

How to Perform Quantile Regression in R?

Authored by
stats writer

December 14, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Perform Quantile Regression in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=107429>

Introduction to Regression Models: Beyond the Mean

Linear regression is perhaps the most ubiquitous method in statistics for modeling the relationship between one or more predictor variables and a corresponding response variable. The fundamental goal of traditional linear regression, relying on the method of Ordinary Least Squares (OLS), is to estimate the conditional **mean** value of the response variable. This approach assumes that the errors are normally distributed and that the relationship across all levels of the predictor is uniform, focusing squarely on the central tendency of the data.

While estimating the mean is often sufficient, there are many practical situations where we are interested in estimating the effect of predictors on the entire conditional distribution of the response, not just its average. For instance, when studying economic inequality, environmental risk assessment, or academic performance, understanding the extremes--the high or low end of the distribution--provides crucial insights that the mean tends to obscure. This necessity for a more comprehensive distributional analysis leads us away from traditional OLS methods.

In response to these limitations, statisticians developed **Quantile Regression**. This powerful technique, pioneered by Roger Koenker and Gilbert Bassett, allows us to model the relationship between a set of predictors and specific quantiles (percentiles) of the response variable distribution. Instead of focusing solely on the 50th percentile (the median, which standard OLS can sometimes approximate), quantile regression enables robust estimation of any percentile value, such as the 10th percentile, the 90th percentile, or the 98th percentile.

Understanding Quantile Regression and Its Advantages

The core distinction of Quantile Regression lies in its objective function. Unlike OLS, which minimizes the sum of squared errors, quantile regression minimizes the sum of asymmetrically weighted absolute residuals. This approach offers several profound statistical advantages, making it particularly useful when dealing with complex, real-world data distributions. First, it is significantly more robust to outliers in the response variable, as minimizing absolute errors is less sensitive to extreme values than minimizing squared errors.

Second, and more importantly, quantile regression does not require strict assumptions about the distribution of the errors, making it a non-parametric technique in terms of the error distribution. This relaxation is crucial when the conditional distribution of the response variable is heterogeneous--meaning the spread or variability of the response changes as the predictor changes (a phenomenon known as heteroscedasticity). In such cases, OLS estimates are still unbiased but become inefficient and may yield misleading standard errors. Quantile regression accurately models how the effect of a predictor changes across different parts of the response distribution.

For example, if we study the effect of hours studied on exam scores, the impact of an additional hour might be very different for a student who is already performing in the top 90% (where returns might diminish) versus a student performing in the bottom 10% (where returns might be high). Quantile regression can estimate these differential effects by fitting separate models for the 10th, 50th, and 90th percentiles, providing a much richer understanding of the relationship than a single mean estimate could ever offer.

The `quantreg` Package in R: Syntax and Parameters

To implement Quantile Regression within the R programming environment, we rely on the excellent **quantreg** package, developed by Koenker himself. This package provides the primary function, **rq()**, which stands for "quantile regression." Mastering the syntax of **rq()** is the key to effectively using this methodology.

The core syntax for the **rq()** function mirrors that of standard linear modeling functions in R (like **lm()**), but includes an essential extra parameter: `tau`. The structure requires specifying the relationship formula, the dataset, and the target quantile. The default setting for the `tau` parameter is 0.5, which estimates the conditional median (the 50th percentile), providing a robust alternative to the OLS mean estimate.

The general syntax utilized for running a quantile regression model is as follows, requiring the installation and loading of the necessary package:

library(quantreg)

```
model <- rq(y ~ x, data = dataset, tau = 0.5)
```

Understanding the components within this function call is critical for generating accurate models. The following list defines the four main parameters used in the formula above:

y: Represents the response variable (the dependent variable) that we are seeking to model.

x: Denotes the predictor variable(s) (the independent variables) used to explain variation in the response.

data: Specifies the name of the dataset or data frame containing the variables.

tau: This crucial parameter defines the specific **quantile** (percentile) of the response distribution we wish to estimate. The value must be a number between 0 and 1. Setting `tau = 0.5` estimates the median, while `tau = 0.9` estimates the 90th percentile, for example.

Case Study: Setting Up the Data Environment

To illustrate the application of R's **rq()** function, we will construct a simulated dataset focused on

student performance. Our goal is to examine how the number of hours a student spends studying affects their final exam score. By utilizing a simulated dataset, we can ensure reproducibility and clearly demonstrate the concepts involved in quantile modeling. We will use 100 observations, representing 100 different students, capturing their study hours and corresponding scores.

