Understanding the Leverage and Its Use in Adjusting Residuals of Linear Models

A Simulation Illustration with R

Longhai Li

2025-10-01

1 Introduction

In statistical modeling, identifying data points that don't fit—the **outliers**—is a critical step. The most reliable tool for this job is the **externally studentized residual**. Its power comes from a simple, intuitive idea: to judge a point fairly, you shouldn't use that point when building your model. This is the core principle of **Leave-One-Out Cross-Validation (LOOCV)**.

This article provides a complete walkthrough of this essential concept. We'll start with the basic linear model, introduce the necessary notation, explore the flaws of simpler residuals, and then formally define and prove the equivalence of the conceptual and computational formulas for studentized residuals. Finally, we'll make it all concrete with a simple example.

2 The Linear Model

Our discussion is based on the standard multiple linear regression model. In matrix form, the relationship between a response vector $\mathbf{Y}$ and a predictor matrix $\mathbf{X}$ is:

$$\mathbf{Y} = \mathbf{X} \ + \tag{1}$$

Where:

- $\mathbf{Y}$ is an n x 1 vector of the observed outcomes.
- $\mathbf{X}$ is the n x p design matrix of predictor variables (where p is the number of coefficients, including the intercept).
- β is the p x 1 vector of unknown true coefficients we want to estimate.
- ϵ is an n x 1 vector of unobservable random errors, assumed to be independent and identically distributed with a mean of 0 and a variance of σ^2.

3 Our Notations

To discuss models fit with all data versus those with one point removed, we need clear notation.

Full Data Model (Using all n observations)

- $\hat{\beta}$: The estimated coefficient vector.
- $\hat{y}_i$: The predicted value for observation i from this model.
- e_i: The **ordinary residual** ($e_i = y_i - \hat{y}_i$).
- $\hat{\sigma}$: The estimated standard deviation of the errors (Residual Standard Error).
- h_{ii}: The **leverage** of observation i, a measure of how much its x-values influence the model.

Leave-One-Out (LOOCV) Model

- $\hat{\beta}_{-i}$: The coefficient vector estimated after **removing** observation i.
- $\hat{y}_{i,-i}$: The predicted value for observation i, from the model fit **without** observation i.
- $e_{i,-i}$: The **deleted residual** ($e_{i,-i} = y_i - \hat{y}_{i,-i}$).
- $\hat{\sigma}_{-i}$: The standard deviation of the errors estimated from the model fit **without** observation i.

4 Non-studentized Residuals

Before getting to the correct solution, it's crucial to understand why simpler methods of standardizing residuals are flawed.

4.1 The Ordinary Residual (e_i): Too Small and x_i Dependent

The most basic residual, e_i, is problematic for two key reasons.

An outlier has an undue influence on the model, pulling the regression line towards itself. This makes its own predicted value, $\hat{y}_i$, artificially close to its actual value, y_i. As a result, its residual, e_i, is **deceptively small** and doesn't reflect the true magnitude of the error.

The variance of an ordinary residual is not constant; it depends on the point's leverage.

$$\text{Var}(e_i) = \sigma^2(1 - h_{ii}) \tag{2}$$

We know

$$\hat{y}_i = \mathbf{x}_i^\top \hat{\beta} = \mathbf{x}_i^\top(\beta + (X^\top X)^{-1} X^\top \epsilon). \tag{3}$$

So the residual is

$$e_i = y_i - \hat{y}_i = \epsilon_i - \mathbf{x}_i^\top (X^\top X)^{-1} X^\top \epsilon. \tag{4}$$

The variance can be derived from the hat matrix $H = X(X^\top X)^{-1}X^\top$. Since $\hat{y} = HY$, we have

$$e = (I - H)\epsilon. \tag{5}$$

Thus,

$$\mathrm{Var}(e) = (I - H)\,\sigma^2\,(I - H)^\top = (I - H)\sigma^2. \tag{6}$$

Therefore, for the ith residual,

$$\mathrm{Var}(e_i) = \sigma^2(1 - h_{ii}). \tag{7}$$

Since leverage (h_{ii}) is always greater than 0, the variance of an ordinary residual is always **less than the true error variance,** σ^2. High-leverage points act as "anchors" for the line and have even smaller variance.

4.2 The LOOCV Residual ($e_{i,-i}$): Too large and x_i-Dependent Variance

The deleted residual, $e_{i,-i}$, solves the "too small" problem. Because the model isn't influenced by the point it's predicting, the residual is an honest measure of prediction error. However, its variance is still not constant. The variance of a deleted residual also depends on leverage, but in the opposite way.

$$\mathrm{Var}(e_{i,-i}) = \frac{\sigma^2}{1 - h_{ii}} \tag{8}$$

From the key identity Equation 17,

$$e_{i,-i} = \frac{e_i}{1 - h_{ii}}. \tag{9}$$

Therefore,

$$\mathrm{Var}(e_{i,-i}) = \frac{\mathrm{Var}(e_i)}{(1 - h_{ii})^2} = \frac{\sigma^2(1 - h_{ii})}{(1 - h_{ii})^2} = \frac{\sigma^2}{1 - h_{ii}}. \tag{10}$$

Since $1 - h_{ii}$ is less than 1, the variance of a deleted residual is always **greater than the true error variance,** σ^2. This is because it has two sources of randomness: the error in the point itself (y_i) and the error in the prediction ($\hat{y}_{i,-i}$).

5 Studentized Residuals

5.1 Studentized LOOCV (Deleted) Residual

The correct solution is to take the LOOCV residual and divide it by its true standard error, which properly accounts for its larger, x-dependent variance. This is the **externally studentized residual**, t_i, defined as follows:

$$t_i = \frac{e_{i,-i}}{\text{SE}(e_{i,-i})} = \frac{e_{i,-i}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{11}$$

This final value is a reliable diagnostic. Under the null hypothesis that the observation is not an outlier, it follows a **Student's t-distribution** with $n - p - 1$ degrees of freedom.

5.2 Studentized Full Data Residuals

Calculating the conceptual formula appears to require fitting n different regression models—a computationally expensive task. Fortunately, a mathematical identity allows us to calculate the exact same value using only the results from the single, full data model.

$$t_i = \frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}} \tag{12}$$

This is not an approximation; it is an **exact algebraic rearrangement** of the conceptual definition.

5.3 Equivalence of Equation 12 and Equation 11

5.3.1 Proof of Equivalence

Let's start with the conceptual definition of the studentized LOOCV residuals Equation 11 and show how it transforms into Equation 12.

- **Start with the conceptual LOOCV definition:**

$$t_i = \frac{e_{i,-i}}{\text{SE}(e_{i,-i})} = \frac{e_{i,-i}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{13}$$

- **Substitute the key identity** into the numerator:

$$t_i = \frac{\frac{e_i}{1-h_{ii}}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{14}$$

- **Simplify the complex fraction.** We can do this by multiplying the numerator by the reciprocal of the denominator:

$$t_i = \frac{e_i}{1-h_{ii}} \cdot \frac{\sqrt{1-h_{ii}}}{\hat{\sigma}_{-i}} \tag{15}$$

- **Cancel the terms.** Since $1-h_{ii} = (\sqrt{1-h_{ii}})^2$, one of the $\sqrt{1-h_{ii}}$ terms in the denominator cancels with the term in the numerator. This leaves us with the computational shortcut formula:

$$t_i = \frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}} \tag{16}$$

This proves that the two formulas are mathematically identical. The computational shortcut is simply a clever algebraic rearrangement of the more intuitive LOOCV definition, allowing for efficient and accurate calculation.

6 List of Residuals

In this article, we will compare the four residuals given as:

Table 1

Short Name	Full Name	Formula
NS-Full	Non-studentized Full-Data Residual	$\frac{e_i}{\hat{\sigma}}$
NS-LOO	Non-studentized LOOCV Residual	$\frac{e_{i,-i}}{\hat{\sigma}_{-i}}$
ST-LOO	Studentized LOOCV Residual	$\frac{e_{i,-i}}{\hat{\sigma}_{-i}/\sqrt{1-h_{ii}}}$
ST-Full	Studentized Full-Data Residual	$\frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}}$

7 Example

7.1 The Linear Model

The simulation uses a **simple linear regression model** to describe the relationship between a single predictor variable, x_i, and a response variable, y_i. The underlying "true" model from which the data is generated is:

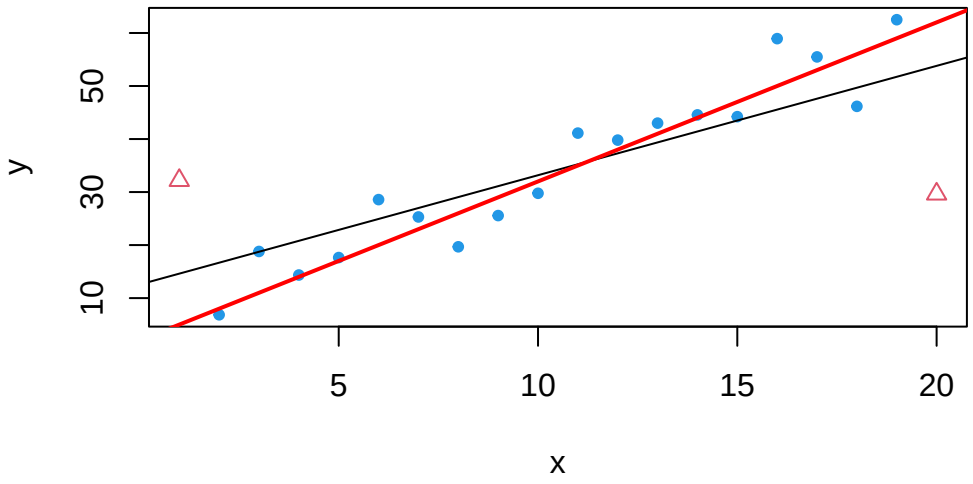
$$y_i = 2 + 3x_i + \epsilon_i$$

This means we have a true intercept of 2, a true slope of 3, and a random error term, ϵ_i, drawn from a normal distribution with a mean of 0 and a standard deviation of 5. Two artificial outliers are added to this data to test the behavior of the different residual types.

```r
# Load libraries
library(dplyr)
library(knitr)

# ------------------------------
# 1) Data and full-model fit
# ------------------------------
set.seed(123)
n <- 20
x <- 1:n
y <- 2 + 3 * x + rnorm(n, mean = 0, sd = 5)
y[1] <- y[1] + 30  # add an outlier at x = 6
y[20] <- y[20] - 30  # add an outlier at x = 11
flag.outlier <- c(2, rep(20, n-2),2)
full_data  <- data.frame(x = x, y = y)

plot(y~x, data=full_data, pch= flag.outlier, col=flag.outlier)
abline(lm(y~x, data=full_data))
abline (a=2, b=3, col="red", lwd=2)
```



7.2 Description of Calculated Columns

The final table compiles several important quantities calculated during the simulation. Here's what each column represents:

- x_i: The predictor variable, which is simply the index of the observation from 1 to 20.
- h_i: The **leverage** of the i-th observation. It measures how influential a point's x-value is in determining the model's fit. A higher value indicates a more influential point.

- e_i: The **ordinary residual**, calculated as the difference between the actual value (y_i) and the predicted value ($\hat{y}_i$) from the model fit on all data.
- $\hat{\sigma}$: The **residual standard error** (or Root Mean Square Error) of the full model, representing the typical size of an ordinary residual.
- $e_{i,-i}$: The **deleted (or LOOCV) residual**. This is the difference between the actual value (y_i) and the value predicted for it by a model that was fit on all other data *except* point i.
- $\hat{e}_{i,-i}$: This column shows the deleted residual calculated using the efficient algebraic shortcut ($e_i/(1-h_{ii})$), verifying it's identical to the brute-force $e_{i,-i}$.
- $\hat{\sigma}_{-i}$: The **LOOCV residual standard error**, calculated from a model that was fit after removing observation i.
- $\tilde{\sigma}_{-i}$: The **LOOCV residual standard error**, calculated from the shortcut formula Equation 18.
- **NS-Full**: The **Non-studentized Full-Data Residual**, calculated as the ordinary residual divided by the full model's standard error ($e_i/\hat{\sigma}$).
- **NS-LOO**: The **Non-studentized LOOCV Residual**, calculated as the deleted residual divided by the corresponding LOOCV standard error ($e_{i,-i}/\hat{\sigma}_{-i}$).
- **ST-LOO**: The **Studentized LOOCV Residual**, calculated using the conceptual formula by dividing the deleted residual by its true standard error.
- **ST-Full**: The **Studentized Full-Data Residual**, calculated using the efficient shortcut formula, which is provided by R's `rstudent()` function.

```
library(kableExtra)

full_model <- lm(y ~ x, data = full_data)
p <- length(coef(full_model))
leverage   <- hatvalues(full_model)
e_full     <- resid(full_model)
sigma_hat_val  <- summary(full_model)$sigma

rss_full <- sum(e_full^2)
df_loo <- n - p - 1
sigma_minus_i_shortcut <- sqrt((rss_full - (e_full^2 / (1 - leverage))) / df_loo)

# ------------------------------
# 2) LOOCV quantities (refit n times)
# ------------------------------
e_del_val <- numeric(n)
sigma_minus_i_val <- numeric(n)

for (i in 1:n) {
  loocv_model <- lm(y ~ x, data = full_data[-i, ])
  yhat_minus  <- predict(loocv_model, newdata = full_data[i, , drop = FALSE])
```

```r
  e_del_val[i]    <- full_data$y[i] - yhat_minus
  sigma_minus_i_val[i] <- summary(loocv_model)$sigma
}

# ------------------------------
# 3) Assemble and round results
# ------------------------------
residuals_df <- data.frame(
  x = full_data$x,
  h = as.numeric(leverage),
  e_i = as.numeric(e_full),
  sigma_hat = as.numeric(sigma_hat_val),
  e_i_minus_i = as.numeric(e_del_val),
  e_i_minus_i_2 = e_full/(1-leverage),
  sigma_minus_i = as.numeric(sigma_minus_i_val),
  sigma_minus_i_shortcut = as.numeric(sigma_minus_i_shortcut),
  `NS-Full` = e_full / sigma_hat_val,
  `NS-LOO` = e_del_val / sigma_minus_i_val,
  `ST-LOO` = e_del_val / (sigma_minus_i_val / sqrt(1 - leverage)),
  `ST-Full` = rstudent(full_model)
) %>%
  mutate(across(where(is.numeric), ~ round(.x, 3)))


# 4) Create simple display names
display_names <- c("$x_i$", "$h_i$", "$e_i$", "$\\hat{\\sigma}$",
                   "$e_{i,-i}$", "$\\hat{e}_{i,-i}$",
                   "$\\hat{\\sigma}_{-i}$", "$\\tilde{\\sigma}_{-i}$",
                   "NS-Full", "NS-LOO", "ST-LOO", "ST-Full")

# 5) Display the table
# Conditional check for output format
if (knitr::is_html_output()) {
  # --- Code for HTML Output (using kableExtra) ---
  knitr::kable(
    residuals_df,
    caption = "Residual variants",
    col.names = display_names,
    align = "r",
    #format="html",
    escape = FALSE # Allows LaTeX and <br/> to render
  )
```

Table 2: Residual variants.

x_i	h_i	e_i	$\hat{\sigma}$	$e_{i,-i}$	$\hat{e}_{i,-i}$	$\hat{\sigma}_{-i}$	$\tilde{\sigma}_{-i}$	NS-Full	NS-LOO	ST-LOO	ST-Full
1	0.186	17.585	9.542	21.595	21.595	8.606	8.606	1.843	2.509	2.264	2.264
2	0.159	-9.827	9.542	-11.680	-11.680	9.468	9.468	-1.030	-1.234	-1.131	-1.131
3	0.135	0.055	9.542	0.064	0.064	9.818	9.818	0.006	0.006	0.006	0.006
4	0.114	-6.448	9.542	-7.274	-7.274	9.677	9.677	-0.676	-0.752	-0.708	-0.708
5	0.095	-5.217	9.542	-5.768	-5.768	9.728	9.728	-0.547	-0.593	-0.564	-0.564
6	0.080	3.649	9.542	3.968	3.968	9.775	9.775	0.382	0.406	0.389	0.389
7	0.068	-1.684	9.542	-1.808	-1.808	9.809	9.809	-0.177	-0.184	-0.178	-0.178
8	0.059	-9.377	9.542	-9.969	-9.969	9.534	9.534	-0.983	-1.046	-1.014	-1.014
9	0.053	-5.548	9.542	-5.861	-5.861	9.720	9.720	-0.581	-0.603	-0.587	-0.587
10	0.050	-3.405	9.542	-3.586	-3.586	9.782	9.782	-0.357	-0.367	-0.357	-0.357
11	0.050	5.881	9.542	6.193	6.193	9.709	9.709	0.616	0.638	0.622	0.622
12	0.053	2.497	9.542	2.638	2.638	9.799	9.799	0.262	0.269	0.262	0.262
13	0.059	3.639	9.542	3.869	3.869	9.776	9.776	0.381	0.396	0.384	0.384
14	0.068	3.126	9.542	3.356	3.356	9.787	9.787	0.328	0.343	0.331	0.331
15	0.080	0.731	9.542	0.795	0.795	9.817	9.817	0.077	0.081	0.078	0.078
16	0.095	13.382	9.542	14.795	14.795	9.206	9.206	1.402	1.607	1.528	1.528
17	0.114	7.874	9.542	8.882	8.882	9.606	9.606	0.825	0.925	0.871	0.871
18	0.135	-3.511	9.542	-4.057	-4.057	9.776	9.776	-0.368	-0.415	-0.386	-0.386
19	0.159	10.766	9.542	12.796	12.796	9.397	9.397	1.128	1.362	1.249	1.249
20	0.186	-24.167	9.542	-29.679	-29.679	7.363	7.363	-2.533	-4.031	-3.638	-3.638

```
} else {
  # --- Code for PDF/Other Output (using kableExtra) ---
  knitr::kable(
    residuals_df,
    caption = "Residual variants.",
    col.names = display_names,
    align = "r",
    format = "latex",
    booktabs = TRUE,
    escape = FALSE # Allows LaTeX and \\ to render
  ) %>%
    kable_styling(
      latex_options = "scale_down"
    )
}
```

```
# Load libraries
library(dplyr)
```

```r
library(tidyr)
library(ggplot2)
library(knitr)


# --------------------------------
# 3) Plotting Code with Updated Names
# --------------------------------

# Prepare data for plotting
plot_df <- residuals_df %>%
  # Use the new, simple column names (R converts '-' to '.')
  select(
    x,
    NS.Full,
    NS.LOO,
    ST.LOO,
    ST.Full
  ) %>%
  pivot_longer(
    cols = -x,
    names_to = "residual_type",
    values_to = "residual_value"
  )

# Update the names in the mapping vectors
shape_map <- c(
  NS.Full = 16,  # solid circle
  NS.LOO  = 1,   # hollow circle
  ST.LOO  = 6,   # asterisk
  ST.Full = 10    # asterisk
)

labels_map <- c(
  NS.Full = "NS-Full",
  NS.LOO  = "NS-LOO",
  ST.LOO  = "ST-LOO",
  ST.Full = "ST-Full"
)

color_map <- c(
  NS.Full = "#1f77b4",  # blue
```

```
  NS.LOO  = "#ff7f0e", # orange
  ST.LOO  = "#2ca02c",  # green
  ST.Full = "#d62728"   # red
)

# Generate the plot
ggplot(
  plot_df,
  aes(x = x, y = residual_value,
      shape = residual_type, color = residual_type, group = residual_type)
) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(size = 3, stroke = 1.2) + # Increased stroke for visibility
  scale_shape_manual(
    values = shape_map,
    breaks = names(labels_map),
    labels = unname(labels_map),
    name = "Residual Type"
  ) +
  scale_color_manual(
    values = color_map,
    breaks = names(labels_map),
    labels = unname(labels_map),
    name = "Residual Type"
  ) +
  labs(
    title = "Four Residual Variants vs x",
    x = "x_i",
    y = "Residual value"
  ) +
  theme_bw() +
  theme(
    legend.position = "right",
    legend.title = element_text(face = "bold")
  )
```

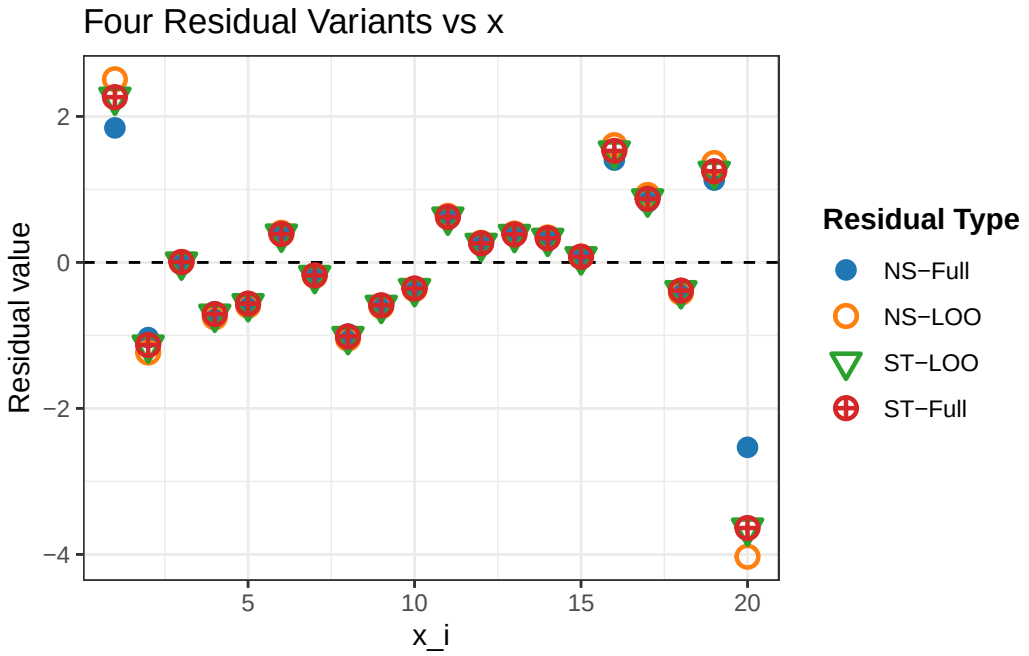


Figure 1: Four residual variants plotted against the predictor variable x.

From the above simulation results, we observe the following important facts:

- **Leverage and Influence:** The simulation confirms that leverage (h_{ii}) measures an observation's influence on the model's coefficients. It shows that points with higher leverage pull the regression line toward them, resulting in smaller, deceptively conservative full-data residuals (e_i).
- **Conservative Residuals:** The study highlights that the ordinary residual (e_i) is a "conservative" measure of error because its value for an outlier is systematically reduced by that same outlier's influence on the model.
- **Identity Verification:** The numerical results validated the key algebraic identity that connects the full-data residual (e_i) to the leave-one-out (deleted) residual ($e_{i,-i}$), as well as the identity for calculating the LOOCV standard error ($\hat{\sigma}_{-i}$) from the full model's statistics. This demonstrates that all key LOOCV errors can be calculated efficiently from a single model fit.
- **Effective Studentization:** The final step of studentization, which uses leverage to properly scale the residuals, is shown to be crucial. It successfully transforms the residuals into a reliable diagnostic tool with a constant variance across all predictor values (x_i), causing them to behave much more like a standard normal or t-distribution.

Appendix: Key Identities for Efficient Calculation of LOOCV Residuals and Noice Variance

The power of modern regression diagnostics comes from algebraic shortcuts that allow us to find the results of a leave-one-out process without the computational cost of refitting the model n times. The following two identities are fundamental to this efficiency.

7.1 Finding the LOOCV Residual ($e_{i,-i}$) from the Ordinary Residual (e_i)

This identity shows that we can find the "pure" leave-one-out residual using only the results from the single model fit on all data.

$$e_{i,-i} = \frac{e_i}{1 - h_{ii}} \tag{17}$$

7.2 Finding the LOOCV Standard Error ($\hat{\sigma}_{-i}$) from the Full-Model Standard Error ($\hat{\sigma}$)

Similarly, this formula provides an efficient shortcut to see how the model's overall error changes when a single point is removed.

$$\hat{\sigma}_{-i} = \sqrt{\frac{(n-p)\hat{\sigma}^2 - \frac{e_i^2}{1-h_{ii}}}{n-p-1}} \tag{18}$$

The derivation of this formula relies on first proving the relationship between the full model's Residual Sum of Squares (RSS) and the leave-one-out version (RSS_{-i}).

1. **Start with the definition** of the leave-one-out residual sum of squares:

$$RSS_{-i} = \sum_{k \neq i} (y_k - \mathbf{x}_k^T \hat{\beta}_{-i})^2$$

2. **Introduce the key identity** that relates the leave-one-out coefficient vector ($\hat{\beta}_{-i}$) to the full model's coefficient vector ($\hat{\beta}$):

$$\hat{\beta}_{-i} = \hat{\beta} - (X^T X)^{-1} \mathbf{x}_i \frac{e_i}{1 - h_{ii}}$$

3. **Substitute this identity** into the expression for a generic leave-one-out residual, $e_{k,-i} = y_k - \mathbf{x}_k^T \hat{\beta}_{-i}$. After simplification, this yields:

$$e_{k,-i} = e_k + h_{ki} \frac{e_i}{1 - h_{ii}}$$

where e_k is the ordinary residual and h_{ki} is the (k, i)-th element of the hat matrix.

4. **Substitute this back into the definition of** RSS_{-i}. After expanding the squared term and performing the summation (which involves considerable but standard matrix algebra), the expression simplifies to the elegant result:

$$RSS_{-i} = RSS - \frac{e_i^2}{1 - h_{ii}}$$

5. **Finally, derive the formula for** $\hat{\sigma}_{-i}$. We know that $\hat{\sigma}_{-i}^2 = \frac{RSS_{-i}}{n-p-1}$ and that $RSS = (n - p)\hat{\sigma}^2$. By substituting the result from Step 4, we arrive at the formula for the variance, and taking the square root gives us the standard error.