

Plot Multiple Linear Regression Results in R

Authored by
Mohammed looti

November 6, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Plot Multiple Linear Regression Results in R*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=11451>

When conducting [Simple Linear Regression](#) (SLR) analysis in [R](#), visualizing the model is straightforward. We are typically working with only a single predictor variable and a single [response variable](#). This two-dimensional structure allows us to easily plot the data points and overlay the calculated fitted regression line.

The Challenge of Visualizing Simple vs. Multiple Regression

The transition from a simple model to a multiple model introduces significant complexity for visualization. In simple linear regression, the model exists within a 2D plane, making the interpretation of the slope and intercept intuitive. However, once we introduce a second predictor, the model space becomes three-dimensional (a plane), and any subsequent predictors push the model into higher-dimensional space that cannot be directly plotted on a standard 2D graph.

This challenge means that standard scatter plots, which are highly effective for SLR, fail to capture the nuances of a [Multiple Linear Regression](#) (MLR) model. MLR models estimate coefficients that represent the relationship between a predictor and the response, **holding all other predictors constant**. This crucial "holding constant" element is lost when attempting a simple projection onto a 2D plane.

Therefore, specialized diagnostic plots are required to understand the individual contribution of each predictor variable within the context of the full MLR model. These plots help statisticians and data scientists assess linearity, identify influential observations, and confirm the direction and strength of the partial relationship for each independent variable.

Understanding Simple Linear Regression Visualization

Before diving into the complexities of MLR visualization, it is helpful to review the basic process for fitting and plotting an SLR model in R. This foundational knowledge highlights why a different approach is necessary when dealing with multiple predictors. The goal here is a simple, direct visualization of the relationship between x and y .

The following R code snippet demonstrates how to create a simple dataset, fit an SLR model using the `lm()` function, generate a scatterplot of the raw data, and subsequently add the fitted regression line using `abline()`. This process results in a clear, immediately interpretable 2D representation of the linear relationship.

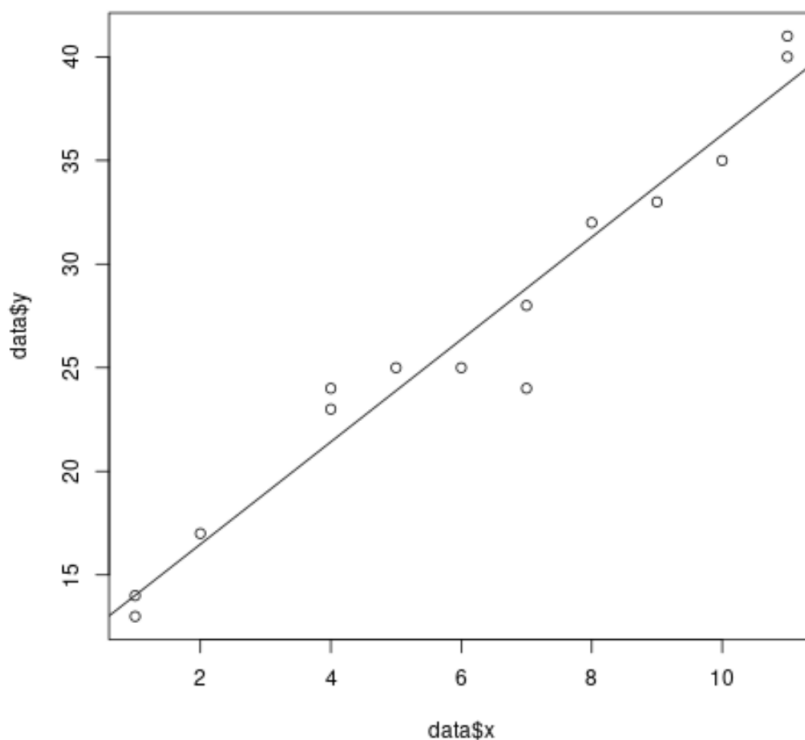
```
# Create example dataset for Simple Linear Regression  
data <- data.frame(x = c(1, 1, 2, 4, 4, 5, 6, 7, 7, 8, 9, 10, 11, 11),  
y = c(13, 14, 17, 23, 24, 25, 25, 24, 28, 32, 33, 35, 40, 41))  
  
# Fit the simple linear regression model: y predicted by x
```

```
model <- lm(y ~ x, data = data)

# Create a base scatterplot of the raw data points
plot(data$x, data$y)

# Add the fitted regression line derived from the model
abline(model)
```

The resulting plot clearly shows the linear trend across the data points, providing immediate confirmation of the model fit in the two-dimensional space. This visualization is straightforward because only one predictor is involved.



Introducing Added Variable Plots (Partial Regression Plots)

When we perform [Multiple Linear Regression](#), it becomes impossible to visualize the results directly using a single regression line because the model operates in a space defined by several predictor variables. Instead, we must use **Added Variable Plots** (AVPs), which are sometimes referred to as "partial regression plots."

These plots are individual visualizations that display the unique relationship between the response variable and one specific predictor variable, *while effectively controlling for the influence of all other predictor variables in the model*. The mathematical basis of an AVP involves plotting the residuals

of the response (when regressed against all other predictors) against the residuals of the target predictor (when regressed against all other predictors).

The primary benefit of using partial regression plots is their ability to isolate the marginal effect of each predictor. This isolation allows us to visually confirm that the relationship between the predictor and the response is linear after accounting for the effects of confounding variables. The slope of the line displayed in an added variable plot is exactly equal to the estimated coefficient for that predictor in the full multiple regression model.

The following example demonstrates how to perform multiple linear regression in R and visualize the results effectively using these essential diagnostic tools.

Practical Example: Plotting Multiple Linear Regression in R

Suppose we fit a Multiple Linear Regression model using the built-in `mtcars` dataset in R. We will model `mpg` (miles per gallon) as the response variable, predicted by `disp` (displacement), `hp` (horsepower), and `drat` (rear axle ratio). We first fit the model and view the statistical summary to understand the estimated coefficients.

Fit the Multiple Linear Regression model

```
model <- lm(mpg ~ disp + hp + drat, data = mtcars)
```

```
# View the detailed results and diagnostics of the model
summary(model)
```

Call:

```
lm(formula = mpg ~ disp + hp + drat, data = mtcars)
```

Residuals:

```
Min 1Q Median 3Q Max
```

```
-5.1225 -1.8454 -0.4456 1.1342 6.4958
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 19.344293 6.370882 3.036 0.00513 **
```

```
disp -0.019232 0.009371 -2.052 0.04960 *
```

```
hp -0.031229 0.013345 -2.340 0.02663 *
```

```
drat 2.714975 1.487366 1.825 0.07863 .
```

```
---
```

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

```
Residual standard error: 3.008 on 28 degrees of freedom
```

Multiple R-squared: 0.775, Adjusted R-squared: 0.7509
F-statistic: 32.15 on 3 and 28 DF, p-value: 3.28e-09

From the results, we can see that the p-values for each of the coefficients suggest a significant or near-significant relationship (less than 0.1). For the sake of demonstrating the visualization technique, we will assume that each of these predictor variables contributes meaningfully and should be included in the model.

To produce the necessary Added Variable Plots, we must rely on the `avPlots()` function, which is contained within the highly useful **car** package (Companion to Applied Regression). This package simplifies the generation of these complex diagnostic plots.

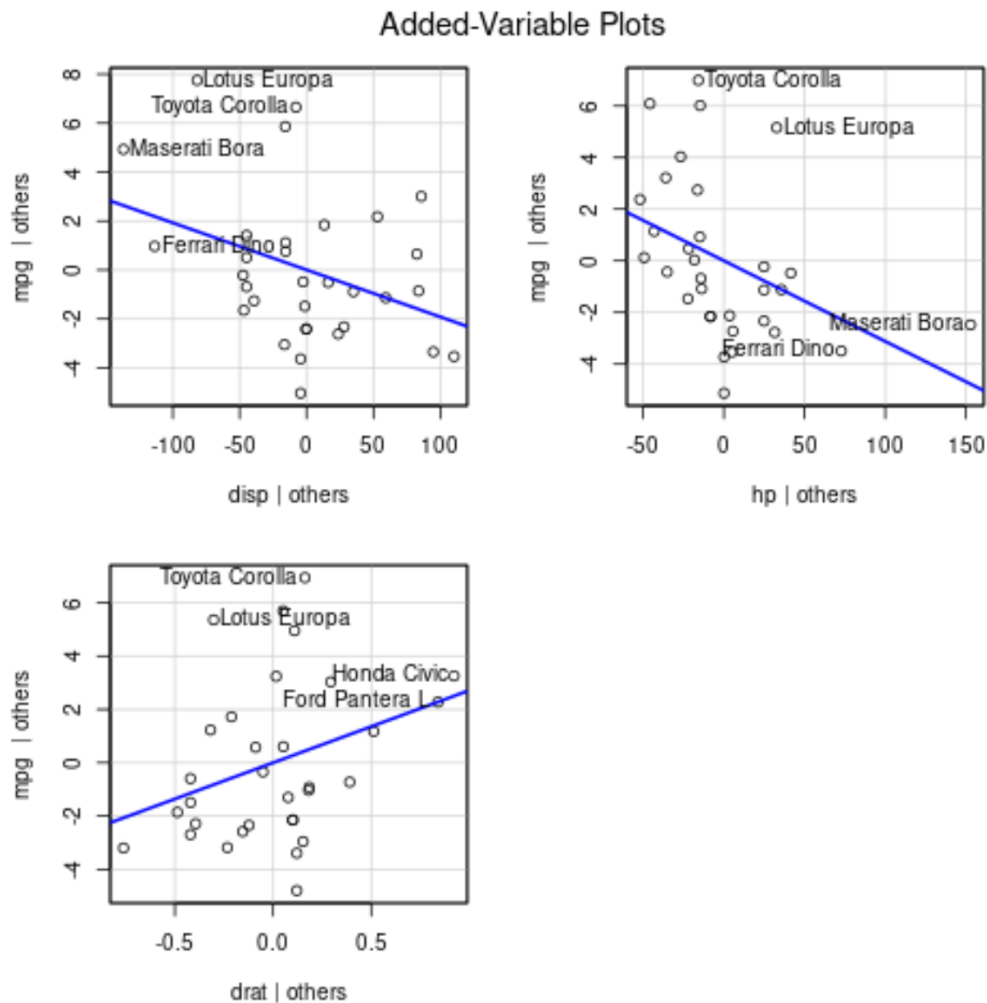
Load the specialized car package

```
library(car)
```

```
# Produce the set of added variable plots for the MLR model
```

```
avPlots(model)
```

The execution of this code yields a composite plot containing three distinct subplots, one for each predictor variable (disp, hp, and drat).



Interpreting the Output and Added Variable Plots

Interpreting these plots correctly is vital for validating the underlying assumptions of the regression model and understanding the partial effects. Each plot adheres to a specific structure designed to highlight the unique contribution of its corresponding predictor:

The x-axis displays the residuals of the predictor variable after controlling for all other predictors in the model.

The y-axis displays the residuals of the [response variable](#) after controlling for all other predictors.

The blue line shows the isolated association between the predictor variable and the response variable, *while holding the value of all other predictor variables constant*.

Furthermore, these plots are crucial for identifying influential observations. The points that are explicitly labeled in each plot typically represent the two observations with the largest Cook's distances and the two observations with the largest partial leverage. Identifying these [outliers](#) is essential for robust model building.

