Understanding Regression with Transformations

Introduction to Transformations in Regression

Transformations in regression models are crucial when the relationship between the variables is not adequately captured by a simple linear model. In this section, we will explore how transformations can help us model the complexities of real-world data.

- Indicator Variables: To capture categorical data within regression models.
- Interaction Terms: To model the effect of combinations of variables.

Intuition Behind Using Transformations

- Transformations such as interaction terms can unveil the hidden structure in the data that a simple linear model may not capture.
- Ignoring such relationships can lead to an underfit model that does not accurately predict or explain the behavior of our outcome variable.
- A transformation does more than just modify the vaoue of Y, it modifies the relationship between X and Y.
- For instance, let's consider the effect of exercise (X) on weight loss (Y), which might be different for individuals with different diets (Z). Ignoring the diet factor may mislead us about the effectiveness of the exercise.

Scatterplot of Two Continuous Variables

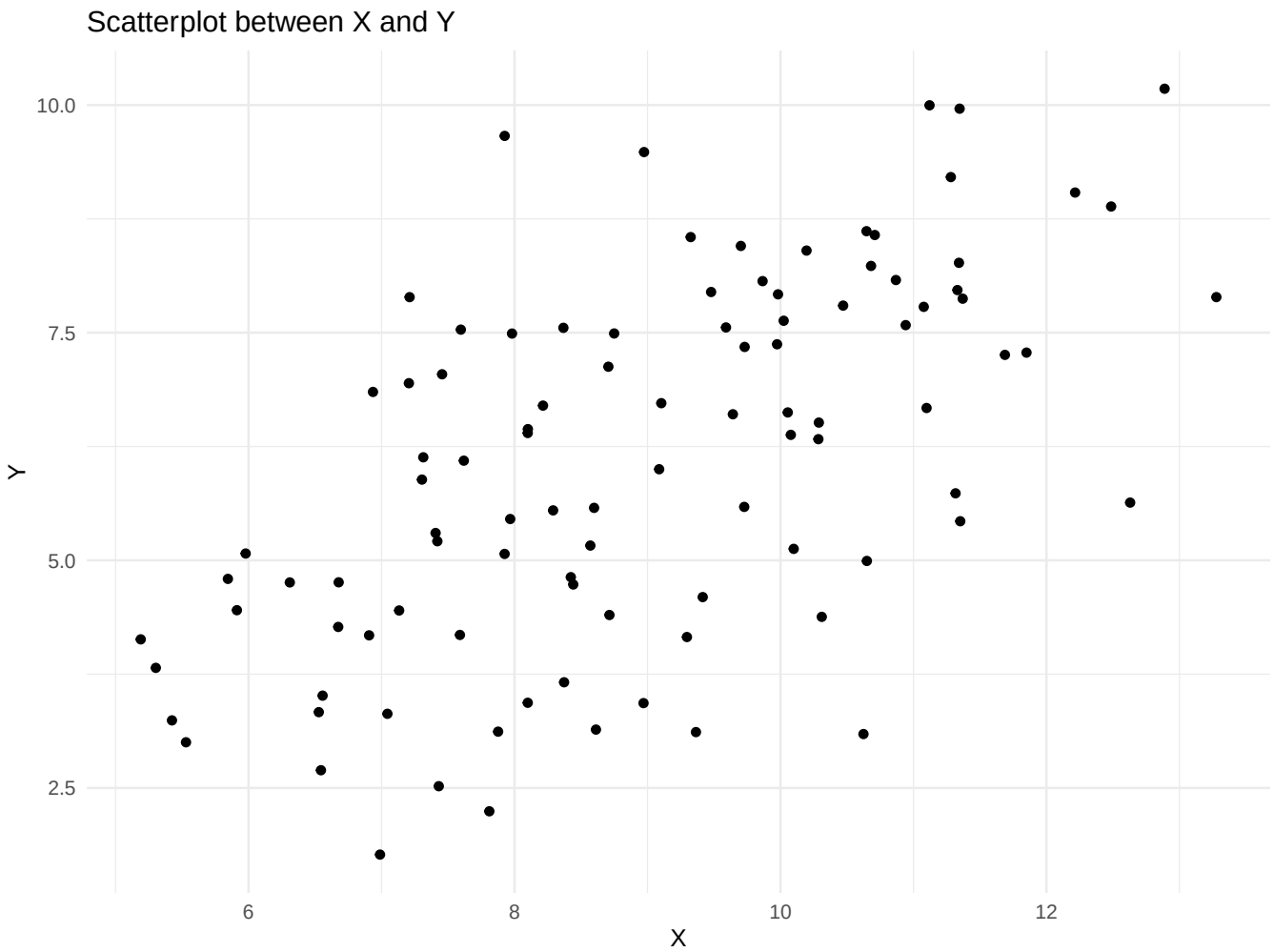

Figure: Scatterplot between variable X and Y.

Scatterplot Colored by Indicator

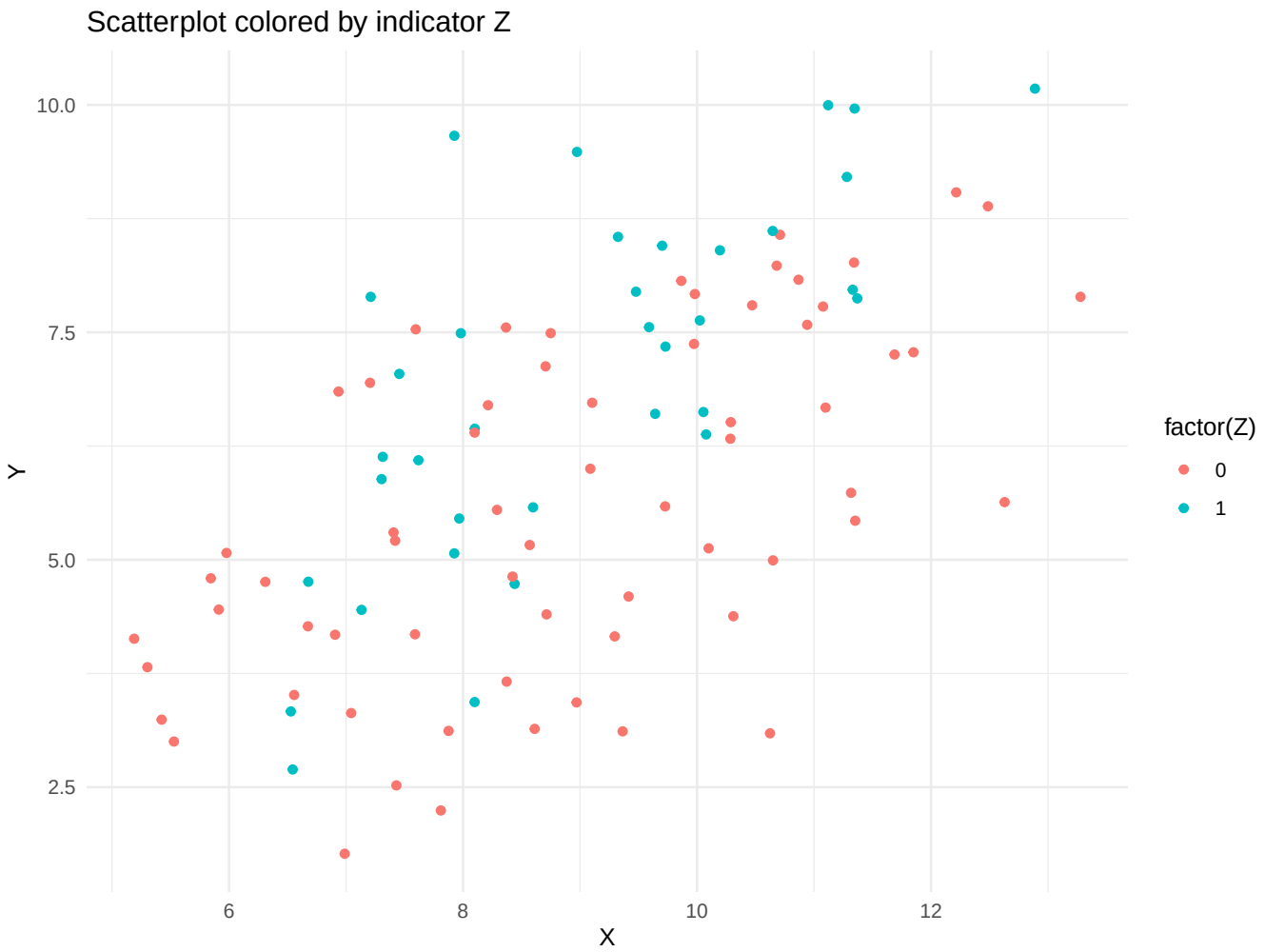

Figure: Scatterplot colored by indicator Z.

Regression Model Equations with Indicator Variables

We can fit the model without the indicator variable at all. The data looks fine without considering it. **Regression without Indicator:**

$$y = \beta_0 + \beta_1 x + \epsilon$$

Regression Line Ignoring the Indicator

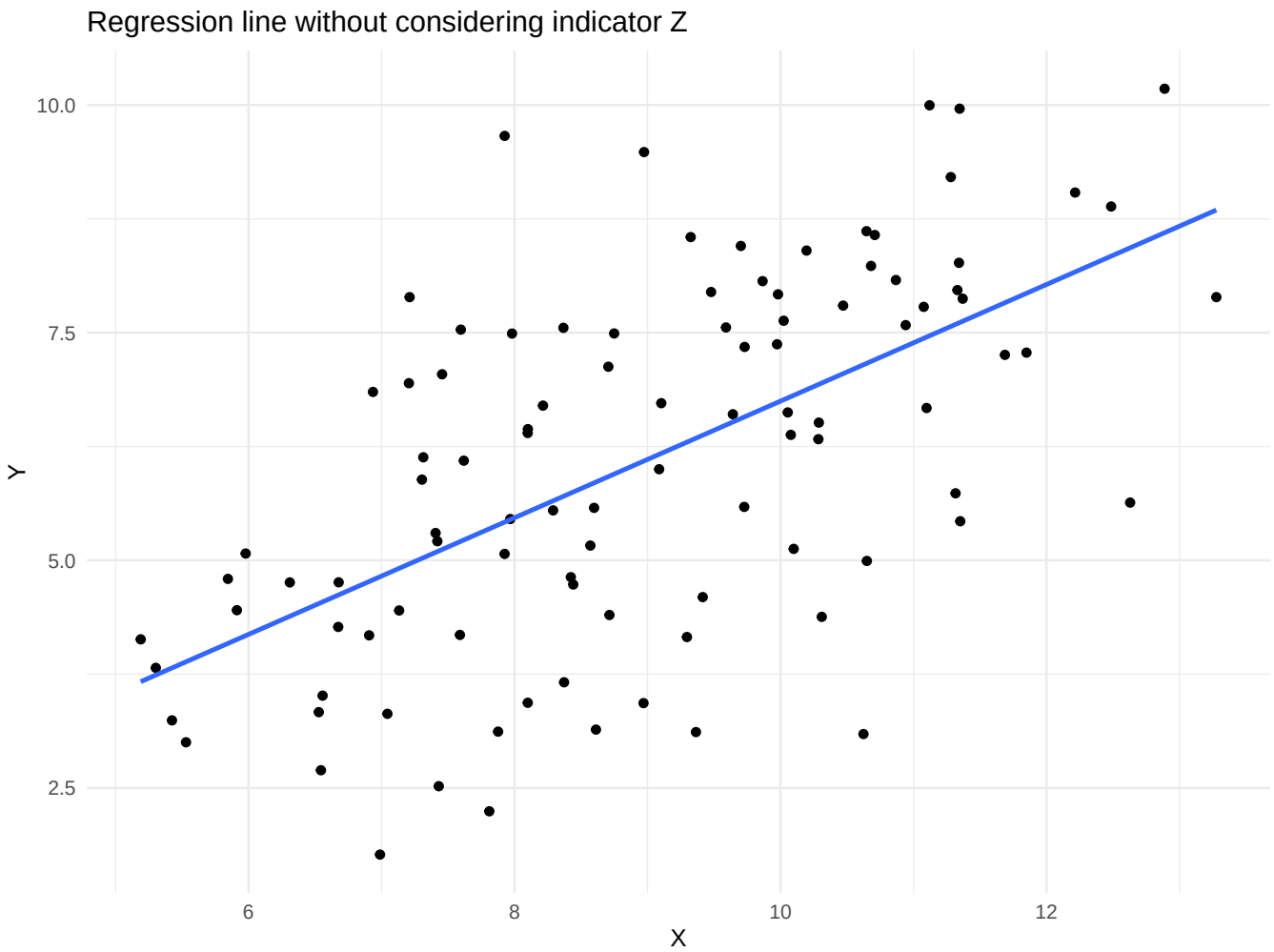

Figure: Regression line without considering indicator Z.

Regression Model Equations with Indicator Variables

When we include an indicator variable Z, which is binary, in our regression model, it effectively allows for different intercepts for the groups defined by Z. The regression line's slope remains the same, but its intercept changes depending on the value of Z. Here are the equations for when Z equals 0 or 1:

With Indicator as Constant:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \epsilon$$

Cases for Z:

- When $Z = 0$: $y = \beta_0 + \beta_1 x + \epsilon$ (Intercept remains β_0)
- When $Z = 1$: $y = (\beta_0 + \beta_2) + \beta_1 x + \epsilon$ (Intercept becomes $\beta_0 + \beta_2$)

The coefficient β_2 represents the shift in the intercept when Z changes from 0 to 1. If β_2 is significant, it indicates that the average value of y when $Z = 1$ is different from when $Z = 0$, after controlling for x.

Adding the Indicator as a Constant

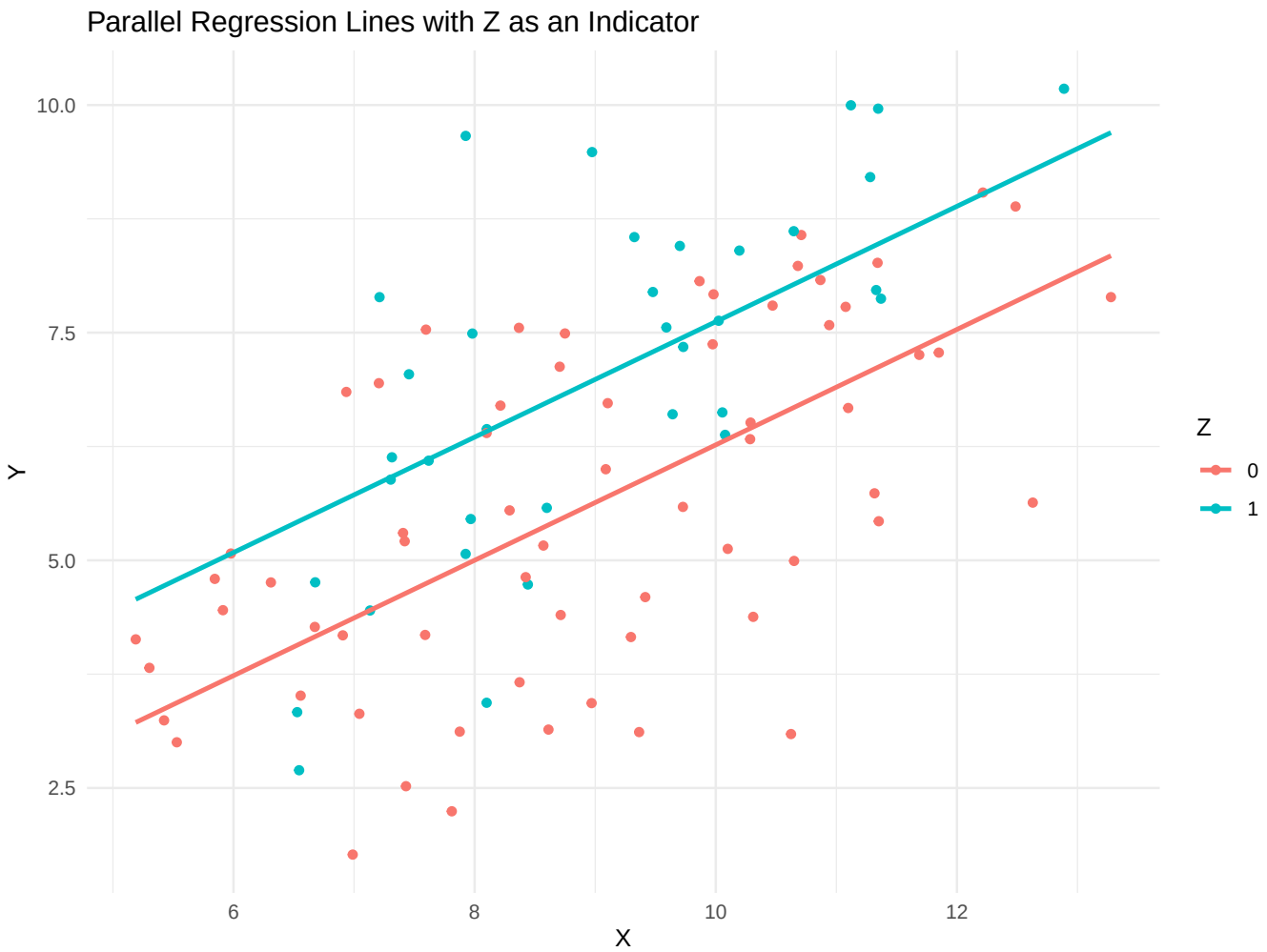

Figure: Regression line with indicator Z as a constant.

Regression Model Equations with Indicator Variables

When including an interaction term between a continuous variable x and a binary indicator variable Z, we allow both the slope and the intercept of the regression line to change with the value of Z. Here's how the model changes:

With Interaction Term:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz + \epsilon$$

Cases for Z with Interaction:

- When $Z = 0$: $y = \beta_0 + \beta_1 x + \epsilon$ (Slope is β_1, intercept is β_0)
- When $Z = 1$: $y = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)x + \epsilon$ (Slope is $\beta_1 + \beta_3$, intercept is $\beta_0 + \beta_2$)

The interaction term $\beta_3 xz$ indicates the additional change in the slope of y with respect to x when $Z = 1$ compared to when $Z = 0$. If β_3 is significant, it suggests that the relationship between x and y is different for different levels of Z.

Adding the Indicator as an Interaction Term

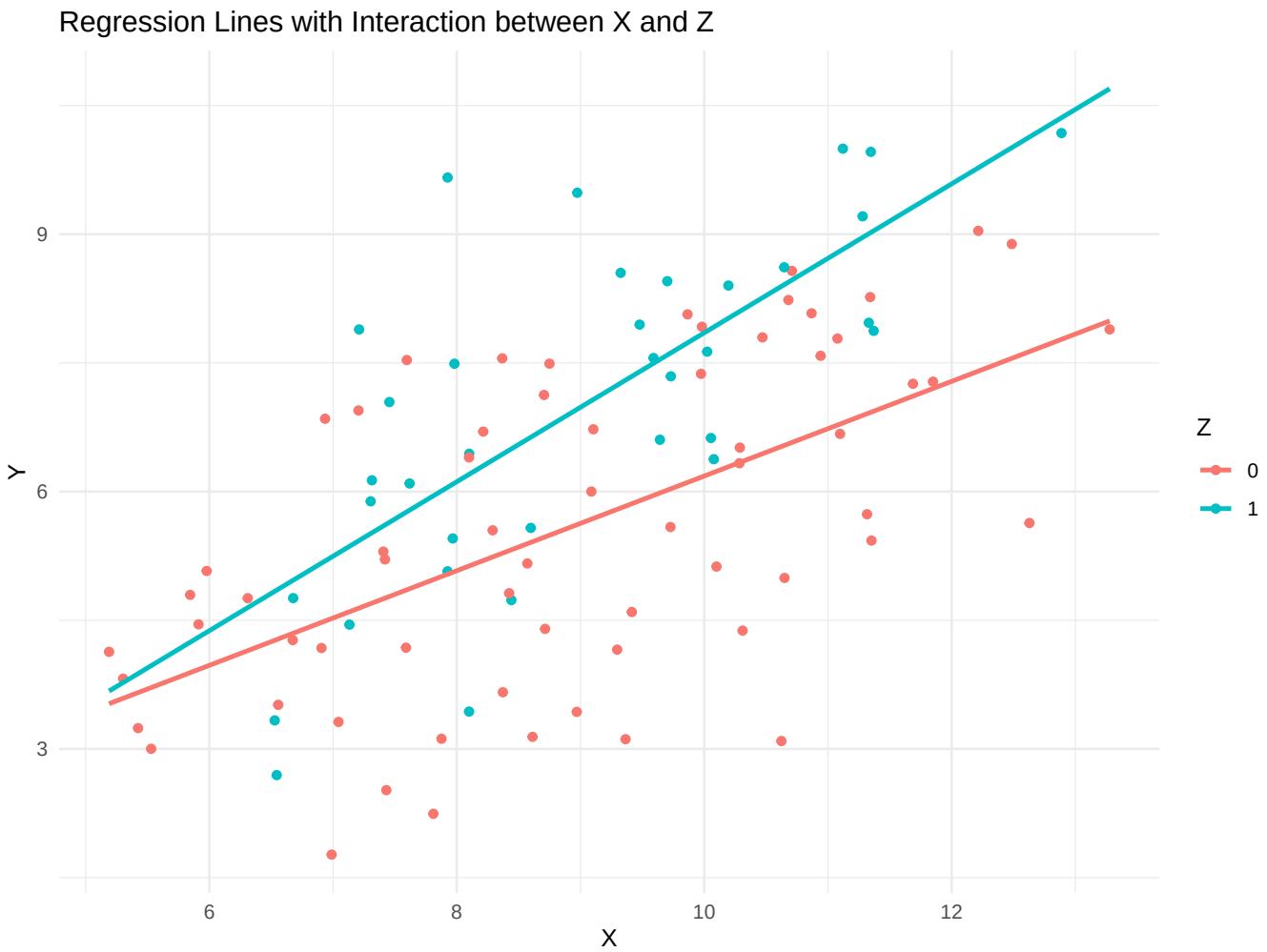

Figure: Regression line with an interaction between X and indicator Z.

Regression Model Equations with Indicator Variables

Regression without Indicator:

$$y = \beta_0 + \beta_1 x + \epsilon$$

With Indicator as Constant:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \epsilon$$

With Interaction Term:

$$y = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz + \epsilon$$

The coefficients β_2 and β_3 will help us understand the influence of the indicator variable and the interaction term, respectively.

Real Scenario Without Interaction

Consider a dataset with students' hours studied (X) and their exam scores (Y). Using a traditional teaching method, the regression model is as follows:

$$y = 50 + 5X$$

This suggests that with no study hours, a student is expected to score 50, and each additional hour of study increases the score by 5 points.

Real Scenario With Interaction

Now let's include an innovative teaching method as an indicator variable (Z), where Z=0 represents the traditional method, and Z=1 represents the innovative method. **Traditional Method:**

$$y = 50 + 5X$$

Innovative Method:

$$y = 60 + 7X$$

Here, not only does the innovative method add an overall 10 points to the expected score, but it also increases the score gains per study hour to 7 points. What are β_0, β_1, β_2, and β_3 in this case?

Python Code for Regression with Indicator and Interaction

```
# Python code for regression with indicators
from sklearn.linear_model import
    LinearRegression
import pandas as pd

# Create the interaction term manually
data['XZ'] = data['X'] * data['Z']

# Define the independent variables for the model
X = data[['X', 'Z', 'XZ']]

# Define the dependent variable
y = data['Y']

# Initialize and fit the linear regression model
model = LinearRegression().fit(X, y)
```

Beyond One Indicator: Exploring Other Interactions

Interactions in regression are not limited to just an indicator variable with a continuous variable. They can occur between:

- A factor with 3+ levels and a continuous variable.
- Two continuous variables.
- Two factor variables.

These interactions can uncover more complex relationships in our data, leading to more accurate and nuanced models.

Factor with 3+ Levels and Continuous Variable Interaction

When we have a factor with three or more levels interacting with a continuous variable, we can model how the relationship between the continuous variable and the outcome changes across the different levels of the factor.

- This can be particularly insightful in fields like medicine, where treatment responses (continuous) may vary significantly across different patient groups (factors with multiple levels, e.g., age groups or disease stages).

Such an interaction allows for tailored strategies and more personalized models.

Interaction Between Two Continuous Variables

Interactions between two continuous variables allow us to model situations where the effect of one continuous variable on the response variable is altered by the level of another continuous variable.

- In economics, for instance, the interaction between interest rates and investment in infrastructure could be studied to understand their combined effect on economic growth.

Including this interaction can reveal synergistic or diminishing effects not observable when considering variables in isolation.

Interaction Between Two Factor Variables

Interactions can also occur between two factor variables. This type of interaction helps us understand how the combination of two categorical factors influences the response variable.

- A classic example would be a study on the effectiveness of teaching methods (factor 1) across different age groups (factor 2) on students' test scores.

It allows us to identify which specific combinations of categories have unique effects.

Higher Order Terms in Regression Models

Higher-order terms in regression refer to both interaction terms and polynomial terms. These can be used to capture more complex, non-linear relationships in the data.

- **Polynomial Terms:** These are powers of a variable (e.g., x^2, x^3), allowing the model to fit a wider range of curvature.
- **Interaction Terms:** These are products of variables (e.g., xy), which model how the effect of one variable on the response changes at different levels of another variable.

The `PolynomialFeatures` function in scikit-learn can generate a new feature matrix including both types of higher-order terms. Here's an example:

Example: Using PolynomialFeatures

```
# Sample data
X = np.array([[1, 2], [3, 4], [5, 6]])

# Instantiate PolynomialFeatures
# degree=2 will generate features up to x^2, y^2, xy
# interaction_only=True will generate only interaction features
poly = PolynomialFeatures(degree=2, include_bias=False, interaction_only=False)

# Fit and transform the data
X_poly = poly.fit_transform(X)

# Output the transformed data
print(X_poly)
```