Understanding and Calculating R-Squared for Generalized Linear Models (GLMs) in R

Authored by

Mohammed looti

October 30, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Understanding and Calculating R-Squared for Generalized Linear Models (GLMs) in R*. PSYCHOLOGICAL STATISTICS. Retrieved from https://statistics.arabpsychology.com/?p=6115

Understanding R-Squared in Linear Models

When constructing a **linear regression** model, the standard measure of goodness-of-fit is **R-squared**, also formally known as the coefficient of determination. This widely adopted statistic provides an intuitive assessment by quantifying the proportion of the total **variance** in the dependent variable that is statistically explained by the set of independent variables included in the model. Essentially, it reveals how well the model accounts for the observed variability in the data.

The theoretical range for **R-squared** is 0 to 1. A value approaching 0 suggests that the predictors explain none of the variability around the mean response, resulting in a poor fit. Conversely, a value near 1 signifies that the model perfectly captures all the variability, indicating an ideal relationship. Consequently, higher R-squared values are generally desirable, as they demonstrate stronger explanatory power and better predictive capability, making this metric an essential tool for initial model evaluation and comparison in traditional linear modeling.

Despite its powerful utility in ordinary least squares regression, the conventional definition of R-squared relies on specific assumptions about the error distribution and the calculation derived from the decomposition of sums of squares. These foundational limitations prevent its direct application to statistical models designed for response variables that are not continuous and normally distributed, such as binary outcomes or count data. This challenge highlights the need for specialized metrics when assessing model fit in broader statistical methodologies.

The Challenge with Generalized Linear Models (GLMs)

While **linear regression** is ideal for continuous data conforming to normality assumptions, many real-world phenomena involve outcomes that violate these requirements. For instance, outcomes might be dichotomous (e.g., pass/fail), represented by counts (e.g., disease occurrences), or bounded by proportions. To effectively model these diverse data types, **Generalized Linear Models (GLMs)** were developed. GLMs offer a flexible framework by accommodating various response distributions derived from the **exponential family** and employing a **link function** to connect the linear predictor to the expected mean of the response.

Key examples of **GLMs** include **logistic regression** for modeling binary events and **Poisson regression** for modeling count data. Because these models utilize non-normal error distributions and the estimation process relies on maximum likelihood, the underlying mathematical mechanism (the sum of squares decomposition) that defines traditional R-squared is invalidated. Consequently, the classic R-squared value loses its meaningful interpretability within the GLM context.

The fundamental incompatibility between the assumptions of traditional R-squared and the structure of **GLMs** necessitates alternative measures for evaluating model performance. This led to

the creation of a family of metrics known as "**pseudo R-squared**" values. These statistics are designed to provide an analogous assessment of explanatory power and overall model fit for GLMs, allowing researchers to quantify the improvement achieved by including predictors, even when the response variable is not continuous.

Introducing McFadden's R-Squared: A Necessary Pseudo-Metric

To effectively gauge the fit of **GLMs**, several **pseudo R-squared** measures have been developed. Among the most popular, reliable, and widely implemented is **McFadden's R-Squared**. Crucially, McFadden's metric does not rely on variance decomposition but instead leverages the **log likelihood** values, which are the core output of the maximum likelihood estimation procedure used to fit GLMs.

McFadden's R-Squared quantifies the proportional improvement in fit achieved by the full model (with predictors) relative to a baseline model that contains only the intercept, known as the **null model**. This value, ranging from 0 to typically less than 1, allows for the comparison of competing GLMs applied to the same dataset. It is essential to remember that McFadden's R-Squared values are generally lower than traditional R-squared values, and direct comparison between the two metrics is statistically inappropriate and misleading.

While there are no universal, strict thresholds for interpreting the absolute value of **McFadden's R-Squared**, context is key. A common guideline suggests that values above 0.2 indicate a reasonable model fit, while values exceeding 0.4 often point toward a robust and strong explanatory model. However, the true utility of this metric lies in its comparative power: it is best used to determine which of several candidate GLMs provides the best relative fit to the data, offering the most significant reduction in unexplained uncertainty.

Decoding the Formula: Log Likelihood and Deviance

The foundation for calculating **McFadden's R-Squared** lies in the principle of **log likelihood**, which is central to the estimation process of **GLMs**. The formula provides an elegant comparison of model performance:

McFadden's R-Squared = 1 - (log likelihoodmodel / log likelihoodnull)

The components of this formula are critical for understanding the metric's meaning:

log likelihoodmodel: This represents the maximized **log likelihood** of the fitted model, which includes all specified predictor variables. A higher log likelihood indicates that the parameters chosen maximize the probability of observing the actual data, signifying a superior fit.
log likelihoodnull: This is the **log likelihood** of the **null model**--the simplest model containing

only the intercept term. It serves as the baseline measure of fit, assuming no relationship between the predictors and the response variable.

The ratio within the formula measures the gain in explanatory power achieved by the full model over the null model. Subtracting this ratio from 1 effectively quantifies the fractional reduction in unexplained variability or uncertainty. For practical calculation in R, it is important to note the relationship between the log likelihood and model **deviance**: Deviance is defined as `-2 * log-likelihood`. This allows the formula to be alternatively, and more commonly, expressed using deviances, which are readily available in R's `glm` output.

Practical Example: Calculating McFadden's R-Squared in R

To illustrate the application of **McFadden's R-Squared**, we will walk through a complete example using the R statistical environment. We will focus on a **logistic regression** model--a standard GLM for binary outcomes--using the well-known **Default** dataset found within the **ISLR package**. This dataset is derived from "An Introduction to Statistical Learning" and is perfect for demonstrating credit risk prediction.

Before fitting the model, we must ensure the necessary package is installed and loaded. The initial exploration of the **Default** data reveals 10,000 observations containing variables such as `default` (the binary response variable), `student`, `balance` (average credit card balance), and `income`. Our objective is to predict the probability of default based on these key predictors.

#install and load ISLR package
install.packages('ISLR')
library(ISLR)

#define dataset
data <- ISLR::Default

#view summary of dataset
summary(data)

default student balance income
No :9667 No :7056 Min. : 0.0 Min. : 772
Yes: 333 Yes:2944 1st Qu.: 481.7 1st Qu.:21340
Median : 823.6 Median :34553
Mean : 835.4 Mean :33517
3rd Qu.:1166.3 3rd Qu.:43808
Max. :2654.3 Max. :73554

#find total observations in dataset

```
nrow(data)
```

```
10000
```

We fit the logistic regression model using R's **`glm()` function**, specifying the binomial family. The resulting model summary contains the necessary components--specifically the Null Deviance and Residual Deviance--required for the manual calculation of McFadden's R-Squared.

```
#fit logistic regression model
model <- glm(default~student+balance+income, family='binomial', data=data)

#view model summary
summary(model)
```

```
Call:
glm(formula = default ~ balance + student + income, family = "binomial",
data = data)

Deviance Residuals:
Min 1Q Median 3Q Max
-2.4691 -0.1418 -0.0557 -0.0203 3.7383

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01 4.923e-01 -22.080 < 2e-16 ***
balance 5.737e-03 2.319e-04 24.738 < 2e-16 ***
studentYes -6.468e-01 2.363e-01 -2.738 0.00619 **
income 3.033e-06 8.203e-06 0.370 0.71152
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2920.6 on 9999 degrees of freedom
Residual deviance: 1571.5 on 9996 degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

In the context of GLMs, the **Null deviance** corresponds to the deviance of the null model (log likelihood null), and the **Residual deviance** corresponds to the deviance of our full model (log

likelihood model). Since `Deviance = -2 * log-likelihood`, the ratio `log likelihoodmodel / log likelihoodnull` is equivalent to `Residual Deviance / Null Deviance`. Utilizing this relationship allows us to perform the calculation directly from the summary output.

#calculate McFadden's R-squared for model
with(summary(model), 1 - deviance/null.deviance)

0.4619194

The resulting **McFadden's R-Squared** value is approximately **0.4619**. This figure is exceptionally high for a **pseudo R-squared** metric, signaling a very strong model fit. It indicates that the predictors--student status, balance, and income--collectively explain a significant portion of the variability in the default outcome, offering robust explanatory power far exceeding the baseline null model.

Streamlining Calculation with the pscl Package

While the manual calculation confirms the underlying principles of the metric, R users typically rely on specialized packages for efficiency and reliability. The **`pscl` package** is the gold standard for this task, offering the dedicated `pR2()` function designed specifically to compute various **pseudo R-squared** statistics for **GLMs**. Using `pR2()` automates the process and minimizes the chance of transcription or calculation errors.

After installing and loading the **`pscl` package**, we simply pass our fitted `glm` object to the `pR2()` function. The output confirms the result obtained through manual calculation, demonstrating that the package provides an accurate and significantly faster workflow for statistical analysis.

#install and load pscl package
install.packages('pscl')
library(pscl)

#calculate McFadden's R-squared for model
pR2(model)

McFadden
0.4619194

Conclusion: Assessing Model Fit in Non-Normal Contexts

In conclusion, traditional **R-squared** remains vital for **linear regression**, but its applicability ceases

when dealing with **Generalized Linear Models (GLMs)**. **McFadden's R-Squared** serves as the robust and essential alternative, derived from **log likelihood** maximization, making it ideal for evaluating models like **logistic regression**. By focusing on the proportional improvement over a null model, McFadden's R-Squared provides critical insight into the overall explanatory power of the covariates.

Whether calculated manually using the model's deviance values or efficiently accessed via the `pR2()` function in the **`pscl` package**, this pseudo R-squared metric is indispensable for model selection and performance assessment in non-normal statistical contexts. It is crucial, however, to interpret **pseudo R-squared** values contextually, using them primarily for comparison among competing models rather than judging absolute fit against the standards of classic linear models.

For those interested in delving deeper into R programming and statistical modeling, the following tutorials offer additional insights into common analytical tasks: