

How can I perform step-by-step logarithmic regression in R?

Authored by
stats writer

April 26, 2024

RECOMMENDED CITATION

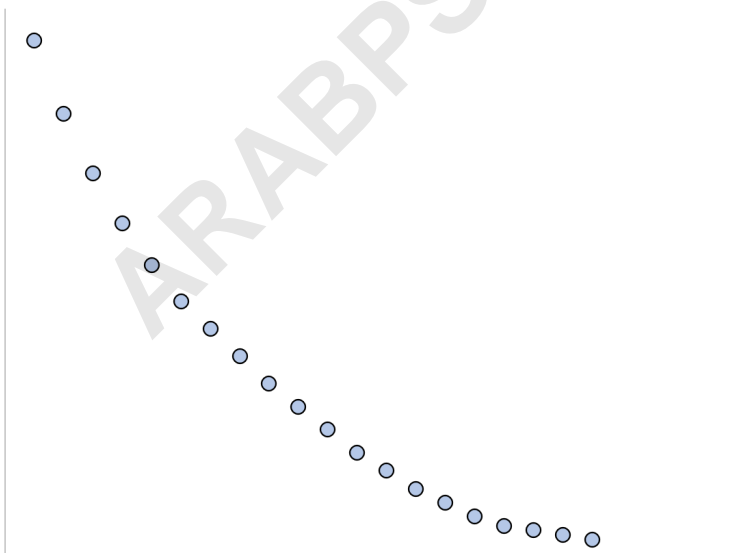
stats writer (2024). *How can I perform step-by-step logarithmic regression in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139436>

Step-by-step logarithmic regression in R refers to the process of fitting a logarithmic curve to a set of data points in order to model the relationship between two variables. This method involves taking the natural logarithm of the data and using it to create a linear model, which can then be used to make predictions and calculate the goodness of fit. The steps involved in this process include importing the data into R, taking the log of the data, plotting the log-transformed data, fitting a linear model, and evaluating the model's performance. By following these steps, users can effectively perform logarithmic regression in R to understand and analyze the logarithmic relationship between their data.

Logarithmic Regression in R (Step-by-Step)

Logarithmic regression is a type of regression used to model situations where growth or decay accelerates rapidly at first and then slows over time.

For example, the following plot demonstrates an example of logarithmic decay:



For this type of situation, the relationship between a

predictor variable and a response variable could be modeled well using logarithmic regression.

The equation of a logarithmic regression model takes the following form:

$$y = a + b \cdot \ln(x)$$

where:

y: The response variable
x: The predictor variable
a, b: The regression coefficients that describe the relationship between x and y

The following step-by-step example shows how to perform logarithmic regression in R.

Step 1: Create the Data

First, let's create some fake data for two variables: x and y :

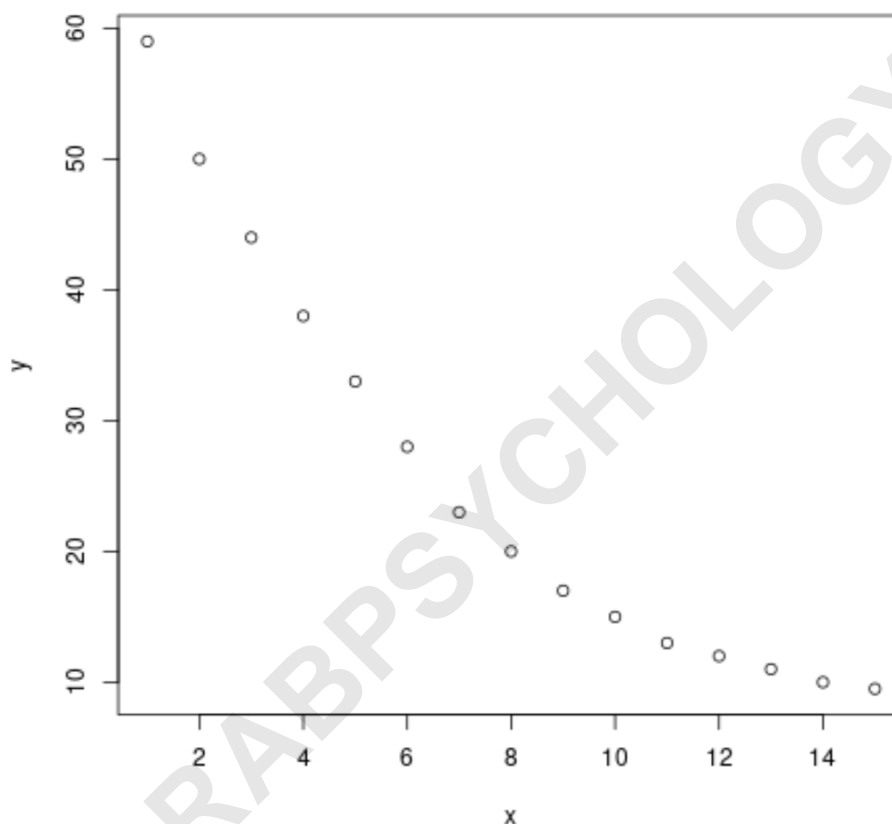
$x=1:15$

$y=c(59, 50, 44, 38, 33, 28, 23, 20, 17, 15, 13, 12, 11, 10, 9.5)$

Step 2: Visualize the Data

Next, let's create a quick to visualize the relationship between x and y :

```
plot(x, y)
```



From the plot we can see that there exists a clear logarithmic decay pattern between the two variables. The value of the response variable, y , decreases rapidly at first and then slows over time.

Step 3: Fit the Logarithmic Regression Model

Next, we'll use the `lm()` function to fit a logarithmic regression model, using the natural log of x as the predictor variable and y as the response variable

#fit the model

```
model <- lm(y ~ log(x))#view the output of the model
summary(model)
```

Call:

```
lm(formula = y ~ log(x))
```

Residuals:

```
Min 1Q Median 3Q Max
-4.069 -1.313 -0.260 1.127 3.122
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 63.0686 1.4090 44.76 1.25e-15 ***
log(x) -20.1987 0.7019 -28.78 3.70e-13 ***
```

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

Residual standard error: 2.054 on 13 degrees of freedom

**Multiple R-squared: 0.9845, Adjusted R-squared: 0.9834
F-statistic: 828.2 on 1 and 13 DF, p-value: 3.702e-13**

The of the model is 828.2 and the corresponding p-value is extremely small (3.702e-13), which indicates that the model as a whole is useful.

Using the coefficients from the output table, we can see that the fitted logarithmic regression equation is:

$$y = 63.0686 - 20.1987 * \ln(x)$$

We can use this equation to predict the response variable, y , based on the value of the predictor variable, x . For example, if $x = 12$, then we would predict that y would be 12.87:

$$y = 63.0686 - 20.1987 * \ln(12) = 12.87$$

Bonus: Feel free to use this online to automatically compute the logarithmic regression equation for a given predictor and response variable.

Step 4: Visualize the Logarithmic Regression Model

Lastly, we can create a quick plot to visualize how well

the logarithmic regression model fits the data:

#plot x vs. y

plot(x, y)

#define x-values to use for regression line

x=seq(from=1,to=15,length.out=1000)

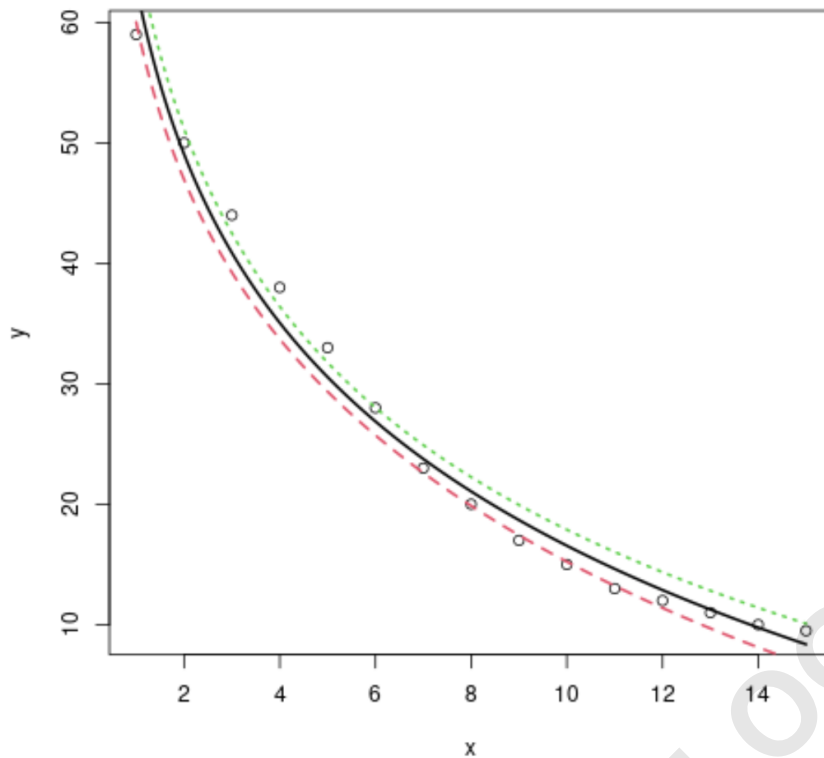
#use the model to predict the y-values based on the x-values

y=predict(model,newdata=list(x=seq(from=1,to=15,length.out=1000)),

interval="confidence")

#add the fitted regression line to the plot (lwd specifies the width of the line)

matlines(x,y, lwd=2)



We can see that the logarithmic regression model does a good job of fitting this particular dataset.