Calculate RMSE in SAS

Authored by

Mohammed looti

November 15, 2025

Evaluating the performance of a predictive model is perhaps the most crucial step in any statistical analysis. One robust and widely accepted method used to assess the effectiveness of a **regression model** is the calculation of the **Root Mean Square Error (RMSE)**. This essential metric provides a clear quantitative measure of the average distance between the **predicted values** generated by the model and the actual, observed values present in the dataset. Understanding the RMSE allows practitioners to gauge how well their model generalizes to new data and provides a benchmark for comparing competing models.

The RMSE is particularly valued because it employs a squaring mechanism during its calculation. By squaring the differences between predictions and observations, the metric inherently gives disproportionately high weight to large errors. Consequently, a model with a very low RMSE is not only generally accurate but is also highly effective at avoiding significant outlier errors, making it a reliable measure of model fit. Simply put, the lower the RMSE value, the better a given model is at accurately capturing the underlying relationship and "fitting" the observed dataset.

To fully appreciate the output provided by statistical software like **SAS**, it is beneficial to examine the mathematical foundation of this metric. The formula used to determine the root mean square error, frequently abbreviated as **RMSE**, is derived from the standard deviation of the residuals (prediction errors). This foundational formula is integral to understanding its interpretation:

RMSE = $\sqrt{\Sigma(Pi - Oi)2 / n}$

In this widely applied formula, each variable holds a specific significance within the context of model evaluation:

Σ is the summation symbol, indicating that all squared errors across the entire dataset must be aggregated.
Pi represents the predicted value generated by the regression equation for the ith observation in the dataset.
Oi represents the observed, or actual, value for the ith observation in the dataset.
n is the total sample size, or the number of observations, used in the calculation.

The following comprehensive, step-by-step example demonstrates the practical application of these concepts, illustrating exactly how to calculate and explicitly extract the RMSE for a **simple linear regression** model using the powerful capabilities of the **SAS** statistical software package.

Step 1: Preparing Your Dataset in SAS

Before fitting any model, the raw data must be properly defined and structured within the **SAS** environment. For this illustrative example, we will construct a compact dataset intended to analyze the relationship between study effort and academic performance. This dataset includes the total

hours studied and the corresponding final exam score achieved by 15 individual students. This structure is foundational for demonstrating how RMSE is applied in a real-world context.

Our objective is to fit a standard **simple linear regression** model where the variable *hours* will serve as the independent predictor variable, and the variable *score* will be designated as the dependent response variable. The choice of simple linear regression is deliberate; it provides a straightforward framework for calculating and interpreting the RMSE without the complexities introduced by multivariate models.

The subsequent **SAS** code block details the necessary syntax to create this dataset. We utilize the `DATA` statement to name the dataset (`exam_data`), the `INPUT` statement to define the variables and their order, and the `DATALINES` statement to input the raw observations directly into the system. This method is common for handling small, self-contained datasets within a SAS program:

```
/*create dataset for analysis*/
data exam_data;
input hours score;
datalines;
1 64
2 66
4 76
5 73
5 74
6 81
6 83
7 82
8 80
10 88
11 84
11 82
12 91
12 93
14 89
;
run;

/*view dataset to ensure correct loading*/
proc print data=exam_data;
```

Obs	hours	score
1	1	64
2	2	66
3	4	76
4	5	73
5	5	74
6	6	81
7	6	83
8	7	82
9	8	80
10	10	88
11	11	84
12	11	82
13	12	91
14	12	93
15	14	89

Step 2: Executing the Simple Linear Regression Model

With the data successfully loaded and verified, the subsequent step involves fitting the statistical model that will generate the predicted scores necessary for calculating the errors. The standard procedure for fitting ordinary least squares (OLS) **regression model**s in **SAS** is `PROC REG`. This powerful procedure determines the line of best fit by minimizing the sum of the squared residuals, which directly relates to the RMSE calculation.

The core syntax requires specifying the dataset and, critically, defining the functional relationship using the `MODEL` statement. In our context, `model score = hours;` instructs SAS to predict the score based on the value of hours studied. Upon execution, `PROC REG` outputs a comprehensive set of diagnostics, including the ANOVA table, parameter estimates (intercept and slope), R-squared values, and, relevantly, the Root MSE. This standard output provides a complete picture of the model's performance and statistical significance.

Executing the following simple code block initiates the regression analysis. While we are ultimately only interested in the RMSE, this initial execution reveals all the standard output statistics, providing context for the fit of the model. Analysts should examine the output to ensure the model's coefficients are significant and that the overall fit (R-squared) is acceptable before diving into error metrics.

/*fit simple linear regression model to the data*/

```
proc reg data=exam_data;
model score = hours;
run;
```

The REG Procedure
Model: MODEL1
Dependent Variable: score

Number of Observations Read	15
Number of Observations Used	15

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	847.26698	847.26698	63.91	<.0001
Error	13	172.33302	13.25639		
Corrected Total	14	1019.60000			

Root MSE	3.64093	R-Square	0.8310
Dependent Mean	80.40000	Adj R-Sq	0.8180
Coeff Var	4.52852		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	65.33395	2.10599	31.02	<.0001
hours	1	1.98237	0.24796	7.99	<.0001

Step 3: Focused Extraction and Interpretation of the RMSE Value

Although the previous step's comprehensive output contains the **RMSE**, it is often necessary in production environments or large batch processes to isolate specific summary statistics for reporting or further computation, avoiding the need to parse extensive output logs. **SAS** provides specific options within `PROC REG` to facilitate this focused extraction, allowing the user to suppress the standard output and save the desired metrics directly into a new SAS dataset.

To achieve this targeted extraction, we modify the `PROC REG` statement using two key options. First, the `NOPRINT` argument is added to the main procedure call, which instructs SAS to suppress the display of all standard regression output (the tables shown in Step 2). Second, we use the

`OUTEST=outest` option, which directs SAS to save the estimation results, including our desired error metric, into a new, temporary dataset named `outest`. Furthermore, the `/ RMSE` sub-option must be included within the `MODEL` statement to ensure that the Root Mean Square Error is specifically calculated and recorded in the output dataset alongside the model coefficients.

Once the model results have been saved to the `outest` dataset, we use `PROC PRINT` to retrieve and display only the RMSE value. SAS automatically labels the RMSE variable saved to the output dataset as `_RMSE_`. By using the statement `VAR _RMSE_;` within `PROC PRINT`, we explicitly tell SAS to display only this specific column, thus providing a clean, concise output focused solely on the measure of error. This method is highly efficient for automated reporting and quality control checks.

```
/*fit simple linear regression model while suppressing standard output*/
proc reg data=exam_data outest=outest noprint;
model score = hours / rmse;
run;
quit;

/*print only the RMSE of the model from the output dataset*/
proc print data=outest;
var _RMSE_;
run;
```

Obs	_RMSE_
1	3.64093

As demonstrated in the focused output above, the calculated **RMSE** value for this specific **regression model** is precisely **3.64093**.

Note: The argument **noprint** in **proc reg** is essential here, as it instructs SAS not to print the entire verbose output of regression results, thereby achieving a clean, single-metric output for the analyst.

Step 4: Interpreting the Result and Conclusion

The calculated RMSE of 3.64093 carries significant meaning in the context of our study. Since the RMSE is expressed in the same units as the response variable (exam score points), this value indicates that, on average, the predictions made by our simple linear regression model deviate

from the actual observed student scores by approximately 3.64 points. This quantifies the typical prediction error and provides a tangible measure of the model's accuracy.

To properly judge whether 3.64 points constitutes a "good" or "poor" fit, the RMSE must be contextualized relative to the magnitude and variability of the response variable. Given that exam scores typically range from 0 to 100, an average error of 3.64 points suggests a reasonably accurate model, especially considering the inherent variability in human performance that cannot be captured solely by study hours. If the RMSE were, for instance, 20 points, the model would be deemed ineffective. Therefore, interpretation always requires domain knowledge and consideration of the data scale.

Ultimately, the primary utility of the **RMSE** is not necessarily its absolute value, but its function as a comparative metric. If we were to develop another model--perhaps a quadratic regression or one incorporating additional predictors--we would calculate its RMSE and compare it directly to 3.64093. The model yielding the lowest RMSE would be statistically preferred as the one that provides the tightest and most robust fit to the data, confirming the efficiency and accuracy of our analysis performed in **SAS**.

Additional Resources

The following tutorials explain how to perform other common tasks in SAS: