Top Linear Regression Interview Questions [Updated 2025]

Last Updated : 23 Jul, 2025

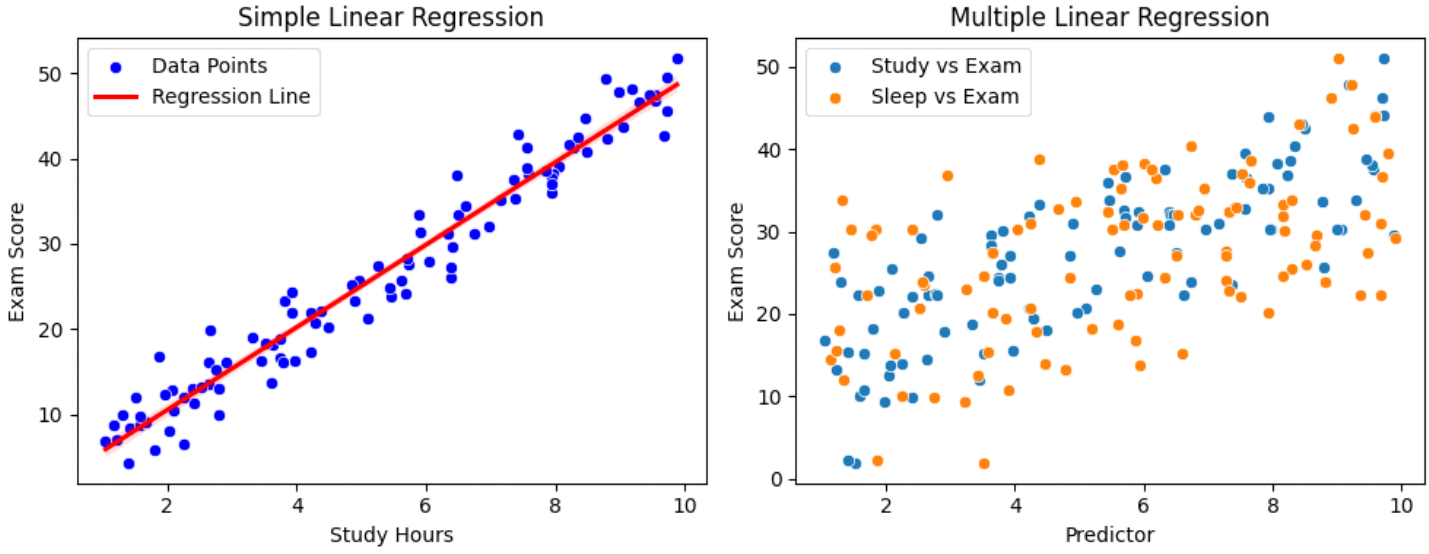
Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data. We have created comprehensive list of the most commonly asked Linear Regression interview questions along with their detailed answers.

1. Difference Between Simple and Multiple Linear Regression?

Simple Linear Regression means there is only **one independent variable** (the factor we're using to make a prediction) and **one dependent variable** (the value we're trying to predict) and we are trying to find the linear relationship between them. **For Example**: Imagine you want to predict a student's final exam score (dependent variable) based on the number of hours they studied (independent variable). Here, there's only one predictor (study hours) and one outcome (exam score).

In **multiple** regression, we have **more than one independent variable**. This allows us to consider more than one factor in predicting the dependent variable.

Let's take an example you're predicting the exam score of students **based on both study hours and the number of hours they sleep. Here, there are two predictors: study hours and sleep hours.**



Simple and Multiple Linear Regression,

2. What are the assumptions of linear regression?

The key assumptions of linear regression are Linearity , Independence ,Homoscedasticity and Normality. Linearity means **t**he relationship between the independent and dependent variables should be linear.

- In Independence we check that each observation (data point) should not depend on other observations.
- Homoscedasticity means the spread (variance) of the errors (residuals) should be consistent across all values of the independent variable(s).
- Normality means Errors (residuals) should follow a normal distribution (bell-shaped curve). If you're predicting a car's mileage based on its engine size, and the prediction **errors mostly fall around zero (with fewer extreme errors), the residuals are normal**.

If the errors are **unevenly distributed (e.g., skewed to one side), this assumption is violated.**

3. What Does a P-Value Indicate?

A **p-value** is a statistical measure used to assess the **significance** of a feature or variable in a machine learning model in regression analysis.

- A **low p-value** (typically below 0.05) indicates that the feature has a **significant relationship** with the target variable and is likely contributing meaningfully to the model's predictions.

- A **high p-value** suggests that the feature **does not significantly contribute** to the model and might be irrelevant or redundant.

While p-values are valuable in traditional statistical modeling (like linear regression), in **machine learning**, we focus on **model performance metrics** (e.g., accuracy, precision, recall) rather than relying solely on p-values to judge a feature's importance.

4. What Is the Role of the Intercept?

The intercept in linear regression **represents the predicted value of the dependent variable when all independent variables are equal to zero**. It is a **key parameter in the regression equation** and helps define the position of the regression line on the graph.

- In **simple linear regression**, it is the value of Y when X=0. For example, if a model predicting exam scores has an intercept of 65, it means a student who studies 0 hours is predicted to score 65.
- In **multiple linear regression**, the intercept is the predicted value of Y when all predictors are zero, though its interpretation may not always be meaningful depending on the context.

Even if the intercept lacks practical interpretability (e.g., predicting height at zero weight), it is still essential for making accurate predictions.

5. How to Handle Categorical Variables in Linear Regression?

To handle categorical variables in linear regression, we first need to convert them into numerical formats **since regression models cannot process non-numeric data**. The two main approaches are **one-hot encoding and label encoding:**

- **One-hot encoding**: creates new **binary columns for each category of a categorical variable**.

For example, if you have a variable called "Color" with three possible values: Red, Blue, and Green, one-hot encoding will create three new columns: "Color_Red", "Color_Blue", and "Color_Green". Each row will have a value of 1 in the column corresponding to its color and 0 in the others.

Can you tell me one **drawback of one-hot encoding? It** Increases **dimensionality when there are many categories (curse of dimensionality).**

- **Label Encoding: It** assigns a unique integer to each category of a variable. Using the same "Color" example, you could label Red as 0, Blue as 1, and Green as 2.

However, for nominal data, **it may introduce unintended ordinal relationships (e.g., the model might interpret Red < Blue < Green).**

Tip : We must also be cautious about the **dummy variable trap in one-hot encoding by dropping one category to avoid multicollinearity**. For high-cardinality variables, target encoding or grouping rare categories can help reduce dimensionality.

7. What is the purpose of residual plots?

Residual plots are used to **assess the validity of a linear regression model by visualizing the differences between observed and predicted values (residuals)**. They help validate key assumptions:

- **Homoscedasticity**: Residuals should have constant variance. A random scatter in the residual plot indicates homoscedasticity, while patterns (e.g., funnel-shaped) suggest heteroscedasticity.
- **Linearity**: A random pattern supports the assumption of linearity. Curved patterns indicate that a non-linear model might be more appropriate.
- **Independence**: Residuals should not exhibit systematic trends, such as autocorrelation in time-series data.

Outliers: Residual plots help identify outliers or influential points that may disproportionately affect the model.

For example, residuals forming a **U-shaped pattern, suggests the model is missing non-linear relationships and needs transformation or additional features.**

8. What steps would you take to evaluate the performance of a linear regression model?

We use the different types of performance metrics to evaluate a linear regression model:

- **R-squared**: This tells us how much of the changes in the dependent variable can be explained with the help of independent variables. if values closer to 1 showing a better fit.
- **Residual Plots**: These plots display the differences between actual and predicted values. A random pattern in residuals suggests that the model's assumptions are met, while non-random patterns may indicate issues.
- **Mean Squared Error (MSE)**: MSE measures the average squared differences between actual and predicted values. A lower MSE indicates better model performance.
- **Adjusted R-squared**: adjusted R-squared based on the number of predictors,

making it useful for comparing models with different numbers of predictors.

9. What is the significance of R-squared- why is it commonly used?

R-squared (R^2) is a statistical measure that helps us understand how well a model understands and explain the relationship between independent variables and dependent variable (the outcome we are trying to predict).
R^2 value can range from 0 to 1. For Example: An R^2 of 0.85 means that 85% of the variation in the dependent variable is explained by the independent variables.
More practical scenarios that are been asked on r-squared:
1. If a model has a high R-squared but performs poorly on test data, what could be the possible reasons?

A high R-squared value **does not always indicate a good model**, especially if it performs poorly on test data. The possible reasons can have:

- **Overfitting:** The model may be too complex, capturing noise or random patterns in the training data rather than the true underlying relationships.
- **Biased R-squared Estimate:** R-squared can be a biased metric when calculated from a sample.
- **Multicollinearity:** High correlations among independent variables can distort the coefficients and make the model unstable, despite a high R-squared value.

