

How to Get Predicted Values and Residuals in Stata

Authored by
stats writer

March 9, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Get Predicted Values and Residuals in Stata*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=134711>

Introduction to Predictive Modeling and Statistical Analysis in Stata

In the contemporary landscape of data science and quantitative research, the ability to accurately model relationships between complex variables is of paramount importance. **Stata** serves as a premier integrated statistical software package that provides researchers with the robust tools necessary for data manipulation, visualization, and advanced econometric modeling. One of the most fundamental yet powerful techniques utilized within this environment is **linear regression**, a method designed to elucidate the underlying connection between one or more **explanatory variables** and a specific **response variable**. By quantifying these relationships, analysts can derive meaningful insights that inform policy decisions, business strategies, and academic inquiries.

The process of fitting a model involves more than simply calculating coefficients; it requires a comprehensive understanding of how well the model represents the observed data. When we execute a regression analysis, we are essentially attempting to find the best-fitting line through a cloud of data points, a process typically achieved through **Ordinary Least Squares (OLS)** estimation. This methodology produces a regression equation that allows us to generate **predicted values** for the response variable based on the input values of our predictors. However, because real-world data rarely adheres perfectly to a mathematical ideal, there will always be a discrepancy between what the model predicts and what is actually observed in the dataset.

These discrepancies are known as **residuals**, and they represent the unexplained variation or "error" within our model. Analyzing these residuals is a critical step in the diagnostic process, as they provide essential clues regarding the validity of the model's assumptions. If a model is well-specified, the residuals should ideally appear as random noise, indicating that the chosen **explanatory variables** have captured the systematic patterns in the data. Conversely, patterns within the residuals can signal issues such as non-linearity or **heteroscedasticity**. This tutorial provides a comprehensive, step-by-step guide on how to utilize **Stata** to obtain both predicted values and residuals, ensuring your statistical conclusions are grounded in rigorous diagnostic checking.

Foundations of Residual Analysis and Model Estimation

Before diving into the procedural steps in **Stata**, it is necessary to establish the theoretical framework that governs the generation of **predicted values**. In a multiple **linear regression** model, the relationship is expressed as an equation where the **response variable** is a function of the intercept, the slopes of the predictors, and an error term. Once the model is estimated, the "predicted value" (often denoted as \hat{y}) is the value of the response variable that the model calculates for a given set of observation inputs. These values represent the model's "best guess" and are crucial for understanding the trend established by the **Ordinary Least Squares** algorithm.

The second pillar of this analysis is the **residual**, which is mathematically defined as the observed value minus the predicted value. Residuals are the primary tool used by statisticians to assess the "goodness of fit" of a regression. By examining the distribution and characteristics of these residuals, we can determine if the model satisfies the Gauss-Markov assumptions, which are required for the OLS estimators to be the Best Linear Unbiased Estimators (BLUE). If these assumptions are violated, the standard errors may be biased, leading to incorrect inferences regarding the significance of the **explanatory variables**.

In **Stata**, the workflow for obtaining these metrics is highly streamlined through post-estimation commands. It is vital to remember that these values are dependent on the specific model currently stored in the software's memory. Therefore, the sequence of operations--fitting the model first and then extracting the diagnostics--is non-negotiable. This structured approach ensures that the **predicted values** and **residuals** accurately reflect the most recent iteration of your statistical analysis, allowing for an iterative refinement of the model as new insights are uncovered during the diagnostic phase.

Step 1: Loading the Dataset and Exploring Variable Distributions

The first practical step in our analysis involves preparing the data environment. For this demonstration, we will utilize the iconic "auto" dataset, which is a built-in resource provided by **Stata** for educational and testing purposes. This dataset contains various attributes of automobiles from 1978, including price, mileage, and physical dimensions. To begin, we use the **sysuse** command to load the data into the active memory. This command is specifically designed to access datasets that are shipped with the software, ensuring that the example is reproducible for any user with access to the platform.

Once the data is loaded, it is a standard best practice to perform an initial exploration using **descriptive statistics**. The **summarize** command provides a concise overview of the dataset, displaying the number of observations, the mean, standard deviation, and the minimum and maximum values for each variable. This step is crucial because it allows the analyst to identify potential outliers or missing data points that could skew the results of the **linear regression**. Understanding the scale and distribution of our **explanatory variables** (mpg and displacement) and our **response variable** (price) ensures that the subsequent model is built on a solid foundation.

sysuse auto

Following the loading of the data, the summary output gives us the necessary context to proceed. For instance, knowing the range of the price variable helps us interpret whether a specific **residual** is large or small in relative terms. In this exercise, we are specifically interested in how the fuel efficiency (mpg) and the engine size (displacement) influence the market price of the vehicle. By

verifying the data integrity through the summary command, we move forward with confidence into the modeling phase.

summarize

```
. sysuse auto
(1978 Automobile Data)
```

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
make	0				
price	74	6165.257	2949.496	3291	15906
mpg	74	21.2973	5.785503	12	41
rep78	69	3.405797	.9899323	1	5
headroom	74	2.993243	.8459948	1.5	5
trunk	74	13.75676	4.277404	5	23
weight	74	3019.459	777.1936	1760	4840
length	74	187.9324	22.26634	142	233
turn	74	39.64865	4.399354	31	51
displacement	74	197.2973	91.83722	79	425
gear_ratio	74	3.014865	.4562871	2.19	3.89
foreign	74	.2972973	.4601885	0	1

Step 2: Constructing the Multiple Regression Equation

With the dataset prepared and the variables understood, we proceed to the core of the analysis: fitting the **linear regression** model. In **Stata**, the **regress** command is the primary tool for estimating the coefficients of an **Ordinary Least Squares** model. By specifying "price" as the dependent variable and "mpg" and "displacement" as the independent variables, we instruct the software to calculate the linear relationship that minimizes the sum of squared **residuals**. This execution generates a detailed output table including the R-squared value, coefficients, and p-values, which are essential for hypothesis testing.

regress price mpg displacement

The resulting output provides the mathematical weights for our regression equation. In this specific model, the estimated price can be calculated using the intercept and the coefficients assigned to each predictor. For example, the negative coefficient for "mpg" suggests that as fuel efficiency increases, the price tends to decrease, holding engine displacement constant. Conversely, the positive coefficient for "displacement" indicates that larger engines are associated with higher vehicle prices. These coefficients form the basis for the **predicted values** that we will generate in

the next step.

```
. regress price mpg displacement
```

Source	SS	df	MS	Number of obs	=	74
Model	173587098	2	86793549.2	F(2, 71)	=	13.35
Residual	461478298	71	6499694.33	Prob > F	=	0.0000
Total	635065396	73	8699525.97	R-squared	=	0.2733
				Adj R-squared	=	0.2529
				Root MSE	=	2549.4

price	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
mpg	-121.1833	72.78844	-1.66	0.100	-266.3193 23.95276
displacement	10.50885	4.58548	2.29	0.025	1.365658 19.65203
_cons	6672.766	2299.72	2.90	0.005	2087.254 11258.28

It is important to analyze the significance of these predictors before relying on the model for predictions. If a variable's p-value is above the standard threshold (usually 0.05), it suggests that the variable may not have a statistically significant relationship with the response variable in the presence of other predictors. However, for the purposes of this tutorial on obtaining **predicted values** and **residuals**, we will proceed with the model as specified to demonstrate the post-estimation mechanics. The regression equation derived from this output is: estimated price = 6672.766 - 121.1833*(mpg) + 10.50885*(displacement).

Step 3: Generating Point Estimates for the Response Variable

Once the model is stored in the system memory, we can use the **predict** command to calculate the **predicted values** for every observation in our dataset. By default, the **predict** command in **Stata** calculates the linear prediction (xb). We must provide a name for the new variable that will store these values; in this instance, we will use **pred_price**. This new variable represents the price that our model expects each car to have, based solely on its mileage and displacement characteristics.

```
predict pred_price
```

Generating these values is immensely useful for validation. By comparing the **pred_price** to the actual observed price, an analyst can see how closely the model tracks the real-world data. This comparison is often the first step in assessing the practical utility of the model. To view these values side-by-side, we use the **list** command. Since viewing all 74 observations might be overwhelming, we restrict the output to the first 10 observations using the **in 1/10** qualifier. This provides a clear, tabular view of how the model's estimations align with reality.

list price pred_price in 1/10

```
. predict pred_price
(option xb assumed; fitted values)

. list price pred_price in 1/10
```

	price	pred_p~e
1.	4,099	5278.305
2.	4,749	7323.933
3.	3,799	5278.305
4.	4,816	6308.834
5.	7,827	8533.113
6.	5,788	6919.01
7.	4,453	6716.69
8.	5,189	6308.834
9.	10,372	7161.377
10.	4,082	6797.827

As we examine the list, we can observe that for some vehicles, the **predicted values** are remarkably close to the actual prices, while for others, there is a significant gap. This gap is the manifestation of the model's error. Understanding where the model succeeds and where it fails is the essence of predictive analytics. These stored values can also be used for future forecasting if new data on "mpg" and "displacement" becomes available for cars not currently in the dataset, showcasing the prospective power of **linear regression**.

Step 4: Isolating the Error Term for Diagnostic Purposes

After calculating the predictions, the next logical step is to isolate the **residuals**. While we could manually subtract the predicted price from the actual price, **Stata** provides a much more efficient method via the **predict** command with the **residuals** option. By executing this command and naming our new variable **resid_price**, the software automatically computes the difference for every row in the dataset. These residuals represent the portion of the vehicle's price that cannot be explained by the mileage or engine displacement alone.

predict resid_price, residuals

The importance of these values cannot be overstated in the context of statistical diagnostics. A **residual** essentially contains all the other factors that influence price--such as brand prestige, interior luxury, or safety features--that were not included in our regression model. By listing the

actual price, the **predicted values**, and the residuals together, we gain a comprehensive view of the model's performance for specific car models.

list price pred_price resid_price in 1/10

- . predict resid_price, residuals
- . list price pred_price resid_price in 1/10

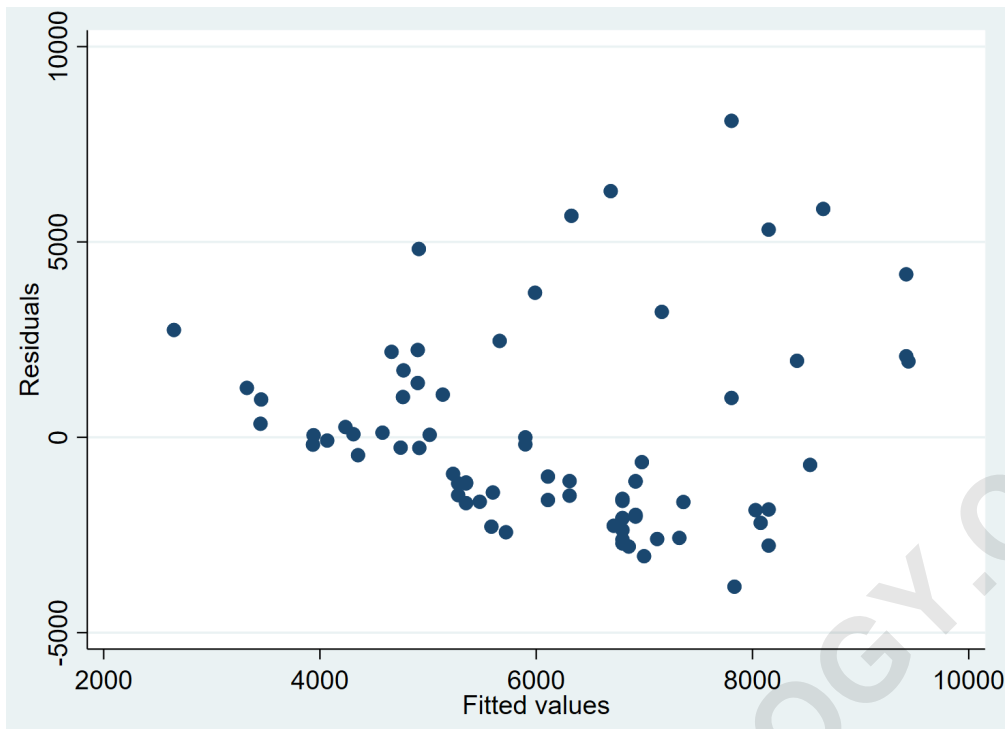
	price	pred_p~e	resid_p~e
1.	4,099	5278.305	-1179.304
2.	4,749	7323.933	-2574.933
3.	3,799	5278.305	-1479.304
4.	4,816	6308.834	-1492.834
5.	7,827	8533.113	-706.1129
6.	5,788	6919.01	-1131.01
7.	4,453	6716.69	-2263.69
8.	5,189	6308.834	-1119.834
9.	10,372	7161.377	3210.623
10.	4,082	6797.827	-2715.827

