Learning Polynomial Regression with SAS: A Step-by-Step Guide

Authored by

Mohammed looti

May 11, 2026

In the realm of statistical analysis, understanding the relationship between variables is paramount. Often, the initial approach involves simple linear regression, a powerful technique that assumes a direct, **straight-line relationship** between a single predictor variable and a response variable. This method is highly effective and widely applicable when the underlying data demonstrates clear linearity.

However, the complexity inherent in real-world data frequently defies simple linearity. Datasets derived from fields such as finance, environmental science, or engineering often exhibit complex, **curvilinear patterns**. When a linear model is mistakenly applied to inherently nonlinear data, the resulting fit is typically poor, leading to inaccurate predictions, biased standard errors, and a fundamental misunderstanding of the true relationship governing the variables. Recognizing this limitation is the first crucial step toward robust modeling.

To address these nonlinear challenges, polynomial regression emerges as a sophisticated and flexible modeling tool. This technique systematically extends the framework of linear regression by incorporating **higher-order polynomial terms** of the predictor variable. By introducing squared, cubed, or even higher powers of the independent variable, the model gains the necessary flexibility to accurately capture and describe intricate curvilinear trends. This article provides a comprehensive guide to implementing and interpreting polynomial regression using the powerful capabilities of the SAS statistical software package.

Theoretical Foundations of Polynomial Modeling

Although it models a nonlinear relationship between the observed variables, polynomial regression is fundamentally classified as a form of **multiple linear regression**. The key distinction lies in the transformation of the independent variable. Instead of relying solely on the original predictor variable (x), we introduce new variables that are powers of x (e.g., x^2, x^3, etc.). Despite these transformations, the model remains linear in its parameters (the coefficients, β), allowing standard ordinary least squares (OLS) estimation techniques to be applied.

Consider a second-degree, or **quadratic**, polynomial model. Its mathematical form is expressed as: $y = \beta0 + \beta1x + \beta2x2 + \varepsilon$. Similarly, a third-degree, or **cubic**, model expands this to include the third power term: $y = \beta0 + \beta1x + \beta2x2 + \beta3x3 + \varepsilon$. These higher-order terms enable the regression line to exhibit bends or curves, allowing it to follow the natural contour of the data points more closely than a simple straight line could ever achieve, thus increasing the model's explanatory power for complex phenomena.

A critical decision in this modeling process is determining the appropriate **degree of the polynomial** (n). While increasing the degree often improves the model's fit to the training data, it simultaneously increases the risk of overfitting. Overfitting occurs when a model captures the noise and specific idiosyncrasies of the sample data rather than the underlying general trend, leading to

poor performance when predicting outcomes for new, unseen data. Therefore, the selection of model complexity must be guided by diagnostic checks, statistical significance tests, and principles of parsimony to ensure the model is both accurate and generalizable.

Data Preparation and Initial Setup in SAS

To demonstrate the practical application of polynomial regression, we must first establish a sample dataset within the SAS environment. This illustrative dataset, which we will name `my_data`, is designed to contain a predictor variable, `x`, and a response variable, `y`, that visually suggest a nonlinear relationship. The initial step involves utilizing the `DATA` and `INPUT` steps to define and populate this foundational dataset.

The following SAS code block shows the exact syntax required to create and load these observations. The `DATA my_data;` statement initiates the dataset creation. The `INPUT x y;` command specifies the variable names. Crucially, the `DATALINES;` statement signals that the data records follow immediately, with each line representing a single observation where the values for `x` and `y` are listed sequentially.

```
/*create dataset*/
data my_data;
input x y;
datalines;
2 18
4 14
4 16
5 17
6 18
7 23
7 25
8 28
9 32
12 29
;
run;

/*view dataset*/
proc print data=my_data;
```

Following the execution of the `DATA` step, it is standard practice to verify the data integrity. The `PROC PRINT` procedure is executed to display the contents of the newly created `my_data` table.

This simple verification step ensures that all observations were read correctly and that the dataset is properly structured, preparing it for the subsequent analytical and visualization stages.

Obs	x	y
1	2	18
2	4	14
3	4	16
4	5	17
5	6	18
6	7	23
7	7	25
8	8	28
9	9	32
10	12	29

Visual Analysis: Identifying the Degree of Curvature

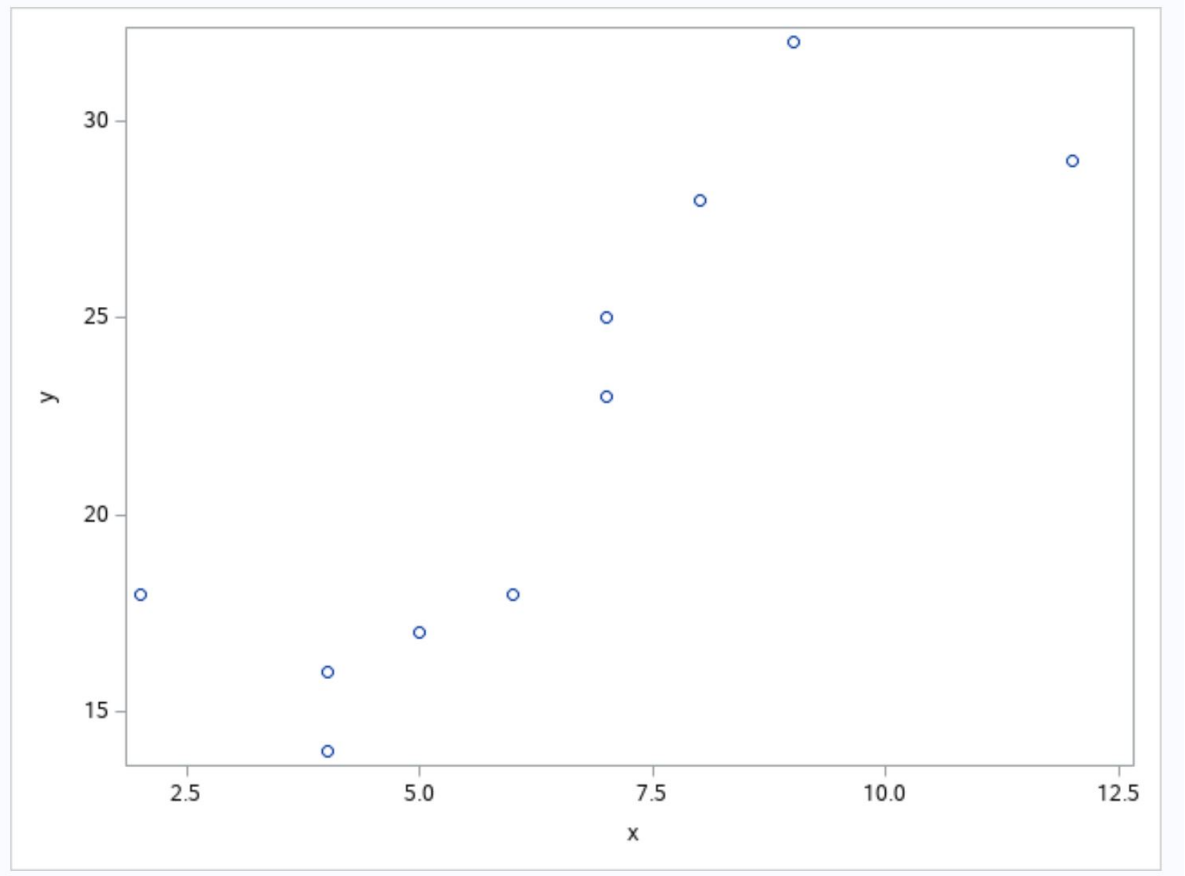
Before any statistical model fitting commences, the single most critical step is effective data visualization. A **scatter plot** of the predictor variable (x) against the response variable (y) offers immediate, invaluable insight into the functional form of the relationship. This visual inspection helps determine whether the pattern is linear, quadratic, exponential, or a higher-order polynomial, thereby preventing the misapplication of an inappropriate model.

We utilize the robust `PROC SGPLOT` procedure in SAS to generate a high-quality scatter plot. The syntax is straightforward: the `SCATTER` statement specifies the variables to be mapped to the X and Y axes. This graphical representation is the primary diagnostic tool used to hypothesize the necessary complexity of our regression equation by revealing potential inflection points and overall curvature.

```
/*create scatter plot of x vs. y*/
proc sgplot data=my_data;
scatter x=x y=y;
run;
```

Upon scrutinizing the resulting scatter plot, a definite nonlinear pattern becomes apparent. The data points initially decrease, reach a minimum around x=4, and then sharply increase, suggesting a curve with multiple changes in direction. Specifically, the data exhibits characteristics highly

consistent with a cubic relationship (a third-degree polynomial) since the curve appears to have two primary bends. This critical visual evidence directs us away from simple linear or quadratic models and confirms that a polynomial regression model incorporating the third power of `x` is the most appropriate choice for accurate modeling.



Constructing the Cubic Regression Model in SAS

Since the visual analysis strongly suggested a cubic relationship, the next operational step is to transform our dataset to include the necessary polynomial variables. This involves creating new columns for `x` squared (x^2) and `x` cubed (x^3) within the SAS environment. We revisit the `DATA` step to perform these crucial calculations, which effectively prepare our independent variables for the regression procedure.

The calculation is straightforwardly handled using the `**` operator for exponentiation: `x2 = x**2;` and `x3 = x**3;`. These computed variables (`x2` and `x3`) are then treated as distinct **independent predictors** in the subsequent multiple regression framework. This transformation is what allows the linear estimation process (OLS) to fit a nonlinear curve to the observed data points, as the model is now linear in terms of the new variables.

With the required polynomial terms now integrated into the dataset, we proceed to fit the model using the `PROC REG` procedure, the standard procedure in SAS for general linear regression. The `MODEL` statement specifies the structure of our cubic polynomial model: the dependent variable (`y`) is modeled as a function of the original predictor (`x`) and the newly created polynomial terms (`x2` and `x3`). The execution of this procedure triggers the estimation of all coefficients, providing the mathematical definition of our best-fit curve.

```
/*create dataset with new predictor variables*/
data my_data;
input x y;
x2 = x**2;
x3 = x**3;
datalines;
2 18
4 14
4 16
5 17
6 18
7 23
7 25
8 28
9 32
12 29
;
run;

/*fit polynomial regression model*/
proc reg data=my_data;
model y = x x2 x3;
run;
```

The REG Procedure
Model: MODEL1
Dependent Variable: y

Number of Observations Read	10
Number of Observations Used	10

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	343.46329	114.48776	80.47	<.0001
Error	6	8.53671	1.42278		
Corrected Total	9	352.00000			

Root MSE	1.19281	R-Square	0.9757
Dependent Mean	22.00000	Adj R-Sq	0.9636
Coeff Var	5.42184		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	37.21341	3.97355	9.37	<.0001
x	1	-14.23806	2.12779	-6.69	0.0005
x2	1	2.64819	0.33621	7.88	0.0002
x3	1	-0.12648	0.01582	-7.99	0.0002

Interpreting Statistical Output and Model Fit

The primary output generated by `PROC REG` is crucial for understanding and validating the constructed model. Among the many tables, the **Parameter Estimates** table stands out as the most informative, detailing the estimated values for the intercept and the slopes associated with each predictor variable (x, x^2, and x^3). These estimated coefficients, along with their associated t-statistics and p-values, define the precise shape and location of the fitted polynomial curve.

By extracting these estimated parameters from the SAS output, we can construct the specific cubic regression equation that best fits our observed data points. The resulting equation is:

y = 37.213 – 14.238x + 2.648x2 – 0.126x3

This equation is the mathematical blueprint of our model, enabling us to make **predictions** for the response variable `y` based on any input value of `x`. For instance, if we wish to predict `y` when `x`

equals 4, we substitute this value into the equation:

y = 37.213 – 14.238(4) + 2.648(4)2 – 0.126(4)3 = 37.213 – 56.952 + 42.368 – 8.064 = **14.565**

The result, 14.565, represents the model's predicted value for `y` when `x` is 4, demonstrating the predictive utility of the fitted equation.

Beyond individual parameter significance, overall model quality must be assessed. The **Adjusted R-squared** value, also provided in the `PROC REG` output, serves as an excellent measure of explanatory power, penalizing the inclusion of unnecessary predictor variables. For this specific cubic model, the Adjusted R-squared is reported as **0.9636**. This exceptionally high value, approaching 1, signifies that approximately 96.36% of the total variation in the response variable `y` is successfully accounted for or explained by the combined set of polynomial predictors (x, x^2, x^3). Such a strong fit confirms that the cubic model was indeed the appropriate choice for capturing the observed nonlinear trend.

Summary and Best Practices for Robust Modeling

This tutorial has successfully demonstrated the methodology for implementing polynomial regression within SAS. We navigated the process from recognizing the limitations of simple linear regression on curvilinear data, through essential data visualization to hypothesize the model structure, and finally, to the construction and interpretation of the cubic model.

The key procedural steps involved transforming the original predictor variable `x` into its higher-order powers (x^2 and x^3) using the SAS `DATA` step, and subsequently using the powerful `PROC REG` procedure to estimate the model coefficients. The interpretation of the parameter estimates yielded the final predictive equation, and the high Adjusted R-squared confirmed the model's strong explanatory capability for this particular dataset.

As a best practice, always prioritize **visual diagnostics** and model selection criteria. While polynomial regression offers immense flexibility, heed the risk of complexity. Experimentation with different polynomial degrees--perhaps starting with quadratic and increasing only if necessary--is advised. Always perform thorough diagnostic checks on residuals to ensure assumptions are met and that the model is robust and avoids the pitfalls of overfitting. A disciplined and methodical approach ensures the resulting statistical insights are reliable and actionable.

Additional Resources for SAS Proficiency

To further enhance your analytical skills and proficiency in using the SAS software suite, we recommend exploring tutorials that cover other essential statistical tasks and advanced procedures: