Question 1: What is Simple Linear Regression?

Answer: Simple Linear Regression is a statistical technique used to model the relationship between two continuous variables: one **independent variable (X)** and one **dependent variable (Y)**. The goal is to find the best-fitting straight line (called the regression line) that predicts the value of the dependent variable based on the independent variable.

The equation for simple linear regression is:

Y=a+bXY = a + bXY=a+bX

Where:

- **Y** is the predicted value (dependent variable)
- **X** is the input value (independent variable)
- **a** is the Y-intercept of the line
- **b** is the slope of the line

The slope (b) shows how much Y changes for a unit change in X. The intercept (a) is the value of Y when X is 0.

Simple linear regression is used in various fields like economics, biology, and machine learning to predict outcomes and analyze trends.

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

Answer: **Key Assumptions of Simple Linear Regression**

Simple Linear Regression relies on several important assumptions to ensure the validity and accuracy of the model. These key assumptions are:

1. **Linearity**
   There is a linear relationship between the independent variable (X) and the dependent variable (Y).

2. **Independence**
   The observations are independent of each other. The residuals (errors) are not correlated.

3. **Homoscedasticity**
   The variance of the residuals is constant across all values of the independent variable. In other words, the spread of errors should be the same for all levels of X.

4. **Normality of Errors**
   The residuals (differences between observed and predicted values) are normally distributed.

5. **No Multicollinearity** *(not applicable for simple linear regression but important in multiple regression)*
   Since simple linear regression involves only one independent variable, this assumption is automatically satisfied.

These assumptions must be checked to ensure the model's predictions and interpretations are reliable.

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

Answer:

Heteroscedasticity refers to the situation where the variance of the residuals (errors) in a regression model is not constant across all levels of the independent variable. In other words, the spread of the residuals increases or decreases with the value of the predictor.

It is important to address heteroscedasticity because:

- It violates a key assumption of linear regression (homoscedasticity).
- It can lead to inefficient estimates and unreliable hypothesis tests.
- It affects the validity of confidence intervals and p-values.

Common methods to address heteroscedasticity include transforming variables, using weighted least squares, or applying robust standard errors.

Question 4: What is Multiple Linear Regression?

Answer:

Multiple Linear Regression is a statistical method used to model the relationship between one dependent variable and two or more independent variables. It helps in predicting the dependent variable by analyzing how it changes with respect to multiple predictors.

The equation is:

Y=a+b1X1+b2X2+···+bnXnY = a + b_1X_1 + b_2X_2 + \dots + b_nX_nY=a+b1X1+b2X2+···+bnXn

Where:

- **Y** = dependent variable
- **X_1, X_2, ..., X□** = independent variables
- **a** = intercept
- **b_1, b_2, ..., b□** = coefficients of the respective variables

Multiple regression provides a more accurate model when multiple factors influence the outcome.

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

Answer:

Polynomial regression is an extension of linear regression where the relationship between the independent variable and the dependent variable is modeled as an nth-degree polynomial.

Unlike linear regression, which fits a straight line, polynomial regression can model curved relationships.

Example:

Y=a+bX+cX2Y = a + bX + cX^2Y=a+bX+cX2

Key differences:

- **Linear Regression** fits a straight line.

- **Polynomial Regression** fits a curved line by adding higher-degree terms (like X2,X3X^2, X^3X2,X3).

It is useful when the data shows a non-linear trend.

Question 6: Python Code to Fit Simple Linear Regression and Plot

python
Copy code

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

# 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])

# Model
model = LinearRegression()
model.fit(X, Y)

# Predictions
Y_pred = model.predict(X)

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

Question 7: Multiple Linear Regression with VIF Check

python

Copy code

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

# Data
data = pd.DataFrame({
    'Area': [1200, 1500, 1800, 2000],
    'Rooms': [2, 3, 3, 4],
    'Price': [250000, 300000, 320000, 370000]
})

X = data[['Area', 'Rooms']]
y = data['Price']

# Fit model
model = LinearRegression()
model.fit(X, y)

# VIF Calculation
X_const = sm.add_constant(X)
vif_data = pd.DataFrame()
vif_data["Feature"] = X_const.columns
vif_data["VIF"] = [variance_inflation_factor(X_const.values, i)
                   for i in range(X_const.shape[1])]

print(vif_data)
```

Question 8: Polynomial Regression (2nd-Degree)

python
Copy code

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

```python
# 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])

# Polynomial features
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Model
model = LinearRegression()
model.fit(X_poly, Y)
Y_pred = model.predict(X_poly)

# Plot
plt.scatter(X, Y, color='blue', label='Actual Data')
plt.plot(X, Y_pred, color='green', label='Polynomial Curve')
plt.title('Polynomial Regression (Degree 2)')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.grid(True)
plt.show()
```

Question 9: Residuals Plot and Heteroscedasticity Check

python
Copy code

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

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

# Model
model = LinearRegression()
```

```
model.fit(X, Y)
Y_pred = model.predict(X)
residuals = Y - Y_pred

# Plot residuals
plt.scatter(X, residuals, color='purple')
plt.axhline(y=0, color='black', linestyle='--')
plt.title('Residuals Plot')
plt.xlabel('X')
plt.ylabel('Residuals')
plt.grid(True)
plt.show()
```

Interpretation:
If residuals show a funnel shape (widening or narrowing), it suggests heteroscedasticity. If they are randomly spread, the assumption of constant variance holds.

Question 10: Handling Heteroscedasticity and Multicollinearity in Regression

Answer:

As a data scientist, to ensure a robust regression model for predicting house prices, I would take the following steps:

To Handle Heteroscedasticity:

1. **Transform the dependent variable** using log, square root, or Box-Cox transformation.
2. **Use Weighted Least Squares (WLS)** to assign weights to data points with different variances.
3. **Apply Robust Standard Errors** to correct statistical inference without changing the model.

To Handle Multicollinearity:

1. **Check Variance Inflation Factor (VIF):** Drop or combine variables with high VIF.

2. **Use Principal Component Analysis (PCA):** Reduce correlated features to independent components.

3. **Regularization Techniques:** Apply Ridge or Lasso Regression to penalize large coefficients.

By addressing these issues, the model will produce more reliable predictions and accurate statistical interpretations.