2. Can a low R-squared still be meaningful in certain applications? Provide examples.

Yes, a low R-squared can still be meaningful in certain contexts:

- **Fields with High Variability:** In areas like social sciences or human behavior studies, where variability is inherently high and difficult to explain, low R-squared values are common.
- **Significant Predictors:** Even with a low R-squared, statistically significant coefficients can indicate meaningful relationships between variables. For example, in medical research, identifying factors that slightly influence disease outcomes can be critical even if the overall model fit is low.

10. How to Handle Missing Data in Linear Regression?

When we have to handle the missing data in linear regression we basically use three approaches:

1. **Imputation**: It Replace missing values with the mean, median, or mode of that feature. For example, fill in missing weights with the average weight.
2. **Dropping Rows/Columns**: Remove rows or columns with missing data if they are minimal, but do this cautiously to avoid losing important information.
3. **Prediction Models**: Use other variables to predict and fill in missing values, ensuring a more complete dataset for analysis.

11. How do you determine if your linear regression model is overfitting?

Overfitting happens when your model **performs exceptionally well on the training data but poorly on new, unseen data. *To identify this, compare the training and test performance metrics, such as R^2 or error*.**
Techniques like cross-validation, simplifying the model, or adding regularization can help address this issue.

- If your model has a **very high R^2** or very low error on the training data but a **much lower R^2** or higher error on the test data, it's a sign of overfitting.
- **Cross-validation**: Using cross-validation (splitting the data into multiple subsets and testing the model on each one) helps to check if the model is consistently performing well on different data sets. If performance varies a lot, overfitting may be happening.
- **Simplifying the Model**: If your model has too many features (predictors), it might be overfitting. Try reducing the number of features and see if the model performs better on unseen data.
- **Regularization**: Techniques like **Ridge** or **Lasso regression** can help prevent overfitting by adding a penalty to overly large model coefficients, encouraging the model to be simpler.

12. How do you handle multicollinearity in a dataset when applying linear regression?

We can handle the multicollinearity with help of:

- Correlation matrix or Variance Inflation Factor (VIF) **to identify and remove variables with high correlation.**
- Utilizing techniques like **Principal Component Analysis (PCA)** to merge correlated variables into a single component.
- Implement **Ridge Regression (L2)** to shrink coefficients or **Lasso Regression (L1)** to eliminate less important predictors.
- Collect more data to enhance variability among predictors, which can stabilize coefficient estimates.
- Use **stepwise regression** to iteratively add or remove predictors based on their significance.

13. How do you deal with highly skewed data in linear regression?

When working with linear regression, skewed data can violate the assumption of normality in the residuals, leading to inaccurate predictions**. E**ffective ways to handle such data:

1. Transformations:

- **Log Transformation:** Apply the natural logarithm to the skewed variable to reduce its variability and bring it closer to a normal distribution effective for right-skewed data.
- **Square Root or Cube Root Transformation:** These transformations can reduce skewness, for small values. Cube root transformations are stronger and work on both zero and negative values.
- **Box-Cox transformation**: More general approach that finds the optimal power transformation to normalize the data. However, it requires all values to be positive.

2. **Removing Outliers**: that disproportionately affect the distribution. It can reduce skewness but should be done cautiously to avoid losing valuable information.
3. **Generalized Linear Models (GLMs)**: For specific cases like right-skewed response variables, GLMs with gamma or inverse Gaussian distributions can effectively model the data without requiring transformations.

14. What is biased and unbiased estimate in Linear Regression

An unbiased estimate in linear regression **accurately reflects the true relationship in the data without systematic error, like the coefficients from OLS regression under ideal conditions**.

A **biased estimate, like those from regularized models (e.g., Ridge), intentionally deviates from the true value to reduce overfitting and improve model performance on new data.**

The choice depends on the trade-off between bias and variance for better generalization.

15. What is the difference between a parametric and non-parametric regression model?

- **Parametric Models**: Assumes a specific functional form for the relationship between the input features and the output (e.g., linear, polynomial). The model learns a fixed number of parameters during training. **Examples**: Linear regression, logistic regression, polynomial regression.
- **Non-Parametric Models**: Makes no assumptions about the functional form of the relationship. It learns the structure of the data dynamically and can adapt to more complex patterns. **Examples:** k-Nearest Neighbors (k-NN), decision trees, support vector regression (with non-linear kernels).

16. What is the difference between forward selection and backward elimination in feature selection for linear regression?

Forward Selection:

1. Begin with an empty model (no predictors).
2. Test **each feature individually** and add the one that improves the model the most **(e.g., reduces error or increases adjusted R^2).**
3. Repeat **until adding features no longer improves the model significantly.**

Preferred when the dataset has many irrelevant features and you want to build the model incrementally.

Backward Elimination:

1. Start with all features in the model.
2. Test the significance of each feature (e.g., using p-values in linear regression).
3. Remove the least significant feature.
4. Repeat until all remaining features are statistically significant.

Useful when most features are expected to be relevant or when the total number of features is manageable.

17. How do you check for heteroscedasticity in a linear regression model?

We plot residuals (actual - predicted) vs. predicted values ; if the **spread of residuals varies (e.g., forms a pattern or cone shape), heteroscedasticity may exist.**

18. What is the difference between ridge and lasso regression?

Ridge and Lasso regression differ in how they handle regularization and feature selection:

- **Ridge Regression** adds an L2 penalty (squared coefficients) to the loss function. It shrinks coefficients to reduce overfitting but doesn't set any to zero, so it keeps all features. It's ideal when all features are important but need their influence reduced.
- **Lasso Regression** adds an L1 penalty (absolute coefficients). This not only shrinks coefficients but can set some to exactly zero, effectively performing feature selection. It's useful when some features are irrelevant.

In short, Ridge is about shrinkage, while Lasso combines shrinkage with feature selection. The choice depends on whether you expect some features to be irrelevant.

19. How can you perform k-fold cross-validation to evaluate a linear regression model in Python?

K-fold cross-validation splits the dataset into k parts and tests the model k times, each time using a different fold as the test set.

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
import numpy as np
X = [[1, 2], [3, 4], [5, 6], [7, 8]]
y = [1, 2, 3, 4]
model = LinearRegression()
scores = cross_val_score(model, X, y, cv=4,
scoring='neg_mean_squared_error')
print(f"Cross-validation scores: {scores}")
print(f"Average MSE: {np.mean(scores)}")
```

20. What is the impact of outliers and how would you address them?

Outliers can **distort the model** by heavily influencing the slope of the regression line. Outliers can **inflate the variance** and affect the accuracy of the model, making it overly sensitive to extreme values.

We use visualization techniques **like box plots or scatter plots, or statistical methods like Z-scores or IQR (Interquartile Range) to detect outliers.**

- If outliers are due to errors or aren't relevant to the problem, we remove them.
- Sometimes applying a transformation (like a log or square root) can reduce the influence of extreme values.
- If outliers are unavoidable, we use regression techniques that are less sensitive to them, such as **Ridge**, **Lasso**, or **Huber regression.**

21. What is bias variance tradeoff in Linear Regression?

The bias-variance trade-off explains how the complexity of a model affects its accuracy.

- **Bias** is the error caused by oversimplifying the model, like assuming a straight-line relationship when the actual pattern is more complex. This leads to underfitting.
- **Variance**, on the other hand, is the error caused by the model being too sensitive to the training data, which happens when the model is too complex and starts capturing noise instead of patterns that causes overfitting.

22. How do you handle categorical variables in linear regression using Python?

```
df_encoded = pd.get_dummies(df, columns=['Category'],
drop_first=True)

# Split data
X = df_encoded.drop('Target', axis=1)
y = df_encoded['Target']
```

23. How does gradient descent work in the context of linear regression?

In linear regression, our goal is to find the best-fitting line that minimizes the **cost function**, typically the **Mean Squared Error (MSE)**. Gradient descent is an optimization algorithm used to minimize this cost function by iteratively updating the model's parameters (the slope and intercept).

- We start **with random values for the parameters (slope and intercept).**
- The **gradient is the derivative of the cost function with respect to the parameters**. It tells us how much the cost function will change if we adjust the parameters by a small amount. **Essentially, it shows us the direction in which the cost function is increasing or decreasing.**
- Using the gradient, we update the parameters by taking small steps in the direction that reduces the cost.

This process is repeated until the parameters converge, meaning the updates become very small and the cost function reaches its minimum value.

Therefore, Gradient descent helps find the optimal parameters for the linear

regression model by iteratively adjusting them to minimize the error.

From <https://www.geeksforgeeks.org/machine-learning/top-linear-regression-interview-questions/>