Working Paper, MSc in Statistics for Data Science, Carlos III University of Madrid
DOI: N/A (internal report)

Advance Regression Term Project

Gerard Palomo & Juan Pablo Uphoff

Abstract
Linear quantile regression extends ordinary least squares (OLS) by modeling conditional quantiles of a response variable as linear functions of predictors. This offers a more complete view of the conditional distribution, revealing heterogeneous effects not captured by mean regression. Unlike OLS, quantile regression makes no strict distributional assumptions and is robust to outliers. We review the formulation, estimation, and inference for linear quantile regression, contrasting it with OLS. A simulation study compares OLS and quantile regression at various quantile levels (0.1 to 0.9) in both univariate and multivariate settings. We examine performance under normal and heavy-tailed error distributions, including scenarios with outliers, and assess the impact of sample size. The results illustrate that quantile regression estimates remain reliable under outlier contamination and uncover distributional effects (e.g. heteroscedasticity) that OLS misses. All simulation code is provided in R for reproducibility.

Keywords
OLS, Quantile Regression, Machine Learning

JEL
C45, C55

Acknowledgements
This report was completed for the course **Advanced Regression and Prediction**, as part of the **MSc in Statistics for Data Science** at **University Carlos III of Madrid**.

***Corresponding author**: , https://www.uc3m.es

Contents

1. Introduction.

Regression analysis typically focuses on modeling the conditional mean of a response given predictors (as in ordinary least squares, OLS). In contrast, quantile regression (QR) aims to model conditional quantiles (e.g., median, quartiles) of the response variable. Koenker and Bassett (1978) first introduced linear quantile regression, generalizing the regression model beyond the mean to an ensemble of conditional quantile functions.

Quantile regression provides several advantages over OLS. First, it allows us to explore the entire conditional distribution of Y given X, not just the mean, thereby revealing heterogeneous effects of predictors at different outcome levels . For example, an education variable might have a larger impact at the lower tail (10th percentile) of income than at the median or upper tail.

Second, quantile regression is robust to outliers in the response data. OLS estimates can be unduly influenced by a few extreme observations because it minimizes squared errors; in contrast, quantile regression (especially median regression) uses absolute-error-based loss, making it less sensitive to extreme Y values.

Third, quantile methods remain valid under skewed or heteroskedastic distributions. OLS relies on homoscedasticity and normality assumptions for optimality and inference, whereas quantile regression imposes no such requirements and can capture distributional changes (e.g., increasing variance) across predictor levels.

There are, of course, other approaches to address OLS limitations. Notably, researchers have developed robust regression methods and penalization techniques. For instance, Fan and Li (2001) proposed nonconcave penalized likelihood methods for variable selection, which can be applied to robust regression models.

In this article, we provide an overview of linear quantile regression theory (Section 2) and present a simulation study (Section 3) comparing quantile regression to OLS under various scenarios. We consider both univariate and multivariate predictor cases, examine performance at multiple quantile levels, and evaluate robustness to outliers and sample size changes. A concluding section summarizes the findings. Throughout, we cite key references, including Koenker and Hallock (2001) for an accessible overview, and Fan and Li (2001) for context on related regression advances. The complete R code for our analysis is provided in the Appendix to ensure reproducibility.

2. 2. Linear Quantile Regression: Theory and Methods

2.1 Definition and Estimation

For a random variable Y, the τ-th quantile is the value q_τ such that $P(Y \leq q_\tau) = \tau$. In a regression setting, **quantile regression (QR)** estimates:

$$Q_Y(\tau \mid X = x) = x^\top \beta(\tau),$$

where $\beta(\tau)$ is a vector of coefficients specific to quantile level τ. For example, $\beta_1(0.5)$ represents the effect of X_1 on the median of Y.

Koenker and Bassett (1978) proposed estimating $\beta(\tau)$ by minimizing the **check loss**:

$$\hat{\beta}(\tau) = \arg\min_{\beta} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta),$$

with $\rho_\tau(u) = u(\tau - \mathbb{I}\{u < 0\})$. For $\tau = 0.5$, this reduces to **least absolute deviations (LAD)** regression.

Each τ is estimated independently using linear programming, and the family $\{\beta(\tau)\}$ forms a **quantile process** describing the full conditional distribution of Y.

2.2 Inference

Under regularity conditions, $\hat{\beta}(\tau)$ is asymptotically normal:

$$\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau)) \xrightarrow{d} N(0, \Sigma(\tau)),$$

with variance estimators obtained via **bootstrapping** or **sandwich estimators** (Koenker, 2005). Practical inference is available via `summary()` in the R package `quantreg`.

2.3 Comparison to OLS

OLS estimates the conditional mean:

$$\mathbb{E}[Y \mid X = x] = x^\top \beta,$$

whereas QR estimates conditional quantiles. When errors are symmetric and homoscedastic, QR and OLS give similar results. Otherwise, QR captures **distributional heterogeneity**, such as increasing variance or skewness.

Moreover, QR is **robust to outliers in Y**, unlike OLS which minimizes squared error and is sensitive to extreme values. QR also allows different slopes across quantiles, offering richer interpretation.

In the next section, we illustrate these theoretical advantages through a simulation study.

3. 3. Simulation Study

This section presents a Monte Carlo simulation study designed to compare the performance of **Ordinary Least Squares (OLS)** and **Quantile Regression (QR)** estimators under controlled, interpretable scenarios. Our aim is to assess how both methods behave across different quantiles of the conditional distribution, particularly under heteroskedasticity, outlier contamination, and varying sample sizes.

3.1 3.1 Data Generating Process (DGP)

We consider the linear model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon,$$

with fixed parameters $\beta_0 = 5$, $\beta_1 = 1.5$, and $\beta_2 = -1$. In the univariate setting, only X_1 is included; in the multivariate setting, both X_1 and X_2 are active predictors.

- **Covariates**: $X_1, X_2 \overset{iid}{\sim} \text{Uniform}(0, 10)$.
- **Error term ε** follows different structures to emulate classical and adversarial settings (see below).
- **Quantiles studied**: $\tau = 0.1$, 0.5, 0.9 to represent lower, central, and upper tails of the conditional distribution.

We focus on: - **Small to large sample regimes**: $n = 50$, 100, 200, and 1000. - **Replications**: Each configuration is simulated over 100 independent datasets to stabilize the evaluation of finite-sample properties.

The sample sizes were chosen to explore both low-n variability and convergence behavior, with $n = 100$ serving as a baseline and $n = 1000$ approximating asymptotic regimes.

We estimate parameters using: - `lm()` for OLS (minimizing squared loss) - `rq()` from the `quantreg` package for QR at various τ values (minimizing asymmetric absolute loss; see Section 2).

3.2 3.2 Simulation Setup

We evaluate the estimators under three error structures:

- **Homoscedastic Gaussian**: $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 2$. This setting satisfies OLS assumptions and serves as a benchmark.
- **Contaminated Gaussian (outliers)**: $\varepsilon_i = Z_i + O_i$ with $Z_i \sim \mathcal{N}(0, 2^2)$ and $O_i = 30$ with probability $p = 0.05$. This mimics a heavy-tailed distribution or a mixture contamination model, testing robustness. Outliers are introduced:
  - randomly (to simulate noise in Y),
  - or conditionally (e.g. to the largest X_1) to induce high-leverage effects.
- **Heteroskedastic**: $\varepsilon_i \sim \mathcal{N}(0, \sigma^2(X_i))$ with $\sigma(X_i) = 1 + 0.2X_1$. This setting creates increasing spread in Y conditional on X_1, violating OLS homoscedasticity.

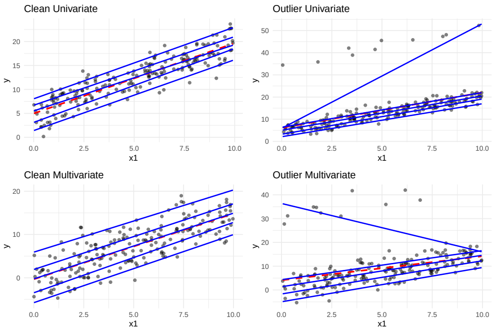


Figure 1. Comparison of slope estimates across settings

3.3 3.3 Evaluation Metrics

For each simulation, we extract the slope estimates (primarily for β_1) and compute:

- **Bias**:

$$\text{Bias}(\hat{\beta}) = \mathbb{E}[\hat{\beta}] - \beta$$

- **RMSE**:

$$\text{RMSE}(\hat{\beta}) = \sqrt{\mathbb{E}[(\hat{\beta} - \beta)^2]}$$

- **Empirical variability**: Standard deviation of estimates across replications.

Along with some graphical support, these evaluations provide insight into estimator consistency, robustness, and efficiency under both ideal and adversarial conditions.

The theoretical basis for QR (Section 2) suggests that: - OLS is optimal under homoscedastic Gaussian errors (Gauss-Markov assumptions), - QR provides robust, distribution-sensitive estimation that remains valid when these assumptions are violated.

This simulation seeks to verify those expectations empirically.

3.4 3.4 Simulation

```
library(gridExtra)
plot_one <- function(data, model, title) {
    ggplot(data, aes(x = x1, y = y)) + geom_point(alpha = 0.5) +
        geom_smooth(method = "lm", se = FALSE, color = "red",
            linetype = "dashed") + geom_quantile(quantiles = tau_le
        color = "blue", size = 0.8) + ggtitle(title) +
        theme_minimal()
}
grid.arrange(plot_one(scenarios$clean_uni, models$clean_uni,
    "Clean Univariate"), plot_one(scenarios$outlier_uni,
    models$outlier_uni, "Outlier Univariate"), plot_one(scenarios$c
    models$clean_multi, "Clean Multivariate"), plot_one(scenarios$c
    models$outlier_multi, "Outlier Multivariate"),
    ncol = 2)
```

```
summarize_slopes <- function(model) {
    slopes <- sapply(model$qrs, function(m) coef(m)["x1"])
    tibble(Method = c(paste0("QR (tau=", tau_levels,
        ")"), "OLS"), Estimate = c(slopes, coef(model$ols)["x1"]))
}
summarize_slopes(models$clean_uni)
```

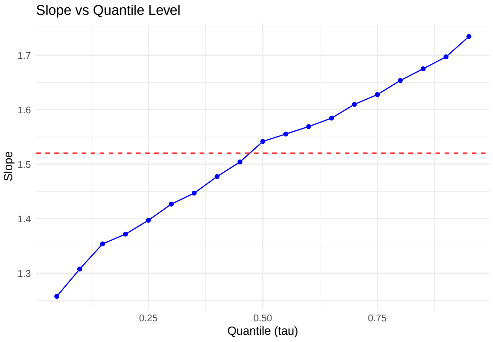


Figure 2. Slope vs Quantile for heteroskedastic case

4. A tibble: 6 x 2

Method Estimate 1 QR (tau=0.05) 1.48 2 QR (tau=0.25) 1.54 3 QR (tau=0.5) 1.38 4 QR (tau=0.75) 1.43 5 QR (tau=0.95) 1.50 6 OLS 1.47

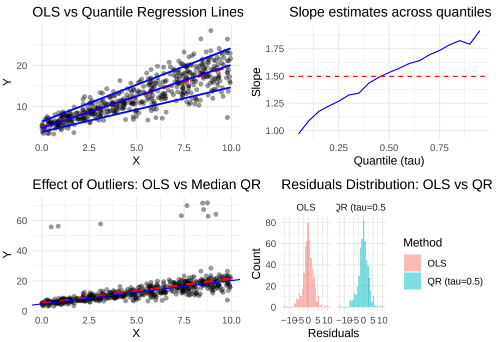
```
set.seed(666)
data_heterosk <- simulate_data(n = 1000, heterosk = TRUE)
fits_heterosk <- lapply(seq(0.05, 0.95, 0.05), function(tau) rq(y ~
    x1, tau = tau, data = data_heterosk))
slopes_heterosk <- sapply(fits_heterosk, function(fit) coef(fit)[2])
ols_slope <- coef(lm(y ~ x1, data = data_heterosk))[2]
plot_df <- tibble(tau = seq(0.05, 0.95, 0.05), slope = slopes_heterosk)
ggplot(plot_df, aes(x = tau, y = slope)) + geom_line(color = "blue") +
    geom_point(color = "blue") + geom_hline(yintercept = ols_slope,
    color = "red", linetype = "dashed") + labs(x = "Quantile (tau)",
    y = "Slope", title = "Slope vs Quantile Level") +
    theme_minimal()
```

```
fit25 <- rq(y ~ x1, tau = 0.25, data = scenarios$clean_uni)
fit50 <- rq(y ~ x1, tau = 0.5, data = scenarios$clean_uni)
fit75 <- rq(y ~ x1, tau = 0.75, data = scenarios$clean_uni)
anova(fit25, fit50, fit75)
```

```
## Quantile Regression Analysis of Deviance Table
##
## Model: y ~ x1
## Joint Test of Equality of Slopes: tau in {  0.25 0.5
##
##   Df Resid Df F value  Pr(>F)
## 1  2      598  2.9672 0.05221 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# kbl(anova_table, caption = 'Joint Test of
# Equality of Slopes at Different Quantiles') %>%
# kable_styling(latex_options = 'striped')
```

```
library(quantreg)
library(gridExtra)
library(dplyr)
# Simulate basic heteroskedastic data
set.seed(666)
n <- 500
X <- runif(n, 0, 10)
epsilon <- rnorm(n, mean = 0, sd = 1 + 0.3 * X)
Y <- 5 + 1.5 * X + epsilon
data <- data.frame(X = X, Y = Y)
# Fit models
ols_fit <- lm(Y ~ X, data = data)
rq_10 <- rq(Y ~ X, tau = 0.1, data = data)
rq_50 <- rq(Y ~ X, tau = 0.5, data = data)
rq_90 <- rq(Y ~ X, tau = 0.9, data = data)
# Plot 1: Scatterplot + regression lines
p1 <- ggplot(data, aes(x = X, y = Y)) + geom_point(alpha = 0.4) +
    geom_smooth(method = "lm", se = FALSE, color = "red",
        linetype = "dashed") + geom_quantile(quantiles = c(0.1,
    0.5, 0.9), color = "blue", size = 0.8) + labs(title = "OLS vs Quantile Regression Lines",
    y = "Y", x = "X") + theme_minimal()
# Plot 2: Slope across quantiles
taus <- seq(0.05, 0.95, by = 0.05)
slopes <- sapply(taus, function(tau) coef(rq(Y ~ X,
    tau = tau))[2])
p2 <- ggplot(data.frame(tau = taus, slope = slopes),
    aes(x = tau, y = slope)) + geom_line(color = "blue") +
    geom_hline(yintercept = coef(ols_fit)[2], linetype = "dashed",
        color = "red") + labs(title = "Slope estimates across quantiles",
    y = "Slope", x = "Quantile (tau)") + theme_minimal()
# Plot 3: Robustness to outliers
data_outlier <- data
idx <- sample(1:n, 10)
data_outlier$Y[idx] <- data_outlier$Y[idx] + 50  # Add strong outliers
ols_fit_outlier <- lm(Y ~ X, data = data_outlier)
rq_50_outlier <- rq(Y ~ X, tau = 0.5, data = data_outlier)
p3 <- ggplot(data_outlier, aes(x = X, y = Y)) + geom_point(alpha = 0.4) +
    geom_smooth(method = "lm", se = FALSE, color = "red",
        linetype = "dashed") + geom_abline(intercept = coef(rq_50_outlier)[1],
    slope = coef(rq_50_outlier)[2], color = "blue") +
    labs(title = "Effect of Outliers: OLS vs Median QR",
        y = "Y", x = "X") + theme_minimal()
# Plot 4: Residuals histograms
residuals_ols <- resid(ols_fit)
residuals_rq50 <- resid(rq_50)
residuals_df <- data.frame(residuals = c(residuals_ols,
    residuals_rq50), Method = rep(c("OLS", "QR (tau=0.5)"),
    each = n))
p4 <- ggplot(residuals_df, aes(x = residuals, fill = Method)) +
    geom_histogram(alpha = 0.5, position = "identity",
        bins = 30) + facet_wrap(~Method) + labs(title = "Residuals Distribution: OLS vs QR",
    x = "Residuals", y = "Count") + theme_minimal()
# Arrange in grid (2x2 layout)
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



5. Discussion & Conclusions.

xxxxx

Extras and considerations: This report has been made with a template in R Markdown, following the guidelines from (**?**), and the most famous resource for the creation of reports in R Markdown, the bookdown package (**?**).

References