Perform Quantile Regression in R

Authored by

Mohammed looti

November 6, 2025

Moving Beyond the Mean: Why Quantile Regression Matters

Traditional linear regression, particularly the method of Ordinary Least Squares (OLS), serves as a cornerstone in statistical analysis, helping us model the relationship between one or more predictor variables and a corresponding response variable. When utilizing OLS, our primary goal is to estimate the conditional **mean** value of the response variable across varying levels of the predictors. This approach is powerful, but it relies on key assumptions, such as constant error variance (homoscedasticity), and fundamentally aims to fit a single line through the central tendency of the data distribution.

However, real-world data frequently defies these neat assumptions. many statistical phenomena exhibit relationships that vary significantly across the data distribution. We often encounter scenarios where the variability of the response changes dramatically depending on the predictor--a condition known as heteroscedasticity. Furthermore, focusing solely on the mean may obscure critical insights related to the tails or extremes of the data, such as modeling poverty levels (low extreme) or peak performance (high extreme). In these complex situations, the conditional mean fails to provide a complete or accurate representation of the underlying process.

To address these limitations, statisticians turn to a robust, non-parametric alternative: quantile regression (QR). Unlike OLS, QR allows us to estimate the conditional relationship for *any* specific quantile or percentile value of the response variable. This flexibility means we can model the 10th percentile, the 90th percentile, or even the median (which corresponds to the 50th percentile), providing a far richer understanding of how predictors influence the entire distribution, not just its center.

Mastering the R Syntax: The quantreg Package and the rq() Function

Implementing quantile regression within the statistical environment of R is streamlined through the specialized quantreg package. This essential package contains the core function, **rq()**, which is specifically engineered to handle the minimization problem required for accurate quantile estimation. Crucially, the **rq()** function delivers statistically robust results even when many conventional distributional assumptions are violated, making it an indispensable tool for advanced data analysis.

The basic structure of the **rq()** function intentionally mimics the syntax used by standard linear modeling functions in R, ensuring a low barrier to entry for users familiar with the language. Before executing the function, the quantreg library must be loaded. The function requires a formula argument (response ~ predictors), the data source, and most importantly, the target quantile.

The typical syntax for calling the function appears as follows:

```
library(quantreg)

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

Understanding the purpose of the primary arguments within **rq()** is vital for tailoring the model to specific research questions. The key parameters define which variables are analyzed and which section of the distribution is being targeted:

y: Defines the **response variable** (dependent variable) that the model seeks to predict.
x: Specifies the **predictor variable(s)** (independent variables) utilized to explain the variation in the response.
data: The name of the R data frame containing both the response and predictor variables.
tau: This is the defining parameter for quantile regression. It specifies the target **quantile** to be estimated and must be a value strictly between 0 and 1. By default, if omitted, **tau = 0.5** is used, which calculates the conditional median regression.

Practical Application (Step 1): Preparing the Heteroscedastic Sample Data

To illustrate the power and implementation of the **rq()** function, we will walk through a detailed, step-by-step example. Our first step involves preparing a suitable dataset for analysis. We will simulate a common educational scenario where we examine the relationship between the number of hours studied and the resulting exam score for a group of 100 hypothetical university students.

Crucially, we simulate the data to include a characteristic often found in high-stakes environments: heteroscedasticity. In this context, students who study more hours are expected to achieve higher average scores, but the variability in their performance also increases (the standard deviation of the error term grows with hours studied). This inherent structure makes the dataset an excellent candidate for quantile regression, as the OLS mean estimate would fail to capture the nuances at the upper performance limits.

The following R code generates this sample data frame, ensuring the example is fully **reproducible** across different sessions by employing the **set.seed()** function. The simulated relationship establishes a baseline score of 60, plus 2 points for every hour studied, with a variable error term:

```
#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
```

Model Estimation (Step 2): Fitting the 90th Percentile Regression

With our heterogeneous dataset prepared, the next phase involves specifying and fitting the target quantile model. We designate **hours** studied as the predictor variable and **exam score** as the response. Our research focus here is on peak performance--specifically, the boundary that separates the top 10% of students from the rest.

To capture this extreme relationship, we set the critical quantile parameter to **tau = 0.9**. This configuration instructs the **rq()** function to estimate the conditional 90th percentile of exam scores as a function of the corresponding hours studied. The resulting regression line will effectively represent the expected score threshold that only 10% of students studying that specific duration would typically surpass.

We proceed by ensuring the quantreg package is loaded and then execute the model fitting using **rq()**, immediately followed by the standard **summary()** command to inspect the estimated coefficients and their confidence intervals:

```
library(quantreg)

#fit model for the 90th percentile
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

Interpretation and Insight: Deciphering the 90th Percentile Coefficients

The output provided by the summary table gives us the estimated coefficients for the specified 90th percentile model, along with their respective lower and upper confidence bounds. These coefficients allow us to construct the specific regression equation that describes the high-performance boundary:

90th percentile of exam score = 60.25 + 2.437 * (hours studied)

To grasp the practical significance of these results, let us consider a student who dedicates eight hours to studying. We can predict the conditional 90th percentile score for that student by plugging the value into our equation: 90th percentile of exam score = 60.25 + 2.437 * (8) = **79.75**.

This estimate has a very specific meaning: based on our model, we anticipate that 90% of students who study for eight hours will achieve a score of 79.75 or lower, while only the top 10% of those students will score higher than 79.75. This type of prediction is exceptionally valuable for policy decisions or interventions focused on extreme outcomes, where relying solely on the conditional mean would lead to potentially misleading evaluations. The columns labeled **lower bd** and **upper bd** define the confidence interval for the coefficients, which are calculated using robust methods inherent to quantile regression, ensuring reliability without requiring restrictive distributional assumptions.

Visualizing the Results (Step 3): Understanding the Conditional Quantile Line

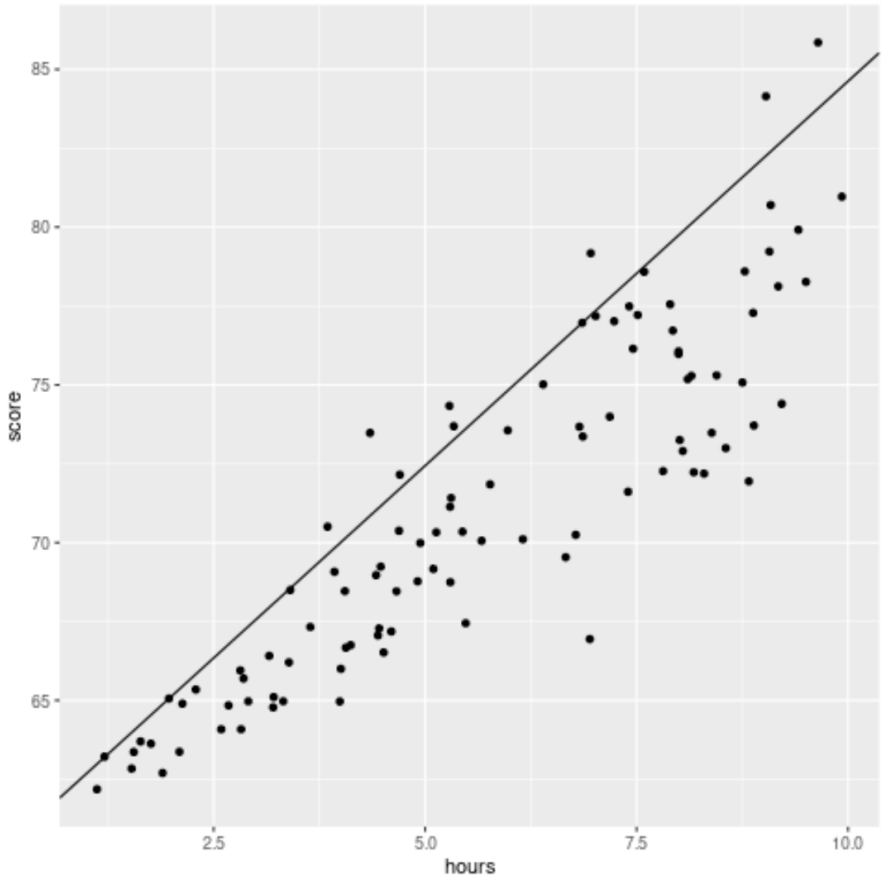
A crucial step in any regression analysis, especially when modeling non-central tendencies, is visualizing the fitted line against the raw data points. This allows for an intuitive understanding of how the model captures the relationship. Using the powerful **ggplot2** package in R, we can generate a scatterplot and overlay the regression line derived from our fitted 90th percentile model.

We employ the **geom_abline()** function, feeding it the intercept (coefficient 1) and the slope (coefficient 2) directly obtained from our stored **rq()** model object. This provides a precise visualization of the estimated boundary:

library(ggplot2)

#create scatterplot with quantile regression line

```
ggplot(df, aes(hours,score)) +
geom_point() +
geom_abline(intercept=coef(model), slope=coef(model))
```



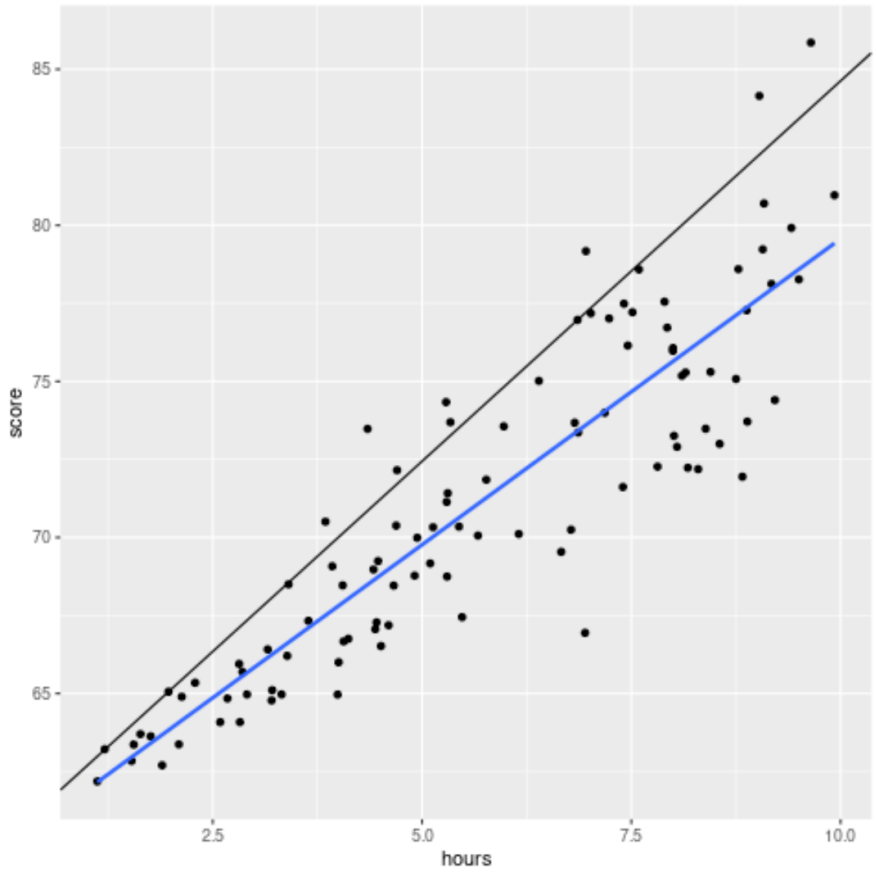
Inspection of the resulting plot clearly shows that this fitted line, unlike a typical OLS fit, does not pass through the "middle" or mean of the data cloud. Instead, the black line accurately traces the estimated 90th percentile boundary across all levels of the predictor variable (hours studied). Visually, we confirm that approximately 90% of the individual data points fall below this calculated boundary, demonstrating the model's success in isolating the performance of the high-achieving student cohort.

OLS vs. QR: A Crucial Comparison

To fully appreciate the flexibility and utility of quantile regression, it is highly instructive to compare its output directly against that of standard linear regression (OLS), which estimates only the conditional mean. We can easily enhance our previous visualization by incorporating the OLS fit using **ggplot2**'s **geom_smooth()** function with the argument **method="lm"**, allowing for a side-by-side comparison:

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)



In this composite visualization, the **black line** represents the fitted quantile regression line (tau = 0.9), while the **blue line** illustrates the simple linear regression line, which estimates the conditional mean score for each study level. As anticipated, the OLS line (blue) passes centrally through the data's core, providing the average estimated value. In stark contrast, the quantile line (black) is positioned substantially higher and exhibits a slightly steeper slope, accurately reflecting the superior performance trend of the top 10% of students. This visual evidence unequivocally highlights the superior robustness and analytical flexibility of quantile methods when the research focus shifts from the average outcome to specific, high-interest locations within the data distribution.

Additional Resources for R Statistics

To further enhance your understanding of advanced statistical modeling and data analysis using R, we encourage you to explore the following relevant tutorials and documentation: