# How can the optim function be utilized in R? Can you provide two examples?

Authored by

**stats writer**

June 29, 2024

The optim function in R is a powerful tool for optimizing parameters in a given function. This function uses various optimization algorithms to find the minimum or maximum value of a function, which is known as the "optimum" value. The optim function can be utilized in R by providing the function to be optimized, along with an initial set of parameters and other arguments. It then iteratively adjusts these parameters to find the optimal solution, based on the chosen optimization algorithm.

Two examples of how the optim function can be used in R are:

1. Parameter Estimation: Suppose we have a dataset and want to find the best-fitting parameters for a specific model. We can use the optim function to minimize the sum of squared errors between the actual data and the model's predictions. This will provide us with the optimal values for the model's parameters, which can then be used for further analysis.

2. Portfolio Optimization: In finance, it is essential to find the optimal allocation of assets in a portfolio to maximize returns while minimizing risk. The optim function can be employed to find the weights of each asset that will result in the highest expected return for a given level of risk. This can help investors make informed decisions about their portfolio composition.

## Use optim Function in R (2 Examples)

## You can use the optim function in R for general-purpose optimizations.

## This function uses the following basic syntax:

## optim(par, fn, data, ...)

## where:

## par: Initial values for the parameters to be optimized overfn: A function to be minimized or maximizeddata: The name of the object in R that contains the data

The following examples show how to use this function in the following scenarios:

1. Find coefficients for a linear regression model.

2. Find coefficients for a quadratic regression model.

Let's jump in!

### Example 1: Find Coefficients for Linear Regression Model

The following code shows how to use the optim() function to find the coefficients for a linear regression model by minimizing the :

```
#create data frame
df <- data.frame(x=c(1, 3, 3, 5, 6, 7, 9, 12),
y=c(4, 5, 8, 6, 9, 10, 13, 17))

#define function to minimize residual sum of squares
min_residuals <- function(data, par) {
with(data, sum((par + par * x - y)^2))
}

#find coefficients of linear regression model
optim(par=c(0, 1), fn=min_residuals, data=df)
```

**$par**

**2.318592 1.162012**

**$value**

**11.15084**

**$counts**

**function gradient**

**79 NA**

**$convergence**

**0**

**$message**

**NULL**

**Using the values returned under $par, we can write the following fitted linear regression model:**

**y = 2.318 + 1.162x**

**We can verify this is correct by using the built-in lm() function in R to calculate the regression coefficients:**

**#find coefficients of linear regression model using lm() function**

**lm(y ~ x, data=df)**

**Call:**
**lm(formula = y ~ x, data = df)**

**Coefficients:**
**(Intercept) x**
**2.318 1.162**

**These coefficient values match the ones we calculated using the optim() function.**

### Example 2: Find Coefficients for Quadratic Regression Model

**The following code shows how to use the optim() function to find the coefficients for a by minimizing the residual sum of squares:**

```
#create data frame
df <- data.frame(x=c(6, 9, 12, 14, 30, 35, 40, 47, 51, 55, 60),
y=c(14, 28, 50, 70, 89, 94, 90, 75, 59, 44, 27))#define function to minimize residual sum of squares
min_residuals <- function(data, par) {
with(data, sum((par + par*x + par*x^2 - y)^2))
```

```
}

#find coefficients of quadratic regression model
optim(par=c(0, 0, 0), fn=min_residuals, data=df)

$par
-18.261320 6.744531 -0.101201

$value
309.3412

$counts
function gradient
218 NA

$convergence
0

$message
NULL
```



**Using the values returned under $par, we can write the following fitted quadratic regression model:**

**y = -18.261 + 6.744x - 0.101x2**

**We can verify this is correct by using the built-in lm() function in R:**

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

#create a new variable for x^2
df$x2 <- df$x^2

#fit quadratic regression model
quadraticModel <- lm(y ~ x + x2, data=df)

#display coefficients of quadratic regression model
summary(quadraticModel)$coef
```

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) -18.2536400 6.185069026 -2.951243 1.839072e-02
x 6.7443581 0.485515334 13.891133 6.978849e-07
x2 -0.1011996 0.007460089 -13.565470 8.378822e-07
```

**These coefficient values match the ones we calculated using the optim() function.**

## Additional Resources