

Learn How to Perform an Anderson-Darling Goodness-of-Fit Test in R

Authored by
Mohammed looti

November 9, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Perform an Anderson-Darling Goodness-of-Fit Test in R*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=14229>

The [Anderson-Darling Test](#) is a powerful and widely respected [goodness of fit test](#) used in statistics. Its primary function is to rigorously measure how well observed data conforms to a specific theoretical cumulative distribution function. While it can be adapted for various distributions, it is most frequently employed to ascertain whether a dataset follows a [normal distribution](#), often referred to as testing for normality. The test places particular emphasis on the tails of the distribution, making it generally more sensitive than tests like the Kolmogorov-Smirnov test, especially when deviations occur at the extremes.

Confirming normality is an essential prerequisite for many parametric [statistical tests](#), including the ubiquitous t-tests, Analysis of Variance (ANOVA), and various regression models. Failing to verify the distributional assumptions can lead to unreliable inferences and potentially flawed conclusions. This guide provides an in-depth exploration of how to execute and interpret the Anderson-Darling Test efficiently using the statistical programming language R.

Understanding the Anderson-Darling Test: Theory and Hypotheses

The core mechanism of the Anderson-Darling (A-D) test relies on comparing the empirical cumulative distribution function (ECDF) of the sample data against the cumulative distribution function (CDF) of the hypothesized distribution (e.g., the normal distribution). The resulting test statistic, denoted as A^2 or simply A , quantifies the magnitude of the discrepancy between these two functions. A higher value of the test statistic A suggests a poorer fit between the data and the specified theoretical distribution.

The test is structured around two competing hypotheses that must be clearly defined before execution. Understanding these hypotheses is fundamental to correctly interpreting the output, particularly the [p-value](#):

Null Hypothesis (H₀): The data follows the specified distribution (e.g., the data is normally distributed).

Alternative Hypothesis (H₁): The data does not follow the specified distribution.

The decision to reject or fail to reject the [Null hypothesis](#) hinges entirely on the calculated p-value relative to a predetermined significance level, often denoted as α (alpha). Common choices for α include 0.10, 0.05, and 0.01. If the p-value is less than α , we reject H₀, concluding that the data significantly deviates from the hypothesized distribution.

Prerequisites for R Implementation

To successfully conduct the [Anderson-Darling Test](#) in R, we must utilize the specialized `ad.test()` function, which is contained within the `nortest` package. If you are working in a fresh R environment, the first necessary step is to install this package from the Comprehensive R Archive

Network (CRAN) and then load it into your current session using the **library()** function. Failure to load the library will result in an error when attempting to call the **ad.test()** function.

The following code block demonstrates the necessary setup steps, followed by the generation of a reproducible dataset. We define a vector of 100 values that are explicitly generated from a standard normal distribution (mean=0, standard deviation=1) using the **rnorm()** function. This controlled environment allows us to verify that the test accurately identifies data generated under ideal conditions.

Example 1: Testing Randomly Generated Normal Data in R

We will now execute the Anderson-Darling test on our synthetic, normally distributed data vector, x . The goal is to confirm whether the test correctly identifies that the data adheres to the normal distribution assumption, which we know to be true based on how the data was generated. The use of **set.seed(1)** ensures that anyone replicating this analysis will obtain the identical results, promoting reproducibility in statistical programming.

The code below illustrates the installation, loading, data generation, and execution of the A-D test:

```
# Install (if not already installed) and load the required nortest library
```

```
install.packages('nortest')
```

```
library(nortest)
```

```
# Make this example reproducible
```

```
set.seed(1)
```

```
# Define vector of 100 values that are normally distributed
```

```
x <- rnorm(100, 0, 1)
```

```
# Conduct Anderson-Darling Test to test for normality
```

```
ad.test(x)
```

```
# Anderson-Darling normality test
```

```
#
```

```
#data: x
```

```
#A = 0.16021, p-value = 0.9471
```

Interpreting the Results: The Test Statistic and p-Value

The output generated by the **ad.test(x)** function provides two critical values necessary for hypothesis testing: the calculated test statistic **A** and the corresponding [p-value](#). These metrics

form the basis for our conclusion regarding the [normal distribution](#) assumption.

A (Test Statistic): This value represents the deviation between the empirical distribution of the sample data and the theoretical normal distribution. A smaller **A** value indicates a better fit. In our first example, **A = 0.16021**.

p-value: This probability measures the likelihood of observing the current data (or data more extreme) assuming the [null hypothesis](#) (normality) is true. If the p-value is very small, it suggests the observed data is highly unlikely if the null hypothesis were true, leading us to reject H₀.

In the first example, the resulting **p-value is 0.9471**. If we establish a standard significance level (α) of 0.05, we must compare the p-value against this threshold. Since 0.9471 is significantly greater than 0.05, we do not have sufficient evidence to reject the null hypothesis. We confidently conclude that the data follows a normal distribution, which perfectly aligns with our knowledge that the data was generated using the **rnorm()** function with a mean of 0 and a standard deviation of 1. This outcome validates the function's ability to correctly identify normally distributed data.

Example 2: Testing Uniformly Distributed Data

To further demonstrate the power of the [Anderson-Darling Test](#), let us consider a scenario where the data clearly violates the assumption of normality. We will generate a new vector of 100 values using R's **runif()** function, which produces a uniform distribution between 0 and 1. A uniform distribution is fundamentally different from a normal distribution, so we anticipate that the A-D test will correctly detect this violation and lead us to reject the null hypothesis of normality.

By executing the test on this new dataset, we demonstrate how the test statistic **A** increases and the [p-value](#) decreases when the data significantly deviates from the hypothesized distribution. The procedure remains identical, requiring only a change in the data generation function:

Make this example reproducible

```
set.seed(1)
```

```
# Define vector of 100 values that are uniformly distributed
```

```
x <- runif(100, 0, 1)
```

```
# Conduct Anderson-Darling Test to test for normality
```

```
ad.test(x)
```

```
# Anderson-Darling normality test
```

```
#
```

```
#data: x
```

```
#A = 1.1472, p-value = 0.005086
```

In this second case, the calculated test statistic **A** equals 1.1472, which is substantially higher than the 0.16021 observed in the previous example. More importantly, the corresponding **p-value is 0.005086**. Since 0.005086 is less than our chosen significance level of 0.05, we have robust evidence to reject the [null hypothesis](#). The formal conclusion is that the data does not follow a normal distribution, which is the expected result given that the data originated from a uniform distribution. This contrast effectively highlights the test's discriminative power.

Applying the Test to Real-World Data: Analyzing a Data Frame Column

The [Anderson-Darling Test](#) is most frequently applied to specific variables within complex datasets rather than synthetic vectors. R provides many built-in datasets for demonstration, one of the most famous being the **iris** dataset, which contains measurements on 150 iris flowers. When working with data frames, we must specify the exact column we wish to test by using the dollar sign operator (`$`).

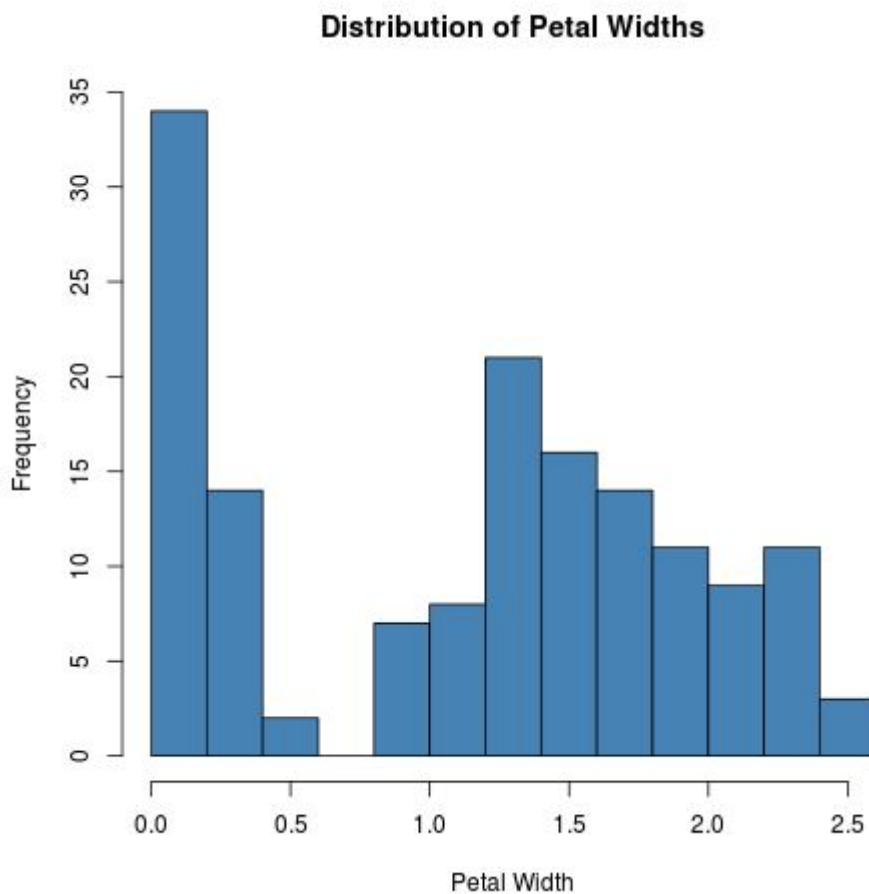
First, we examine the structure of the data frame to identify the variables we might want to test for normality:

```
# View first six lines of the built-in iris dataset  
head(iris)
```

```
# Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
#1 5.1 3.5 1.4 0.2 setosa  
#2 4.9 3.0 1.4 0.2 setosa  
#3 4.7 3.2 1.3 0.2 setosa  
#4 4.6 3.1 1.5 0.2 setosa  
#5 5.0 3.6 1.4 0.2 setosa  
#6 5.4 3.9 1.7 0.4 setosa
```

Our objective is to determine if the variable *Petal.Width* is normally distributed. Before conducting any formal test, it is highly recommended practice to visually inspect the data's distribution. We generate a histogram to get an initial sense of the shape of the data. If the distribution appears multimodal or heavily skewed, it is unlikely to meet the normality assumption.

```
hist(iris$Petal.Width, col = 'steelblue', main = 'Distribution of Petal Widths',  
xlab = 'Petal Width')
```



The histogram clearly suggests that the distribution of *Petal.Width* is not bell-shaped; instead, it exhibits a distinct bimodal pattern. To confirm this visual assessment with statistical rigor, we apply the `ad.test()` function directly to the specified column:

```
# Conduct Anderson-Darling Test to test for normality  
ad.test(iris$Petal.Width)
```

```
# Anderson-Darling normality test  
#  
#data: iris$Petal.Width  
#A = 5.1057, p-value = 1.125e-12
```

The results confirm our visual inspection. The test statistic **A** is extremely high (5.1057), and the resulting [p-value](#) is exceptionally small (1.125e-12, or virtually zero). Since this p-value is orders of magnitude smaller than any conventional significance level (e.g., 0.05), we decisively reject the null hypothesis. We conclude that we have overwhelming evidence that the *Petal.Width* variable in the **iris** dataset does not follow a [normal distribution](#), advising caution if using this variable in

parametric [statistical tests](#) that rely on this assumption.