How to Calculate Leverage Statistics in R

Authored by

stats writer

December 16, 2025

Understanding and calculating **leverage statistics** is a fundamental step in validating any statistical regression model. Leverage quantifies how far an observation's predictor variables are from the mean of those variables across the entire dataset. Observations that exhibit high leverage have the potential to exert a disproportionately large influence on the estimated regression coefficients and, consequently, on the overall conclusions drawn from the model. While the original content referenced the `leverage` function from the `MASS` package, the standard procedure for calculating leverage (or hat values) for a fitted linear model in R often utilizes the built-in `hatvalues()` function, which we will demonstrate here.

High leverage observations are not inherently problematic, but they necessitate careful scrutiny. If an observation falls far outside the pattern established by the majority of the data points in the predictor space, it requires investigation to determine if it is merely unusual or if it represents an error or a unique characteristic of the phenomenon being modeled. Identifying these points is essential for robust model development, ensuring that the model's parameters are stable and accurately reflect the underlying relationship between the variables. This tutorial provides a comprehensive guide to calculating, interpreting, and visualizing these crucial leverage statistics within the R environment.

The Importance of Leverage in Model Diagnostics

In statistical modeling, we strive to create models that generalize well and whose parameters are not overly sensitive to individual data points. **Leverage** helps us measure this sensitivity by focusing exclusively on the independent variables. An observation possesses high leverage if its input values (the X variables) are extreme relative to the center of the data cloud. This position in the predictor space gives that observation geometric "leverage," meaning if the corresponding response variable (Y value) is also unusual, the point can dramatically pull the regression line toward itself.

It is crucial to differentiate between high leverage and simple large residuals. A large residual indicates that the observation's response variable (Y) is poorly predicted by the model, suggesting it might be an outlier in the response space. Conversely, high leverage indicates an outlier in the predictor space. An observation can have high leverage yet fit the model perfectly (small residual), or it can have low leverage but a very large residual. The most concerning scenario is an observation that has both high leverage and a large residual--this is known as an **influential observation**, capable of radically changing the model coefficients.

Analyzing leverage is thus one of the first steps in assessing the quality of a regression model, allowing analysts to detect potential issues before making inferences. Ignoring highly influential points can lead to models that inaccurately represent the population and fail spectacularly when applied to new, unseen data. By calculating and visualizing these statistics, we gain immediate

insight into the distribution of the data and the stability of our parameter estimates in R.

Step 1: Building a Robust Regression Model in R

To illustrate the process of calculating leverage, we will first establish a baseline linear regression using a well-known dataset available directly within R: the **mtcars** dataset. This dataset contains fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models). For this example, we will model miles per gallon (`mpg`) as the response variable, predicted by engine displacement (`disp`) and horsepower (`hp`). This setup provides a practical context for identifying observations that might disproportionately affect the relationship between vehicle performance metrics and fuel efficiency.

The following code snippet loads the dataset into the R environment and then fits the multiple regression model using the `lm()` function. The `lm()` function is the standard tool in R for fitting linear models. Following the model fitting, we examine the summary output to understand the initial fit quality, including the coefficients, standard errors, and overall model statistics like R-squared, although these statistics do not directly measure leverage itself.

#load the dataset
data(mtcars)

#fit a regression model
model <- lm(mpg~disp+hp, data=mtcars)

#view model summary
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 model summary confirms that both displacement and horsepower are negatively associated with fuel efficiency, which aligns with common engineering knowledge. The adjusted R-squared

value of 0.7309 indicates that approximately 73% of the variance in `mpg` is explained by the predictors. However, this overall statistical fit does not confirm the absence of influential observations; therefore, the next critical step involves calculating the specific influence of each individual data point.

Step 2: Calculating the Leverage for each Observation

The standard function used in R to obtain the hat values, which represent the measure of leverage, is `hatvalues()`. These hat values (denoted mathematically as h_{ii}) are derived from the Hat Matrix ($mathbf{H}$), which projects the response vector onto the column space of the design matrix. The diagonal elements of this matrix are the leverage statistics we are interested in. The higher the hat value, the greater the leverage an observation has on the fitted values of the model.

By applying `hatvalues()` to our fitted `model` object, we obtain a vector containing the leverage score for every observation in the mtcars dataset. It is beneficial to convert this output into a data frame for easier viewing and sorting, especially when dealing with smaller datasets where specific observations can be easily identified by their row names (which correspond to the car models).

#calculate leverage for each observation in the model
hats <- as.data.frame(hatvalues(model))

#display leverage stats for each observation
hats

hatvalues(model)
Mazda RX4 0.04235795
Mazda RX4 Wag 0.04235795
Datsun 710 0.06287776
Hornet 4 Drive 0.07614472
Hornet Sportabout 0.08097817
Valiant 0.05945972
Duster 360 0.09828955
Merc 240D 0.08816960
Merc 230 0.05102253
Merc 280 0.03990060
Merc 280C 0.03990060
Merc 450SE 0.03890159
Merc 450SL 0.03890159
Merc 450SLC 0.03890159
Cadillac Fleetwood 0.19443875
Lincoln Continental 0.16042361

Chrysler Imperial 0.12447530
Fiat 128 0.08346304
Honda Civic 0.09493784
Toyota Corolla 0.08732818
Toyota Corona 0.05697867
Dodge Challenger 0.06954069
AMC Javelin 0.05767659
Camaro Z28 0.10011654
Pontiac Firebird 0.12979822
Fiat X1-9 0.08334018
Porsche 914-2 0.05785170
Lotus Europa 0.08193899
Ford Pantera L 0.13831817
Ferrari Dino 0.12608583
Maserati Bora 0.49663919
Volvo 142E 0.05848459

The resulting output provides the leverage value for all 32 observations. We can immediately observe that the scores vary significantly, ranging from approximately 0.0389 up to 0.4966. The highest value belongs to the "Maserati Bora," suggesting its combination of `disp` and `hp` is the most unusual or extreme among all vehicles in this dataset. Analyzing this distribution of leverage is essential before proceeding to the interpretive phase.

Interpreting the Leverage Threshold and Identifying High-Leverage Observations

After calculating the hat values, the next crucial step is determining which observations qualify as having "high leverage." While there is no single, universally agreed-upon threshold, a common rule of thumb suggests investigating any observation where the leverage value (h_{ii}) exceeds $2p/n$ or $3p/n$, where p is the number of parameters in the model (including the intercept) and n is the number of observations. In our specific model using the mtcars dataset, we have $n=32$ observations and $p=3$ parameters (Intercept, `disp`, and `hp`).

Applying the conventional $2p/n$ rule: $2 times 3 / 32 approx 0.1875$. Using the stricter $3p/n$ rule: $3 times 3 / 32 approx 0.28125$. Based on these guidelines, any observation with a leverage value significantly greater than 0.1875 warrants careful examination, as its presence in the dataset could potentially distort the regression coefficients. Although the original text referenced a threshold of 2, this is generally incorrect for the scaled hat values (which typically range from $1/n$ to 1); for hat values, the $2p/n$ rule is the statistically sound approach.

To systematically identify the highest leverage points, we should sort the observations in descending order of their hat values. This allows us to quickly pinpoint the cars that are spatially furthest from the center of the predictor data.

#sort observations by leverage, descending
hats),]

0.49663919 0.19443875 0.16042361 0.13831817 0.12979822 0.12608583
0.12447530 0.10011654 0.09828955 0.09493784 0.08816960 0.08732818
0.08346304 0.08334018 0.08193899 0.08097817 0.07614472 0.06954069
0.06287776 0.05945972 0.05848459 0.05785170 0.05767659 0.05697867
0.05102253 0.04235795 0.04235795 0.03990060 0.03990060 0.03890159
0.03890159 0.03890159

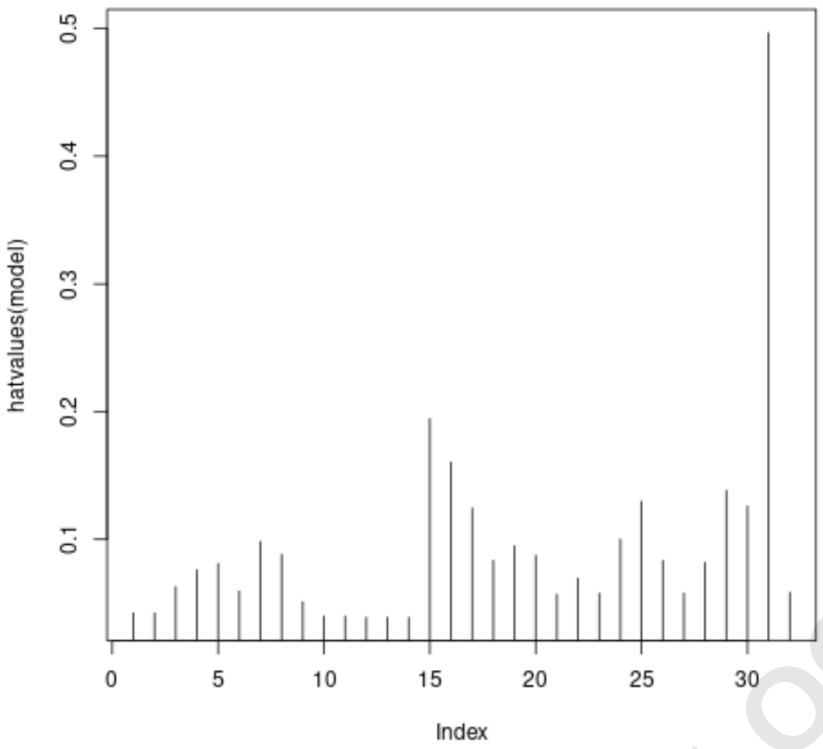
By examining the sorted output, we can definitively identify the observations that exceed our critical threshold of 0.1875. The "Maserati Bora" stands out dramatically with a leverage value of 0.4966. This is significantly higher than the mean leverage ($p/n approx 0.09375$). The "Cadillac Fleetwood," with a score of 0.1944, also surpasses the $2p/n$ threshold. This indicates that these two vehicles possess predictor variable combinations (`disp` and `hp`) that are highly unusual compared to the rest of the sample, potentially making them highly influential in determining the slope of our regression model.

Step 3: Visualizing the Leverage for each Observation

While numerical sorting is precise, visualization offers an immediate and intuitive understanding of the distribution of leverage across the dataset. Creating a simple plot of the hat values allows us to quickly confirm which observations are pushing the boundaries of the predictor space. In R, we can use the base plotting function `plot()` combined with the `hatvalues()` output, specifying the type as 'h' (for histogram-like vertical lines).

This visualization is essential for demonstrating the relative magnitude of influence across the dataset. Typically, the visualization would benefit from having the critical threshold line ($2p/n$) overlaid, but even without it, the distinct height of the highest leverage point (Maserati Bora) is immediately apparent, confirming our numerical findings.

#plot leverage values for each observation
plot(hatvalues(model), type = 'h')



The resulting graph displays the index of each observation on the x-axis (from 1 to 32) and the corresponding leverage statistic on the y-axis. As seen in the generated plot, the largest vertical line corresponds to the observation at index 31 (Maserati Bora), clearly demonstrating its extreme position. This visual confirmation is vital for presenting diagnostic results, as it powerfully highlights the potential impact of specific data points.

Conclusion and Next Steps for Influential Observations

Calculating and assessing leverage is a mandatory procedure for ensuring the validity of a linear regression model. We successfully identified two high-leverage observations in the mtcars dataset based on the $2p/n$ threshold rule. While high leverage alone does not dictate that a point must be removed, it signals that the observation's predictor values are unique and that the corresponding regression line passes very close to that point.

The next logical step is to combine the leverage analysis with residual analysis to identify truly **influential observations** using metrics like Cook's Distance or DFFITS. These combined metrics measure how much the model coefficients would change if that specific point were removed. If an observation has high leverage and high influence, researchers must investigate the data point's source (e.g., potential measurement error) or consider performing a sensitivity analysis by fitting the model both with and without the point to assess the stability of the conclusions.

By mastering the use of the `hatvalues()` function in R and applying sound statistical thresholds

like $2p/n$, analysts can create more robust models that are less susceptible to distortion by extreme data points, leading to more reliable and trustworthy statistical inferences.