Understanding and Calculating SMAPE (Symmetric Mean Absolute Percentage Error) in R

Authored by

Mohammed looti

November 5, 2025

Introduction to SMAPE and its Importance in Time Series Analysis

The accurate evaluation of models is the cornerstone of effective time-series analysis and forecasting. Among the variety of metrics available, the **Symmetric Mean Absolute Percentage Error (SMAPE)** stands out as a highly robust and frequently utilized tool. Its fundamental purpose is to quantify the **predictive accuracy** of models by meticulously assessing the deviation between predicted outcomes and the true, observed data points. Unlike traditional metrics which can struggle under specific conditions, SMAPE was specifically engineered to overcome common pitfalls, particularly those arising when actual values approach or equal zero.

In the realm of quantitative analysis, selecting the appropriate metric is paramount. While metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) provide error measurements in the absolute units of the data, percentage errors--such as SMAPE--offer context-independent evaluation. This crucial feature allows analysts to seamlessly compare model performance across diverse scales, domains, and datasets, providing a standardized measure of fit. A lower value for the calculated SMAPE consistently indicates superior model performance and predictive power. For example, a SMAPE result of 7% suggests that the model's forecasts deviate from the actual observations by an average of 7% across the entire analyzed sample set.

Key Advantages: Why SMAPE Excels Over Standard Percentage Metrics

A primary benefit that elevates SMAPE above its alternatives, such as the standard Mean Absolute Percentage Error (MAPE), is its ability to provide a normalized and inherently bounded percentage error. This normalization makes the metric intuitive to interpret and highly scalable. The calculation achieves this stability by incorporating the average of both the actual and the forecasted values in the denominator. This mechanism ensures that the resulting percentage error remains bounded, generally between 0% and 200%.

The concept of **symmetry** is central to SMAPE's design and effectiveness. Traditional percentage error metrics often exhibit inherent bias, meaning they may penalize positive errors (overestimation) and negative errors (underestimation) unequally. By contrast, SMAPE symmetrically balances the error calculation based on both the actual value and the forecasted value. This balanced approach ensures that the metric is robust against directional bias, providing a more fair and reliable assessment of model fit, regardless of whether the forecast consistently overshoots or undershoots the true outcomes.

Crucially, SMAPE resolves the critical issue faced by MAPE: the potential for infinite or undefined errors. If the actual value is zero, the MAPE formula results in an undefined division. Since SMAPE utilizes the sum of the absolute actual and forecasted values in its denominator, it effectively handles zero or near-zero observations, making it a much safer and more reliable metric for volatile or intermittent time series data. This stability is why SMAPE is often the preferred metric in

large-scale machine learning competitions involving diverse datasets.

The Formal Mathematical Definition of SMAPE

Translating the concept of symmetric error measurement into a tangible calculation requires a precise mathematical formulation. The process involves summing the absolute differences between the forecast and actual values, normalizing this difference by the average of their absolute values, and subsequently averaging this ratio across the entire sample size. The mathematical formula for calculating the SMAPE is typically defined as follows, often multiplied by 100 to yield a percentage:

SMAPE = (1/n) * Σ(|forecast - actual| / ((|actual| + |forecast|)/2)) * 100

To ensure clarity when implementing this formula in code, it is essential to understand the definition of each component:

Σ - This is the summation operator, indicating that the operation within the parentheses must be summed across all observations in the dataset.
n - This variable denotes the sample size, or the total number of data points being analyzed within the series.
actual - Represents the observed or true data value at a specific point in time.
forecast - Represents the predicted data value generated by the model for that same point in time.

When preparing to implement this logic in programming environments, such as the R programming language, the division by 2 in the denominator is frequently simplified. Programmers often invert this division and apply it as a multiplication by 2 in the numerator (2 * |forecast - actual|), streamlining the code structure while maintaining complete mathematical equivalence to the original definition. The subsequent sections will detail two highly effective methods for calculating this metric efficiently within R.

Implementation in R: Method 1 - Utilizing the Metrics Package

For data scientists working within the R environment, the most efficient and often recommended method for calculating standardized error metrics involves leveraging specialized external libraries. The **Metrics** package is a powerful, well-maintained tool that provides a suite of common evaluation statistics, including a dedicated, optimized function for computing SMAPE. This approach minimizes the necessity for manual formula transcription, significantly reducing the potential for coding errors inherent in custom implementations.

Prior to execution, it is mandatory to ensure that the **Metrics** package is installed on the system and loaded into the current R session using the `library()` function. The package exposes a clean

and straightforward function, `smape()`, which expects two primary input vectors: the vector containing the actual observed values, and the vector containing the forecasted (predicted) values. Utilizing this pre-optimized function ensures both compliance with established statistical standards and maximum computational reliability, making it the preferred choice for production environments.

The following R code block illustrates the process: loading the required library, defining realistic sample data vectors, and executing the `smape()` function to derive the final error metric for the comparison between the actual and forecast series.

library(Metrics)

# Define the vector of actual observed values
actual <- c(12, 13, 14, 15, 15, 22, 27)

# Define the vector of forecasted (predicted) values
forecast <- c(11, 13, 14, 14, 15, 16, 18)

# Calculate the Symmetric Mean Absolute Percentage Error (SMAPE)
smape(actual, forecast)

0.1245302

The output generated by the package reveals a raw SMAPE value of 0.1245302. When properly converted and expressed as a percentage, the SMAPE for this specific model evaluation stands at approximately **12.45%**. This figure quantifies the average magnitude of the forecast error relative to the mean value of the actual and predicted data points, providing an immediate, interpretable measure of model accuracy.

Implementation in R: Method 2 - Developing a Custom Function

While package utilization is standard practice, many experienced quantitative analysts and data scientists frequently choose to construct their own customized functions. Creating a bespoke implementation of SMAPE provides several distinct advantages, including granular control over error handling logic, precise management of potential edge cases (such as zero values), and the ability to seamlessly integrate the metric calculation within advanced optimization loops or proprietary modeling frameworks.

This approach necessitates the direct translation of the rigorous mathematical definition of SMAPE into executable R code. We define a new function, often named descriptively like `find_smape`, which must accept two arguments--the actual values (`a`) and the forecasted values (`f`)--and return the final calculated percentage error. The function implementation must scrupulously adhere

to the symmetric formula, correctly handling absolute values and the crucial normalization step.

The custom function provided below meticulously implements the SMAPE formula. Note the multiplication by 2 in the numerator: this is the standard programming simplification that correctly accounts for the division by 2 in the original denominator, ensuring mathematical fidelity and accurate calculation.

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

To validate the precision of this custom implementation, we apply the newly defined function to the identical dataset utilized in Method 1. Successful execution that yields a matching result confirms that our custom code is mathematically sound and consistent with the established functionality of the **Metrics** package.

```
# Define the vector of actual observed values
actual <- c(12, 13, 14, 15, 15,22, 27)

# Define the vector of forecasted (predicted) values
forecast <- c(11, 13, 14, 14, 15, 16, 18)

# Calculate SMAPE using the custom function
find_smape(actual, forecast)
```

12.45302

As anticipated, the result derived from the custom `find_smape` function is precisely **12.45302%**. This outcome perfectly aligns with the result obtained from the streamlined `smape()` function in the Metrics package, decisively confirming the mathematical accuracy and validity of the self-written R implementation.

Interpreting and Contextualizing the SMAPE Score

Understanding the implications of a calculated SMAPE value is paramount for effective decision-making in model evaluation. Since SMAPE is inherently expressed as a percentage, it offers a direct, easily communicated measure of the average forecasting error relative to the overall scale and magnitude of the data series. This percentage provides immediate insight into the reliability of the predictive model, without requiring knowledge of the original data units.

When tasked with evaluating and comparing several competing forecasting models, the model that

consistently produces the lowest SMAPE score is conventionally deemed to possess the highest **predictive accuracy**. However, this interpretation must always be contextualized against the specific domain and inherent volatility of the time series. For example, in a stable environment like long-term inventory planning, a 10% SMAPE might be considered unacceptable, whereas the same 10% error might be viewed as highly successful when forecasting highly volatile financial market trends or complex economic indicators.

It is vital to reiterate the fundamental difference between SMAPE and MAPE during interpretation. SMAPE calculates the error relative to the average of both the actual and the predicted values, providing its characteristic stability and boundedness. In contrast, MAPE calculates the error strictly relative to the actual value, making it susceptible to distortion when actual values are small. This inherent averaging mechanism within SMAPE provides the metric's robustness and symmetry, ensuring stable evaluations even when dealing with datasets exhibiting intermittent demand or values fluctuating near zero.

Conclusion: Best Practices for SMAPE Calculation in R

The calculation of the **Symmetric Mean Absolute Percentage Error** in the R environment can be executed with high reliability through two distinct, validated methods: either by utilizing the specialized `smape()` function provided within the powerful Metrics package, or by constructing a precise custom function that strictly adheres to the metric's mathematical definition. As demonstrated, both methods consistently yield identical and accurate results when applied to the same data.

In most standard data analysis workflows, leveraging established and community-supported packages such as Metrics is highly recommended. This approach prioritizes computational speed, ease of use, and verified reliability. Custom function development should generally be reserved for scenarios requiring deep integration, specialized error handling, or highly specific zero-handling logic that goes beyond the package defaults. Regardless of the choice of method, SMAPE remains an essential, symmetric, and bounded metric, critical for performing comprehensive model evaluation across a wide range of real-world forecasting applications. Always ensure rigorous data preparation, confirming that your actual and forecast vectors are perfectly aligned in length and time correspondence to guarantee a statistically valid calculation.