

How to Easily Calculate SMAPE in R Using the Forecast Package

Authored by
stats writer

December 6, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Calculate SMAPE in R Using the Forecast Package*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106053>

The Symmetric Mean Absolute Percentage Error (SMAPE) is a robust metric used widely in time series analysis and demand planning. In the statistical programming language **R**, SMAPE can be computed efficiently using various dedicated functions. While packages like the renowned **forecast** package offer utilities like the `accuracy()` function (where you might specify the metric as "SMAPE"), a more direct and often preferred approach utilizes specialized packages designed purely for evaluating model performance. Regardless of the method, the function requires two critical inputs: vectors representing the **predicted (forecasted)** values and the **observed (actual)** values. The standard output is the calculated SMAPE value, which is usually expressed as a percentage, offering an intuitive measure of forecasting error. Understanding the syntax is key: `accuracy(forecast, observed, metric="SMAPE")` provides one route, but we will explore other highly effective methods in this comprehensive guide.

Introduction to SMAPE and Its Importance in Forecasting

The **symmetric mean absolute percentage error (SMAPE)** is an essential tool for assessing the quality and reliability of predictive models, particularly those involved in time series forecasting. Unlike the standard Mean Absolute Percentage Error (MAPE), which can suffer from issues when actual values are near zero, SMAPE provides a more balanced assessment of error by normalizing the absolute difference between the forecast and actual values based on their average. This symmetrical nature makes SMAPE a favored metric in commercial applications, such as retail and supply chain management, where underestimations and overestimations are weighted equally relative to the scale of the data. High-quality forecasting relies heavily on selecting models that minimize this error metric, thereby ensuring that business decisions based on these predictions are as accurate as possible.

The core challenge in predictive accuracy assessment is selecting a metric that is both mathematically sound and intuitively interpretable by stakeholders. SMAPE successfully bridges this gap. Since it is represented as a percentage, a SMAPE value of, say, 10% clearly indicates that the average forecast deviates from the actual observation by 10% of their combined average. This clarity allows data scientists to communicate model performance effectively to non-technical audiences. Furthermore, the minimization of SMAPE often serves as a primary objective function during the training and calibration of sophisticated forecasting algorithms, making its correct implementation in **R** a fundamental skill for any data practitioner.

Deconstructing the SMAPE Formula

To accurately calculate and interpret **SMAPE**, it is crucial to understand the mathematical formula that defines it. The symmetric mean absolute percentage error is derived by calculating the average of the ratios of the absolute error to the average of the actual and forecasted values, and

then expressing this result as a percentage. This calculation ensures that the resulting error is bounded and symmetrical, addressing common pitfalls of simpler percentage error metrics.

The standard mathematical expression for SMAPE is defined as follows:

The **symmetric mean absolute percentage error (SMAPE)** is used to measure the predictive accuracy of models. It is calculated as:

$$\text{SMAPE} = (1/n) * \sum(|\text{forecast} - \text{actual}| / ((|\text{actual}| + |\text{forecast}|)/2)) * 100$$

where the following components are defined:

Σ - This symbol represents the mathematical operation of summation, indicating that we must sum the results across all data points in the sample.

n - This variable denotes the total number of observations or the sample size within the dataset being evaluated.

actual - This refers to the true, observed value at a given point in time.

forecast - This refers to the predicted value generated by the forecasting model for the corresponding point in time.

A key takeaway from this formula is that the denominator includes both the actual and the forecast values. This averaging prevents the extreme percentage errors that occur in traditional MAPE when the actual value is very close to zero. Consequently, the smaller the numerical value calculated for SMAPE, the greater the predictive accuracy of the statistical model under evaluation. This metric is thus invaluable for comparing different forecasting methodologies objectively.

Prerequisites: Setting Up Your R Environment for Metrics

Before diving into the calculation methods, ensure that your **R** environment is correctly set up. Calculating SMAPE requires specific external packages that contain optimized functions for performance metrics. While the base installation of R provides powerful statistical capabilities, accessing specialized functions like those for SMAPE often necessitates installing and loading dedicated community-contributed packages from the Comprehensive R Archive Network ([CRAN](https://cran.r-project.org/)).

The primary packages we will focus on are the **Metrics** package, which offers a direct `smape()` function, and potentially the **forecast** package, which includes the broader `accuracy()` function. If these packages are not already installed on your system, you must use the `install.packages("PackageName")` command in your R console. Once successfully installed, they must be loaded into the current session using the `library(PackageName)` function. Failure to load a required library will result in errors when attempting to call the functions contained within it.

For the examples that follow, we will be heavily utilizing the **Metrics** package, as it provides a

clean, dedicated function specifically for this purpose. The initial setup requires just a few lines of code to ensure all necessary tools are accessible, allowing for smooth and efficient execution of the error calculation steps.

Method 1: Calculating SMAPE Using the Specialized Metrics Package

The most straightforward and generally recommended method for calculating **SMAPE** in R is by leveraging the specialized functions available in the **Metrics** package. This package is specifically designed to provide a wide range of standard evaluation metrics, ensuring that the calculations are performed efficiently and correctly according to standard statistical definitions. Using a dedicated function like `smape()` eliminates the need to manually code the complex SMAPE formula, significantly reducing the risk of implementation errors.

To begin, we must load the **Metrics** library and then define our sample datasets. For demonstration purposes, we will define two simple vectors: one representing the **actual**, observed values and another representing the **forecasted** values generated by a hypothetical model. These vectors must be of the same length, as SMAPE requires a one-to-one comparison between each forecast and its corresponding actual observation.

The `smape()` function is extremely simple to use, requiring only two arguments: the vector of actual values and the vector of forecasted values, in that specific order. The output is a single numerical value representing the symmetric mean absolute percentage error, typically presented as a decimal, which should then be multiplied by 100 to express it as a percentage.

Here is the practical implementation of this approach using sample data:

library(Metrics)

```
#define actual values: These represent the observed historical data points.
```

```
actual <- c(12, 13, 14, 15, 15, 22, 27)
```

```
#define forecasted values: These represent the predictions made by the model.
```

```
forecast <- c(11, 13, 14, 14, 15, 16, 18)
```

```
#calculate SMAPE by calling the dedicated function
```

```
smape(actual, forecast)
```

```
0.1245302
```

Upon execution, we observe that the symmetric mean absolute percentage error for this specific set of forecasts is approximately 0.1245302. When translated into a more user-friendly percentage, we can see that the symmetric mean absolute percentage error for this model is **12.45%**. This

result provides immediate insight into the model's performance characteristics.

Method 2: Implementing SMAPE with a Custom R Function

Another highly flexible way to calculate **SMAPE** is to create our own function. While using pre-built packages like **Metrics** is convenient, defining a custom function offers deeper control over the calculation process, the ability to integrate specific customizations or constraints, and independence from external package dependencies. For those integrating R scripts into larger systems or simply seeking a clearer understanding of the underlying mathematics, writing a custom function is highly valuable.

To create this custom function in R, we must map the mathematical definition of the symmetric mean absolute percentage error directly into R code syntax. The function must accept two arguments--the actual values (`a`) and the forecasted values (`f`)--and then apply the core SMAPE formula. The implementation involves using vectorized operations (like `abs()` and `sum()`) to efficiently handle the inputs.

The critical part of the implementation involves ensuring the correct algebraic representation of the formula. Specifically, the denominator term in the core formula, which is $\frac{|actual| + |forecast|}{2}$, is simplified in R code by multiplying the numerator by 2, resulting in $\frac{2 * |forecast - actual|}{|actual| + |forecast|}$.

Here is the structure for defining and using a robust custom SMAPE function:

```
find_smape <- function(a, f) {  
  return (1/length(a) * sum(2*abs(f-a) / (abs(a)+abs(f))*100))  
}
```

We can then use this function to calculate the SMAPE between a vector of actual values and forecasted values, using the same sample dataset to ensure consistency:

```
#define actual values for comparison  
actual <- c(12, 13, 14, 15, 15, 22, 27)  
  
#define forecasted values for comparison  
forecast <- c(11, 13, 14, 14, 15, 16, 18)  
  
#calculate SMAPE using the custom function  
find_smape(actual, forecast)  
  
12.45302
```

Once again the SMAPE turns out to be **12.45%**, which matches the results from the previous example, confirming the mathematical equivalence of both implementation methods.

Comparing SMAPE with Other Forecasting Error Metrics

While **SMAPE** is highly valued for its robustness and intuitive percentage representation, it is essential to understand how it stands in comparison to other popular error metrics, such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Choosing the right metric depends heavily on the specific domain, the distribution of the errors, and the business penalty associated with large forecast errors.

MAE is scale-dependent, meaning its value changes if the units of measurement change (e.g., switching from units to hundreds of units). SMAPE, being a percentage error, is inherently scale-independent, making it ideal for comparing model performance across different datasets or time series with vastly different scales. This is a critical advantage in large organizations that manage diverse product lines or services where a uniform metric is preferred.

Furthermore, SMAPE's primary advantage over standard MAPE lies in its symmetrical nature. Standard MAPE suffers from an asymmetry bias: it penalizes negative errors (overestimation) much more heavily than positive errors (underestimation), particularly when actual values are small or near zero. By using the average of the actual and forecast values in the denominator, SMAPE mitigates this bias, providing a fairer assessment of predictive accuracy. However, practitioners must be mindful of potential division-by-zero errors if both the actual and forecasted values are zero simultaneously, requiring careful data handling.

Interpreting SMAPE Results and Practical Applications

Interpreting the **SMAPE** value is straightforward due to its percentage format. A SMAPE of 0% indicates a perfect forecast where predicted values exactly match actual observations. As the SMAPE value increases, the deviation between the forecast and the reality grows, signaling poorer model performance. The typical goal in forecasting is to select the model that yields the lowest SMAPE across the validation or test dataset, thereby ensuring maximum reliability.

In practical applications, benchmarks for acceptable SMAPE vary significantly by industry. For instance, in highly volatile financial markets, a SMAPE of 15% might be considered excellent, whereas in stable manufacturing or utility processes, a SMAPE exceeding 5% might indicate significant operational problems requiring model recalibration. It is crucial to benchmark the calculated SMAPE against historical performance or industry standards to determine if the error rate is satisfactory and operationally viable.

SMAPE is particularly useful in environments that prioritize balanced errors, such as inventory

management. An error in inventory forecasting results in either holding excess stock (costly storage) or suffering stockouts (lost sales). Since SMAPE treats under- and overestimation errors symmetrically, it encourages models that minimize the total operational disruption caused by forecasting inaccuracies, making it a critical Key Performance Indicator (KPI) for supply chain professionals.

Conclusion: Mastering SMAPE Calculation in R

Mastering the calculation of the Symmetric Mean Absolute Percentage Error (SMAPE) in the R environment is a fundamental step toward professionalizing forecasting and model evaluation. We have demonstrated two robust methods: utilizing the optimized and readily available ``smape()`` function within the **Metrics** package, and developing a custom R function that provides complete control over the mathematical implementation. Both methods consistently produce accurate results, confirming the robust nature of the SMAPE metric.

Data scientists are encouraged to choose the method that best fits their workflow--relying on the package function for rapid analysis or developing a custom function for integration into bespoke software pipelines. Regardless of the choice, understanding the symmetrical nature of SMAPE ensures that models are evaluated fairly, leading to improved decision-making and higher predictive accuracy across various business domains. By effectively implementing and interpreting SMAPE, R users can elevate the quality and trustworthiness of their statistical forecasting models.