# Linear Regression Analysis

## Analysis Knowledge Check (5 points total)

### 1. Formula for Simple Linear Regression (1 point):

The equation for simple linear regression is as follows:  
y = β₀ + β₁x + ε  
Where:  
- y: Dependent variable (outcome)  
- x: Independent variable (predictor)  
- β₀: Y-intercept (constant term)  
- β₁: Slope of the line (coefficient of x)  
- ε: Random error term

### 2. Mean Squared Error (MSE) Definition and Importance (1 point):

The Mean Squared Error (MSE) quantifies the average of the squared deviations between actual and predicted values:  
MSE = (1/n) Σ (yᵢ - ŷᵢ)²  
We use MSE because:  
- It amplifies larger errors by squaring them, ensuring greater penalization.  
- The function is smooth and convex, making it ideal for optimization via gradient descent.

### 3. Gradient Descent Rule Explained (1 point):

Gradient descent updates model parameters iteratively to minimize the cost function:  
θⱼ := θⱼ - α ∂J(θ)/∂θⱼ  
Where:  
- α: Learning rate controlling the step size  
- ∂J(θ)/∂θⱼ: Derivative of the cost function with respect to parameter θⱼ

### 4. Deriving ∂J/∂β₁ in Linear Regression (1 point):

The partial derivative of the cost function with respect to β₁ is calculated as follows:  
1. Start with the MSE function:  
 J(β₁) = (1/n) Σ (yᵢ - (β₀ + β₁xᵢ))²  
2. Differentiate with respect to β₁:  
 ∂J/∂β₁ = -(2/n) Σ xᵢ (yᵢ - ŷᵢ)  
This result shows that the gradient depends on the weighted residuals.

### 5. When to Choose Multiple Linear Regression (1 point):

Multiple linear regression is used when the dependent variable y is influenced by more than one predictor variable (x₁, x₂, ..., xₖ), improving the model's explanatory and predictive power.

## Practical “Learn by Doing” Exercise (5 points)

Objective: Build a simple linear regression model and interpret its results.

### 1. Data Collection (1 point):

Dataset: Study hours (x) versus exam scores (y):  
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
y = [50, 55, 60, 65, 70, 75, 80, 85, 90, 95]

### 2. Model Creation (2 points):

Using scikit-learn:  
import numpy as np  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error  
  
# Data  
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).reshape(-1, 1)  
y = np.array([50, 55, 60, 65, 70, 75, 80, 85, 90, 95])  
  
# Fit model  
model = LinearRegression()  
model.fit(X, y)  
  
# Parameters  
intercept = model.intercept\_  
coefficient = model.coef\_[0]  
  
print(f"Intercept (β₀): {intercept}")  
print(f"Coefficient (β₁): {coefficient}")  
  
Output:  
Intercept: β₀ = 45.0, Coefficient: β₁ = 5.0

### 3. Model Evaluation (1 point):

Code:  
# Predictions and MSE  
predictions = model.predict(X)  
mse = mean\_squared\_error(y, predictions)  
  
print(f"MSE: {mse}")  
  
Output:  
MSE = 0.0  
Comment: The zero MSE value indicates a perfect fit due to the dataset's ideal linear relationship.

### 4. Interpreting the Coefficient (1 point):

In this model, β₁ = 5.0 signifies that for every additional hour of study, a student's exam score increases by an average of 5 points.

## Total Points: 10/10