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

A Simulation Illustration with R

Longhai Li

2025-10-06

Table of contents

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:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

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

$$\hat{y} = HY,$$

we have

$$e = (I - H)\epsilon. \tag{1}$$

Thus,

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

Therefore, for the ith residual,

$$\mathrm{Var}(e_i) = \sigma^2(1 - h_{ii}). \tag{3}$$

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.

$$\text{Var}(e_{i,-i}) = \frac{\sigma^2}{1-h_{ii}} \tag{4}$$

From the key identity Equation 13,

$$e_{i,-i} = \frac{e_i}{1-h_{ii}}. \tag{5}$$

Therefore,

$$\text{Var}(e_{i,-i}) = \frac{\text{Var}(e_i)}{(1-h_{ii})^2} = \frac{\sigma^2(1-h_{ii})}{(1-h_{ii})^2} = \frac{\sigma^2}{1-h_{ii}}. \tag{6}$$

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:

$$t_i = \frac{e_{i,-i}}{\text{SE}(e_{i,-i})} = \frac{e_{i,-i}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{7}$$

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.

$$t_i = \frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}} \tag{8}$$

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

5.3 Equivalence of Equation 8 and Equation 7

5.3.1 Proof of Equivalence

Let's start with the conceptual definition of the studentized LOOCV residuals Equation 7 and show how it transforms into Equation 8.

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

$$t_i = \frac{e_{i,-i}}{\text{SE}(e_{i,-i})} = \frac{e_{i,-i}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{9}$$

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

$$t_i = \frac{\frac{e_i}{1-h_{ii}}}{\frac{\hat{\sigma}_{-i}}{\sqrt{1-h_{ii}}}} \tag{10}$$

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

$$t_i = \frac{e_i}{1-h_{ii}} \cdot \frac{\sqrt{1-h_{ii}}}{\hat{\sigma}_{-i}} \tag{11}$$

- **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:

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

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.

Of course. Here is the modified `.qmd` file with the **standardized residual** (which you've labeled **STD-Full**) added to the table, the descriptions, and the plot.

The main changes include:

1. Adding the `STD-Full` column to the `residuals_df` data frame using R's `rstandard()` function.
2. Updating the list of calculated columns to include a description of **STD-Full**.
3. Modifying the plotting code to include **STD-Full** with its own distinct color and shape.

---

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	$\frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}}$
STD-Full	Standardized (Internal Studentized) Residual	$\frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$

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:

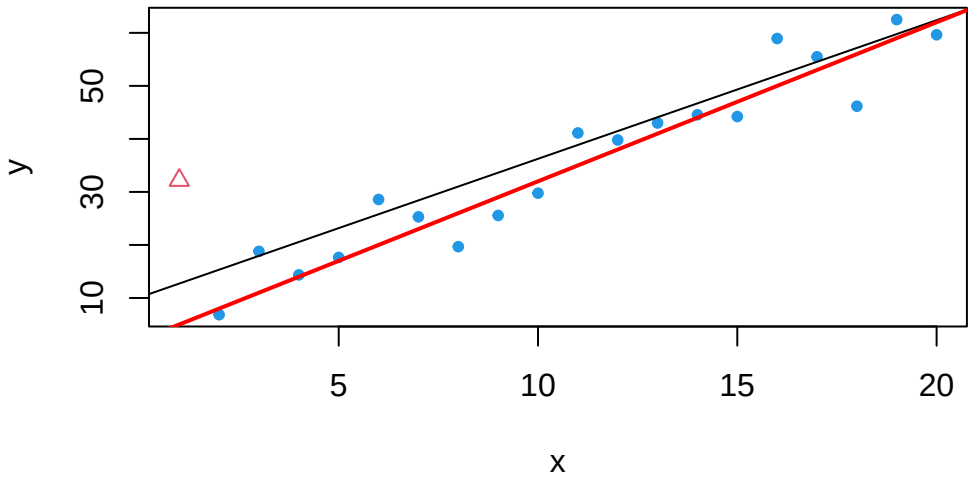
$$y_i = 2 + 3x_i + \epsilon_i$$

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. One artificial outlier is added to this data to test the behavior of the different residual types. 5 unrelated predictors are added to the dataset.

```
# 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
#y[11] <- y[11] - 30
o.index <- c(1)
flag.outlier <- rep(20, n)
flag.outlier[o.index] <- 2
full_data  <- data.frame(x = x, replicate(5, rnorm (n)), y = y)

plot(y~x, data=full_data, pch= flag.outlier, col=flag.outlier)
fit <- lm(y~., data=full_data)
abline (fit)
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 **?@eq-sigma_**-i.
- **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 ($e_{i,-i}/\hat{\sigma}_{-i}$).
- **STD-Full**: The **Standardized (or Internally Studentized) Residual**, calculated as the ordinary residual divided by its estimated standard error ($e_i/(\hat{\sigma}\sqrt{1-h_{ii}})$). This is provided by R's `rstandard()` function.

- **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 ~ ., 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 ~ ., data = full_data[-i, ])
  yhat_minus  <- predict(loocv_model, newdata = full_data[i, , drop = FALSE])
  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,
  `STD-Full` = rstandard(full_model), # <-- ADDED STANDARDIZED RESIDUAL
  `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", "STD-Full", "ST-LOO", "ST-Full") # <-- ADDED LABEL

# 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
  )
} 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"
    )
}
```

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	STD-Full	ST-LOO	ST-Full
1	0.247	17.638	7.57	23.434	23.434	5.257	5.257	2.330	4.457	2.686	3.867	3.867
2	0.241	-7.909	7.57	-10.418	-10.418	7.431	7.431	-1.045	-1.402	-1.199	-1.222	-1.222
3	0.364	-3.271	7.57	-5.147	-5.147	7.790	7.790	-0.432	-0.661	-0.542	-0.527	-0.527
4	0.555	1.553	7.57	3.490	3.490	7.851	7.851	0.205	0.445	0.308	0.297	0.297
5	0.257	-1.515	7.57	-2.038	-2.038	7.863	7.863	-0.200	-0.259	-0.232	-0.223	-0.223
6	0.400	-0.400	7.57	-0.666	-0.666	7.878	7.878	-0.053	-0.085	-0.068	-0.066	-0.066
7	0.253	0.247	7.57	0.331	0.331	7.879	7.879	0.033	0.042	0.038	0.036	0.036
8	0.303	-8.972	7.57	-12.871	-12.871	7.243	7.243	-1.185	-1.777	-1.419	-1.484	-1.484
9	0.347	-8.394	7.57	-12.852	-12.852	7.287	7.287	-1.109	-1.764	-1.372	-1.425	-1.425
10	0.573	-5.696	7.57	-13.342	-13.342	7.467	7.467	-0.752	-1.787	-1.152	-1.168	-1.168
11	0.117	7.175	7.57	8.127	8.127	7.565	7.565	0.948	1.074	1.009	1.009	1.009
12	0.441	0.998	7.57	1.786	1.786	7.870	7.870	0.132	0.227	0.176	0.170	0.170
13	0.359	1.298	7.57	2.026	2.026	7.865	7.865	0.171	0.258	0.214	0.206	0.206
14	0.395	0.698	7.57	1.153	1.153	7.875	7.875	0.092	0.146	0.119	0.114	0.114
15	0.228	-1.221	7.57	-1.583	-1.583	7.869	7.869	-0.161	-0.201	-0.184	-0.177	-0.177
16	0.402	7.998	7.57	13.365	13.365	7.292	7.292	1.057	1.833	1.366	1.418	1.418
17	0.433	3.996	7.57	7.048	7.048	7.729	7.729	0.528	0.912	0.701	0.687	0.687
18	0.414	-3.369	7.57	-5.746	-5.746	7.776	7.776	-0.445	-0.739	-0.581	-0.566	-0.566
19	0.258	3.073	7.57	4.141	4.141	7.812	7.812	0.406	0.530	0.471	0.457	0.457
20	0.414	-3.930	7.57	-6.707	-6.707	7.739	7.739	-0.519	-0.867	-0.678	-0.663	-0.663

```r
# Load libraries
library(dplyr)
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,
    STD.Full, # <-- ADDED FOR PLOTTING
    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
  STD.Full = 2,   # hollow triangle <-- ADDED
  ST.LOO   = 6,   # asterisk
  ST.Full  = 10   # asterisk
)

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

color_map <- c(
  NS.Full  = "#1f77b4",  # blue
  NS.LOO   = "#ff7f0e", # orange
  STD.Full = "#9467bd",  # purple <-- ADDED
  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 = "Five Residual Variants vs x",
  x = "x_i",
  y = "Residual value"
) +
theme_bw() +
theme(
  legend.position = "right",
  legend.title = element_text(face = "bold")
)
```

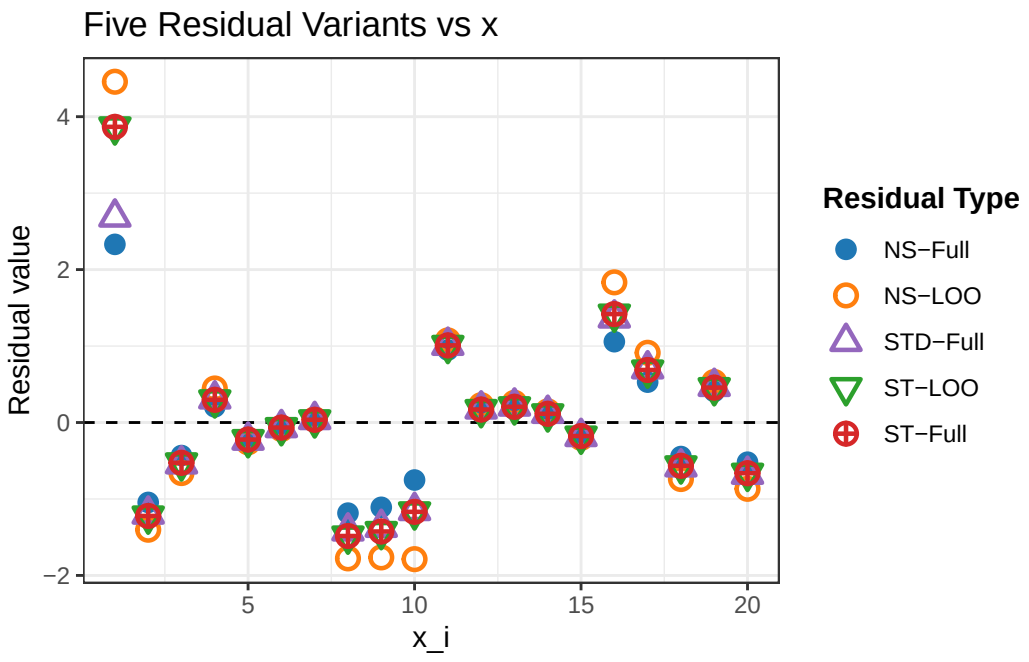


Figure 1: Five residual variants plotted against the predictor variable x.

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.

8 Cook's Distance

8.1 Definition from the change in coefficients

Let $\hat{\beta}$ be the OLS estimate on all n cases and $\hat{\beta}_{(-i)}$ the estimate after deleting case i.
With p parameters (including intercept) and $\hat{\sigma}^2 = \text{MSE}$,

$$D_i = \frac{\left(\hat{\beta} - \hat{\beta}_{(-i)}\right)^\top (X^\top X) \left(\hat{\beta} - \hat{\beta}_{(-i)}\right)}{p\,\hat{\sigma}^2}.$$

This measures how far the whole coefficient vector moves (in the $X^\top X$ metric) when case i is removed, scaled **per parameter**.

8.2 Express D_i via the studentized LOOCV residual: t_i

Let h_{ii} be the leverage and define the LOOCV quantities $e_{i,-i}$ and $\hat{\sigma}_{-i}$ from the model refit **without** case i.
The externally studentized residual is

$$t_i = \frac{e_i}{\hat{\sigma}_{-i}\sqrt{1-h_{ii}}} = \frac{e_{i,-i}\sqrt{1-h_{ii}}}{\hat{\sigma}_{-i}}, \qquad \text{since} \quad e_{i,-i} = \frac{e_i}{1-h_{ii}}.$$

Then Cook's distance can be written as

$$D_i = \frac{n-p}{p}\,\frac{h_{ii}}{1-h_{ii}}\,\frac{t_i^2}{(n-p-1)+t_i^2}\,.$$

8.3 Exact null distribution

Under the classical linear model,

$$t_i^2 \;\sim\; F_{1,\,\nu}, \qquad \nu = n-p-1.$$

Let

$$c_i = \frac{n-p}{p}\cdot\frac{h_{ii}}{1-h_{ii}}, \qquad W_i = \frac{t_i^2}{\nu+t_i^2}.$$

Because $t_i^2 \sim F_{1,\nu}$,

$$W_i \sim \mathrm{Beta}\Big(\tfrac{1}{2},\tfrac{\nu}{2}\Big), \qquad \frac{D_i}{c_i} = W_i \in [0,1]\,.$$

This yields exact, per–case p-values and critical values:

$$p_i = \Pr\left(W_i \geq \frac{D_i}{c_i}\right) = S_{\mathrm{Beta}}\left(\frac{D_i}{c_i};\,\tfrac{1}{2},\,\tfrac{n-p-1}{2}\right),$$

$$d_{i,\alpha} = c_i\; q_{\mathrm{Beta}}\Big(1-\alpha;\tfrac{1}{2},\tfrac{n-p-1}{2}\Big).$$

where S_{Beta} and q_{Beta} stand for the survival and quantile functions of Beta distribution.

8.4 The rough $4/n$ rule (average-leverage simplification)

Approximating a "typical" case by **average leverage** $h_{ii} \approx p/n$ gives

$$c_i = \frac{n-p}{p} \cdot \frac{p/n}{1-p/n} \approx 1.$$

The 95th percentile of W_i is

$$\text{qbeta}\Big(0.95; \tfrac{1}{2}, \tfrac{n-p-1}{2}\Big) \;\approx\; \frac{F_{1,\nu,0.95}}{\nu + F_{1,\nu,0.95}} \;\approx\; \frac{3.84}{n-p-1} \;\approx\; \frac{4}{n} \quad (\text{when } p \ll n).$$

So $4/n$ is a **rule-of-thumb** 95% cutoff for an **average-leverage** point; the exact leverage–aware cutoff is $d_{i,\alpha}$ above (larger for high h_{ii}, smaller for low h_{ii}).

8.5 Comparing $4/n$ rules with the actual critical values

```
library(dplyr)
library(tidyr)
library(knitr)
library(kableExtra)

# Exact 95% Cook's D cutoff under average leverage h_ii = p/n (so c_i = 1):
# d_{i,0.95} = qbeta(0.95; 1/2, (n - p - 1)/2), valid when nu = n - p - 1 > 0
cook_crit_avg <- function(n, p, alpha = 0.05) {
  nu <- n - p - 1
  if (nu <= 0) return(NA_real_)
  stats::qbeta(1 - alpha, shape1 = 0.5, shape2 = nu / 2)
}

# Grids (edit as needed)
n_vals <- c(20, 30, 50, 80, 100, 150, 200, 500)
p_vals <- c(2, 3, 5, 10, 15, 20, 30, 50)

df <- tidyr::crossing(n = n_vals, p = p_vals) %>%
  mutate(valid = p <= n - 2,
         nu    = n - p - 1L) %>%
  rowwise() %>%
  mutate(
    cook_crit_95 = if (valid) cook_crit_avg(n, p, 0.05) else NA_real_,
```

```
    `4/n`        = 4 / n,
    ratio        = cook_crit_95 / `4/n`,
    `p/n`        = p / n
  ) %>%
  ungroup() %>%
  filter(valid) %>%
  select(n, p, `p/n`, nu, cook_crit_95, `4/n`, ratio) %>%
  mutate(
    `p/n`         = round(`p/n`, 3),
    cook_crit_95  = round(cook_crit_95, 6),
    `4/n`         = round(`4/n`, 6),
    ratio         = round(ratio, 4)
  )


if (knitr::is_html_output()) {
  # HTML → Quarto prints `df` as a paged table and uses tbl-cap
  library(DT)
  DT::datatable(
    df,
    rownames = FALSE,
    options = list(pageLength = 10, scrollX = TRUE),
    caption = htmltools::tags$caption(
      style = 'caption-side: top; text-align: left;',
      htmltools::HTML("Exact 95% Cook&#39;s D critical value (average leverage $h_{ii}=p/n \\
    )
  )
} else {
  # Non-HTML (PDF, DOCX) → fall back to kable

  knitr::kable(
    df,
    align = "r",
    booktabs = TRUE,
    caption = "Exact 95% Cook's D critical value (average leverage $h_{ii}=p/n \\Rightarrow
  )
}
```

