**ASSIGNMENT SOLUTION**

# Question 1: What is Simple Linear Regression?

## Answer

Simple Linear Regression is a foundational concept in statistics and machine learning used to model the relationship between two quantitative variables—one independent (predictor) and one dependent (response). Simple Linear Regression fits a straight line through a set of data points to predict the dependent variable based on the independent variable.

## The relationship is expressed by the equation:

$y = \beta 0 + \beta 1 x$ where:

$y$ is the predicted value (dependent variable)

$x$ is the independent variable

$\beta 0$ is the intercept (value of $y$ when $x = 0$ )

$\beta 1$ is the slope (how much $y$ changes for a one-unit change in $x$ )

## Key Concepts

- Line of Best Fit: The goal is to find the line that minimizes the sum of squared differences between actual and predicted values (called residuals).
- Positive vs. Negative Slope:
- Positive slope: As $x$ increases, $y$ increases.
- Negative slope: As $x$ increases, $y$ decreases

## Assumptions:

- Linearity: The relationship between variables is linear.
- Homoscedasticity: Constant variance of errors.
- Independence: Observations are independent.
- Normality: Errors are normally distributed

## Real-Life Example

Imagine a company wants to predict sales based on advertising spend. By applying simple linear regression, they can estimate how much sales will increase for every additional dollar spent on ads

# Question 2: What are the key assumptions of Simple Linear Regression?

## Answer

- The classical linear regression model (CLRM) relies on several key assumptions to ensure the validity and reliability of its results. These assumptions are critical for obtaining unbiased, consistent, and efficient estimates of the regression coefficients. Below is a discussion of the fundamental assumptions:

## Linearity

The relationship between the dependent variable and the independent variables must be linear in parameters. This means the model should be expressed as a linear combination of the predictors and their coefficients. If the true relationship is non-linear, transformations or non-linear models may be required.

## Homoscedasticity

The variance of the residuals (errors) should remain constant across all levels of the independent variables. This is known as homoscedasticity. If the variance changes (heteroscedasticity), it can lead to inefficient estimates and unreliable hypothesis tests. Residual plots are often used to detect violations of this assumption.

## Independence of Errors

The residuals should be independent of each other. This implies that the error for one observation should not influence the error for another. Violations of this assumption, such as autocorrelation, are common in time-series data and can lead to biased standard errors.

## Normality of Errors

The residuals should follow a normal distribution. This assumption is particularly important for hypothesis testing and constructing confidence intervals. Deviations from normality can be assessed using Q-Q plots or statistical tests like the Shapiro-Wilk test.

## No Multicollinearity

The independent variables should not be highly correlated with each other. Multicollinearity inflates the standard errors of the coefficients, making it difficult to determine the individual effect of each predictor. Variance Inflation Factor (VIF) is commonly used to detect multicollinearity.

## Random Sampling

The data should be collected through a random sampling process to ensure that the observations are representative of the population. This helps avoid biases in the estimates.

## Zero Conditional Mean

The expected value of the error term, given the independent variables, should be zero. This ensures that the predictors are not correlated with the error term, avoiding endogeneity issues that can lead to biased estimates.

## No Heteroskedasticity or Serial Correlation

The error terms should have constant variance (homoscedasticity) and should not exhibit serial correlation. Serial correlation occurs when errors are correlated across observations, which is common in time-series data.

## Normally Distributed Errors

The error terms should be normally distributed to ensure valid statistical inference. This assumption is crucial for small sample sizes, as it allows the use of t-tests and F-tests.

These assumptions form the foundation of the classical linear regression model. Violations of these assumptions can lead to biased, inconsistent, or inefficient estimates, and addressing them is essential for reliable model performance.

# Question 3: What is heteroscedasticity, and why is it important to address in regression models?

## Answer

Heteroscedasticity, sometimes spelled heteroskedasticity, refers to the unequal scatter of residuals or error terms in regression analysis. Specifically, it indicates a systematic change in the spread of the residuals over the range of measured values. This phenomenon violates one of the key assumptions of ordinary least squares (OLS) regression, which assumes that the residuals have constant variance, known as homoscedasticity.

## Detection and Causes

Heteroscedasticity can be detected using a fitted value vs. residual plot. In such a plot, if the residuals become more spread out as the fitted values increase, forming a "cone" shape, heteroscedasticity is likely present. Common causes include datasets with a large range of observed values, incorrect model specification, or skewness in the distribution of a regressor.

## Effects on Regression Analysis

When heteroscedasticity is present, it increases the variance of the regression coefficient estimates, making the results of the analysis unreliable. This can lead to incorrect conclusions about the statistical significance of the terms in the model. Additionally, the OLS estimators are no longer the Best Linear Unbiased Estimator (BLUE), and hypothesis tests like the t-test and F-test become invalid due to inconsistencies in the covariance matrix of the estimated regression coefficients.

## Fixing Heteroscedasticity

There are several methods to address heteroscedasticity:

- Transform the Dependent Variable: Applying a transformation, such as taking the logarithm of the dependent variable, can often reduce heteroscedasticity
- Redefine the Dependent Variable: Using a rate or ratio instead of the raw value can help. For example, predicting the number of flower shops per capita instead of the total number of flower shops
- Weighted Regression: Assigning weights to each data point based on the variance of its fitted value can help. This method gives smaller weights to data points with higher variances, reducing their impact on the regression model

## Why Its important

Heteroscedasticity violates one of the key assumptions of Ordinary Least Squares (OLS) regression: that residuals have constant variance. When this assumption is broken:

- Unreliable Estimates: The standard errors of coefficients become biased, leading to incorrect confidence intervals and hypothesis tests.
- Misleading Significance: You might think a variable is statistically significant when it's not—or vice versa.
- Inefficient Predictions: The model may still be unbiased, but it's no longer the most efficient (i.e., it doesn't minimize variance as well as it could).

# Question 4: What is Multiple Linear Regression?

## Answer

Multiple Regression is a special kind of regression model that is used to estimate the relationship between two or more independent variables and one dependent variable. It is also called Multiple Linear Regression(MLR).

It is a statistical technique that uses several variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the independent variables and dependent variables. It is used extensively in econometrics and financial inference.

We generally use the Multiple Regression to know the following.

How strong the relationship is between two or more independent variables and one dependent variable. The estimate of the dependent variable at a certain value of the independent variables. For example,

A public health researcher is interested in social factors that influence heart disease. In a survey of 500 towns' data is gathered on the percentage of people in each town who smoke, on the percentage of people in each town who bike to work, and on the percentage of people in each town who have heart disease.

As we have two independent variables and one dependent variable, and all the variables are quantitative, we can use multiple regression to analyze the relationship between them.

## Basic Condition for Multiple Regression

The basic conditions for Multiple Regression are listed below.

There must be a linear relationship between the independent variable and the outcome variables. It considers the residuals to be normally distributed. It assumes that the independent variables are not highly correlated with each other.

# Question 5: What is polynomial regression, and how does it differ from linear regression?

## Answer

Polynomial regression is a powerful extension of linear regression that allows you to model non-linear relationships between variables by fitting a polynomial equation to the data. Polynomial regression models the relationship between the independent variable $x$ and the dependent variable $y$ using an equation of the form:

$y = a\,0 + a\,1\,x + a\,2\,x\,2 + a\,3\,x\,3 + \cdots + a\,n\,x\,n\,a\,0$ , $a\,1$ , ... , $a\,n$ are coefficients

$n$ is the degree of the polynomial

As $n$ increases, the model becomes more flexible and can fit more complex curves

## When to Use Polynomial Regression

- When your data shows a curved trend that linear regression can't capture
- For modeling growth patterns, seasonal effects, or complex behaviors
- But be cautious: higher-degree polynomials can overfit the data and reduce generalizability

## Real-Life Example

Imagine you're modeling the trajectory of a ball. The path isn't a straight line—it curves due to gravity. Linear regression would fail here, but polynomial regression (say, degree 2) would capture the parabolic arc beautifully.

## When to Use Linear Regression:

The relationship between variables is linear.

Simplicity and interpretability are important.

The dataset is relatively small, reducing the risk of overfitting.

It serves as a good baseline model for initial analysis.

Example Use Case:

Predicting house prices based on square footage and location. The relationship between these features and price is often linear, making linear regression a straightforward and interpretable choice

## When to Use Polynomial Regression:

The relationship between variables is non-linear.

The dataset has sufficient size to avoid overfitting.

Flexibility is needed to capture intricate patterns in the data.

Residual plots or scatterplots indicate a non-linear trend.

Example Use Case: Modeling electricity consumption based on temperature. The relationship is often U-shaped, with consumption increasing during extreme temperatures and decreasing at moderate temperatures. Polynomial regression effectively captures this non-linear pattern

## Question 6: Implement a Python program to fit a Simple Linear Regression model to the following sample data:

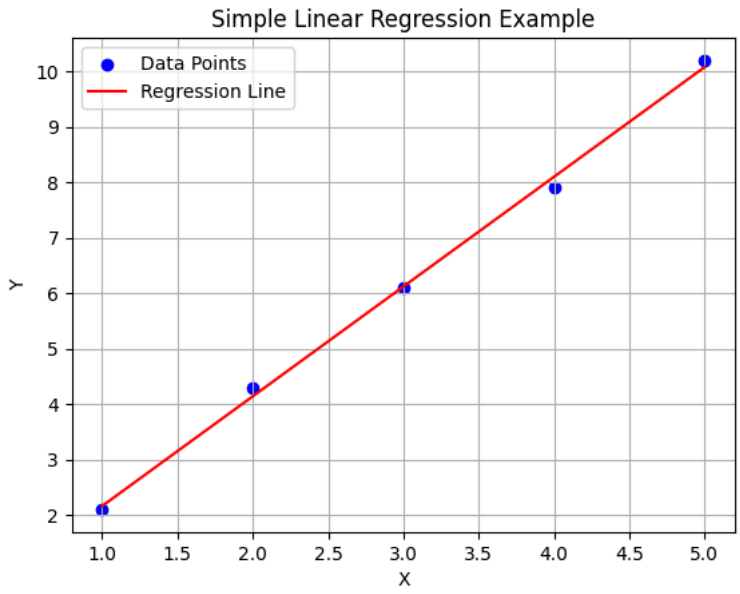
● X = [1, 2, 3, 4, 5]

● Y = [2.1, 4.3, 6.1, 7.9, 10.2]

Plot the regression line over the data points.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
Y = np.array([2.1, 4.3, 6.1, 7.9, 10.2])

# Create and fit the model
model = LinearRegression()
model.fit(X, Y)

# Predictions
Y_pred = model.predict(X)

# Plot
plt.scatter(X, Y, color='blue', label='Data Points')
plt.plot(X, Y_pred, color='red', label='Regression Line')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Simple Linear Regression Example')
plt.legend()
plt.grid(True)
```

```
plt.show()

# Coefficients
model.intercept_, model.coef_[0]
```




```
(np.float64(0.17999999999999794), np.float64(1.9800000000000004))
```

## Question 7: Fit a Multiple Linear Regression model on this sample data:

- Area = [1200, 1500, 1800, 2000]
- Rooms = [2, 3, 3, 4]
- Price = [250000, 300000, 320000, 370000]

Check for multicollinearity using VIF and report the results.

## Answer

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
import statsmodels.api as sm

# Sample data
Area = np.array([1200, 1500, 1800, 2000])
Rooms = np.array([2, 3, 3, 4])
Price = np.array([250000, 300000, 320000, 370000])

# Prepare the data
X = pd.DataFrame({
    'Area': Area,
    'Rooms': Rooms
})
Y = Price

# Fit Multiple Linear Regression
model = LinearRegression()
model.fit(X, Y)

# Add constant for statsmodels VIF
X_with_const = sm.add_constant(X)

# Calculate VIF
vif_data = pd.DataFrame()
vif_data['feature'] = X_with_const.columns
vif_data['VIF'] = [variance_inflation_factor(X_with_const.values, i)
                   for i in range(X_with_const.shape[1])]

# Results
model.intercept_, model.coef_, vif_data
```

```
(np.float64(103157.89473684214),
 array([   63.15789474, 34736.84210526]),
   feature        VIF
 0   const  34.210526
 1    Area   7.736842
 2   Rooms   7.736842)
```

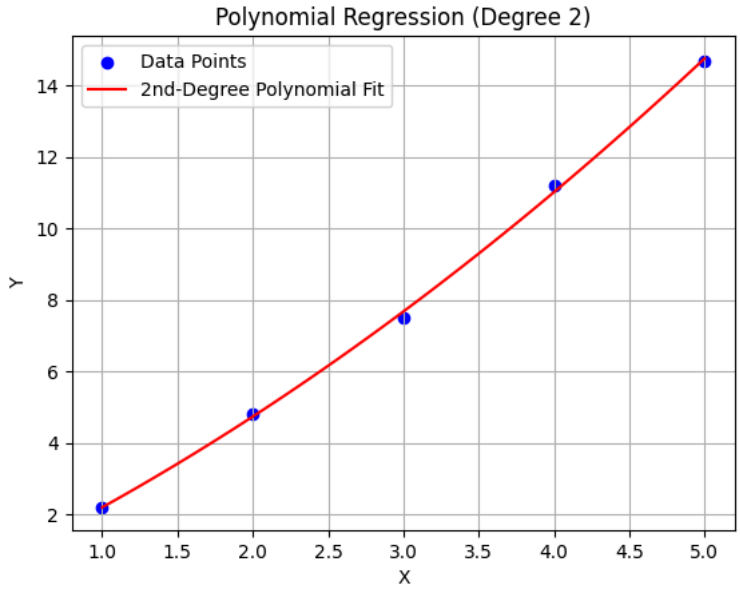
## Question 8: Implement polynomial regression on the following data:

- X = [1, 2, 3, 4, 5]
- Y = [2.2, 4.8, 7.5, 11.2, 14.7]

Fit a 2nd-degree polynomial and plot the resulting curve. (Include your Python code and output in the code box below.)

## Answer

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
Y = np.array([2.2, 4.8, 7.5, 11.2, 14.7])

# Transform to 2nd-degree polynomial features
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Fit the polynomial regression model
model = LinearRegression()
model.fit(X_poly, Y)

# Predictions for smooth curve
X_range = np.linspace(1, 5, 100).reshape(-1, 1)
X_range_poly = poly.transform(X_range)
Y_pred = model.predict(X_range_poly)

# Plot
plt.scatter(X, Y, color='blue', label='Data Points')
plt.plot(X_range, Y_pred, color='red', label='2nd-Degree Polynomial Fit')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Polynomial Regression (Degree 2)')
plt.legend()
plt.grid(True)
plt.show()

# Coefficients
model.intercept_, model.coef_
```




```
(np.float64(0.06000000000000938), array([0.  , 1.94, 0.2 ]))
```

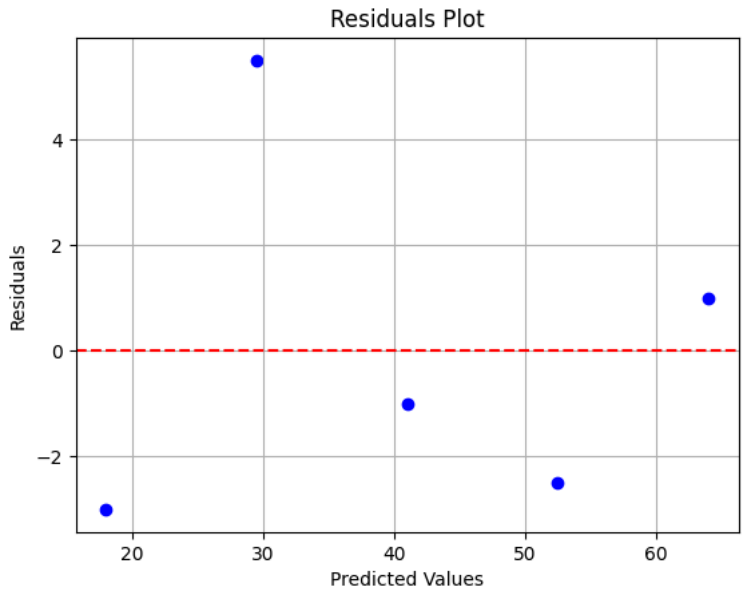
## Question 9: Create a residuals plot for a regression model trained on this data:

- X = [10, 20, 30, 40, 50]
- Y = [15, 35, 40, 50, 65]

Assess heteroscedasticity by examining the spread of residuals. (Include your Python code and output in the code box below.)

### Answer

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Sample data
X = np.array([10, 20, 30, 40, 50]).reshape(-1, 1)
Y = np.array([15, 35, 40, 50, 65])

# Fit Linear Regression
model = LinearRegression()
model.fit(X, Y)

# Predictions and residuals
Y_pred = model.predict(X)
residuals = Y - Y_pred

# Plot residuals
plt.scatter(Y_pred, residuals, color='blue')
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residuals Plot')
plt.grid(True)
plt.show()

# Residuals
residuals
```




```
array([-3. ,  5.5, -1. , -2.5,  1. ])
```

## Question 10: Imagine you are a data scientist working for a real estate company. You need to predict house prices using features like area, number of rooms, and location. However, you detect heteroscedasticity and multicollinearity in your regression model. Explain the steps you would take to address these issues and ensure a robust model.

### Answer:

Goal: Predict house prices using features like area, number of rooms, and location. Challenges: Heteroscedasticity: The spread of residuals is not constant — larger houses may have wider price variability. Multicollinearity: Predictors like area and number of rooms are likely correlated.

## 1. How to Address Heteroscedasticity

Why it's a problem:

Heteroscedasticity doesn't bias coefficients, but it makes standard errors unreliable → bad p-values → misleading significance tests. Common solutions:

1. Transform the dependent variable (Y):

Take the log of house prices. This often stabilizes variance.

New model:

2. Use robust standard errors:

Instead of changing the model, adjust the calculation of standard errors to be robust to heteroscedasticity. E.g., in Python: statsmodels → HC0, HC3 robust covariance estimators.

3. Weighted Least Squares (WLS):

Give less weight to observations with higher variance. More complex, but effective for severe heteroscedasticity.

4. Check for omitted variables:

Sometimes heteroscedasticity appears because important predictors are missing (e.g., neighborhood income level).

## 2. How to Address Multicollinearity

Why it's a problem:

High correlation among predictors inflates standard errors → unstable estimates → hard to interpret individual effects. Common solutions:

1. Check VIF:

Drop or combine predictors with high VIF (typically > 10 is a red flag). Example: If area and rooms are highly correlated, maybe replace them with a composite feature like area per room.

2. Use dimensionality reduction:

Principal Component Analysis (PCA): Combines correlated variables into uncorrelated components. This sacrifices interpretability but can stabilize the model.

3. Regularization:

Use Ridge Regression (L2 penalty) to shrink coefficients of correlated predictors. Or Lasso Regression (L1 penalty) to shrink some coefficients to zero → automatic feature selection. Example: sklearn.linear_model.Ridge or Lasso.

4. Domain knowledge:

Maybe you realize area is more meaningful than rooms, so you drop rooms altogether.

## 3. Model Diagnostics and Validation

After fixing issues:

- Re-check residual plots to confirm variance is more stable.
- Re-calculate VIF to verify multicollinearity is under control.
- Use cross-validation to check generalization.
- Compare model performance ($R^2$, RMSE) with and without fixes.

## 4. Final Deliverable

Provide a clear, interpretable model:

- Communicate the impact of each predictor.
- Explain any transformations used (e.g., log price).
- Justify any features dropped or combined.