In generating this data, we introduce a degree of heteroscedasticity--the variability in scores will increase as the hours studied increase. This is a realistic scenario: students who study little might all score similarly low, but students who study many hours might either score extremely high or only moderately high due to external factors, creating greater variance at the high end. This inherent characteristic makes the dataset ideal for showcasing the power of Quantile Regression over traditional OLS.

We begin by setting a seed for reproducibility and then generating the predictor variable (hours studied) and the response variable (exam score) based on a simple linear relationship plus heterogeneous noise, before assembling them into a data frame named `df`. The initial steps involve basic data frame manipulation in R, ensuring the dataset is correctly structured for the subsequent regression analysis.

#make this example reproducible

```
set.seed(0)
```

```
#create data frame
```

```
hours <- runif(100, 1, 10)
```

```
score <- 60 + 2*hours + rnorm(100, mean=0, sd=.45*hours)
```

```
df <- data.frame(hours, score)
```

```
#view first six rows
```

```
head(df)
```

```
hours score
```

```
1 9.070275 79.22682
```

```
2 3.389578 66.20457
```

```
3 4.349115 73.47623
```

```
4 6.155680 70.10823
```

```
5 9.173870 78.12119
```

```
6 2.815137 65.94716
```

Executing Quantile Regression for the 90th Percentile

With the data successfully entered and structured within the `df` data frame, the next step is to execute the quantile regression model. For this demonstration, we are specifically interested in

modeling the **upper end** of the performance distribution. Therefore, we will fit a model designed to predict the expected 90th percentile of exam scores (the scores achieved by high-performing students) based on the number of hours studied. This choice of `tau = 0.9` allows us to analyze the marginal effect of study hours specifically on the students who generally score well.

We load the **quantreg** package and then call the **rq()** function. In the formula syntax, `score` is the response, `hours` is the predictor, the data is `df`, and we set `tau = 0.9`. After fitting the model object (named `model`), we use the `summary()` function to generate detailed statistics regarding the coefficients. Unlike OLS, which relies on t-tests, the summary output for quantile regression often utilizes rank-based tests or bootstrapping for inference, providing robust confidence intervals.

Running the analysis reveals the estimated intercept and the slope coefficient, which specifically describe the linear relationship at the 90th quantile of the score distribution. This coefficient informs us how much the 90th percentile score is expected to increase for every one-unit increase in hours studied.

library(quantreg)

```
#fit model
```

```
model <- rq(score ~ hours, data = df, tau = 0.9)
```

```
#view summary of model
```

```
summary(model)
```

```
Call: rq(formula = score ~ hours, tau = 0.9, data = df)
```

```
tau: 0.9
```

```
Coefficients:
```

```
coefficients lower bd upper bd
```

```
(Intercept) 60.25185 59.27193 62.56459
```

```
hours 2.43746 1.98094 2.76989
```

Interpreting the Quantile Regression Output

The summary output provides the estimated coefficients for the 90th percentile model. This information is used to construct the estimated Quantile Regression equation. From the results above, we extract the intercept (60.25185) and the coefficient for `hours` (2.43746). These values allow us to formulate the following regression equation specific to the 90th percentile of exam scores:

90th percentile of exam score = $60.25 + 2.437 * (\text{hours})$

The interpretation of these coefficients is nuanced. The intercept (60.25) suggests that a student who studies zero hours is predicted to score 60.25 points, provided they are performing at the 90th percentile of all zero-hour students. More importantly, the slope coefficient (2.437) means that for students performing at the 90th percentile level, every additional hour studied is associated with an increase of 2.437 points in their expected exam score. This magnitude of change is specific to the upper tail of the score distribution.

We can use this equation for prediction. For instance, to predict the 90th percentile score for a student who studies 8 hours, we substitute this value into the equation: 90th percentile of exam score = $60.25 + 2.437 * (8)$. This calculation yields an estimated score of **79.75**. This does not mean the average score for an 8-hour student is 79.75; rather, it means 90% of students who study 8 hours are expected to score below 79.75, and 10% are expected to score above it. The output also handily provides the upper and lower confidence limits for both the intercept and the predictor variable `hours`, allowing for formal statistical inference on these quantile effects.

Visualizing and Comparing Quantile Regression Results

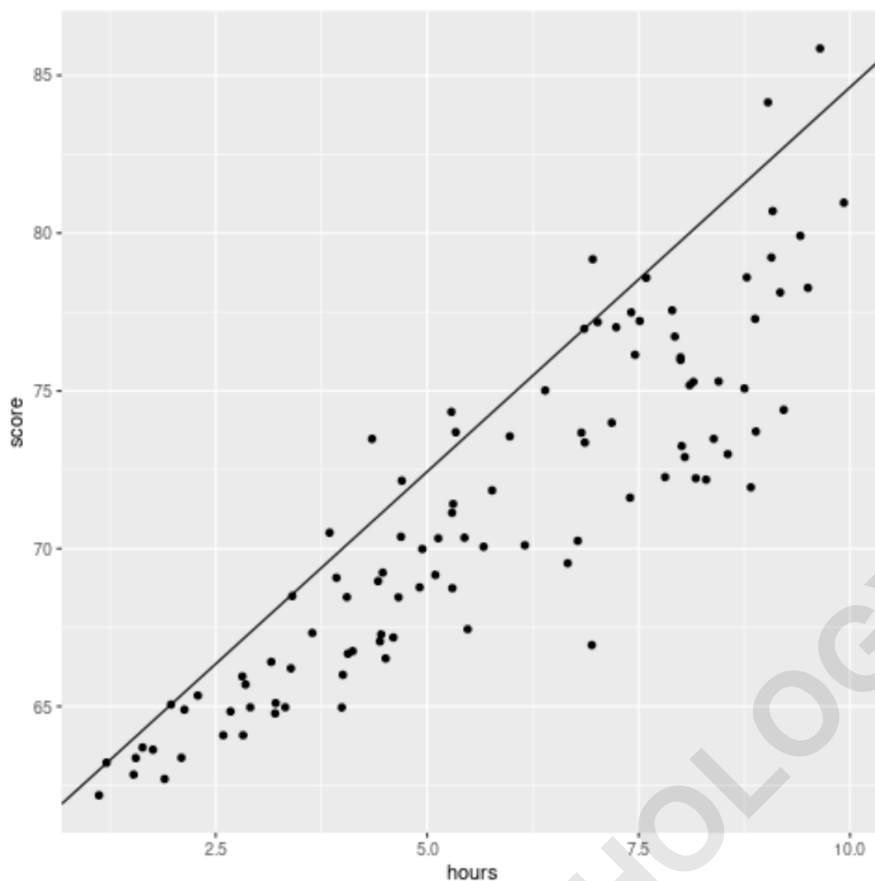
Visualization is an essential component of statistical analysis, particularly for [Quantile Regression](#), as it helps illustrate how the fitted line differs from the traditional mean-based line. We can use the powerful **ggplot2** package in [R](#) to create a scatterplot of the raw data and overlay the fitted quantile regression line. This visual representation demonstrates precisely where the 90th percentile line falls relative to the data points.

To plot the line, we use the `geom_abline()` function, pulling the intercept and slope coefficients directly from the `model` object we created earlier.

library(ggplot2)

```
#create scatterplot with quantile regression line
ggplot(df, aes(hours,score)) +
  geom_point() +
  geom_abline(intercept=coef(model), slope=coef(model))
```

The resulting plot clearly shows the positioning of the fitted line:



Crucially, notice that this fitted line does not pass through the "heart" or center of the data points, which is the path a traditional [linear regression](#) line would follow. Instead, the line consistently lies near the upper boundary of the scattered points, reflecting that it goes through the estimated 90th percentile at each level of the [predictor variable](#) (hours studied). This confirms that we are modeling the high performance boundary, not the average performance.

Contrasting Quantile and Simple Linear Regression

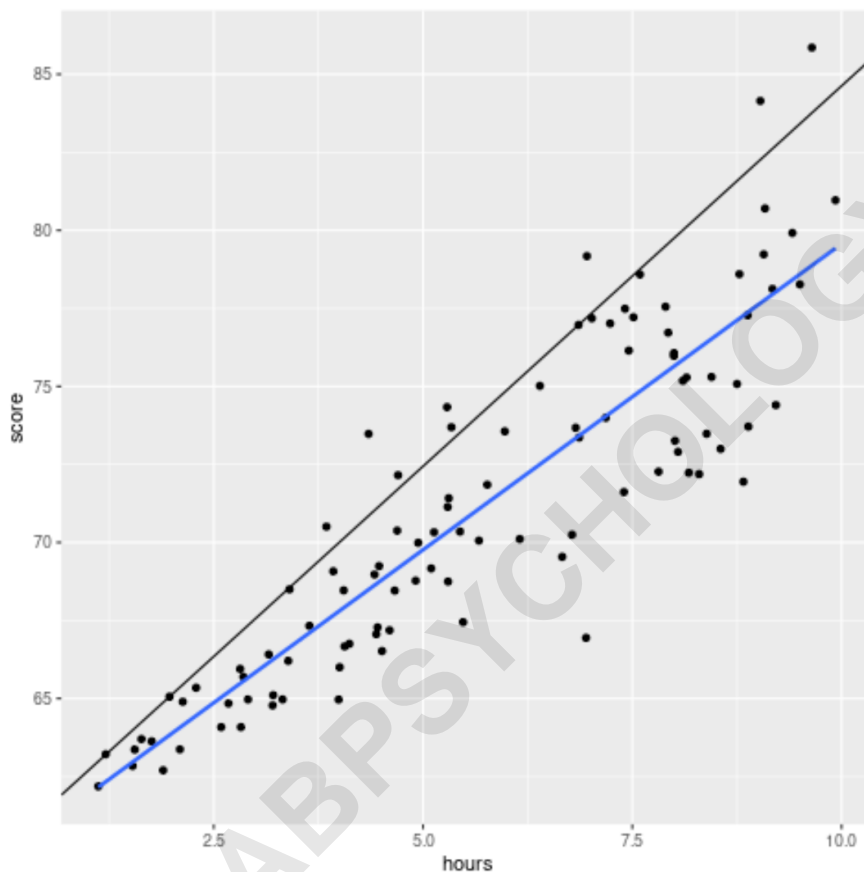
To fully appreciate the utility of [Quantile Regression](#), it is highly informative to compare its results directly against those derived from standard [Linear Regression](#) (OLS). We can achieve this visualization by adding the traditional OLS mean regression line to our existing scatterplot using the `geom_smooth()` argument with `method="lm"`. This comparison highlights the difference between modeling the mean versus modeling a specific quantile in a heteroscedastic dataset.

The traditional linear model estimates the expected mean score, whereas our quantile model estimates the expected 90th percentile score. Observing both lines simultaneously allows us to see the dispersion and the non-uniform effect of the predictor variable across the score distribution.

library(ggplot2)

```
#create scatterplot with quantile regression line and simple linear regression line
ggplot(df, aes(hours,score)) +
  geom_point() +
  geom_abline(intercept=coef(model), slope=coef(model)) +
  geom_smooth(method="lm", se=F)
```

The resulting plot provides a clear visual distinction between the two modeling approaches:



In this figure, the black line represents the fitted Quantile Regression line for the 90th percentile, while the blue line displays the Simple Linear Regression line, which estimates the conditional mean value for the response variable. As expected in this simulated scenario, the blue OLS line goes straight through the center of the data, showing the average estimated value of exam scores at each level of study hours. The black QR line, however, remains fixed at the upper boundary, demonstrating the effectiveness of quantile regression in isolating and modeling specific segments of the conditional distribution, especially when the variance changes across the data range.

Conclusion: When to Choose Quantile Regression

The ability to model conditional quantiles rather than just the conditional mean makes Quantile

Regression is an indispensable tool in the statistical toolbox, particularly when studying heterogeneous populations or focusing on extremes. If the relationship between predictors and the response is not uniform across all levels of the response (i.e., heteroscedasticity is present), or if you are concerned about the influence of outliers, quantile regression provides robust and efficient estimates.

In summary, you should choose quantile regression over traditional Linear Regression whenever:

The error terms are known or suspected to be **non-normal**.

The data exhibits **heteroscedasticity** (non-constant variance).

The analysis requires estimating the impact of predictors on specific parts of the distribution (e.g., the bottom 10% or the top 5%).

The presence of **outliers** makes mean estimation unreliable.

By utilizing the powerful **rq()** function from the **quantreg** package in R, analysts can move beyond simple mean estimation to gain a deeper, percentile-specific understanding of complex data relationships.