**Akhilesh Kumar LR\_Assignment (Only Answer)**

**Section - 1 : Data Inspection and cleaning**

Q.1. - import pandas as pd

file\_path = 'insurance.csv'

data = pd.read\_csv(file\_path)

print(data.head())

print("\nColumns in the dataset:")

print(data.columns)

print("\nMissing values:")

print(data.isnull().sum())

print("\nData types:")

print(data.dtypes)

**Output -**

age sex bmi children smoker region charges

0 19 female 27.900 0 yes southwest 16884.92400

1 18 male 33.770 1 no southeast 1725.55230

2 28 male 33.000 3 no southeast 4449.46200

3 33 male 22.705 0 no northwest 21984.47061

4 32 male 28.880 0 no northwest 3866.85520

Columns in the dataset:

Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')

Missing values:

age 0

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

Data types:

age int64

sex object

bmi float64

children int64

smoker object

region object

charges float64

dtype: object

Q.2. -import pandas as pd

df = pd.read\_csv('insurance.csv')

print("Missing values before handling:")

print(data.isnull().sum())

for column in data.select\_dtypes(include=['float64', 'int64']).columns:

data[column] = data[column].fillna(data[column].mean())

for column in data.select\_dtypes(include=['object']).columns:

data[column] = data[column].fillna(data[column].mode()[0])

print("\nDuplicate records before handling:", data.duplicated().sum())

data = data.drop\_duplicates()

print("\nMissing values after handling:")

print(data.isnull().sum())

print("\nDuplicate records after handling:", data.duplicated().sum())

**Output -**

Missing values before handling:

age 0

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

Duplicate records before handling: 1

Missing values after handling:

age 0

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

Duplicate records after handling: 0

**Section - 2 : Exploratory Data Analysis**

Q.3. - import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

data = pd.read\_csv('insurance.csv')

numerical\_columns = data.select\_dtypes(include=['float64', 'int64']).columns

for column in numerical\_columns:

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

sns.histplot(data[column], kde=True)

plt.title(f'{column} - Histogram')

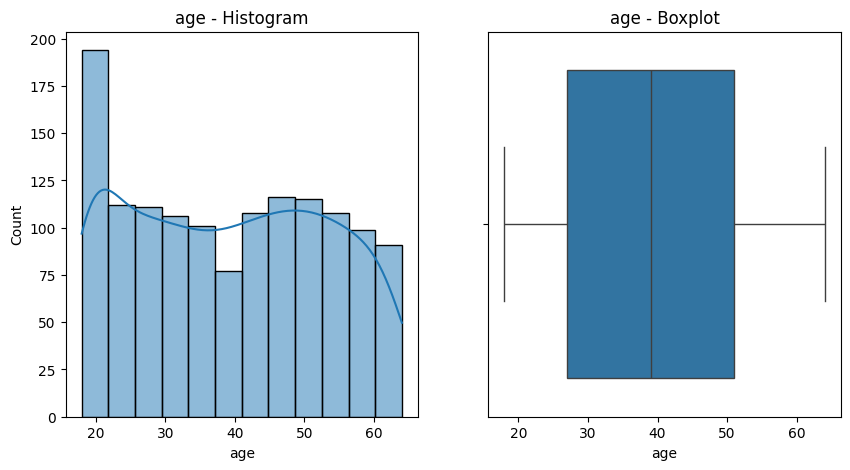
plt.subplot(1, 2, 2)

