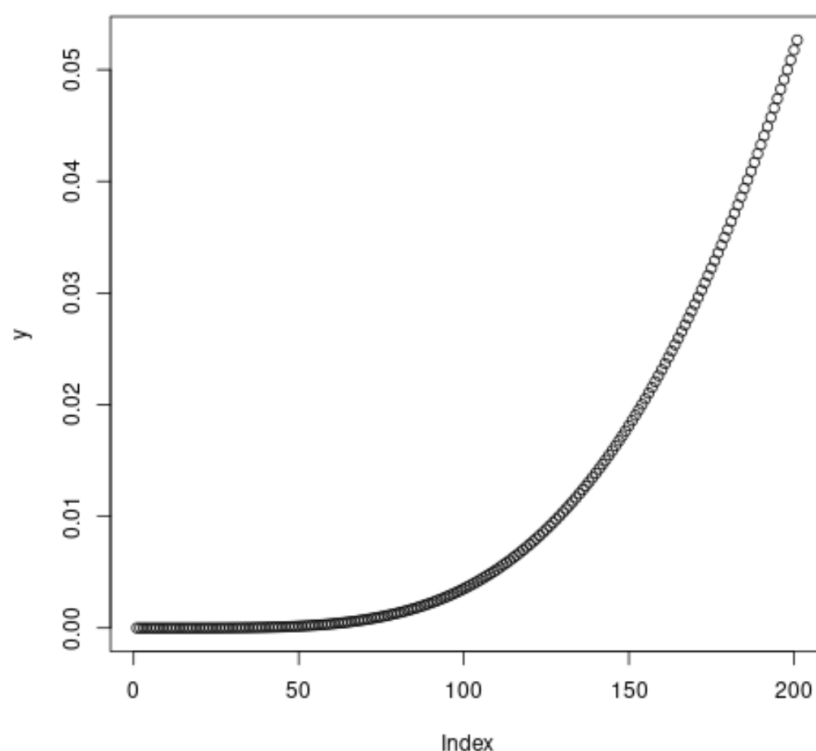
While `dgamma()` provides relative likelihood, `pgamma()` is used to calculate absolute probability. This function determines the cumulative probability up to a specific value q . If we denote X as a gamma-distributed random variable, **`pgamma(q, shape, rate)`** precisely yields $P(X \leq q)$.

This functionality is crucial for risk analysis, determining confidence intervals, and calculating the likelihood that a particular measurement (such as an insurance payout or a waiting time) will fall below a predefined critical threshold. The output of this function, when plotted, defines the Cumulative Distribution Function (CDF). Using the same parameterization (`shape=5`) as in the density example, we can observe the characteristic S-shape of the CDF, showing how the probability accumulates as the quantile value increases.

```
# Define x-values, representing the quantiles (q) to check
x <- seq(0, 2, by=0.01)

# Calculate the cumulative probability (y-values) for each x-value
y <- pgamma(x, shape=5)

# Create cumulative density plot (CDF)
plot(y)
```



Example 3: Determining Quantiles using qgamma()

The `qgamma()` function performs the inverse operation of `pgamma()`, allowing us to move from a known probability back to the corresponding data value. Given a probability p (which must range between 0 and 1), `qgamma()` returns the value that marks that specific percentile of the distribution.

This inverse capability is indispensable for tasks requiring the definition of critical boundaries. For instance, statisticians use it to calculate the median (by setting $p=0.5$), determine critical values for hypothesis testing at a specific significance level, or establish regulatory thresholds for safety or risk management. The resulting quantile plot demonstrates the relationship between the cumulative probability and the required data value, showing the rapid increase in value as the probability approaches certainty ($p=1$).

Define x-values as probabilities (p), ranging from 0 to 1

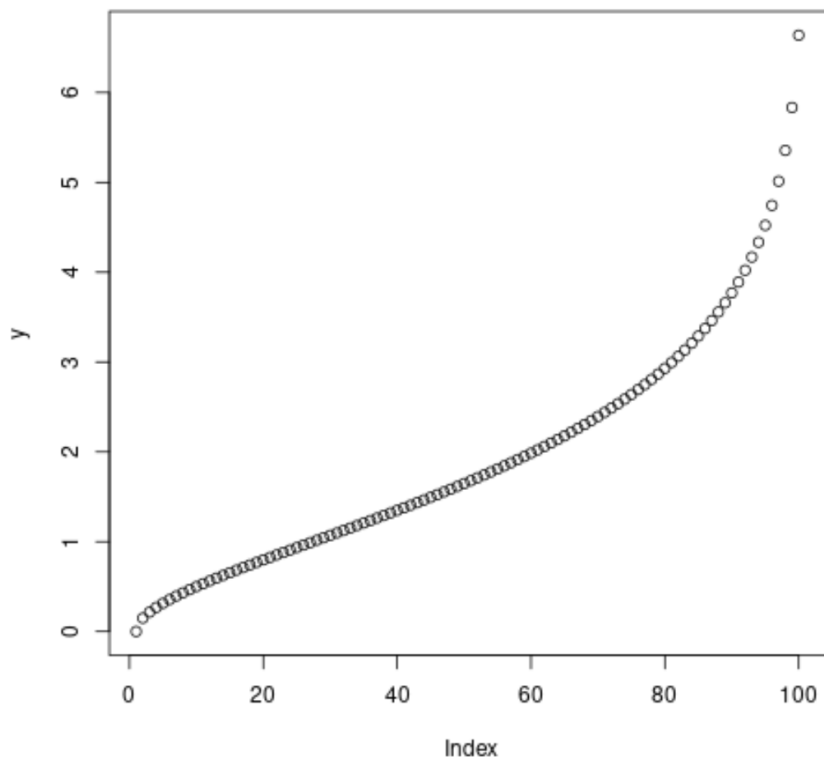
```
x <- seq(0, 1, by=0.01)
```

Calculate the quantile (y-value) corresponding to each probability

```
y <- qgamma(x, shape=5)
```

Create quantile plot

```
plot(y)
```



Example 4: Generating Random Samples with `rgamma()` for Simulation

For applications in simulation, modeling complex systems, or conducting power analysis, the ability to generate synthetic data that accurately mirrors a theoretical distribution is essential. The `rgamma()` function fulfills this need by generating a specified number (n) of random observations that strictly conform to the **gamma distribution** defined by the supplied shape and rate parameters.

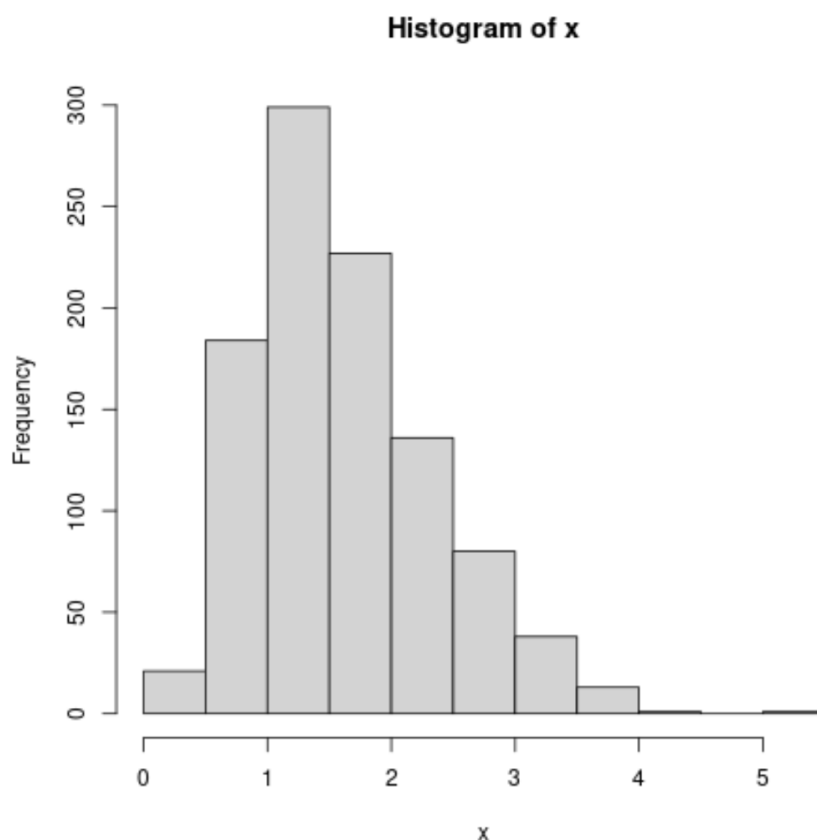
In the following code block, we generate 1,000 random values using a shape parameter of 5 and a rate parameter of 3. Crucially, we use `set.seed(0)` to ensure that the results are reproducible, a best practice for all statistical simulations. We then visualize these synthetic data points using a histogram. This visualization serves as a crucial sanity check, confirming that the frequency distribution of the generated sample closely approximates the theoretical curve derived using `dgamma()`.

```
# Make this example reproducible by setting a seed
set.seed(0)
```

```
# Generate 1,000 random values that follow gamma distribution
x <- rgamma(n=1000, shape=5, rate=3)
```

```
# Create histogram to view the frequency distribution of the generated values
```

hist(x)



Extending Knowledge to Other Statistical Distributions in R

The standardized framework demonstrated throughout this article--the use of the `d`, `p`, `q`, and `r` prefixes--is not unique to the gamma distribution. It is the universal standard for virtually all [statistical distributions](#) available within R's extensive ecosystem, including the Normal, Exponential, Chi-squared, and Beta distributions.

By mastering these four core functions as applied to the gamma distribution, you have established a robust and transferable skill set for handling any continuous or discrete probability model in R. This foundational knowledge allows for efficient exploration, analysis, and simulation across the entire spectrum of statistical modeling tasks.

The following tutorials build upon this foundation by explaining how to use other common statistical distributions in R, reinforcing the consistency of the `d/p/q/r` paradigm: