Dummy variables in Regression. Anti-inflammatory medication remaining in devices from different batches

Diego Ramos Crespo

April 23, 2025

1 Introduction

The dataset used for the current project is located in the 'bootstrap' package with the name 'hormone'. It contains the amount in milligrams of anti-inflammatory hormone remaining in devices after certain number of hours. The data frame contains three columns and 27 rows, the first one with the name 'Lot' that provides information about the batch from where the samples were taken; it considers the lot 'A', 'B' and 'C'. The next column is the number of hours followed by the amount of hormone in milligrams (mg).

Figure 1 depicts the data that will be used for fitting the model.

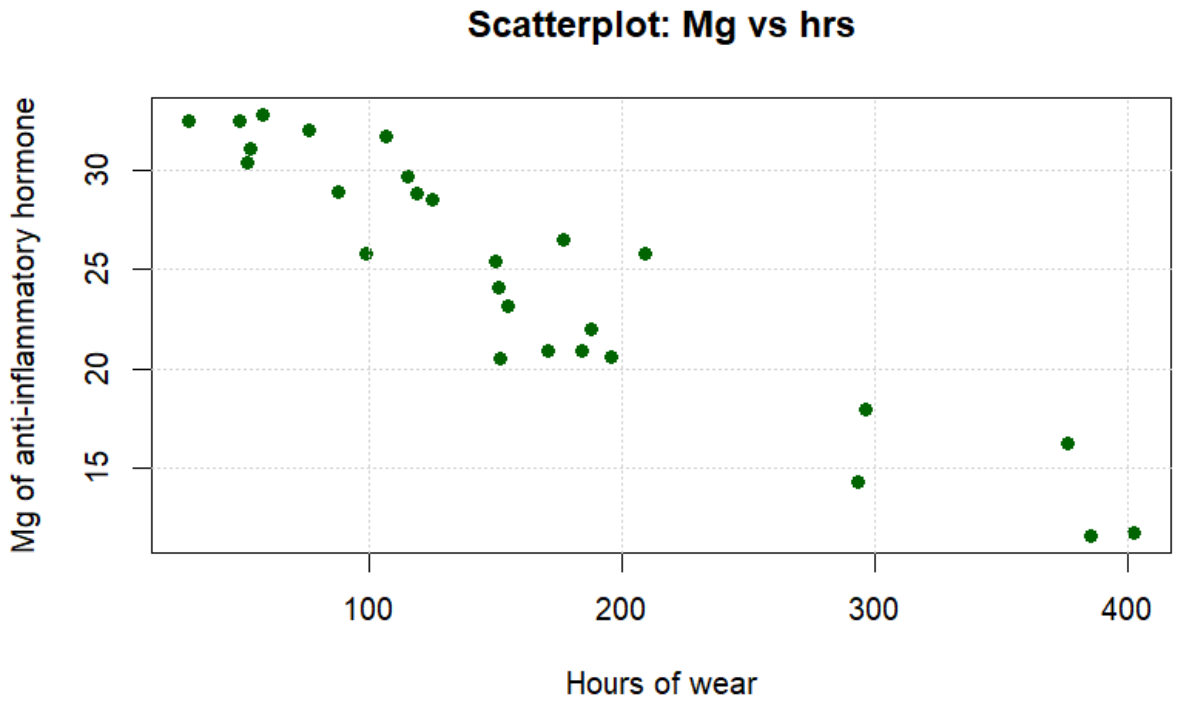

Figure 1: Plot: Milligrams of anti-inflammatory hormone in 27 devices against hours of wear.

1.1 Encode variables

Since the data frame includes a categorical variable, it may be useful to incorporate this information to assess whether it improves the model's performance. However, linear regression models cannot directly process categorical inputs. To address this problem, we transform the categorical variable into dummy variables by assigning binary values (0 or 1) to indicate the absence or presence of

each category.

Two new columns are added: `batch B` and `batch C`. A value of 1 in either column indicates that the sample belongs to that batch. If both values are 0, it means the sample belongs to `batch A`.