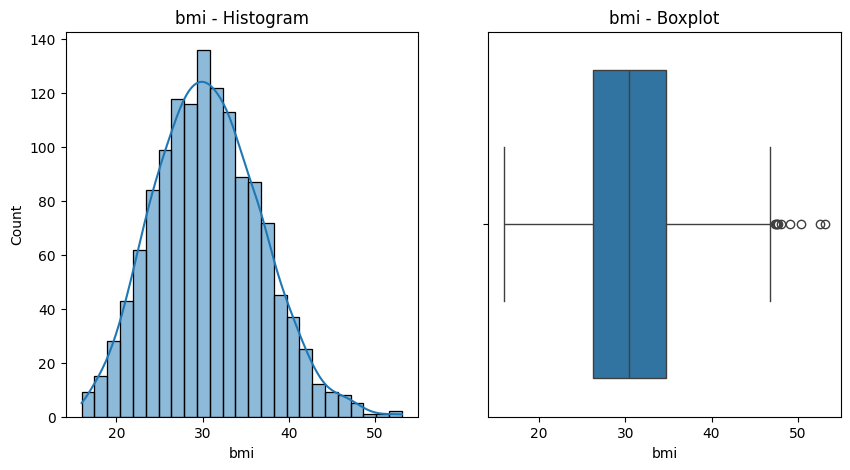
sns.boxplot(x=data[column])

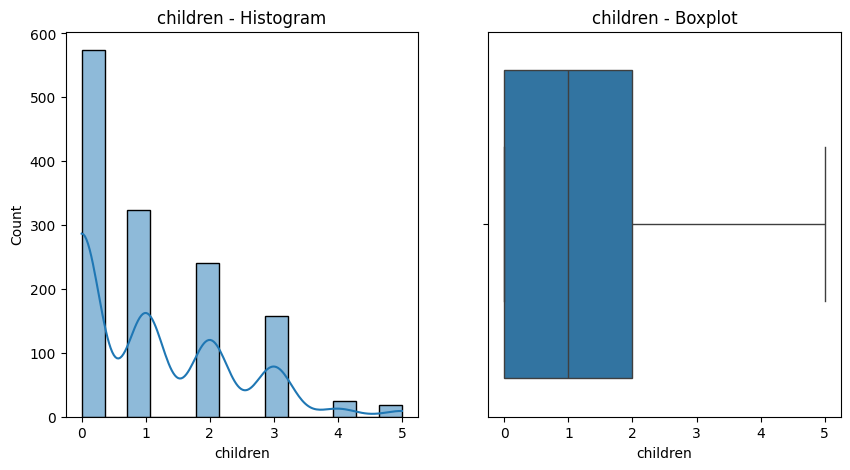
plt.title(f'{column} - Boxplot')

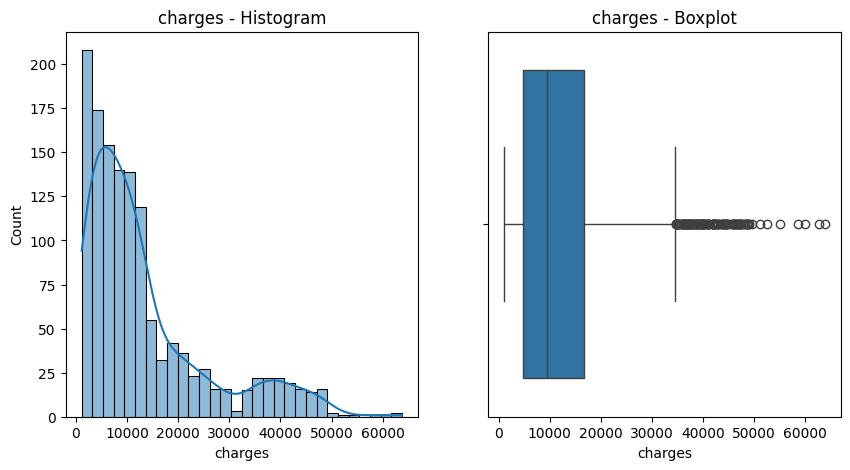
plt.show()

**Output -**

****

****

****

****

Q.4. - import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('insurance.csv')

independent\_variables = ['age', 'bmi', 'children', 'smoker', 'region']

target\_variable = 'charges'

for variable in independent\_variables:

plt.figure(figsize=(8, 5))

sns.scatterplot(x=data[variable], y=data[target\_variable])

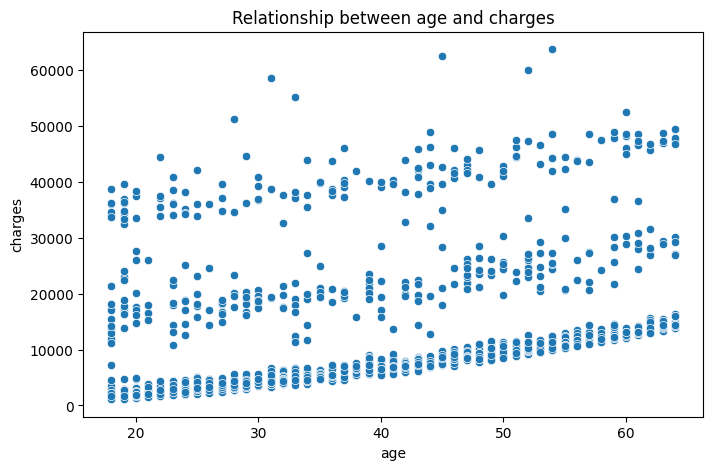
plt.title(f'Relationship between {variable} and {target\_variable}')

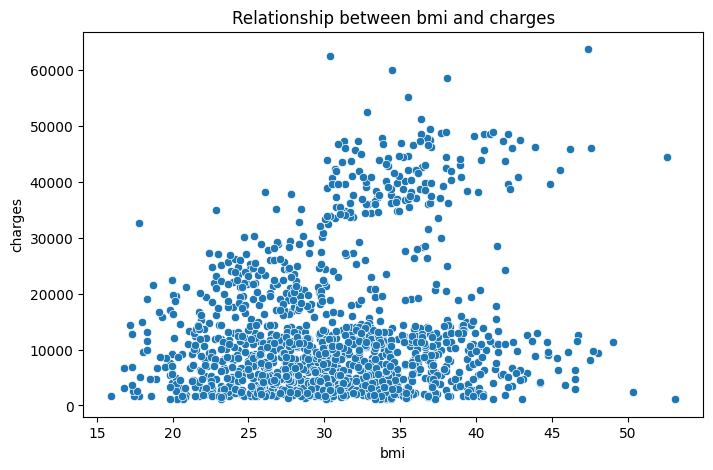
plt.xlabel(variable)

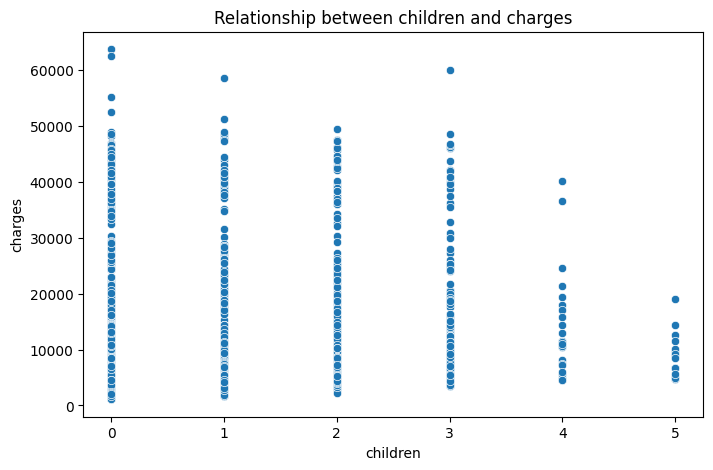
plt.ylabel(target\_variable)

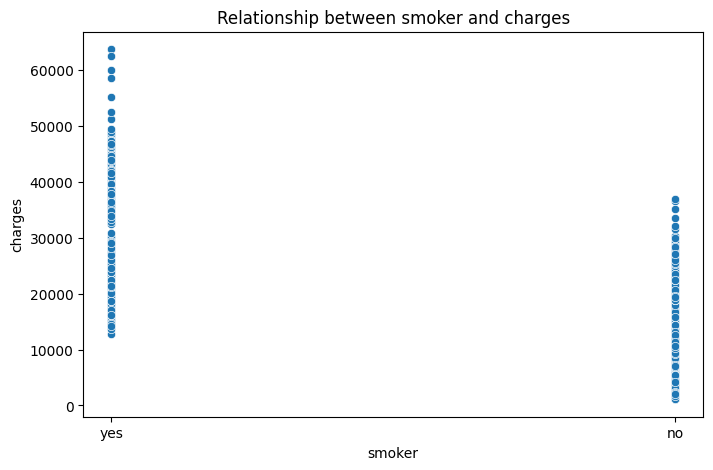
plt.show()

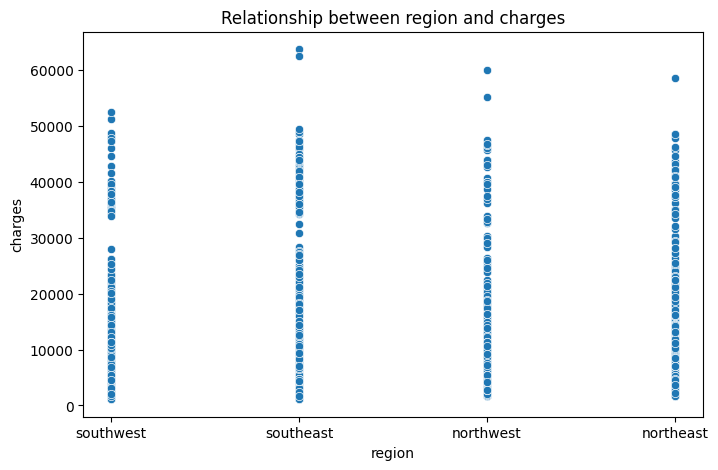
**Output -**

****

****

****

****

****

Q.5. - import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

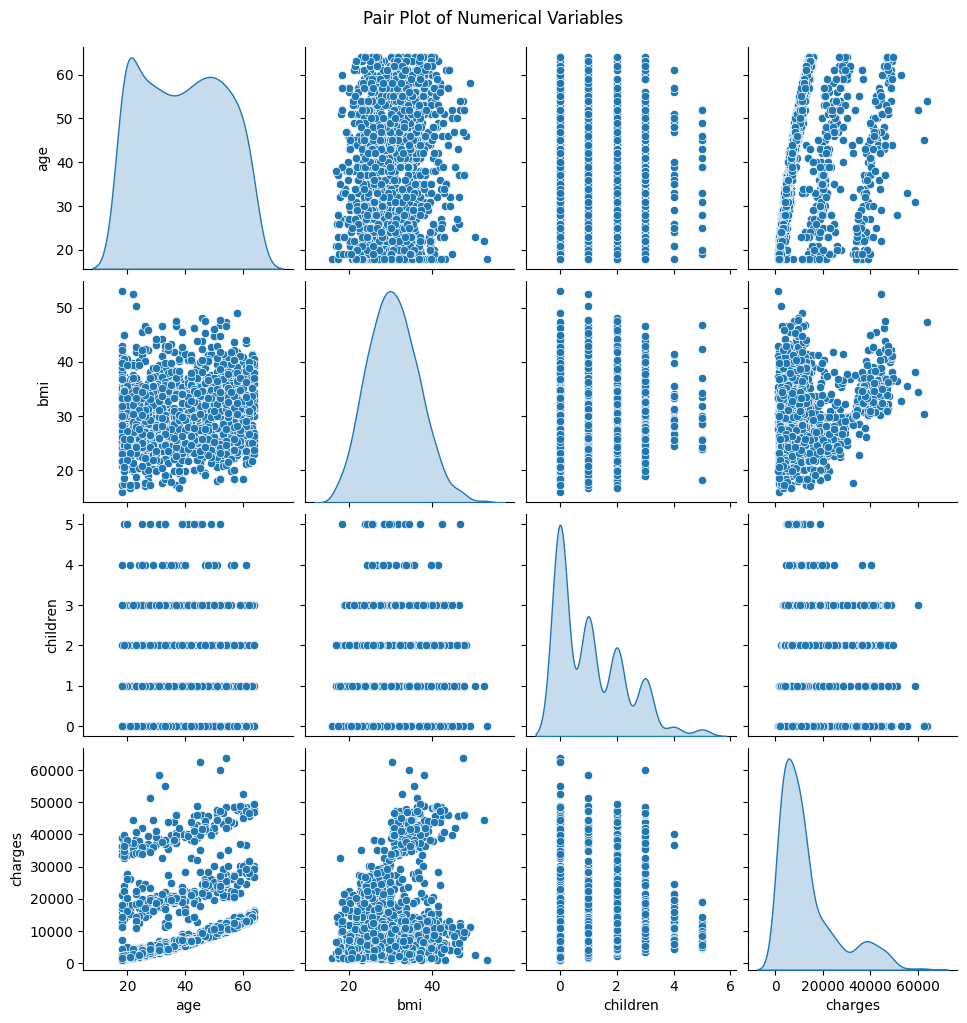
data = pd.read\_csv('insurance.csv')

sns.pairplot(data[numerical\_columns], diag\_kind='kde')

plt.suptitle('Pair Plot of Numerical Variables', y=1.02)

plt.show()

**Output -**

****

Q.6. - import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('insurance.csv')

correlation\_matrix = data[numerical\_columns].corr()

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.title('Correlation Matrix Heatmap')

plt.show()

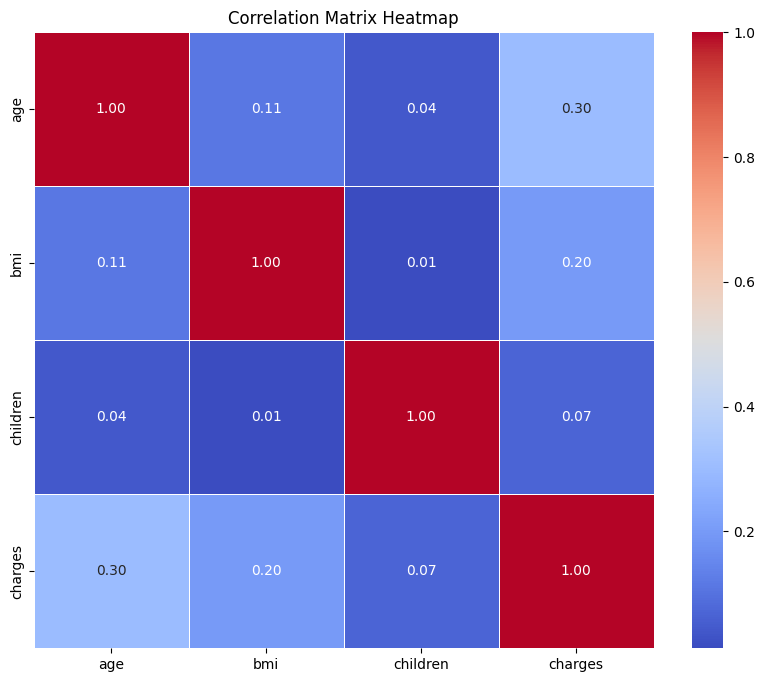
threshold = 0.75

high\_correlation = correlation\_matrix[(correlation\_matrix > threshold) & (correlation\_matrix < 1.0)]

print("Highly correlated features (correlation > 0.75):")

print(high\_correlation.dropna(how='all').dropna(axis=1, how='all'))

**Output -**

****

Highly correlated features (correlation > 0.75):

Empty DataFrame

Columns: [ ]

Index: [ ]

**Section 3 : Model Building and Evaluation**

Q.7. - from sklearn.model\_selection import train\_test\_split

X = df.drop("target\_column", axis=1)

y = df["target\_column"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print("Training set size:", X\_train.shape)

print("Testing set size:", X\_test.shape)

**Output -**

Q.8. - import numpy as np

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

data = pd.DataFrame({

'X': [1, 2, 3, 4, 5],

'Y': [2, 4, 6, 8, 10]

})

X = data[['X']] # Feature

Y = data['Y'] # Target

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, Y\_train)

print(f"Coefficient: {model.coef\_[0]}")

print(f"Intercept: {model.intercept\_}")

**Output -**

Coefficient: 2.0

Intercept: 0.0

Q.9. - from sklearn.metrics import r2\_score, mean\_absolute\_error, mean\_squared\_error

import numpy as np

Y\_pred = model.predict(X\_test)

r\_squared = r2\_score(Y\_test, Y\_pred)

mae = mean\_absolute\_error(Y\_test, Y\_pred)

mse = mean\_squared\_error(Y\_test, Y\_pred)

rmse = np.sqrt(mse)

print(f"R-squared: {r\_squared}")

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Mean Squared Error (MSE): {mse}")

print(f"Root Mean Squared Error (RMSE): {rmse}")

**Output -**

R-squared: nan

Mean Absolute Error (MAE): 0.0

Mean Squared Error (MSE): 0.0

Root Mean Squared Error (RMSE): 0.0

Q.10. - 1. Mean Absolute Error (MAE)

* What it means: The average absolute difference between actual and predicted values.
* Why it's important: It's intuitive—if MAE = 5, the model is wrong by 5 units on average.
* Ideal value: Closer to 0 is better.
* What it tells you about accuracy: Lower MAE means the model consistently makes small errors, indicating better accuracy.

2. Mean Squared Error (MSE)

* What it means: The average of the squared differences between actual and predicted values.
* Why it's important: Penalizes large errors more than small ones (because of squaring).
* Ideal value: Again, lower is better.
* What it tells you about accuracy: High MSE indicates the model makes occasional large mistakes. Good for spotting outlier errors.

3. Root Mean Squared Error (RMSE)

* What it means: Square root of MSE, brings the error back to the same scale as the data.
* Why it's important: Easier to interpret than MSE, gives a sense of how far off predictions are in the units of the target variable.
* Ideal value: The closer to 0, the better.
* What it tells you about accuracy: Like MAE, but more sensitive to big errors. A high RMSE = model struggles with larger mistakes.

4. R² Score (Coefficient of Determination)

* What it means: Proportion of the variance in the dependent variable that is predictable from the independent variables.
* Why it's important: Shows how well the model fits the data overall.
* Range: Between 0 and 1 (can be negative if the model is really poor).
* Ideal value: Closer to 1.
* What it tells you about accuracy:
* 0.90 – 1.0 → Excellent fit
* 0.70 – 0.89 → Good
* 0.50 – 0.69 → Moderate
* < 0.50 → Weak

**Section 4 - Model Interpretation and Conclusion**

Q.11. - Key Features and Their Significance

Age:- Older individuals generally incur higher insurance charges. As age increases, health risks tend to rise, leading to higher premiums.

BMI (Body Mass Index):- A higher BMI indicates a greater risk of health complications (like obesity-related issues), leading to higher charges.

Particularly significant in policies sensitive to lifestyle and health.

Children (Number of Dependents):- The number of dependents can influence the charges in some policies, reflecting the insured family's needs.

Region:- Geographical location impacts charges due to varying healthcare costs, regulations, and risk factors across regions.

Smoking Status:- Smokers typically face significantly higher charges compared to non-smokers due to the well-documented health risks associated with smoking (e.g., heart disease, lung cancer).

This is often the most impactful feature due to its strong correlation with higher medical expenses.

Sex:- While sex may slightly impact charges due to statistical health trends, its effect is typically less significant compared to other features.

Q.12. - 1. Linearity Assumption :

Linear regression assumes a linear relationship between the features (independent variables) and the target (dependent variable). If the dataset contains non-linear patterns, the model may underperform.

2. Multicollinearity :

If features are highly correlated, the model struggles to distinguish the individual impact of each feature, which can lead to unstable coefficients and reduced interpretability.

3. Outliers :

Linear regression is sensitive to outliers, as they can disproportionately influence the model, skewing predictions and reducing accuracy.

4. Feature Independence :

Linear regression assumes that the features are independent. If there are interactions between features (e.g., smoking status and BMI combined affecting insurance charges), the model won’t capture them effectively.

5. Homoscedasticity :

The model assumes that the variance of the errors (residuals) is constant across all levels of the independent variables. If this isn’t true (heteroscedasticity), predictions may be less reliable.

6. Normal Distribution of Errors :

Linear regression assumes that errors are normally distributed. Deviations from this assumption can affect the reliability of statistical tests associated with the model.