

# How to Easily Interpret glm Output in R

Authored by  
**stats writer**

December 2, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Interpret glm Output in R*. PSYCHOLOGICAL SCALES.  
Retrieved from <https://scales.arabpsychology.com/?p=103748>

The R Programming Language provides robust statistical tools, chief among them the Generalized Linear Model (GLM) implemented via the `glm()` function. GLMs are a flexible class of statistical models that extend ordinary least squares regression to accommodate response variables that follow error distributions other than a normal distribution, making them essential for analyzing binary, count, or categorical data. Understanding the comprehensive output generated by the `glm()` function is paramount for drawing valid statistical conclusions.

The model summary provides critical information necessary for evaluating model performance and interpreting the relationship between predictors and the response. Key components include the estimated regression coefficients for each explanatory variable, the estimated intercept, and associated statistical measures such as standard errors and p-values. Furthermore, the output furnishes crucial diagnostics, including measures of fit like the Akaike Information Criterion (AIC) and Null and Residual Deviance, which are indispensable for model comparison and selection.

## Understanding the `glm()` Function Syntax

The standard implementation of the `glm()` function in R is designed for fitting generalized linear models across various data types. Its core flexibility stems from the ability to specify the error distribution appropriate for the response variable, moving beyond the normality assumptions of standard linear models.

This function uses the following standardized syntax:

**`glm(formula, family=gaussian, data, ...)`**

The primary arguments necessary for robust model specification are detailed below:

**formula:** This argument defines the structure of the statistical model, specifying the relationship between the response variable and the predictor variables (e.g., `y ~ x1 + x2`).

**family:** This is arguably the most crucial argument, determining the distribution of the response variable and the link function used. While `gaussian` is the default (equivalent to standard linear regression), common alternatives include `binomial` (for binary outcomes, often used in logistic regression), `poisson` (for count data), and `Gamma`.

**data:** This specifies the name of the R data frame containing the variables referenced in the formula.

Although GLMs encompass a wide range of models, the function is frequently employed to fit logistic regression models by setting the `family` argument to `binomial`, allowing for the modeling of probabilities and binary outcomes.

## Practical Example: Interpreting GLM Output in R using Logistic Regression

To demonstrate the interpretation process, we will utilize a practical scenario involving the built-in `mtcars` dataset in R. This dataset provides characteristics of 32 automobiles. Our goal is to fit a logistic regression model to predict whether a car has an automatic transmission (`am = 0`) or a manual transmission (`am = 1`) based on two continuous predictors: engine displacement (`disp`) and horsepower (`hp`).

First, we inspect the initial rows of the data to understand the variables involved and ensure data integrity:

```
#view first six rows of mtcars dataset
```

```
head(mtcars)
```

```
mpg cyl disp hp drat wt  qsec vs am gear carb
Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4
Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2
Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1
```

We are specifically interested in using the continuous predictor variables **disp** (displacement) and **hp** (horsepower) to model the likelihood that a vehicle uses a manual transmission, represented by the binary response variable **am**.

We proceed by fitting the model using the **glm()** function, explicitly setting `family=binomial` to handle the binary nature of the `am` response variable. The summary output then provides the necessary metrics for model evaluation and interpretation.

```
#fit logistic regression model
```

```
model <- glm(am ~ disp + hp, data=mtcars, family=binomial)
```

```
#view model summary
```

```
summary(model)
```

Call:

```
glm(formula = am ~ disp + hp, family = binomial, data = mtcars)
```

Deviance Residuals:

Min 1Q Median 3Q Max

```
-1.9665 -0.3090 -0.0017 0.3934 1.3682
```

Coefficients:

```
Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.40342 1.36757 1.026 0.3048
disp -0.09518 0.04800 -1.983 0.0474 *
hp 0.12170 0.06777 1.796 0.0725 .
```

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 43.230 on 31 degrees of freedom

Residual deviance: 16.713 on 29 degrees of freedom

AIC: 22.713

Number of Fisher Scoring iterations: 8

The following sections meticulously break down the crucial components of this output, from the residuals to the overall measures of fit.

## Interpreting Deviance Residuals

The summary output begins with the **Deviance Residuals**. These residuals serve a similar purpose to standard residuals in linear regression, quantifying the difference between the observed response values and the values predicted by the model. However, in GLMs, they are calculated based on the contribution of each observation to the overall model deviance, using the chosen distribution's properties.

Ideally, these residuals should be symmetrically distributed around zero. The presented range (Min, 1Q, Median, 3Q, Max) helps assess the model's systematic errors. A median close to zero, as seen in our example (-0.0017), suggests that the model is fitting the data reasonably well without significant skewness in the residuals. Large positive or negative values indicate poor fit for those specific observations, potentially highlighting outliers or areas where the model is fundamentally mis-specified, requiring further diagnostic checks.

## Analyzing Regression Coefficients and Statistical Significance

The core of model interpretation lies in the **Coefficients** table. For a generalized linear model using the binomial family (logistic regression), the **Estimate** value represents the average change in the **log odds** of the response variable being 1 (manual transmission), associated with a one-unit

increase in the predictor variable, holding all other predictors constant. It is crucial to remember that these are not direct linear changes in probability, but rather changes on the log-odds scale defined by the link function.

In our example: the coefficient for `disp` (displacement) is **-0.09518**. This negative value indicates that as engine displacement increases by one unit (e.g., one cubic inch), the log odds of having a manual transmission decrease by 0.09518. Conversely, the coefficient for `hp` (horsepower) is **0.12170**, suggesting that increasing horsepower increases the log odds of having a manual transmission, all else being equal. To convert these log-odds coefficients back to interpretable odds ratios, one must exponentiate the estimates ( $e^{\text{Estimate}}$ ).

The **Std. Error** column provides the standard error of the estimated regression coefficients, which is essential for calculating the **z value**. The  $z$ -statistic (or Wald statistic) is the ratio of the Estimate to its Standard Error. For the `disp` variable, the  $z$ -value is calculated as  $-0.09518 / 0.04800 = -1.983$ . This statistic tests the null hypothesis that the true coefficient is zero, meaning the predictor has no causal effect on the response variable.

The final column, **Pr(>|z|)**, provides the two-sided p-value associated with the corresponding z value. This value determines the statistical significance of each predictor. If the p-value is below a pre-determined significance level ( $\alpha$ ), typically 0.05, we reject the null hypothesis and conclude that the predictor is statistically significant. For `disp`, the p-value is 0.0474, establishing displacement as a significant predictor of transmission type in this model. For `hp`, the p-value is 0.0725, which falls outside the conventional 0.05 threshold but is significant at the 10% level, prompting further consideration of its practical relevance.

## Evaluating Model Utility using Null and Residual Deviance

In GLMs, the concept of **Deviance** substitutes the sum of squared errors used in standard linear regression. Deviance essentially measures how well the model fits the data, with lower values indicating a better fit relative to the saturated model (a model that perfectly predicts the data). The summary output presents two crucial deviance measures used for assessing overall model usefulness: Null Deviance and Residual Deviance.

The **Null Deviance** reflects the fit of the "null model"--a model that includes only the intercept term but no predictor variables. It represents the baseline level of unexplained variability in the response data. For our example, the Null Deviance is 43.230 on 31 degrees of freedom.

The **Residual Deviance** measures the lack of fit for the specific model we constructed, which includes  $p$  predictor variables. A smaller Residual Deviance relative to the Null Deviance suggests that the inclusion of the predictors has significantly improved the model fit. Our model shows a Residual Deviance of 16.713 on 29 degrees of freedom.

To formally test if our predictors, as a group, significantly improve the model compared to the Null Model, we perform a likelihood ratio test. This is achieved by computing the Chi-Square ( $\chi^2$ ) statistic, defined as the difference between the Null Deviance and the Residual Deviance:

$$\chi^2 = \text{Null deviance} - \text{Residual deviance}$$

The degrees of freedom for this test equal  $p$ , the number of predictors added (2 in this case). For our model:

$$\text{Calculation: } \chi^2 = 43.230 - 16.713 = 26.517$$

$$\text{Degrees of Freedom: } p = 2$$

We assess this  $\chi^2$  value using the Chi-squared distribution. A calculated  $\chi^2$  value of 26.517 with 2 degrees of freedom yields an extremely low p-value (approximately 0.000002). Since this p-value is far below the typical significance threshold of 0.05, we confidently reject the null hypothesis and conclude that our model, including `disp` and `hp`, is statistically useful and provides a significantly better fit than a model based solely on the intercept.

## Model Comparison using Akaike Information Criterion (AIC)

The **Akaike Information Criterion (AIC)** is a crucial metric provided in the `glm()` output, specifically designed for comparing non-nested or otherwise distinct statistical models. AIC serves as an estimate of the predictive accuracy of the model, penalizing complexity (number of parameters,  $K$ ) while rewarding goodness of fit (likelihood).

Therefore, when evaluating multiple candidate models for the same dataset, the model exhibiting the **lowest AIC value** is generally preferred, as it represents the best balance between maximizing predictive power and maintaining parsimony.

The AIC value is calculated based on the following mathematical relationship:

$$\text{AIC} = 2K - 2\ln(L)$$

Where the components are defined as:

**K:** Represents the total number of parameters estimated in the model, including the intercept. For our example,  $K = 3$ .

**$\ln(L)$ :** Denotes the maximized log-likelihood of the model. The log-likelihood quantifies how likely the observed data is under the fitted model, representing the goodness of fit.

It is important to understand that the absolute value of the AIC (22.713 in our example) holds little intrinsic meaning. Its utility is purely comparative. Researchers typically fit several models--for

instance, adjusting predictor variables or interaction terms--and then use the AIC score to objectively select the model that provides the most efficient and powerful explanation of the data without overfitting.

## Understanding Fisher Scoring Iterations

Unlike Ordinary Least Squares regression, generalized linear models often cannot be solved analytically. Instead, R uses an iterative maximum likelihood estimation technique, typically based on the **Fisher Scoring** algorithm (which is a numerically stable variant of the Newton-Raphson method). This process iteratively refines the parameter estimates until they converge to a stable maximum likelihood solution.

The `Number of Fisher Scoring iterations` reported (8 in our case) indicates how many steps the algorithm required to converge upon the final regression coefficients estimates. A high number of iterations or a failure to converge (often indicated by an error or warning) suggests potential numerical issues, such as severe multicollinearity, sparsity in the data (especially common in logistic regression with small sample sizes), or a condition known as complete separation. Observing a moderate number of iterations, such as 8, confirms that the model successfully converged and reached stable parameter estimates.

## Conclusion and Next Steps in GLM Modeling

Mastering the interpretation of the `glm()` output is fundamental to conducting sound statistical analysis in R. By carefully examining the Deviance Residuals, the statistically significant regression coefficients and their associated p-values, and the overall model fit diagnostics like Null Deviance and AIC, analysts can rigorously assess the relationships between predictors and diverse types of response variables.

Further exploration into GLMs should include converting log-odds to odds ratios for enhanced interpretability in binary models, and investigating specialized diagnostics tailored to specific distributions (e.g., assessing overdispersion in Poisson or Gamma models).

We encourage readers to explore additional resources that cover advanced usage of the **glm()** function in R, including considerations for quasi-likelihood models and handling convergence issues:

Advanced tutorials on model selection using BIC and adjusted AIC criteria.

Guidance on diagnosing and correcting common errors encountered when fitting models with the **glm()** function.

The following tutorials explain how to handle common errors when using the **glm()** function: