Learn to Calculate DFFITS for Regression Analysis in R

Authored by

Mohammed looti

November 6, 2025

In the expansive domain of statistics and advanced data analysis, ensuring the reliability of predictive tools, particularly regression models, is paramount. A critical step involves rigorously assessing whether individual observations unduly skew the overall model results. The presence of outliers or points exhibiting high leverage can dramatically distort coefficient estimates, leading to fundamentally unreliable conclusions and poor predictive performance.

To quantify this potential risk, data scientists rely on various influence diagnostics. Among the most informative is the metric known as **DFFITS**, an acronym for "Difference in Fits." This calculation measures the severity of change in a model's prediction when a specific data point is temporarily excluded from the fitting process. Utilizing **DFFITS** is essential for robust model diagnostics, enabling practitioners to pinpoint exactly which observations exert substantial influence on the model's predictive outcome.

This comprehensive, step-by-step guide is designed to equip analysts with the necessary tools to calculate, rigorously interpret, and effectively visualize the **DFFITS** metric for every observation within a standard linear model. We will execute this process using the powerful, open-source statistical programming language, R, ensuring clarity and reproducibility throughout the diagnostic workflow.

The Theoretical Foundation of DFFITS

The **DFFITS** metric provides a standardized measure of how much the predicted response ($\hat{y}_i$) for the i-th observation shifts when that observation is entirely removed from the dataset used to train the model. Crucially, this difference is standardized by the estimated standard error of the fitted value, which allows the resulting scores to be compared meaningfully across different models and datasets, regardless of the scale of the response variable.

An observation yielding a large **DFFITS** value, whether positive or negative, signifies that this particular data point holds considerable influence over the parameters of the regression equation and, consequently, its predictions. If a data point registers a high absolute **DFFITS** score, its removal would instigate a substantial and measurable alteration in the predicted value for that very point, highlighting its disproportionate weight in the fitting process.

Identifying these highly influential data points is a necessary step in the modeling lifecycle. Researchers can then undertake focused investigation to determine the cause of the influence--whether it stems from potential data entry errors, unique measurement conditions, or represents a genuine, yet powerful, outlier that necessitates careful deliberation during the final interpretation of the model results.

Step 1: Constructing the Linear Regression Model in R

The prerequisite for calculating any influence diagnostic, including **DFFITS**, is the successful establishment of a well-defined standard linear regression model. For demonstrative purposes, we will employ the widely recognized, built-in R dataset, **mtcars**. This dataset documents various performance characteristics for 32 distinct automobiles. Our specific objective is to model and predict the vehicle's miles per gallon (mpg) based on two key characteristics: engine displacement (disp) and horsepower (hp).

The following R code chunk initiates the analysis by loading the dataset, fitting the specified linear model using the fundamental `lm()` function, and subsequently providing a comprehensive summary detailing the resulting coefficient estimates, statistical significance, and overall measures of model fit:

 # Load the standard mtcars dataset
data(mtcars)

# Fit the multiple regression model: mpg ~ displacement + horsepower
model <- lm(mpg~disp+hp, data=mtcars)

# Display the detailed summary of the resulting model
summary(model)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.735904 1.331566 23.083 < 2e-16 ***
disp -0.030346 0.007405 -4.098 0.000306 ***
hp -0.024840 0.013385 -1.856 0.073679 .

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

Residual standard error: 3.127 on 29 degrees of freedom
Multiple R-squared: 0.7482, Adjusted R-squared: 0.7309
F-statistic: 43.09 on 2 and 29 DF, p-value: 2.062e-09

The output confirms the successful construction of our predictive model. The summary indicates that the two chosen predictors, displacement and horsepower, collectively account for approximately 75% of the variance observed in MPG, as evidenced by the Multiple R-squared statistic. With the model established, we can now proceed to the core task of calculating influence measures.

Step 2: Automated Calculation of DFFITS Scores in R

Once the model object (named `model` in our example) has been created using `lm()`, deriving the **DFFITS** values for every observation becomes a remarkably efficient process in R, thanks to its robust suite of built-in diagnostic tools. We utilize the specialized dffits() function, which is specifically designed for this purpose and requires only the fitted model object as its primary input argument.

The execution of this function generates a numerical vector, where each entry represents the calculated **DFFITS** score corresponding to a specific row (or observation) within the original **mtcars** dataset. To facilitate easier visual inspection, comparison, and subsequent data manipulation, we immediately convert this resultant vector into a standardized data frame format.

 # Calculate DFFITS scores for each observation in the fitted model
dffits <- as.data.frame(dffits(model))

Display the calculated DFFITS scores, indexed by car model
dffits

dffits(model)
Mazda RX4 -0.14633456
Mazda RX4 Wag -0.14633456
Datsun 710 -0.19956440
Hornet 4 Drive 0.11540062
Hornet Sportabout 0.32140303
Valiant -0.26586716
Duster 360 0.06282342
Merc 240D -0.03521572
Merc 230 -0.09780612
Merc 280 -0.22680622
Merc 280C -0.32763355
Merc 450SE -0.09682952
Merc 450SL -0.03841129
Merc 450SLC -0.17618948
Cadillac Fleetwood -0.15860270
Lincoln Continental -0.15567627
Chrysler Imperial 0.39098449
Fiat 128 0.60265798
Honda Civic 0.35544919
Toyota Corolla 0.78230167
Toyota Corona -0.25804885

Dodge Challenger -0.16674639
AMC Javelin -0.20965432
Camaro Z28 -0.08062828
Pontiac Firebird 0.67858692
Fiat X1-9 0.05951528
Porsche 914-2 0.09453310
Lotus Europa 0.55650363
Ford Pantera L 0.31169050
Ferrari Dino -0.29539098
Maserati Bora 0.76464932
Volvo 142E -0.24266054

Establishing and Applying the DFFITS Influence Threshold

While a raw list of **DFFITS** values provides the influence score for each observation, it is crucial to establish an objective benchmark to determine which scores are statistically significant and genuinely influential. A widely accepted heuristic threshold for flagging potentially problematic data points in statistical model diagnostics is defined by the formula: $2sqrt{p/n}$.

In this standardized formula:

p: Represents the count of predictor variables (or regressors) incorporated into the model, explicitly excluding the intercept term.
n: Represents the total number of observations (data points) utilized in the process of fitting the model.

Any observation whose absolute **DFFITS** value exceeds this calculated threshold is conventionally flagged as highly influential and warrants immediate and thorough review. Given our specific model setup, we have $p=2$ predictor variables (disp and hp) and $n=32$ total observations. We calculate this critical threshold value directly within R:

 # Extract the number of predictor variables (p) from the model object
p <- length(model$coefficients)-1

# Determine the total number of observations (n)
n <- nrow(mtcars)

# Calculate the DFFITS threshold value using the 2*sqrt(p/n) rule
thresh <- 2*sqrt(p/n)

thresh

0.5

The resulting calculated threshold for our **mtcars** model is precisely **0.5**. Consequently, any observation possessing an absolute **DFFITS** score greater than 0.5 is automatically classified as highly influential. To facilitate the focused identification of these points, we organize the calculated values by sorting them in descending order based on the magnitude of the score. This action brings the most influential observations immediately to the forefront for focused data analysis.

 # Sort observations by DFFITS score in descending order
dffits),]

0.78230167 0.76464932 0.67858692 0.60265798 0.55650363 0.39098449
0.35544919 0.32140303 0.31169050 0.11540062 0.09453310 0.06282342
0.05951528 -0.03521572 -0.03841129 -0.08062828 -0.09682952 -0.09780612
-0.14633456 -0.14633456 -0.15567627 -0.15860270 -0.16674639 -0.17618948
-0.19956440 -0.20965432 -0.22680622 -0.24266054 -0.25804885 -0.26586716
-0.29539098 -0.32763355

Upon reviewing the sorted numerical results, it is evident that five specific observations--Toyota Corolla (0.78), Maserati Bora (0.76), Pontiac Firebird (0.67), Fiat 128 (0.60), and Lotus Europa (0.55)--possess **DFFITS** values that significantly exceed the 0.5 threshold. These vehicles are deemed highly influential. Their temporary exclusion from the model training process would substantially alter the coefficients and significantly change the predicted MPG for the remaining set of automobiles.

Step 3: Visual Interpretation of Influential Observations

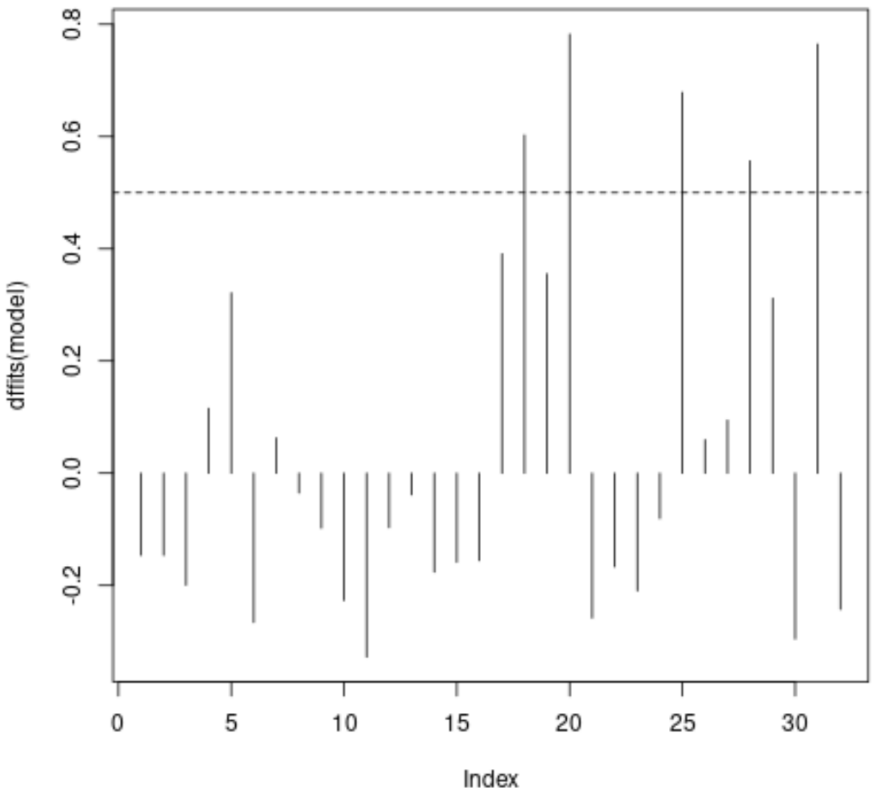
While numerical sorting is effective, visualization offers the most immediate and intuitive method for identifying influential points in relation to the calculated threshold lines. An index plot of the **DFFITS** values allows analysts to quickly grasp the distribution of influence across the dataset.

The `plot()` function, when applied directly to the vector generated by `dffits(model)`, produces a straightforward index plot. We utilize the `type='h'` argument to render distinct vertical lines for enhanced clarity. Subsequently, the `abline()` function is used to overlay two critical horizontal dashed lines corresponding to the positive threshold (**thresh**) and the negative threshold (**-thresh**).

 # Generate the index plot of DFFITS values for each observation
plot(dffits(model), type = 'h')

Add horizontal reference lines at the positive and negative thresholds

```
abline(h = thresh, lty = 2)
abline(h = -thresh, lty = 2)
```



In this generated visualization, the x-axis corresponds to the sequential index of each observation within the **mtcars** dataset, while the y-axis represents the calculated **DFFITS** score. Any vertical line segment that visibly extends beyond the dashed horizontal boundary lines (at $y=0.5$ and $y=-0.5$) clearly signifies an observation that warrants specific, in-depth attention due to its pronounced impact on the model's predictive capabilities. The plot confirms the numerical findings, visually highlighting the five highly influential points.

Conclusion and Advanced Diagnostic Measures

The systematic calculation and evaluation of **DFFITS** represent a fundamental and indispensable step in establishing the robustness and validity of any statistical model. By proactively identifying and thoroughly analyzing data points that yield high **DFFITS** scores, data analysts can effectively mitigate the risk of obtaining skewed or misleading results, thereby constructing more stable and reliable predictive models.

It is important to recognize that **DFFITS** specifically focuses on quantifying the impact of an observation on the overall fitted values (predictions). For a truly comprehensive assessment of

influence, analysts should employ a suite of complementary diagnostics. Metrics such as Cook's Distance and DFBETAS provide additional, valuable insights into how individual observations specifically affect the estimation of the model coefficients and the overall quality of the fit. Integrating multiple diagnostic measures provides the most complete and nuanced picture of data influence.

For individuals interested in expanding their knowledge of advanced model diagnostics within the R environment, the following resources offer excellent supplementary reading materials and official documentation references:

The official R documentation on influence measures, detailing the functions available in the `stats` package.
Detailed explanations and practical applications of Cook's Distance in regression analysis.
Tutorials focusing on comprehensive residual analysis, model assumptions, and methods for addressing issues like heteroscedasticity.