```python
# linear_regression using Boston Housing Data Set

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
data = pd.read_csv("boston_housing.csv")
print(data.head())
print(data.shape)
print(data.info())

# Select feature and target
#X = data[['RM']]  # Feature: average number of rooms
#y = data['MEDV']  # Target: house price
# Select features and target
X = data.drop('MEDV', axis=1)  # All columns except 'MEDV'
y = data['MEDV']               # Target column
```

```python
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Evaluation
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R² Score:", r2)
```

```
# Visualization with RM vs MEDV (for illustration)
# Create a new model using only RM for visualization
rm_model = LinearRegression()
rm_model.fit(data[['RM']], data['MEDV'])
data['Predicted'] = rm_model.predict(data[['RM']])

# Plot
sns.set(style='whitegrid')
sns.scatterplot(x='RM', y='MEDV', data=data, label='Actual data', color='blue', alpha=0.5)
sns.lineplot(x='RM', y='Predicted', data=data, label='Regression Line (RM only)', color='red')
plt.title("Linear Regression - RM vs MEDV (Visualization)")
plt.xlabel("Average Number of Rooms (RM)")
plt.ylabel("Median Home Value (MEDV)")
plt.legend()
plt.show()
```