```
# Lab 4: Linear Regression
# 0. Install and Import Libraries

# Install ucimlrepo if not done
!pip install ucimlrepo

from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.7-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in /usr/local/lib/python3.12/dist-packages (from ucimlrepo) (2025.1
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0-
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlrepo
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.0.0->ucimlre
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->panda
Downloading ucimlrepo-0.0.7-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.7
```

```
# Lab 4: Linear Regression
# 1. Load Dataset and Inspect

from ucimlrepo import fetch_ucirepo

chess_king_rook_vs_king_pawn = fetch_ucirepo(id=22)

# Features and target extraction
X = chess_king_rook_vs_king_pawn.data.features
y = chess_king_rook_vs_king_pawn.data.targets

# Metadata and variable information
print("Metadata:\n", chess_king_rook_vs_king_pawn.metadata)
print("\nVariables:\n", chess_king_rook_vs_king_pawn.variables)

# Display sample features and targets
print("Sample features:\n", X.head())
print("Sample targets:\n", y.head())
```

```
2   bkon8  Feature  Categorical        None        None  None             no
3   bkona  Feature  Categorical        None        None  None             no
4   bkspr  Feature  Categorical        None        None  None             no
5   bkxbq  Feature  Categorical        None        None  None             no
6   bkxcr  Feature  Categorical        None        None  None             no
7   bkxwp  Feature  Categorical        None        None  None             no
8   blxwp  Feature  Categorical        None        None  None             no
9   bxqsq  Feature  Categorical        None        None  None             no
10  cntxt  Feature  Categorical        None        None  None             no
11  dsopp  Feature  Categorical        None        None  None             no
12  dwipd  Feature  Categorical        None        None  None             no
13  hdchk  Feature  Categorical        None        None  None             no
14  katri  Feature  Categorical        None        None  None             no
15  mulch  Feature  Categorical        None        None  None             no
16  qxmsq  Feature  Categorical        None        None  None             no
17  r2ar8  Feature  Categorical        None        None  None             no
18  reskd  Feature  Categorical        None        None  None             no
19  reskr  Feature  Categorical        None        None  None             no
20  rimmx  Feature  Categorical        None        None  None             no
21  rkxwp  Feature  Categorical        None        None  None             no
22  rxmsq  Feature  Categorical        None        None  None             no
23  simpl  Feature  Categorical        None        None  None             no
24  skach  Feature  Categorical        None        None  None             no
25  skewr  Feature  Categorical        None        None  None             no
26  skrxp  Feature  Categorical        None        None  None             no
27  spcop  Feature  Categorical        None        None  None             no
28  stlmt  Feature  Categorical        None        None  None             no
29  thrsk  Feature  Categorical        None        None  None             no
30  wkcti  Feature  Categorical        None        None  None             no
31  wkna8  Feature  Categorical        None        None  None             no
32  wknck  Feature  Categorical        None        None  None             no
```

Toggle Gemini

```
f     f     f     f     t     f     t     f     f     f     f  ...     f
f     f     f     f     f     f     f     f     t     f     f  ...     f
f     f     f     f     f     f     f     f     f     f     f  ...     f

  skrxp spcop stlmt thrsk wkcti wkna8 wknck wkovl wkpos
f     f     f     f     f     f     f     t     t     n
f     f     f     f     f     f     f     t     t     n
f     f     f     f     f     f     f     t     t     n
f     f     f     f     f     f     f     t     t     n
f     f     f     f     f     f     f     t     t     n

[5 rows x 35 columns]
Sample targets:
   wtoeg
f   won
f   won
f   won
f   won
f   won
```

```
# Lab 4: Linear Regression
# 2. Train-Test Split

# For simple regression, select one independent variable (e.g., first column)
X_single = X.iloc[:, [0]]  # Replace 0 with appropriate feature index for regression relevance

# Apply one-hot encoding to the selected feature
X_single_encoded = pd.get_dummies(X_single, drop_first=True)

# Encode the target variable
y_encoded = y['wtoeg'].apply(lambda x: 1 if x == 'won' else 0)


X_train, X_test, y_train, y_test = train_test_split(X_single_encoded, y_encoded, test_size=0.2, random_state=42)
```

```
# Lab 4: Linear Regression
# 3. Model Fitting

reg = LinearRegression()
reg.fit(X_train, y_train)

print("Intercept (B0):", reg.intercept_)
print("Slope (B1):", reg.coef_)
```

```
Intercept (B0): 0.5200506542845083
Slope (B1): [-0.00133408]
```

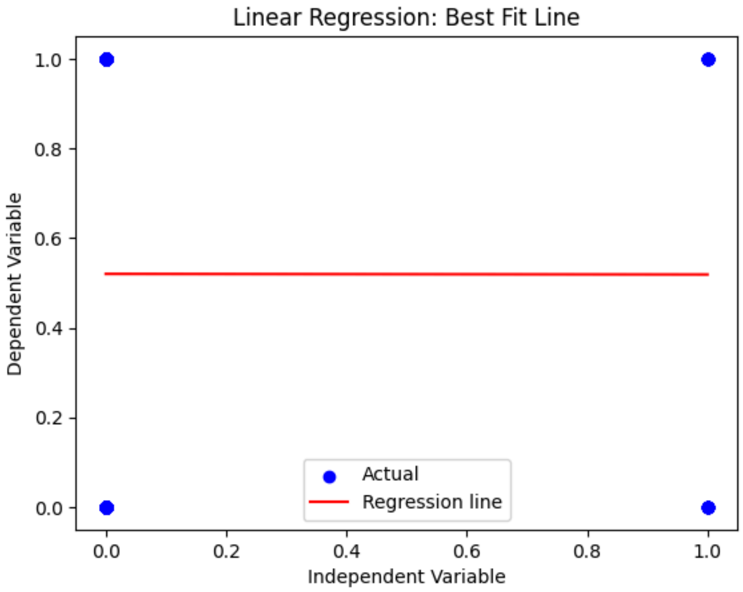
Toggle Gemini

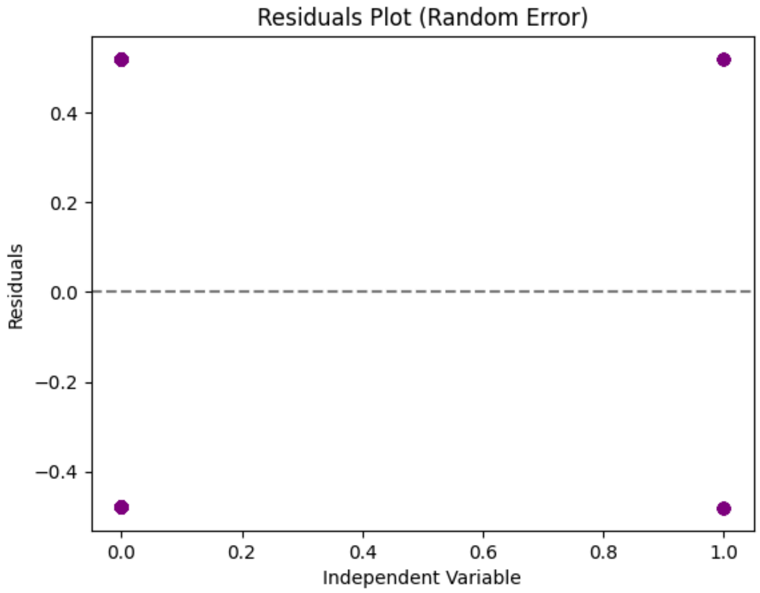
```
# Lab 4: Linear Regression
# 4. Prediction, Visualization, and Residuals

# Predict on train and test set
y_pred = reg.predict(X_test)

# Scatter plot predicted vs actual
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.plot(X_test, y_pred, color='red', label='Regression line')
plt.xlabel('Independent Variable')
plt.ylabel('Dependent Variable')
plt.title('Linear Regression: Best Fit Line')
plt.legend()
plt.show()

# Residuals
residuals = y_pred - y_test
plt.scatter(X_test, residuals, color='purple')
plt.axhline(y=0, linestyle='--', color='gray')
plt.xlabel('Independent Variable')
plt.ylabel('Residuals')
plt.title('Residuals Plot (Random Error)')
plt.show()
```





Toggle Gemini

```
Start coding or generate with AI.
```