

How to Create a Q-Q Plot in SAS

Authored by
stats writer

November 19, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Create a Q-Q Plot in SAS*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=96975>

Generating a Q-Q plot in SAS is a fundamental skill for data analysts seeking to validate distributional assumptions. The process centers around the powerful **PROC UNIVARIATE** procedure. Initially, the user must invoke the procedure, specifying the target data set and utilizing the **QQPLOT** statement. Essential steps follow, requiring the identification of the variable to be analyzed and, optionally, specifying advanced options related to quantiles, such as custom theoretical distributions or specific plotting parameters. Executing this concise program yields the visualization crucial for assessing adherence to a theoretical model, most commonly the normal distribution.

Introduction to Quantile-Quantile (Q-Q) Plots

A **Q-Q plot**, which stands for "quantile-quantile" plot, serves as an indispensable graphical tool in statistical analysis. Its primary function is to determine whether an empirical dataset originates from a specified theoretical probability distribution. By comparing the quantiles of the observed data against the theoretical quantiles of the hypothesized distribution, the plot provides a powerful visual assessment of the fit, moving beyond simple numerical tests. This visual approach allows analysts to quickly identify potential problems such as skewness, heavy tails, or bimodality that violate the underlying distributional assumptions necessary for many parametric tests.

Although Q-Q plots can test against any distribution--such as the Uniform, Exponential, or Weibull distributions--their most frequent and critical application involves checking for normal distribution. This check is vital because numerous statistical methodologies, including t-tests, ANOVA, and linear regression, rely fundamentally on the assumption that the residuals or the data itself are normally distributed. Failure to meet this assumption can lead to invalid statistical inferences and unreliable model parameters. Consequently, mastering the creation and interpretation of the Q-Q plot is crucial for maintaining the integrity of statistical modeling.

The interpretation of the plot is intuitively straightforward: if the sample data aligns perfectly with the chosen theoretical distribution, the plotted points will fall precisely along a straight diagonal reference line, often called the line of identity. This line represents where the theoretical quantile equals the observed quantile. However, if the data deviates significantly from this straight line--showing pronounced curvature, S-shapes, or isolated clusters--it indicates a high likelihood that the data does not follow the hypothesized distribution. Minor deviations, especially at the extremes (the tails), are often tolerated, but severe or systematic departures mandate the use of non-parametric methods or data transformation.

Essential SAS Procedure: **PROC UNIVARIATE**

Within the SAS programming environment, the **PROC UNIVARIATE** procedure is the most efficient and robust method for generating quantile-quantile plots. This procedure is specifically designed

for comprehensive univariate analysis, providing not only graphical output but also descriptive statistics, tests for normality, and outlier detection. To isolate the Q-Q plot visualization from the extensive textual output this procedure typically generates, analysts frequently incorporate the **NOPRINT** option within the procedure statement. This ensures that only the desired plot is produced, streamlining the output review process.

The core syntax for creating a Q-Q plot against the default theoretical distribution (the normal distribution) is surprisingly concise. It requires calling **PROC UNIVARIATE**, specifying the dataset, and then using the dedicated **QQPLOT** statement followed by the variable name under investigation. This simple structure allows SAS to automatically calculate the necessary empirical and theoretical quantiles, generating the graph instantly upon execution.

Consider the following fundamental code block. This template is the starting point for any Q-Q plot analysis in SAS, demonstrating the minimum required commands to execute the plot generation efficiently:

```
proc univariate data=my_data noprint;  
qqplot my_variable;  
run;
```

It is important to emphasize the function of the **NOPRINT** option here. By default, **PROC UNIVARIATE** generates detailed output tables, including moments, basic statistical measures, and tests of location. Since the focus of this exercise is purely on the graphical assessment of distribution shape, **NOPRINT** suppresses all these extraneous tables, resulting in a cleaner output log. The subsequent examples will build upon this essential structure, showcasing its application in practical scenarios involving both normally and non-normally distributed data.

Example 1: Q-Q Plot for Normally Distributed Data

To demonstrate the ideal outcome of a Q-Q plot, we will first generate a synthetic dataset that adheres strictly to a normal distribution. This controlled scenario provides a baseline for interpreting plots derived from real-world data, where assumptions are often less perfectly met. We utilize the **RANNOR** function within a SAS DATA step to create 1,000 observations, simulating a population with a mean (μ) of 10 and a standard deviation (σ) of 2. This function is essential for generating random numbers that follow the standardized normal distribution, which we then scale and shift to meet our specified parameters.

The subsequent step involves applying the standard **PROC UNIVARIATE** syntax to this newly created dataset, `normal_data`. We instruct SAS to generate the Q-Q plot for the variable `x` using the **QQPLOT** statement. Since the data was explicitly generated to be normal, we expect the

resulting plot points to exhibit a strong linear relationship with the theoretical line, confirming the data's distributional properties.

The complete SAS code block, encompassing both the data generation and the plot creation, is presented below. Notice the use of descriptive comments (starting with `/*`) to clearly delineate the purpose of each section of the script, enhancing readability and maintainability for future analysis.

```
/*generate 1000 values that follow normal distribution with mean 10 and sd 2 */
```

```
data normal_data;
```

```
do i = 1 to 1000;
```

```
x = 10 + 2*rannor(1);
```

```
output;
```

```
end;
```

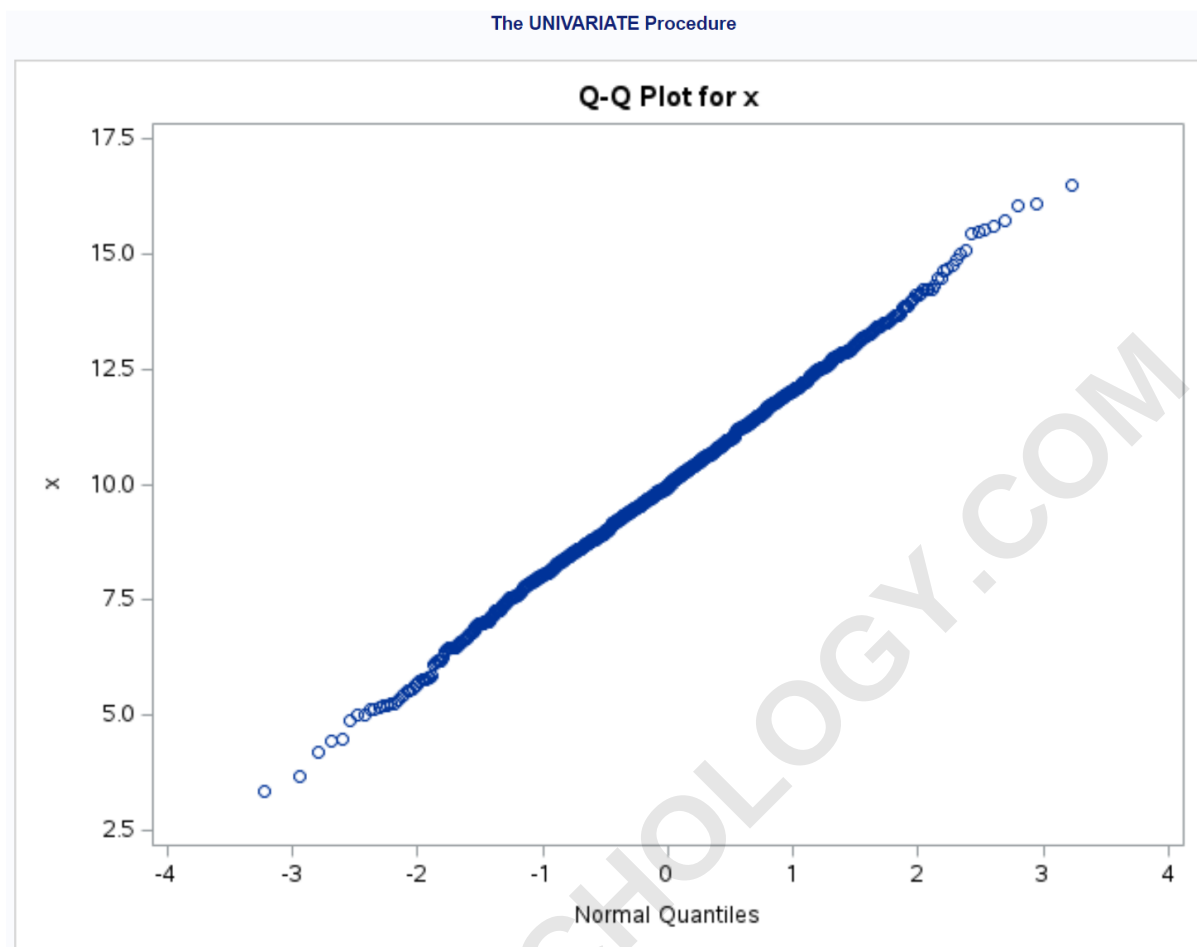
```
run;
```

```
/*create q-q plot*/
```

```
proc univariate data=normal_data noprint;
```

```
qqplot x;
```

```
run;
```



Interpreting the Results of Normal Data Simulation

Upon visual inspection of the generated Q-Q plot, the evidence overwhelmingly supports the hypothesis of normality. The vast majority of the data points cluster tightly along the central diagonal reference line. This straight alignment confirms that the empirical quantiles of our sample data closely match the theoretical quantiles derived from the standard normal distribution. This outcome serves as the classic visual signature for data that is suitable for parametric statistical procedures.

While the plot shows excellent linearity in the core of the distribution, it is common to observe slight, minor deviations near the extreme edges, or "tails," of the plot. These small variances are generally considered normal sampling variability, especially when dealing with finite datasets like our 1,000 observations. Unless these tail deviations are severely pronounced or systematic--such as forming a heavy curvature--they typically do not invalidate the overall assumption of normality.

In summary, based on the clear, near-perfect linearity displayed in the output image, we can confidently conclude that the synthetic dataset is normally distributed. This analysis validates the

use of the **QQPLOT** statement within **PROC UNIVARIATE** for accurately assessing distributional fit in SAS. The subsequent example will contrast this ideal scenario with data drawn from a fundamentally different distribution.

Example 2: Analyzing Non-Normal Data (Exponential Distribution)

To illustrate how a Q-Q plot clearly identifies deviations from normality, we now generate a second dataset, `exp_data`, using the **RANEXP** function. This function generates random numbers that follow an exponential distribution, which is inherently skewed and distinctly non-normal. We again generate 1,000 observations for consistency, ensuring the scale parameter is set to 1. This contrast is crucial: while the first example showed a linear pattern, this example demonstrates the classic curvilinear shape associated with skewed data when tested against the normal model.

The procedure for generating the plot remains identical to Example 1, emphasizing the simplicity of the SAS syntax regardless of the underlying data structure. We pass `exp_data` to **PROC UNIVARIATE** and request a **QQPLOT** of variable `x`. Because the exponential distribution is characterized by its long right tail, we anticipate a severe, systematic deviation from the diagonal reference line, manifesting as a curve rather than a straight path. This deviation occurs because the empirical quantiles of the highly skewed exponential data are fundamentally different from those expected under the symmetric normal model.

The full SAS script for generating and plotting the exponentially distributed data is detailed below. Observe that the logic of the plot procedure itself is universal; only the data generation step changes, highlighting the flexibility of **PROC UNIVARIATE**.

```
/*generate 1000 values that follow an exponential distribution*/
```

```
data exp_data;
```

```
do i = 1 to 1000;
```

```
x = ranexp(1);
```

```
output;
```

```
end;
```

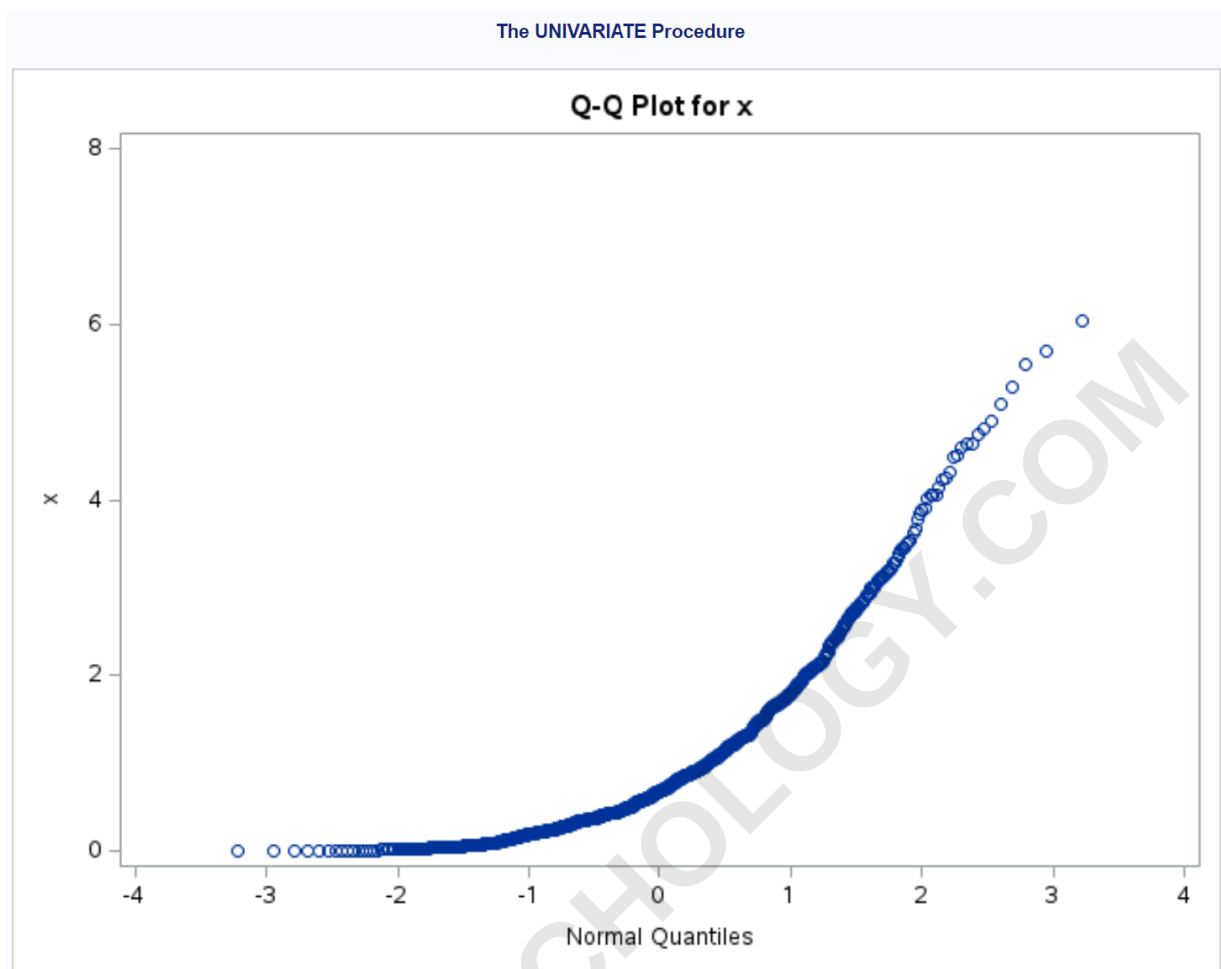
```
run;
```

```
/*create q-q plot*/
```

```
proc univariate data=exp_data noprint;
```

```
qqplot x;
```

```
run;
```



Interpreting the Non-Normal Q-Q Plot

As clearly visualized in the second Q-Q plot, the data points dramatically depart from the theoretical straight diagonal line. Instead of clustering along the line, the points form a distinct upward-curving pattern. This severe, non-linear distortion is the definitive visual signal that the dataset is **not normally distributed**. Specifically, this upward curvature indicates a strong positive skew, which is characteristic of the underlying exponential distribution.

The deviation occurs because the sample quantiles (plotted on the y-axis) rise much faster than the expected normal quantiles (plotted on the x-axis) in the upper range of the data. When the data is heavily skewed, the observed tail values are significantly larger than what would be predicted by a symmetric normal model. Understanding this systematic curvature allows statisticians to diagnose the type of departure from normality--whether it is due to skewness, heavy tails (kurtosis), or modality issues--even before running formal statistical tests.

Since we intentionally generated the data using an exponential distribution function (RANEXP), the result aligns perfectly with statistical expectation. For data exhibiting this level of non-normality,

analysts in SAS must either apply transformations (such as logarithmic or square root) to attempt to normalize the data, or pivot to robust non-parametric tests that do not require the assumption of normality.

Advanced Customization of Q-Q Plots in SAS

While the basic **QQPLOT** statement defaults to testing against the normal distribution, **PROC UNIVARIATE** in SAS offers extensive customization options, allowing analysts to test data against other theoretical distributions. The procedure supports fitting distributions such as the Lognormal, Exponential, Weibull, and Gamma distributions, among others. To specify a different theoretical distribution, you simply include the distribution name after the **QQPLOT** statement. For instance, testing the `exp_data` from the previous example against its true distribution--the exponential--would result in a plot that is highly linear, confirming that while it is not normal, it is indeed exponential.

Furthermore, SAS provides options to adjust the calculation of the empirical quantiles. By default, **PROC UNIVARIATE** uses one of five methods to estimate the quantiles, which can sometimes influence the appearance of the plot, especially with smaller sample sizes. Analysts can use the **RANKADJ** and **RANKMULT** options, or the **PLOT** statement combined with the **ODS GRAPHICS** settings, to control the plotting parameters, including specifying the color, marker size, and adding confidence bands. Confidence bands are particularly useful as they provide a visual boundary; if the data points fall largely outside these bands, the null hypothesis (that the data comes from the specified distribution) is rejected more strongly.

For example, if an analyst wished to plot the quantiles of a variable named `response` against a Lognormal distribution, the syntax would be modified as follows: `qqplot response / lognormal;`. This flexibility ensures that the Q-Q plot remains a versatile diagnostic tool, adaptable to diverse distributional assumptions encountered across various scientific and business domains. Understanding these options moves the user from basic visual confirmation to rigorous distributional testing within the SAS environment.

Conclusion and Further Statistical Applications

The ability to generate and accurately interpret a Q-Q plot in SAS using **PROC UNIVARIATE** is fundamental for robust statistical practice. Whether validating the assumption of a normal distribution for linear models or testing data against specific theoretical models like the exponential distribution, the Q-Q plot offers superior diagnostic clarity compared to relying solely on numerical tests of normality such as Shapiro-Wilk or Kolmogorov-Smirnov. A straight line signifies distributional congruence, whereas systematic curvature immediately alerts the analyst to violations of assumptions, guiding subsequent methodological choices.

In practical data analysis, Q-Q plots are often used in conjunction with histograms and box plots to provide a comprehensive view of the data's shape. When non-normality is detected, the analyst might consider techniques such as rank transformation or non-parametric alternatives, or perhaps investigate the possibility of outliers or heterogeneity that could be driving the deviation. SAS provides all the necessary tools to handle these complex situations efficiently, ensuring that statistical modeling is built upon sound distributional foundations.

For those seeking to expand their SAS proficiency, exploring the detailed documentation for **PROC UNIVARIATE** is highly recommended, particularly regarding options for confidence limits and testing against alternative distributions. The following tutorials provide guidance on performing other common and essential data manipulation and analysis tasks within the SAS environment:

How to Perform a Two-Sample T-Test in SAS

Guide to Running Linear Regression Analysis in SAS

Calculating Descriptive Statistics using PROC MEANS