Table 3: Exact 95% Cook's D critical value (average leverage $h_{ii} = p/n \Rightarrow c_i = 1$) vs heuristic $4/n$.

n	p	p/n	nu	cook_crit_95	4/n	ratio
20	2	0.100	17	0.207508	0.200000	1.0375
20	3	0.150	16	0.219284	0.200000	1.0964
20	5	0.250	14	0.247316	0.200000	1.2366
20	10	0.500	9	0.362487	0.200000	1.8124
20	15	0.750	4	0.658372	0.200000	3.2919
30	2	0.067	27	0.134893	0.133333	1.0117
30	3	0.100	26	0.139791	0.133333	1.0484
30	5	0.167	24	0.150733	0.133333	1.1305
30	10	0.333	19	0.187366	0.133333	1.4052
30	15	0.500	14	0.247316	0.133333	1.8549
30	20	0.667	9	0.362487	0.133333	2.7187
50	2	0.040	47	0.079282	0.080000	0.9910
50	3	0.060	46	0.080951	0.080000	1.0119
50	5	0.100	44	0.084510	0.080000	1.0564
50	10	0.200	39	0.094944	0.080000	1.1868
50	15	0.300	34	0.108314	0.080000	1.3539
50	20	0.400	29	0.126058	0.080000	1.5757
50	30	0.600	19	0.187366	0.080000	2.3421
80	2	0.025	77	0.048973	0.050000	0.9795
80	3	0.038	76	0.049605	0.050000	0.9921
80	5	0.062	74	0.050920	0.050000	1.0184
80	10	0.125	69	0.054533	0.050000	1.0907
80	15	0.188	64	0.058698	0.050000	1.1740
80	20	0.250	59	0.063551	0.050000	1.2710
80	30	0.375	49	0.076141	0.050000	1.5228
80	50	0.625	29	0.126058	0.050000	2.5212
100	2	0.020	97	0.039025	0.040000	0.9756
100	3	0.030	96	0.039425	0.040000	0.9856
100	5	0.050	94	0.040251	0.040000	1.0063
100	10	0.100	89	0.042476	0.040000	1.0619
100	15	0.150	84	0.044961	0.040000	1.1240
100	20	0.200	79	0.047756	0.040000	1.1939
100	30	0.300	69	0.054533	0.040000	1.3633
100	50	0.500	49	0.076141	0.040000	1.9035
150	2	0.013	147	0.025880	0.026667	0.9705
150	3	0.020	146	0.026056	0.026667	0.9771
150	5	0.033	144	0.026414	0.026667	0.9905
150	10	0.067	139	0.027355	0.026667	1.0258

n	p	p/n	nu	cook_crit_95	4/n	ratio
150	15	0.100	134	0.028364	0.026667	1.0637
150	20	0.133	129	0.029452	0.026667	1.1044
150	30	0.200	119	0.031897	0.026667	1.1961
150	50	0.333	99	0.038248	0.026667	1.4343
200	2	0.010	197	0.019359	0.020000	0.9680
200	3	0.015	196	0.019457	0.020000	0.9729
200	5	0.025	194	0.019657	0.020000	0.9828
200	10	0.050	189	0.020173	0.020000	1.0086
200	15	0.075	184	0.020717	0.020000	1.0358
200	20	0.100	179	0.021291	0.020000	1.0645
200	30	0.150	169	0.022540	0.020000	1.1270
200	50	0.250	149	0.025536	0.020000	1.2768
500	2	0.004	497	0.007707	0.008000	0.9634
500	3	0.006	496	0.007723	0.008000	0.9653
500	5	0.010	494	0.007754	0.008000	0.9692
500	10	0.020	489	0.007833	0.008000	0.9791
500	15	0.030	484	0.007914	0.008000	0.9892
500	20	0.040	479	0.007996	0.008000	0.9995
500	30	0.060	469	0.008166	0.008000	1.0207
500	50	0.100	449	0.008529	0.008000	1.0661

Appendix: Key Identities

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.

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.

$$e_{i,-i} = \frac{e_i}{1 - h_{ii}} \tag{13}$$

Finding the LOOCV Standard Error from the Full-Model Standard Error

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

$$\hat{\sigma}_{-i} = \sqrt{\frac{(n-p)\hat{\sigma}^2 - \frac{e_i^2}{1-h_{ii}}}{n-p-1}} \tag{14}$$

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.