How to Perform Quadratic Regression in R: A Step-by-Step Guide

Authored by

stats writer

March 14, 2026

ARABPSYCHOLOGY.COM

RECOMMENDED CITATION

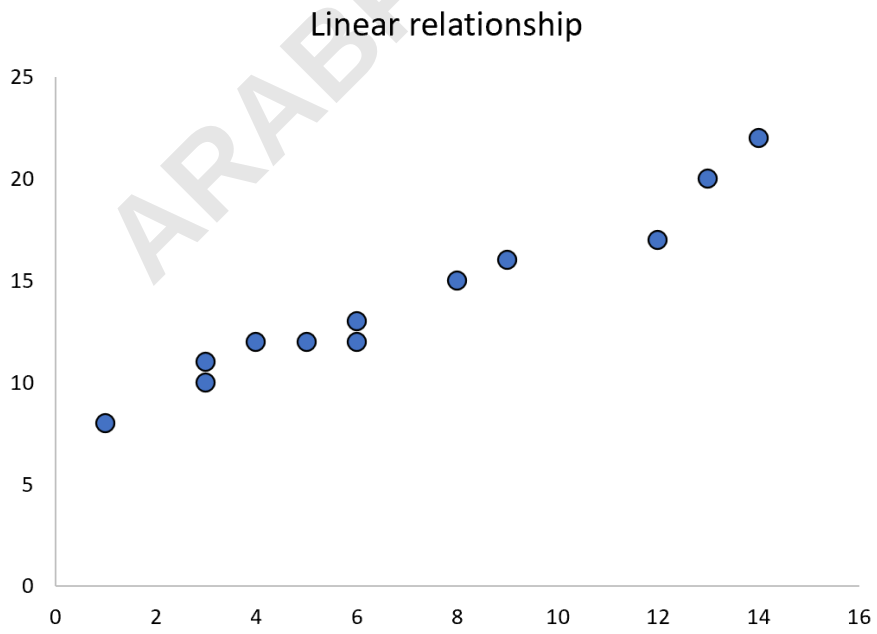
stats writer (2026). *How to Perform Quadratic Regression in R: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=135634

Quadratic regression is a sophisticated **statistical technique** employed to model the intricate relationship between a **dependent variable** and one or more **independent variables**, specifically when that relationship is hypothesized to be **curved** rather than linear. In the context of the **R programming language**, performing a quadratic regression involves utilizing the **lm function**, which is primarily designed for fitting **linear regression models**. To accommodate the curvature, the independent variables are transformed into their **quadratic form**, often through the **poly function** or by manually creating a **squared term**. This expansion of the model allows for the inclusion of **polynomial terms**, which significantly enhances the **goodness of fit** for datasets that display a **parabolic** trajectory. By analyzing the resulting **regression coefficients** and **p-values**, researchers can make precise **predictions** and derive nuanced conclusions about the underlying **data patterns**.

Perform Quadratic Regression in R

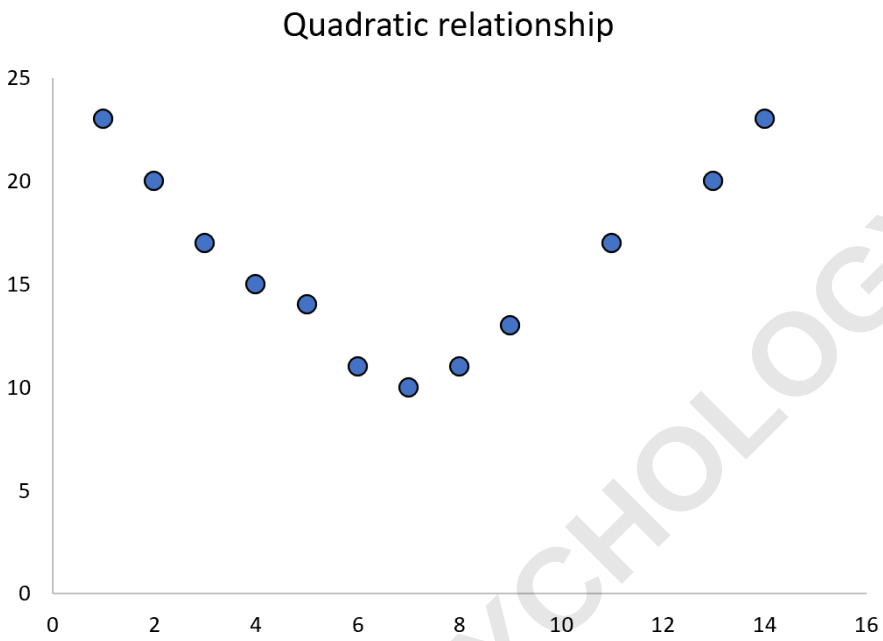
Theoretical Foundations of Quadratic Modeling

In the realm of **statistical analysis**, researchers often begin by assuming a **linear relationship** between variables. When two variables exhibit a direct, straight-line correlation, **simple linear regression** serves as an excellent tool to quantify and predict their interactions. This method seeks to find the "line of best fit" that minimizes the sum of the **squared residuals**, providing a clear mathematical description of how a change in one variable influences another. However, real-world data is rarely perfectly linear, and forcing a straight line through **curvilinear data** can lead to significant **modeling errors** and misleading conclusions.



ARABPSYCHOLOGY.COM

When the relationship between variables demonstrates a bend or a "U-shape," **quadratic regression** becomes the more appropriate analytical framework. This form of **polynomial regression** introduces a squared term into the **regression equation**, allowing the model to capture the **acceleration** or deceleration of the **dependent variable** as the **independent variable** increases. By accounting for this **non-linearity**, the model provides a much more accurate representation of the **data distribution**, particularly in fields like **economics**, **biology**, and **psychology** where diminishing returns or optimal peaks are common.



ARABPSYCHOLOGY.COM

This comprehensive tutorial is designed to guide you through the practical application of **quadratic regression** using the **R environment**. We will explore how to identify **non-linear patterns**, implement the necessary **data transformations**, and interpret the resulting **statistical output**. By the end of this guide, you will be equipped to handle **curved datasets** with confidence, ensuring your **predictive modeling** is both robust and scientifically sound.

Step 1: Data Preparation and Environmental Setup

The first step in any **data analysis** workflow is the accurate collection and **input of data** into the software environment. In this scenario, we are investigating the hypothetical relationship between the number of **hours worked** per week and the self-reported **happiness level** of individuals. Happiness is measured on a **continuous scale** from 0 to 100, providing a **quantitative metric** for our study. We have collected observations from 11 distinct individuals to serve as our **sample population** for this **regression analysis**.

To begin the process in **R**, we must organize this information into a **data frame**, which is the

standard **data structure** for handling tabular information. A **data frame** allows us to store different types of variables in a single object, making it easy to pass the data into **statistical functions**. Ensuring that your **data cleaning** and entry are precise is vital, as any errors at this stage will propagate through the **model fitting** and **hypothesis testing** phases.

Hours	Happiness
6	14
9	28
12	50
14	70
30	89
35	94
40	90
47	75
51	59
55	44
60	27

The following **R script** demonstrates how to construct this **data frame** manually. We define two **vectors**--one for the **predictor variable** (hours) and one for the **response variable** (happiness)--and combine them. Once the object is created, viewing the **dataset** helps verify that the observations match the intended **experimental design**.

```
#create data
data <- data.frame(hours=c(6, 9, 12, 14, 30, 35, 40, 47, 51, 55, 60),
happiness=c(14, 28, 50, 70, 89, 94, 90, 75, 59, 44, 27))

#view data
data

   hours happiness
1      6        14
2      9        28
3     12        50
4     14        70
5     30        89
6     35        94
7     40        90
8     47        75
9     51        59
10    55        44
```

ARABPSYCHOLOGY.COM

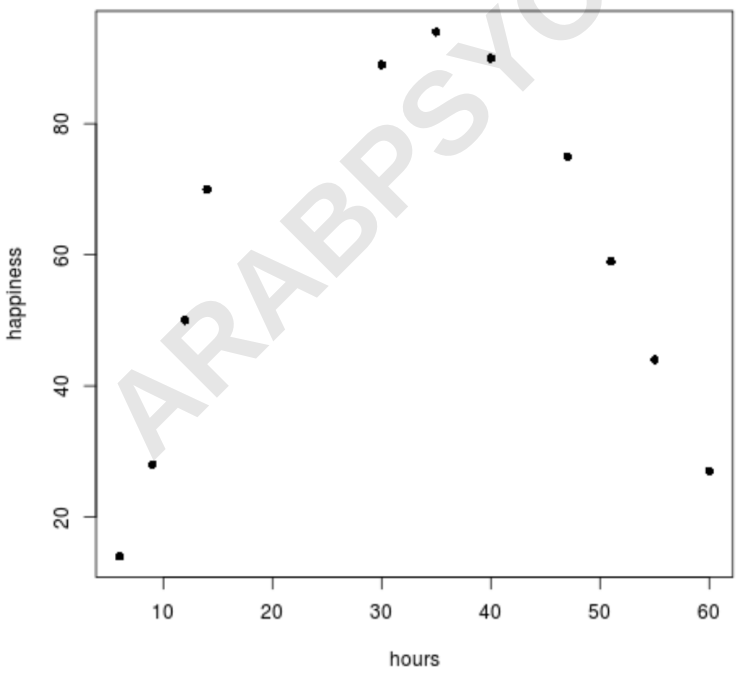
11 60 27

Step 2: Exploratory Data Analysis and Visualization

Before proceeding to **model construction**, it is essential to perform **Exploratory Data Analysis (EDA)**. The primary tool for this is the **scatterplot**, which provides a visual summary of the **association** between our variables. By plotting the **independent variable** on the **x-axis** and the **dependent variable** on the **y-axis**, we can immediately identify **trends**, **outliers**, and the overall **functional form** of the relationship. This visual check prevents the mistake of applying a **linear model** to data that clearly requires a **polynomial approach**.

In **R**, the base `plot()` function is a powerful way to generate these **visualizations**. By setting **graphical parameters** such as `pch=16` (to create solid circles), we can produce a clean, professional **data visualization**. If the points form a straight line, we proceed with **simple linear regression**; however, if the points rise and then fall (or vice versa), it is a definitive signal that a **quadratic term** is necessary to capture the **variance** in the data.

#create scatterplot
plot(data$hours, data$happiness, pch=16)



Upon inspecting the generated **scatterplot**, it becomes evident that the relationship is non-linear. The happiness levels increase as **work hours** move toward the middle of the range but begin to

ARABPSYCHOLOGY.COM

decline sharply as hours increase further. This "inverted U" shape is a classic example of a **quadratic trend**, suggesting that there is an **optimal value** for work hours, after which happiness experiences **diminishing returns**.

Step 3: Evaluating the Limitations of Linear Regression

To demonstrate why **quadratic regression** is superior for this specific dataset, we will first attempt to fit a **simple linear regression model**. This model assumes that for every unit increase in hours worked, happiness changes by a constant amount. We use the **lm function** in **R** to define the relationship and then utilize the **summary function** to extract **statistical metrics** such as the **intercept**, **slope**, and **coefficient of determination**.

The **summary output** provides a wealth of information, including **standard errors** and **t-statistics**. In a linear model, we look for a low **p-value** (typically < 0.05) to indicate **statistical significance**. More importantly, we examine the **R-squared value**, which represents the proportion of **variance** in the **dependent variable** that is explained by the **independent variable**. A low **R-squared** indicates that the model is failing to capture the underlying pattern of the data.

#fit linear model
linearModel <- lm(happiness ~ hours, data=data)

#view model summary
summary(linearModel)

Call:
lm(formula = happiness ~ hours)

Residuals:
Min 1Q Median 3Q Max
-39.34 -21.99 -2.03 23.50 35.11

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.4531 17.3288 2.796 0.0208 *
hours 0.2981 0.4599 0.648 0.5331

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.72 on 9 degrees of freedom
Multiple R-squared: 0.0446, Adjusted R-squared: -0.06156
F-statistic: 0.4201 on 1 and 9 DF, p-value: 0.5331

ARABPSYCHOLOGY.COM

The results of the **linear model** are revealing. The **Multiple R-squared** value is a mere **0.0446**, meaning that only **4.46%** of the **variability** in happiness is explained by a linear relationship with hours worked. This is an exceptionally poor fit. Furthermore, the **p-value** for the 'hours' variable is 0.5331, which fails to reach the threshold for **significance**. This confirms that a **linear approach** is fundamentally inadequate for this **non-linear dataset**.

Step 4: Implementing the Quadratic Regression Model

To address the **non-linearity** identified in the **EDA**, we must now transition to a **quadratic regression model**. The mathematical form of this model is **$y = \beta 0 + \beta 1x + \beta 2x^2$**, where **$x^2$** represents the **quadratic term**. In **R**, we can implement this by creating a new variable in our **data frame** that contains the **squared values** of the hours worked. This allows us to treat the **parabolic relationship** as a form of **multiple linear regression**.

By adding the **squared term**, we enable the **lm function** to calculate a **curve** that can bend to follow the data points. This process of **feature engineering**--transforming existing variables to improve **model performance**--is a cornerstone of **predictive analytics**. Once the new variable is created, we update the **formula** in our **regression model** to include both the original linear term and the new **quadratic term**.

```
#create a new variable for hours2
data$hours2 <- data$hours^2

#fit quadratic regression model
quadraticModel <- lm(happiness ~ hours + hours2, data=data)

#view model summary
summary(quadraticModel)
```

```
Call:
lm(formula = happiness ~ hours + hours2, data = data)

Residuals:
Min 1Q Median 3Q Max
-6.2484 -3.7429 -0.1812 1.1464 13.6678

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -18.25364 6.18507 -2.951 0.0184 *
hours 6.74436 0.48551 13.891 6.98e-07 ***
hours2 -0.10120 0.00746 -13.565 8.38e-07 ***
---
```

ARABPSYCHOLOGY.COM

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.218 on 8 degrees of freedom
Multiple R-squared: 0.9602, Adjusted R-squared: 0.9502
F-statistic: 96.49 on 2 and 8 DF, p-value: 2.51e-06

The improvement in **model fit** is dramatic. The **Multiple R-squared** has increased to **0.9602**, indicating that the **quadratic model** explains **96.02%** of the **variance** in happiness. Additionally, the **p-values** for both the linear and quadratic terms are extremely low ($p < 0.001$), signifying that both terms are **statistically significant** contributors to the model. This high **predictive power** demonstrates the effectiveness of **polynomial expansion** in **regression analysis**.

Step 5: Visualizing the Model Fit and Predictions

After calculating the **regression coefficients**, it is helpful to visualize how the resulting **curved line** fits against the original **data points**. To do this in **R**, we generate a sequence of **predictor values** across the full range of our data using the **seq function**. We then use the **predict function** to calculate the expected happiness levels for each of these values based on our **quadratic model**.

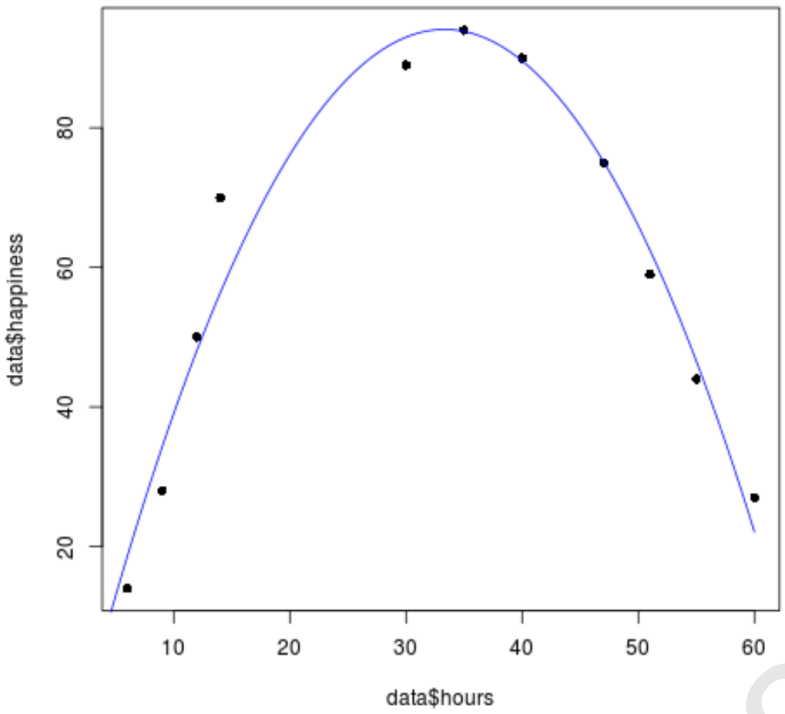
Plotting this **regression curve** allows us to visually verify the **accuracy** of our model. We use the `lines()` function to overlay the **predicted values** on the existing **scatterplot**. If the model is successful, the blue curve should pass through the center of the **data clusters**, capturing the peak happiness level and the subsequent decline. This **graphical representation** is essential for communicating **statistical results** to stakeholders who may not be familiar with **regression coefficients**.

#create sequence of hour values
hourValues <- seq(0, 60, 0.1)

#create list of predicted happines levels using quadratic model
happinessPredict <- predict(quadraticModel,list(hours=hourValues, hours2=hourValues^2))

#create scatterplot of original data values
plot(data$hours, data$happiness, pch=16)
#add predicted lines based on quadratic regression model
lines(hourValues, happinessPredict, col='blue')

ARABPSYCHOLOGY.COM



ARABPSYCHOLOGY.COM

The visual evidence confirms the **statistical output**: the **quadratic regression line** follows the **observed data** with remarkable precision. Unlike the **linear model**, which would have passed through the middle of the "arc" and missed every data point, the **curved model** accounts for the **dynamic relationship** between labor and well-being. This **model validation** step ensures that our **mathematical assumptions** align with the **empirical evidence**.

Step 6: Interpretation of Coefficients and Equation

The final stage of **quadratic regression** is the **interpretation** of the **model parameters**. The **summary output** provides us with three critical **estimates**: the **intercept** (β0), the **linear coefficient** (β1), and the **quadratic coefficient** (β2). Each of these numbers plays a specific role in defining the shape and position of the **parabola** on the coordinate plane. Understanding these values allows us to construct a **functional equation** for future **forecasting**.

In our specific **R output**, the **intercept** is approximately -18.25, the **linear term** for hours is 6.74, and the **quadratic term** for hours squared is -0.10. The negative sign on the **squared term** is particularly important, as it indicates that the **parabola** opens downward, creating a peak (or maximum) rather than a valley. This aligns with our **observation** that happiness increases up to a point before decreasing.

Coefficients:
Estimate Std. Error t value Pr(>|t|)

(Intercept) -18.25364 6.18507 -2.951 0.0184 *
hours 6.74436 0.48551 13.891 6.98e-07 *
hours2 -0.10120 0.00746 -13.565 8.38e-07 *

By combining these **coefficients**, we arrive at the following **fitted regression equation**: **Happiness = -0.1012(hours)² + 6.7444(hours) - 18.2536**. This **mathematical model** serves as a **predictive tool**. By plugging any number of work hours into the **variable** placeholders, we can calculate the **expected value** for happiness. This capability is invaluable for **scenario analysis** and **decision-making** in organizational behavior and **resource management**.

Step 7: Practical Application and Predictive Analysis

To see the **quadratic regression model** in action, we can apply it to specific **data points** within our range. For instance, if an individual works a **60-hour work week**, we can calculate their **predicted happiness**. By squaring the hours (3600), multiplying by the **quadratic coefficient**, adding the **linear component**, and adjusting for the **intercept**, we arrive at a specific **quantitative prediction**. This process demonstrates the **practical utility** of the model in **forecasting** individual outcomes.

For an individual working 60 hours, the calculation is: **Happiness = -0.1012(60)² + 6.7444(60) - 18.2536 = 22.09**. This low score reflects the **negative impact** of excessive work hours as captured by the **curved model**. Conversely, we can look at a more moderate work schedule, such as **30 hours per week**, to see how the **prediction** changes based on the **model's trajectory**.

For the 30-hour worker, the result is significantly different: **Happiness = -0.1012(30)² + 6.7444(30) - 18.2536 = 92.99**. This **predicted value** suggests that a 30-hour week is much closer to the **optimal balance** for happiness in this **sample population**. Through these **predictive exercises**, we see how **quadratic regression** provides a deeper, more accurate understanding of **complex variables** than **linear methods** ever could.

ARABPSYCHOLOGY.COM