There was used the `'ifelse'` function to create the dummy variables. This built-in R function allows to enter an array and compare each element in that array. In case the condition is true, then it fills a new array with a specified value in the parameter `'yes'`, otherwise it fills with another personalized value defined in the `'no'` parameter until the array reaches the same shape as the input. An example of its use is depicted below.

```
ifelse(hormone$Lot == 'B', 1, 0)
```

After the data pre-processing, we get a new data frame with four columns, shown in Table 1.

Hours	**Batch_B**	**Batch_C**	**Amount**
107	0	1	31.7
196	0	0	20.6
152	0	0	20.5
296	1	0	18.0
171	0	0	20.9
188	0	1	22.0
151	1	0	24.1
402	1	0	11.8
293	0	0	14.3
52	0	0	30.4

Table 1: Sample of data with encoded batch variables

1.2 Models

In this section, two linear regression models are constructed. The first model considers only the variable `Hours`, while the second incorporates the dummy variables created in Section 1.1, with the goal of comparing both models and assessing the contribution of categorical information.

The first model is represented by Equation 1.2.1:

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 \tag{1.2.1}$$

Where:

- X_1: Hours of wear

The results are presented in Table 1.2 show that the p-value is extremely small, which provides strong evidence that the slope coefficients are significantly different from zero. Additionally, Table 1.2 displays a high coefficient of determination ($R^2 = 86.88\%$), indicating that the model explains the majority of the variability in the response variable. The adjusted R^2 is also high (86.36%), reinforcing the model's explanatory power even after accounting for the number of predictors. Furthermore, the F-statistic value of 165.6 with a p-value of 1.58×10^{-12} confirms that the model as a whole is statistically significant, meaning that the explanatory variable has a strong linear relationship with the dependent variable.

Table 2: Residual summary

Min	1Q	Median	3Q	Max
-4.9357	-1.7282	-0.0229	1.7388	3.7323

Table 3: Regression coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	34.1675	0.8672	39.40	$< 2 \times 10^{-16}$ ***
Hours	-0.0574	0.0045	-12.87	1.58×10^{-12} ***

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

Table 4: Model summary

Residual Std. Error	2.378 on 25 degrees of freedom
Multiple R-squared	0.8688
Adjusted R-squared	0.8636
F-statistic	165.6 on 1 and 25 DF
p-value	1.584×10^{-12}

The model is depicted in Figure 2.

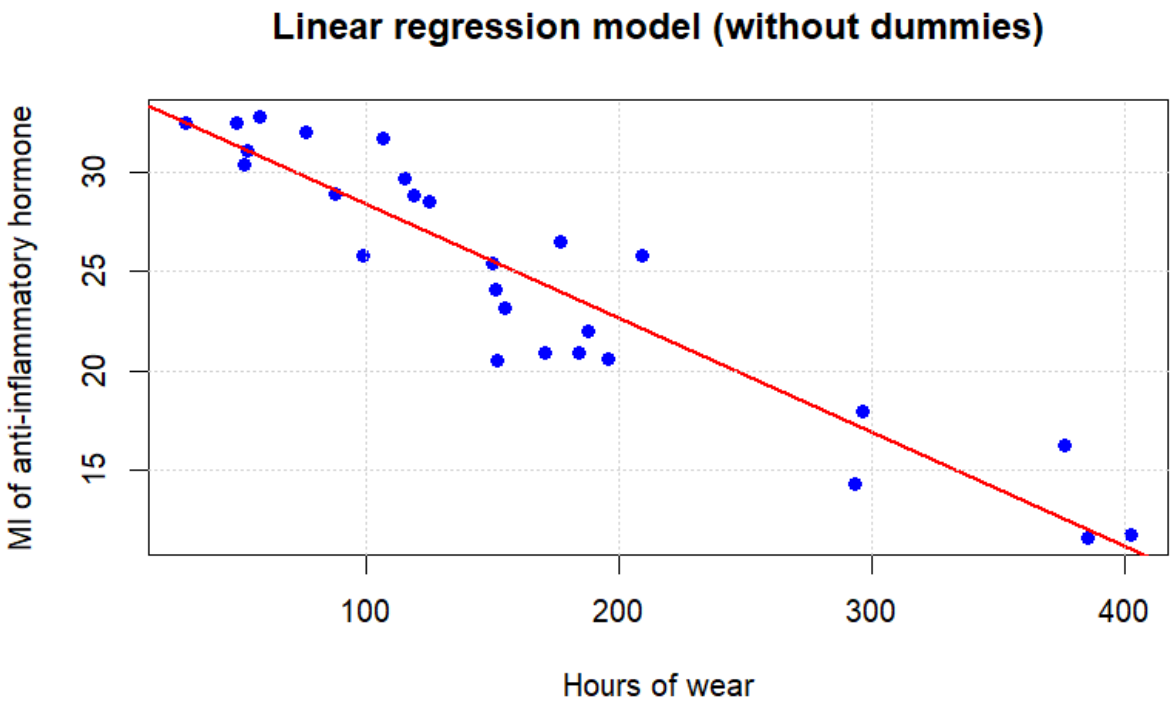

Figure 2: Scatterplot and regression line computed using the 'Hour of wear' variable

The second model is given by Equation 1.4.1. In this case, the binary variables serve as indicators. When a sample belongs to a specific batch, the corresponding dummy variable takes the value 1, activating the associated partial slope in the equation. For the other batches, the dummy variables are zero, removing the contribution of unrelated categories and making the model more specific to each group.

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \widehat{\beta}_2 X_B + \widehat{\beta}_3 X_C \tag{1.2.2}$$

Where:

- X_1: Hours of wear
- X_A: Sample from batch 'A' (reference category)
- X_B: Sample from batch 'B'
- X_C: Sample from batch 'C'

The results presented in Table 9 show that all predictor variables are statistically significant, with extremely small p-values. This provides strong evidence that the slope coefficients differ significantly from zero. Specifically, the variable `Hours` has a negative and highly significant effect on the response variable. Additionally, the batch variables `Batch_B` and `Batch_C` show significant positive effects. Table 10 shows a very high coefficient of determination ($R^2 = 94.5\%$), indicating that the model explains almost all of the variability. The adjusted R^2 is also high (93.78%), and this confirms the model's robustness. Furthermore, the F-statistic value of 131.8 with a p-value of 1.25×10^{-14} indicates that the model as a whole is statistically significant, showing a strong linear relationship between the predictors and the response variable.

Table 5: Residual summary

Min	1Q	Median	3Q	Max
-2.9245	-1.0626	-0.1304	0.8544	2.8061

Table 6: Regression coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	32.1316	0.7483	42.941	$< 2 \times 10^{-16}$ ***
Hours	-0.0601	0.0035	-17.310	1.10×10^{-14} ***
Batch_B	3.9735	0.8097	4.907	5.87×10^{-5} ***
Batch_C	3.4657	0.7691	4.506	0.000159 ***

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

Table 7: Model summary

Residual Std. Error	1.605 on 23 degrees of freedom
Multiple R-squared	0.945
Adjusted R-squared	0.9378
F-statistic	131.8 on 3 and 23 DF
p-value	1.254×10^{-14}

Figure 3 shows the regression line found for every batch. This makes the regression analysis more specific for every category in comparison to the previous model.

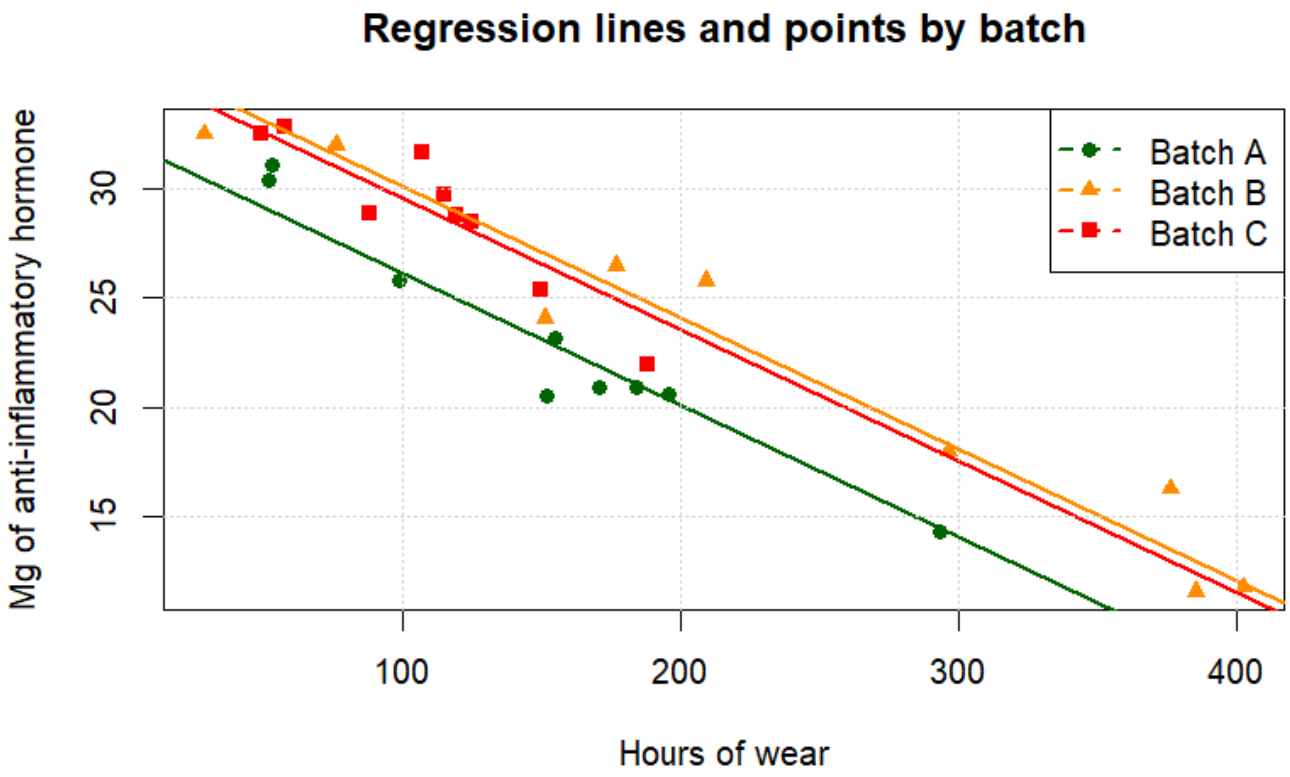

Figure 3: Scatterplot and regression lines using dummy variables to separate every batch.

Incorporating `Batch_B` and `Batch_C` as dummy variables allows the model to capture the effects of being in different batches, helping to improve the model's accuracy and explainability. It was shown that the value of the R^2 and adjusted R^2 metrics increase when these new attribute are considered. Moreover, the p-value of the F statistic support this affirmation. Therefore, it would be meaningful to consider the dummy variables in the model.

1.3 Comparison of models

The following image (Figure 4)visually compares the regression line of model 1 (blue line, with no dummies) against model 2 (one regression line per batch). It shows how the inclusion of dummy variables allows each lot to have a different intercept, getting even closer to its cloud of points. Besides, the points that belong to each batch have a different shape and color to the rest in order to make it easier to notice the differences.

LOOCV Implementation in R:

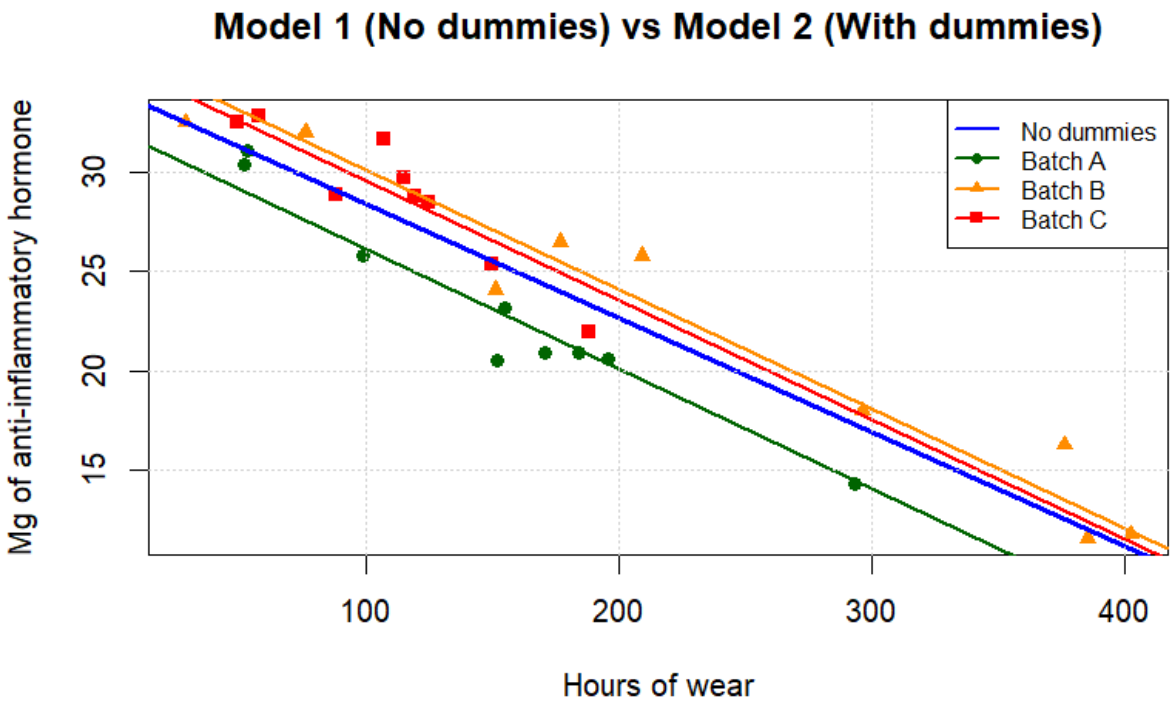

Figure 4: Comparison between regression line of model 1 (with just one predictor variable) and model 2 (considering dummy variables).

The following code shows the implemented LOOCV function to evaluate both models.

```
# Leave-One-Out Cross Validation (LOOCV)
loocv <- function(df, col_resp, cols_independent) {
  n <- nrow(df)  # Number of rows in the dataset
  squared_error <- numeric(n)  # Initialize vector to store squared errors

  cols_names <- colnames(df)  # Get all column names
  y_name <- cols_names[col_resp]  # Identify the name of the response variable
  x_names <- cols_names[cols_independent]  # Identify names of independent variables

  # Construct a formula for the linear model.
  formula_text <- paste(y_name, "~", paste(x_names, collapse = "+"))
  formula <- as.formula(formula_text)
```

```
  # Perform LOOCV: loop through each row, leaving one out each time
  for (i in 1:n) {
    df_cv <- df[-i, ]  # Exclude the i-th row for training
    row <- df[i, , drop = FALSE]  # Keep the excluded row for testing

    model <- lm(formula, data = df_cv)  # Fit linear model on training data
    y_hat <- predict(model, newdata = row)  # Predict response for left-out observation
    y_real <- df[i, y_name]  # Actual response value

    squared_error[i] <- (y_real - y_hat)^2  # Compute squared error
  }

  return(mean(squared_error))  # Return average squared error
}
```

The function `loocv` was called to perform Leave-One-Out Cross Validation (LOOCV) on two models. However, it is important to mention that the input parameters are the **data frame** that contains the information, `'col_resp'`, that correspond to the number of the column that contains the response variable, and `'cols_independent'`, that obtains to the number of columns that will be used for training the model. The function just accepts numerical values.

```
# Call functions for both CV
cv1 <- loocv(hormone_df, col_resp=4, cols_independent=1)
cv2 <- loocv(hormone_df, col_resp=4, cols_independent=c(1,2,3))

cat("CV1 (Model 1):", cv1, "\n") # 6.027698
cat("CV2 (Model 2):", cv2, "\n") # 3.092669
# Ratio of both Cross-Validation
cat("CV1/CV2:", cv1/cv2) # 1.949028
```

As stated above, to determine which model performs better, we applied Leave-One-Out Cross Validation (LOOCV) to both models. LOOCV involves training the model on all data points

except one, predicting the left-out observation, and repeating this process for every data point. The average of the squared prediction errors is then used as a measure of model performance.

Using the `loocv` function, the calculated the cross-validation scores were:

- **CV1 (Model 1)**: 6.027698, which corresponds to the model using only the variable `hrs`.
- **CV2 (Model 2)**: 3.092669, which includes `hrs` and two dummy variables representing batches B and C.

The ratio between the two scores is:

$$\frac{\text{CV1}}{\text{CV2}} = \frac{6.027698}{3.092669} \approx 1.949$$

This ratio indicates that the prediction error for model 1 is nearly twice that of model 2. Therefore, model 2 provides significantly better generalization performance. The inclusion of batch information as dummy variables in Model 2 helps explain variability in the response variable, reducing the prediction error. Thus, based on the LOOCV scores, **Model 2 is the better model**.

1.4 Model 3: Batch A

In the current analysis, we are considering only Batch A as the predictor for the amount of anti-inflammatory hormone. This allows us to isolate the effects of Batch A on the regression model, while simplifying the comparison to other batches. The model we are building here includes only the predictor `'Hours'` (the number of hours of wear) and the indicator for Batch A. This analysis aims to evaluate the influence of Batch A specifically, and whether distinguishing it from other batches improves the predictive power of the model.

Once Model 3 is built, we will compute the Leave-One-Out Cross-Validation (LOOCV) score to assess its predictive performance. By comparing the LOOCV scores from all three models, we can determine which model performs best in terms of predictive accuracy. Additionally, we will examine whether it is statistically meaningful to distinguish between the two batches that had similar slopes in model 2.

To summarize, the model we are analyzing here is a linear regression model that includes the predictor variable `Hours` (the number of hours of wear) and the categorical variable `Batch_A` (the batch indicator for Batch A), as shown by the following regression formula:

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \widehat{\beta}_2 X_A \tag{1.4.1}$$

Where:

- X_1: Hours of wear
- X_A: Sample from batch 'A'

The regression results are summarized in the following tables:

Table 8: Residual summary

Min	1Q	Median	3Q	Max
-2.6495	-1.1397	-0.1053	0.8182	2.8570

Table 9 provides the estimated coefficients for the intercept, `Hours`, and `Batch_A`. The coefficients are highly significant, with p-values less than 2×10^{-16} for `Intercept` and `Hours`, and 7.28×10^{-6} for `Batch_A`, indicating that all predictors significantly contribute to the model.

Table 9: Regression coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	35.6797	0.6355	56.147	$< 2 \times 10^{-16}$ ***
Hours	-0.0591	0.0030	-19.801	2.25×10^{-16} ***
Batch_A	-3.6980	0.6495	-5.693	7.28×10^{-6} ***

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

Finally, the model summary, shown in Table 10, provides key statistics to evaluate the overall fit of the model, including the residual standard error, R-squared, adjusted R-squared, F-statistic, and the p-value for the F-test.

Table 10: Model summary

Residual Std. Error	1.583 on 24 degrees of freedom
Multiple R-squared	0.9442
Adjusted R-squared	0.9395
F-statistic	203 on 2 and 24 DF
p-value	9.12×10^{-16}

The model has a high R-squared value of 0.9442, indicating that approximately 94.42% of the variation in the response variable (amount of hormone) is explained by the predictor variables `Hours` and `Batch_A`. The F-statistic is also highly significant, with a p-value of 9.12×10^{-16}, suggesting that the model is statistically significant.

Based on the Cross-Validation (CV) results, we can have several conclusions about the models.

Model 3, with a LOOCV score of 2.814787, performs better than model 1 and is almost as good as model 2. The CV score for Model 3 suggests that it is able to generalize better to new data, indicating better predictive accuracy.

Looking at the performance of Model 3 compared to Model 2, we see that the ratio of CV3 to CV2 (0.9101482) is very close to 1. This suggests that distinguishing between the two lots with similar slopes, as done in model 2, may not be statistically meaningful. The predictive performance of model 3, which simplifies the distinction by using a single dummy variable for the batches, is almost identical to that of model 2. Thus, it may not be necessary to maintain a more complex model when a simpler one (model 3) yields similar results.

Further analysis of the ratio between Model 3 and Model 1 (CV3/CV1 = 0.4669754) shows that the simpler Model 3 performs significantly better than model 1. This result implies that reducing model complexity, by combining similar categories, leads to improved predictive accuracy compared to model 1, which may have overfitted due to its more complex structure.

In conclusion, Model 3 appears to offer the best balance between predictive accuracy and simplicity. By grouping the batches with similar slopes into a single category, we achieve almost identical performance to model 2, but with fewer parameters making it the best model for this analysis.