

# How to Perform Lowess Smoothing in R (Step-by-Step)

Authored by  
**stats writer**

December 7, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Perform Lowess Smoothing in R (Step-by-Step)*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106608>

Lowess smoothing, often referred to interchangeably with **LOESS** (Locally Estimated Scatterplot Smoothing), is a powerful and flexible technique in statistics utilized for fitting a smooth, non-parametric curve to a series of data points in a scatterplot. Unlike traditional methods which attempt to fit a single global model (like simple linear regression), Lowess excels when the relationship between variables is complex, varying locally, or explicitly non-linear regression. This method is crucial for data visualization, trend identification, and making predictions without imposing strict distributional assumptions on the underlying data.

The core mechanism of Lowess involves calculating weighted regression fits over localized subsets of the data. This allows the fitted curve to adapt to the local structure of the relationship, capturing nuances that a standard polynomial regression might miss. Implementing this technique is straightforward in the **R** programming environment, primarily through the built-in `lowess()` function. To effectively utilize this function, the user must provide the input vectors for the x and y coordinates, alongside a critical parameter known as the smoother span (often denoted as  $\hat{f}$ ). The result is a highly accurate, smoothed representation of the data trend, which can be easily overlaid onto the original data visualization.

In the realm of exploratory data analysis (EDA), **Lowess** stands for "locally weighted scatterplot smoothing." This methodology is fundamental to visualizing and summarizing the relationship between two variables when a simple model is inadequate. It helps statisticians and data scientists identify the underlying pattern, especially when the noise level is high or the relationship changes slope significantly across the range of the predictor variable. By providing an immediate visual representation of the central tendency, it becomes much easier to detect anomalies, curvature, and structural breaks in the data.

## Understanding Locally Weighted Scatterplot Smoothing (Lowess/LOESS)

The mathematical foundation of Lowess relies on the principle of local polynomial regression. Instead of calculating one regression line across the entire dataset, Lowess calculates many weighted least squares regression lines within small, overlapping windows of the data. For any point on the x-axis where we wish to calculate the smoothed value ( $\hat{y}$ ), the algorithm considers only the data points within a defined neighborhood around that x-value. The size of this neighborhood is directly controlled by the smoother span parameter,  $\hat{f}$ .

Within each neighborhood, the data points are assigned weights based on their distance from the focal point. Points closer to the focal point receive higher weights, meaning they influence the local regression fit more heavily, while points farther away receive lower weights. This localized, distance-based weighting scheme ensures that the resulting curve reflects the true local behavior of the data without being overly influenced by distant outliers or trends that exist outside that immediate region. This process is repeated for a sufficient number of points along the x-axis, and

these local fits are then connected to form the final smooth curve.

One of the primary benefits of Lowess is its robustness. Since it is a non-parametric method, it makes very few assumptions about the underlying distribution of the errors or the functional form of the relationship. This makes it an ideal tool for initial data exploration and diagnosing potential issues, such as heteroscedasticity or non-linearity, before committing to a more restrictive parametric model. Understanding this localized approach is key to appreciating why Lowess often provides a superior visual fit compared to simple regression lines, particularly when dealing with noisy empirical data.

## Why Use Lowess Smoothing? Applications and Benefits

Lowess smoothing serves several critical functions in statistical analysis and data visualization. Its primary advantage is providing a highly flexible estimate of the conditional mean of the response variable given the predictor variable. When performing exploratory analysis on a dataset where the relationship is suspected to be curved or highly variable--for instance, in economic time series, environmental data, or biological growth curves--Lowess quickly reveals the underlying structure.

A significant benefit of using Lowess is its ease of interpretation. The resulting curve is intuitive: it visually traces the average path of the data points. This makes it invaluable for presentations and communication, allowing non-technical audiences to immediately grasp the trend. Furthermore, Lowess is inherently less susceptible to distortion from extreme outliers compared to global polynomial fits because the influence of any single point is localized, preventing it from pulling the entire curve far from the majority of the data.

In the context of modeling, Lowess is often used as a preprocessing step. By applying the smoothing function, analysts can effectively separate the systematic trend (the signal) from the random noise (the residuals). The residuals derived from a Lowess fit can then be analyzed independently to check for remaining patterns or autocorrelation. This dual capability--visualization of the signal and isolation of the noise--makes it an indispensable tool in any robust data analysis toolkit within R or any other statistical environment.

## The R Function: `lowess()` Syntax Explained

To perform Lowess smoothing in R, we utilize the base function `lowess()`. This function is streamlined and requires only a few core arguments to execute the smoothing operation efficiently. Understanding the syntax and the role of each argument is essential for effective implementation and control over the resulting curve. The general structure of the function call is as follows, incorporating the necessary input vectors and the crucial smoothing parameter:

```
lowess(x, y, f = 2/3, iter = 3)
```

The parameters dictate how the local fitting process is executed. The key mandatory and optional arguments are defined below. Note that the default value for the smoother span,  $\mathbb{f}$ , is set conservatively to two-thirds (approximately 0.67), which often provides a reasonable starting point for many datasets. The `iter` argument controls the number of robustifying iterations used to minimize the influence of potential outliers.

**$\mathbb{x}$ :** A numerical vector containing the predictor variable values. This represents the horizontal axis data points.

**$\mathbb{y}$ :** A numerical vector containing the response variable values. This represents the vertical axis data points, corresponding element-wise to the  $\mathbb{x}$  vector.

**$\mathbb{f}$ :** The value for the smoother span. This is a proportion (a fraction between 0 and 1) that specifies the proportion of points in the dataset that influence the smooth at each local calculation. Larger values of  $\mathbb{f}$  include more points in the local regressions, leading to a smoother curve.

**`iter`:** The number of robustifying iterations. By default, `lowess()` performs three iterations to refine the curve, making the fit more resistant to outliers.

## Step 1: Preparing and Visualizing the Data in R

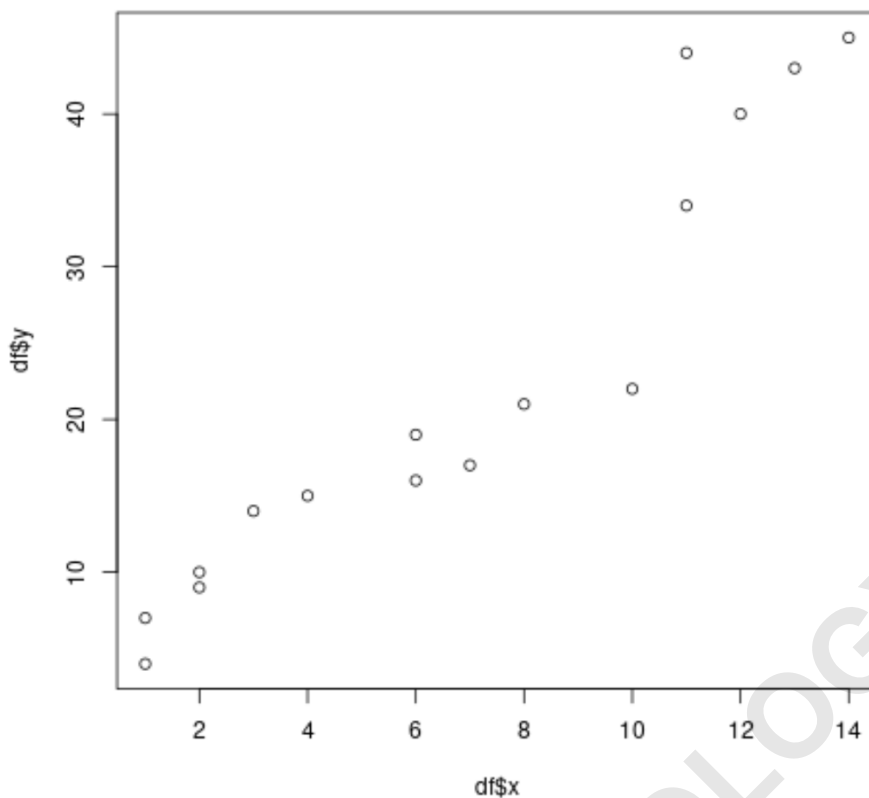
Before applying any smoothing technique, it is standard practice to create and visualize the raw data. This initial scatterplot provides the baseline visual representation against which the smoothed curve will be compared. For demonstration purposes, we will first create a small, artificial dataset in R that exhibits a clear non-linear trend, which is ideal for showcasing the power of Lowess.

The following code snippet generates a data frame, `df`, containing 16 observations. The relationship between  $\mathbb{x}$  and  $\mathbb{y}$  visibly increases but contains inherent noise and some notable curvature, especially towards the latter half of the data range. This complexity mandates a flexible fitting procedure like Lowess to accurately capture the true underlying pattern.

```
df <- data.frame(x=c(1, 1, 2, 2, 3, 4, 6, 6, 7, 8, 10, 11, 11, 12, 13, 14),  
y=c(4, 7, 9, 10, 14, 15, 19, 16, 17, 21, 22, 34, 44, 40, 43, 45))
```

Once the data frame is loaded into the R environment, the next essential step is plotting the data using the base plotting function. This visualization immediately confirms the scatter of the points and the apparent non-linearity, justifying the use of a localized smoothing technique over simple linear regression. We plot the raw  $\mathbb{x}$  and  $\mathbb{y}$  vectors directly from the newly created data frame `df`.

```
plot(df$x, df$y)
```



## Step 2: Implementing Default Lowess Smoothing

With the initial `scatterplot` established, we can proceed to apply the `lowess()` function. Since `lowess()` outputs a list containing the smoothed x and y coordinates, we typically use the `lines()` function in R to overlay this calculated curve directly onto the existing plot. By default, R's `lowess()` function uses a smoother span ( $\hat{h}$ ) of  $2/3$  (approximately 0.67).

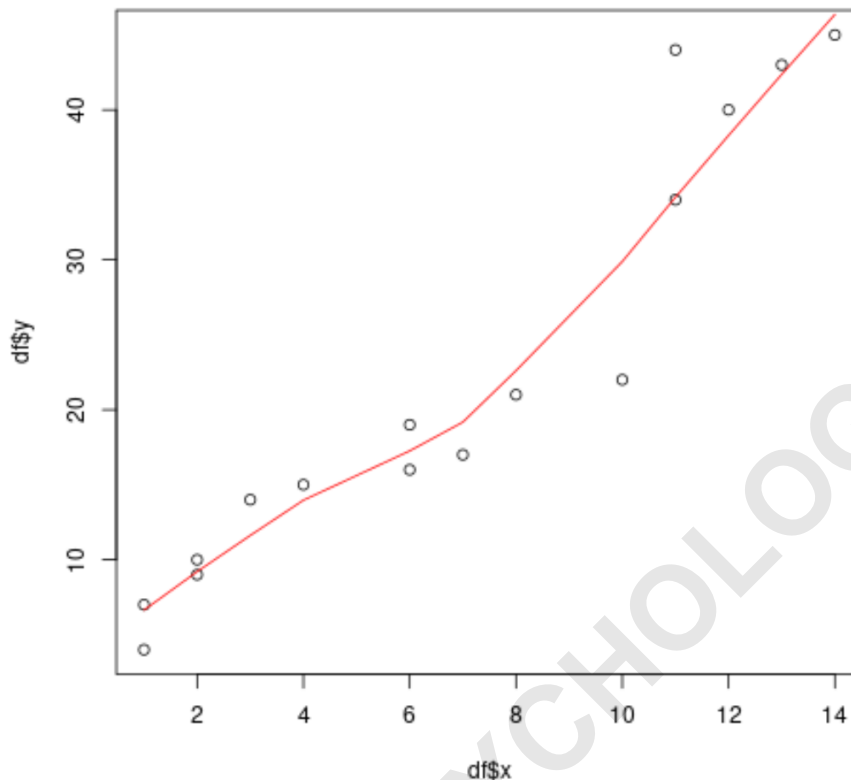
The following code block first recreates the scatterplot and then uses the `lines()` function, passing the output of `lowess(df$x, df$y)` as the input coordinates. We explicitly set the color of the line to 'red' to make the smoothed curve clearly distinguishable from the raw data points. This visualization provides the first insight into the underlying non-linear trend estimated by the Lowess algorithm.

```
#create scatterplot  
plot(df$x, df$y)
```

```
#add lowess smoothing curve to plot  
lines(lowess(df$x, df$y), col='red')
```

Observing the resulting graph, the default Lowess curve successfully tracks the increasing trend

while dampening the local fluctuations caused by individual data points. This curve provides a robust estimate of the central tendency, showing a clear acceleration in the relationship between  $x$  and  $y$  as  $x$  increases, particularly in the range from 10 to 14. This default application provides a solid benchmark for judging the effects of modifying the smoothing parameter.



### Step 3: Fine-Tuning the Smoother Span (The Critical Parameter $\epsilon$ )

The single most important parameter in Lowess smoothing is  $\epsilon$ , the smoother span. This value directly controls the trade-off between the smoothness of the resulting curve and its fidelity to the local data structure. A fundamental decision in applying Lowess is determining the optimal span that balances bias and variance. A smaller span results in a less biased fit (it follows the data points very closely, capturing rapid changes) but introduces higher variance (it is highly sensitive to noise and outliers). Conversely, a larger span reduces variance (creating a very smooth, stable curve) but introduces higher bias (it might miss subtle but important local features).

We can adjust the  $\epsilon$  argument within the `lowess()` function to deliberately increase or decrease the degree of smoothing applied to the data. It is crucial to iterate through different values of  $\epsilon$  to find the value that best reflects the underlying process without overfitting the noise. Generally, values closer to 1 (e.g., 0.8 or 0.9) yield very smooth results, while values closer to 0 (e.g., 0.1 or 0.2) produce highly wiggly curves that closely hug the data points.

In the following example, we illustrate the effect of varying the smoother span by plotting three Lowess curves simultaneously on the same scatterplot: the default ( $f=0.67$ , red), a very local fit ( $f=0.3$ , purple), and an extremely smooth fit ( $f=3$ , although the parameter is typically restricted between 0 and 1, inputting values greater than 1 results in the smoother using all available data, often clamped at 1 in practice for the calculation of the proportion, but R often handles this by effectively maximizing the smoothing, resulting in a nearly linear approximation, represented here by 'steelblue'). This visual comparison is key for diagnostic purposes, allowing the analyst to select the appropriate degree of smoothing based on the context of the data.

**#create scatterplot****plot(df\$x, df\$y)**

#add lowess smoothing curves

lines(lowess(df\$x, df\$y), col='red')

lines(lowess(df\$x, df\$y, f=0.3), col='purple')

lines(lowess(df\$x, df\$y, f=3), col='steelblue')

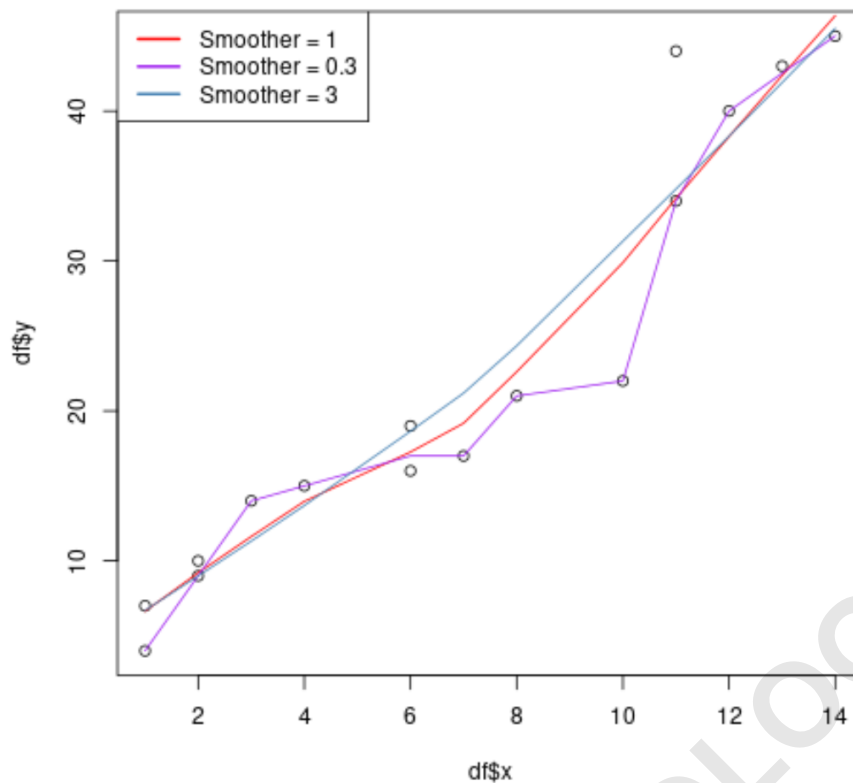
#add legend to plot

legend('topleft',

col = c('red', 'purple', 'steelblue'),

lwd = 2,

c('Smoother = 2/3 (Default)', 'Smoother = 0.3 (Local Fit)', 'Smoother = Max (Global Fit)'))



## Interpreting Lowess Output and Best Practices

Interpreting the output of a `Lowess` analysis requires careful consideration of the context and the chosen `smoother span`. If the span is too small (e.g.,  $\epsilon=0.3$  in our example, purple line), the curve exhibits high variance, demonstrating strong oscillations that follow the noise in the data, potentially leading to overfitting. This may be acceptable if the analyst believes the underlying process truly exhibits rapid fluctuations. However, for identifying a stable, general trend, this is usually too rough.

Conversely, when the span is very large (approaching 1 or using an oversized value like  $\epsilon=3$ , the steelblue line), the curve becomes extremely smooth, often resembling a simple linear regression line. While stable, this smooth curve might mask important local features or non-linearities, leading to high bias and underfitting the true complex relationship. The default value (red line,  $\epsilon=2/3$ ) typically offers a good balance, capturing the overall curvature without being overly sensitive to every minor deviation.

A key best practice when employing Lowess smoothing in `R` is to systematically vary the  $\epsilon$  parameter and analyze the resulting visualizations. This process, often subjective, should be guided by domain knowledge. The goal is to select the smallest span that still yields a curve that appears smooth and captures the essential systematic movement without incorporating noise.

Furthermore, always check the residuals (the differences between the raw y values and the smoothed y values) to ensure no remaining systematic patterns exist, which would indicate that the chosen smoother span might be too large. By following these steps, Lowess smoothing transforms noisy data into meaningful, visually compelling trends.

ARABPSYCHOLOGY.COM