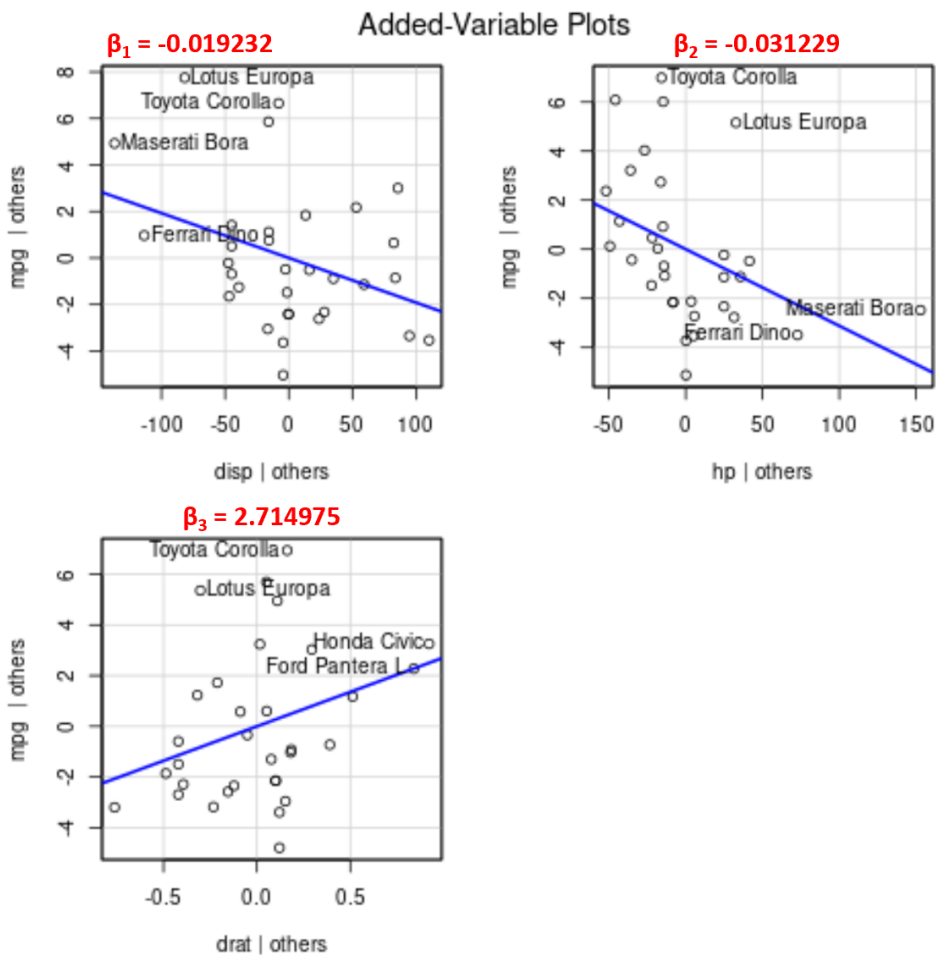
A key validation step is matching the visual slope of the line in the plot to the sign and magnitude of the estimated coefficient from the model summary. Recall the estimated coefficients:

disp: -0.019232

hp: -0.031229

drat: 2.714975

Notice how the angle of the line is positive (sloping upward) in the added variable plot for *drat*, perfectly aligning with its positive coefficient. Conversely, the lines for both *disp* and *hp* slope downward (negative angle), which confirms the negative signs of their estimated coefficients. This visual check provides strong evidence that the model is behaving as expected.



Conclusion: Leveraging Partial Plots for Model Insight

Although we cannot plot a single fitted regression line on a 2D plot when working with multiple predictor variables, Added Variable Plots allow us to observe the relationship between each individual predictor variable and the response variable while holding the effects of other predictor

variables constant. This ability to decompose the multivariate relationship is what makes these plots so powerful.

In summary, AVPs are indispensable tools in regression diagnostics. They allow analysts to:

Visually assess the linearity of the relationship for each predictor.

Confirm the direction (sign) and relative strength of the coefficient estimates.

Identify influential data points or [outliers](#) that might disproportionately affect the model fit.

By effectively utilizing the `avPlots()` function within the `car` package, researchers using [R](#) can gain significantly deeper insights into the performance and validity of their [Multiple Linear Regression](#) models.