When we look at the **resid_price** column, we may see positive and negative numbers. A positive residual indicates that the actual price of the car is higher than what the model predicted (the car is "overpriced" relative to its specs), while a negative residual indicates the car is cheaper than the model suggests. In a perfectly specified **Ordinary Least Squares** model, the sum of these residuals will always be zero, reflecting the fact that the OLS line is the mathematical center of the data points.

Step 5: Diagnostic Visualization via Residual Plots

Data visualization is often more intuitive than examining raw numbers, and in regression analysis, the residual-versus-fitted plot is the "gold standard" for diagnostic checking. By creating a **scatter plot** with the **predicted values** on the x-axis and the **residuals** on the y-axis, we can visually inspect the data for violations of regression assumptions. In **Stata**, this is easily achieved using the **scatter** command. This visual representation allows us to identify if the error terms have a constant variance, a property known as homoscedasticity.

scatter resid_price pred_price



If the model is performing correctly and the OLS assumptions hold, the points on this **scatter plot** should be randomly dispersed around the horizontal axis (where the residual is zero) without any discernible pattern. However, as we observe the plot generated from the auto dataset, we might notice that the spread of the residuals increases as the **predicted values** increase. This "fan-shaped" or "funnel" pattern is a classic indicator of **heteroscedasticity**, which suggests that the model's predictive accuracy varies across different levels of the response variable.

The presence of **heteroscedasticity** is a significant finding. It implies that while our coefficient estimates may still be unbiased, our standard errors are likely incorrect, which could lead us to believe a variable is statistically significant when it is not. This highlights why obtaining residuals in **Stata** is not just a computational exercise, but a necessary component of responsible statistical reporting. Without this visual check, an analyst might proceed with a flawed model, leading to potentially erroneous conclusions.

Advanced Inference and Addressing Model Specification Errors

When the residual plot suggests issues like **heteroscedasticity**, the next step is to move beyond visual inspection toward formal statistical testing. One common method is the **Breusch-Pagan test**, which can be executed in **Stata** using the `estat hettest` command following a regression. This test provides a p-value that helps determine if the non-constant variance of the **residuals** is statistically significant. If the test confirms our visual suspicions, we must take remedial action to ensure the reliability of our statistical inferences.

One of the most common solutions to address the issue of non-constant variance is the use of **robust standard errors**. In **Stata**, this is achieved by adding the **vce(robust)** option to the **regress** command. This adjustment modifies the calculation of the standard errors to be "robust" against heteroscedasticity, providing more accurate p-values and confidence intervals. While this does not change the **predicted values** themselves, it ensures that our conclusions about the relationships between the **explanatory variables** and the price are valid.

In conclusion, obtaining **predicted values** and **residuals** is a vital workflow for any researcher using **Stata** for **linear regression**. These metrics allow for a deeper understanding of model performance, facilitate the identification of outliers, and enable the diagnostic testing required to satisfy the rigorous assumptions of classical statistics. By following the steps outlined in this guide--from data loading to diagnostic plotting--you can ensure that your econometric models are both accurate and statistically sound, providing a reliable basis for your data-driven insights.