

How to Use dt, qt, pt, & rt Functions in R for Data Manipulation

Authored by
stats writer

December 30, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Use dt, qt, pt, & rt Functions in R for Data Manipulation*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=110015>

The collection of functions `dt()`, `qt()`, `pt()`, and `rt()` in **R** are core components of the statistical programming environment, specifically designed for working with probability distributions. While similar function sets exist for the Normal, Binomial, and other distributions (e.g., `dnorm`, `pbinom`), this comprehensive guide focuses exclusively on their application to the **Student t distribution**. Understanding these four functions--which handle density, probability, quantiles, and random generation, respectively--is fundamental for statistical inference, hypothesis testing, and constructing confidence intervals in **R**.

The **Student t distribution** is indispensable in statistics, particularly when dealing with small sample sizes or when the population standard deviation is unknown. This detailed tutorial provides an expert overview of how to efficiently utilize the four primary functions in **R** related to this distribution: `dt()`, `qt()`, `pt()`, and `rt()`.

dt() Function: Density Function

The `dt()` function calculates the value of the **probability density function (pdf)** for the **Student t distribution** at a specific point x . This value represents the relative likelihood of a random variable falling near x . To define the shape of the t-distribution precisely, the function requires the number of **degrees of freedom**, denoted by df .

The standard syntax for utilizing `dt()` in **R** is straightforward, requiring only the point of interest (x) and the shape parameter (df):

dt(x, df)

While `dt()` does not directly provide probabilities (as the area under a continuous PDF curve is required for that), it is crucial for understanding the curve's height at various points. Below are practical examples demonstrating how to retrieve the density value for given arguments:

#find the value of the Student t distribution pdf at x = 0 with 20 degrees of freedom

dt(x = 0, df = 20)

0.3939886

#by default, R assumes the first argument is x and the second argument is df

dt(0, 20)

0.3939886

#find the value of the Student t distribution pdf at x = 1 with 30 degrees of freedom

dt(1, 30)

0.2379933

Although `dt()` is rarely used for calculating cumulative probability, its primary utility lies in visualization. It is the necessary tool for generating the smooth curve representing the **Student t distribution**. The following code snippet details the process of creating and styling such a plot in **R**:

```
#Create a sequence of 100 equally spaced numbers between -4 and 4
```

```
x <- seq(-4, 4, length=100)
```

```
#create a vector of values that shows the height of the probability distribution
```

```
#for each value in x, using 20 degrees of freedom
```

```
y <- dt(x = x, df = 20)
```

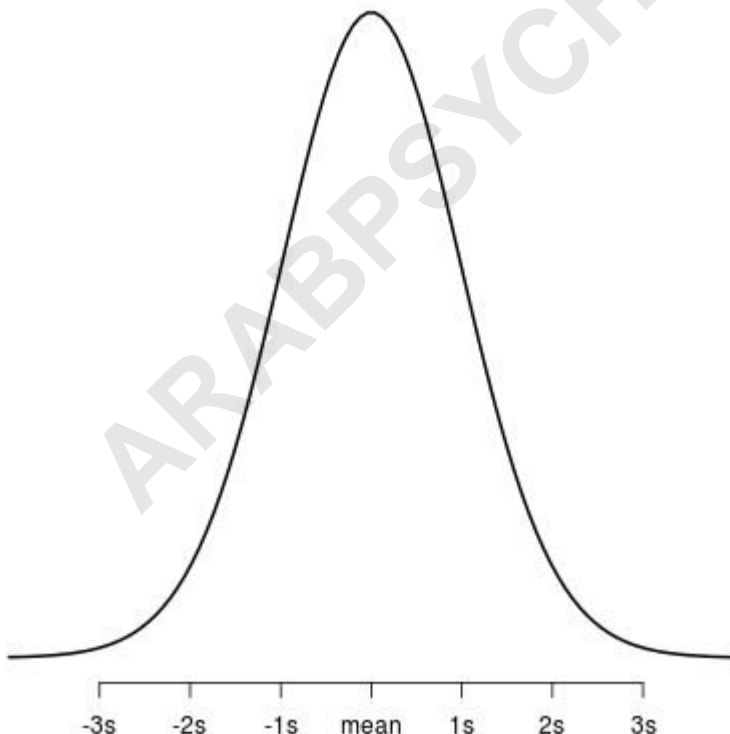
```
#plot x and y as a scatterplot with connected lines (type = "l") and add
```

```
#an x-axis with custom labels
```

```
plot(x,y, type = "l", lwd = 2, axes = FALSE, xlab = "", ylab = "")
```

```
axis(1, at = -3:3, labels = c("-3s", "-2s", "-1s", "mean", "1s", "2s", "3s"))
```

Executing this code visualizes the density curve, clearly showing the symmetrical, bell-shaped form characteristic of the **Student t distribution** with 20 **degrees of freedom**:



pt() Function: Cumulative Probability

The `pt()` function is arguably the most frequently used of the four functions for practical hypothesis testing and finding p-values. It computes the value of the **cumulative density function (cdf)** for the **Student t distribution**. In essence, `pt()` returns the probability that a random variable drawn from the distribution will be less than or equal to a specified value x , given the requisite **degrees of freedom**, df .

The base syntax for calculating the area to the left (the lower tail) of the specified x value is as follows:

pt(x, df)

If, however, you need to calculate the area to the right of x (the upper tail), you can efficiently achieve this by setting the optional argument `lower.tail = FALSE`. This is often necessary when calculating the probability associated with upper critical regions in statistical tests:

pt(x, df, lower.tail = FALSE)

The following examples illustrate how to solve common probability questions related to the t-distribution using the `pt()` function:

Example 1: Finding the Left Tail Probability. Determine the area to the **left** of a t-statistic with a value of -0.785 , assuming 14 **degrees of freedom**.

pt(-0.785, 14)

```
# 0.2227675
```

Example 2: Finding the Right Tail Probability. Calculate the area to the **right** of the t-statistic (-0.785) with 14 degrees of freedom. This can be calculated in two equivalent ways: using the complement rule or setting the lower tail argument to `FALSE`.

#Method 1: Using the complement rule (1 - area to the left)

1 - pt(-0.785, 14)

```
# 0.7772325
```

#Method 2: Directly calculating the area to the right using lower.tail = FALSE

pt(-0.785, 14, lower.tail = FALSE)

```
# 0.7772325
```

Example 3: Finding Two-Tailed Probability. Find the total area in a **Student t distribution** with 14 degrees of freedom that lies to the left of -0.785 or to the right of 0.785. This is typically used in two-tailed hypothesis testing.

```
pt(-0.785, 14) + pt(0.785, 14, lower.tail = FALSE)
```

```
# 0.4455351
```

qt() Function: Quantile Function (Inverse CDF)

The `qt()` function serves as the inverse of `pt()`. It is also known as the **quantile** function or the inverse **cumulative density function (cdf)**. Instead of inputting a t-score (x) to get a probability (p), you input the probability (p) to retrieve the corresponding **t-score** or critical value that defines that **quantile**.

This function is indispensable for establishing confidence intervals and determining **critical values** for hypothesis tests, especially when relying on specific significance levels (α). The probability p represents the cumulative area to the left of the desired **t-score**. The necessary syntax is:

```
qt(p, df)
```

Here, p is the probability (or percentile) and df is the number of **degrees of freedom**. The following code demonstrates how to find the **t-score** corresponding to various high **quantiles**, a process vital for calculating upper critical points:

```
#find the t-score of the 99th quantile (0.99 probability to the left) of the Student t distribution with df = 20
```

```
qt(.99, df = 20)
```

```
# 2.527977
```

```
#find the t-score of the 95th quantile (0.95 probability to the left) of the Student t distribution with df = 20
```

```
qt(.95, df = 20)
```

```
# 1.724718
```

```
#find the t-score of the 90th quantile (0.90 probability to the left) of the Student t distribution with df = 20
```

```
qt(.9, df = 20)
```

```
# 1.325341
```

It is important to recognize that the critical values generated by the `qt()` function in **R** are precisely the same values one would find listed in traditional **Student t distribution tables** or through other specialized statistical software. This consistency confirms `qt()` as a highly reliable method for statistical decision-making.

rt() Function: Random Generation

The `rt()` function is used for generating random numbers that adhere to the statistical properties of the **Student t distribution**. This is an essential function for simulations, Monte Carlo methods, and visualizing the distribution of samples. It requires two primary arguments: *n*, the number of random variates to generate, and *df*, the **degrees of freedom** defining the specific shape of the distribution.

The syntax for generating a sample of random variables from the t-distribution is:

rt(n, df)

A crucial aspect demonstrated by `rt()` is the relationship between the **degrees of freedom** and the distribution's variance. As the degrees of freedom increase, the t-distribution approaches the standard Normal distribution. Conversely, fewer degrees of freedom result in "heavier tails," meaning the distribution is wider and more prone to extreme values.

The examples below illustrate the generation of random samples and how to visualize the impact of varying degrees of freedom using histograms:

#generate a vector of 5 random variables that follow a Student t distribution

#with df = 20

rt(n = 5, df = 20)

```
# -1.7422445 0.9560782 0.6635823 1.2122289 -0.7052825
```

#generate a vector of 1000 random variables that follow a Student t distribution

#with df = 40 (Narrow Distribution)

```
narrowDistribution <- rt(1000, 40)
```

#generate a vector of 1000 random variables that follow a Student t distribution

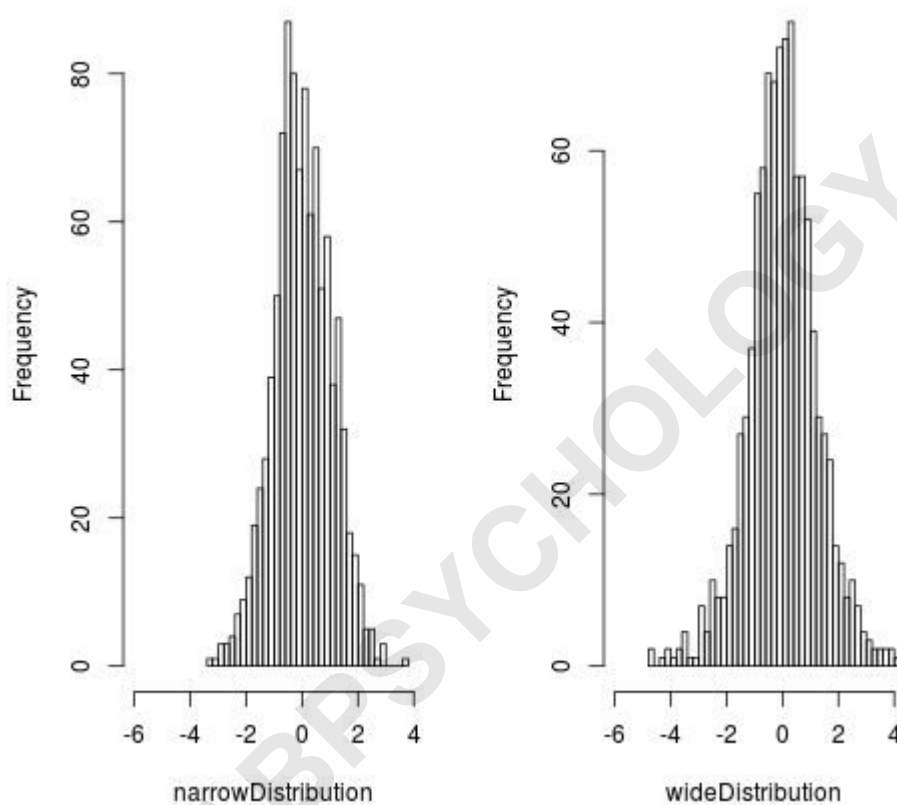
#with df = 5 (Wide Distribution)

```
wideDistribution <- rt(1000, 5)
```

#generate two histograms to view these two distributions side by side, and specify

```
#50 bars in histogram,
par(mfrow=c(1, 2)) #one row, two columns
hist(narrowDistribution, breaks=50, xlim = c(-6, 4))
hist(wideDistribution, breaks=50, xlim = c(-6, 4))
```

The comparison of the two resulting histograms visually confirms the influence of degrees of freedom:



As clearly illustrated, the distribution generated with fewer **degrees of freedom** ($df = 5$) is substantially more spread out, possessing wider tails compared to the distribution with higher **degrees of freedom** ($df = 40$). This visual difference underscores the necessity of correctly specifying the *df* parameter when conducting statistical analysis using the t-distribution.

Summary of T-Distribution Functions in R

The family of functions--`dt()`, `pt()`, `qt()`, and `rt()`--provides a complete toolkit for analyzing and manipulating the Student t distribution within **R**. Whether you are plotting the density curve, calculating a precise p-value, or determining critical boundaries for statistical significance, these functions ensure robust and accurate statistical inference. Mastery of these four functions is

essential for any professional working with inferential statistics in the R environment.

d (Density): Calculates the height of the **probability density function (pdf)**.

p (Probability): Calculates the **cumulative density function (cdf)** (area under the curve).

q (Quantile): Calculates the critical value or **t-score** for a given probability.

r (Random): Generates a random sample of variates from the distribution.

Further Reading:

[A Guide to dnorm, pnorm, qnorm, and rnorm in R](#)

[A Guide to dbinom, pbinom, qbinom, and rbinom in R](#)

ARABPSYCHOLOGY.COM