In [ ]:
```
# Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

In [ ]:
```
# Loading data
df = pd.read_csv('temperatures.csv')
print("(row,column)",df.shape)
print("The first 5 rows are : ",df.head())
```

```
(row,column) (117, 18)
The first 5 rows are :     YEAR    JAN    FEB    MAR    APR    MAY    JUN    JU
L    AUG    SEP    OCT  \
0  1901  22.40  24.14  29.07  31.91  33.41  33.18  31.21  30.39  30.47  29.97
1  1902  24.93  26.58  29.77  31.78  33.73  32.91  30.92  30.73  29.80  29.12
2  1903  23.44  25.03  27.83  31.39  32.91  33.00  31.34  29.98  29.85  29.04
3  1904  22.50  24.73  28.21  32.02  32.64  32.07  30.36  30.09  30.04  29.20
4  1905  22.00  22.83  26.68  30.01  33.32  33.25  31.44  30.68  30.12  30.67

     NOV    DEC  ANNUAL  JAN-FEB  MAR-MAY  JUN-SEP  OCT-DEC
0  27.31  24.49   28.96    23.27    31.46    31.27    27.25
1  26.31  24.04   29.22    25.75    31.76    31.09    26.49
2  26.08  23.65   28.47    24.24    30.71    30.92    26.26
3  26.36  23.63   28.49    23.62    30.95    30.66    26.40
4  27.52  23.82   28.30    22.25    30.00    31.33    26.57
```

In [ ]:
```
df_long = df.melt(id_vars='YEAR', var_name='Month', value_name='Temp')
df_long.dropna(inplace=True)

df_long['MonthNum'] = df_long['Month'].map({
    'JAN':1,'FEB':2,'MAR':3,'APR':4,'MAY':5,'JUN':6,
    'JUL':7,'AUG':8,'SEP':9,'OCT':10,'NOV':11,'DEC':12
})

# Filter out rows where MonthNum is NaN (seasonal columns)
df_long = df_long.dropna(subset=['MonthNum'])

df_long.head()
```

Out[ ]:

	YEAR	Month	Temp	MonthNum
0	1901	JAN	22.40	1.0
1	1902	JAN	24.93	1.0
2	1903	JAN	23.44	1.0
3	1904	JAN	22.50	1.0
4	1905	JAN	22.00	1.0

In [ ]:
```
X = df_long[['MonthNum']]  # input feature
y = df_long['Temp']        # target variable

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42
)

print(X_train.shape, X_test.shape)
```

```
(1053, 1) (351, 1)
```

In [ ]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[ ]:

▾ LinearRegression

```
LinearRegression()
```

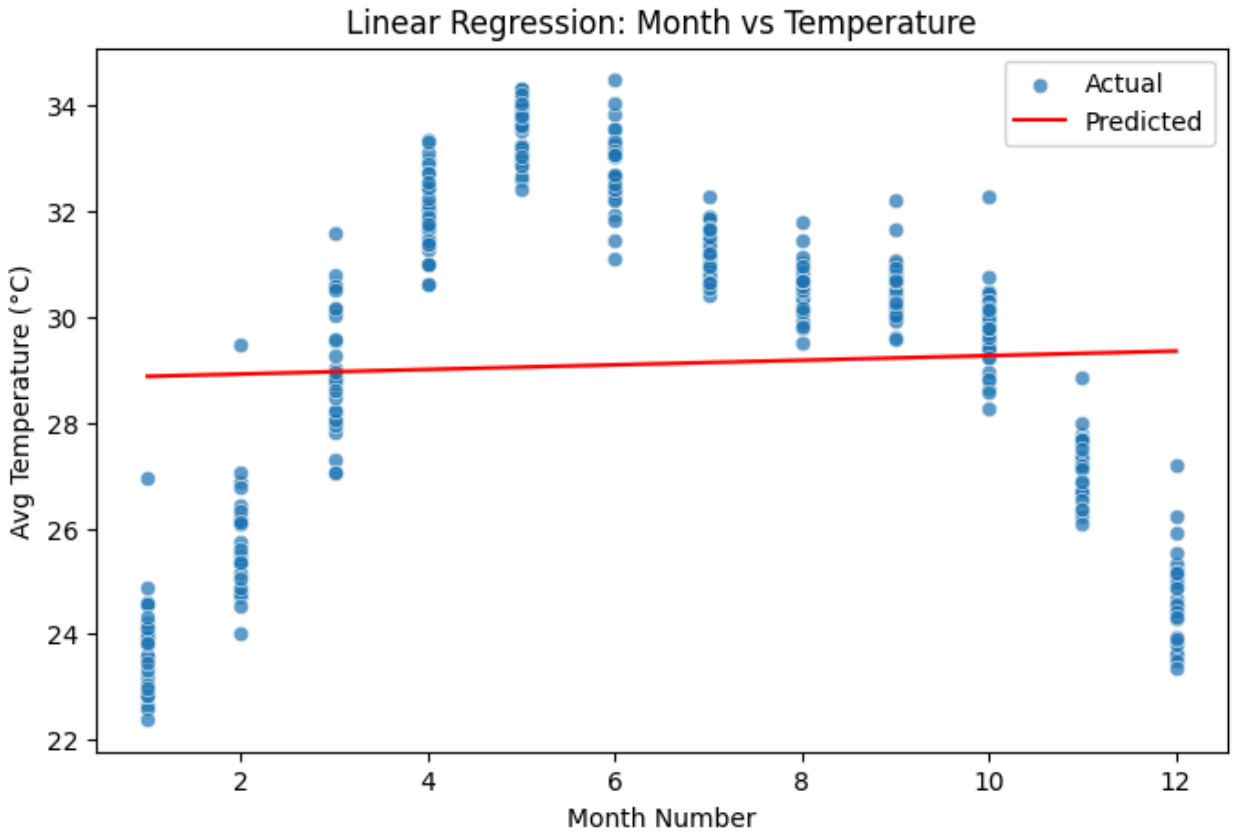
In [ ]:
```
y_pred = model.predict(X_test)
```

In [ ]:
```
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse:.4f}")
print(f"MAE: {mae:.4f}")
print(f"R²: {r2:.4f}")
```

```
MSE: 9.6476
MAE: 2.6478
R²: -0.0105
```

In [ ]:
```
plt.figure(figsize=(8,5))
sns.scatterplot(x=X_test['MonthNum'], y=y_test, label='Actual', alpha=0.7)
sns.lineplot(x=X_test['MonthNum'], y=y_pred, color='red', label='Predicted')
plt.xlabel('Month Number')
plt.ylabel('Avg Temperature (°C)')
plt.title('Linear Regression: Month vs Temperature')
plt.legend()
plt.show()
```


Linear Regression: Month vs Temperature
Actual
Predicted
Avg Temperature (°C)
Month Number
34
32
30
28
26
24
22
2
4
6
8
10
12

In [ ]: