Lab Work:

```
# Importing necessary libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error, r2_score


# Step 1: Load the dataset

# Students need to download the dataset from Kaggle or UCI:

# - Github: https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv

# Write your code to load datasets from CSV or local files using Pandas.

# [Reference 1: Pandas Documentation on Reading Data]
(https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html#io-read-csv)


# Step 2: Exploratory Data Analysis (EDA)

# Write your code for printing the first 5 rows of the dataset

# [Reference 2: Pandas Documentation on DataFrame Methods]
(https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#dataframe)


# Write your code for printing summary statistics of the dataset

# [Reference 3: Article on Exploratory Data Analysis]
(https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15)


# Write your code to check for missing values

# [Reference 4: Handling Missing Data in Python]
(https://www.analyticsvidhya.com/blog/2021/04/handling-missing-values-in-pandas/)
```

# Step 3: Data Visualization

# Plot correlation heatmap

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

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')
```

# Saving the heatmap image

```
plt.savefig('correlation_heatmap.png')  # Save heatmap as an image

plt.show()
```

# Write your code for visualizing the relationship between 'rm' (average number of rooms) and 'medv'

# [Reference 5: Guide on Data Visualization Using Seaborn]
(https://seaborn.pydata.org/tutorial.html)

# Step 4: Prepare Data for Linear Regression

# Write your code after reading about data splitting using train_test_split and implement it here.

# [Reference 6: Scikit-Learn Documentation on Model Selection]
(https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)

# Step 5: Train Linear Regression Model

# Write your code for implementing the Linear Regression model

# [Reference 7: Linear Regression in Scikit-Learn]
(https://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares)

# Step 6: Make predictions on the test set

# Research and write code for making predictions with the trained model

# [Reference 8: Making Predictions with Scikit-Learn]
(https://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html)

# Step 7: Evaluate the Model

# Write your code for evaluating model performance

# [Reference 9: Evaluating a Regression Model]
(https://scikit-learn.org/stable/modules/model_evaluation.html)

```
# Step 8: Visualize the Linear Regression Results

# Implement the plotting of actual vs predicted values for comparison

# [Reference 10: Matplotlib Plotting Guide]
(https://matplotlib.org/stable/tutorials/introductory/pyplot.html)


# Assuming predictions are made (y_pred) and actual data is y_test

plt.figure(figsize=(8, 6))

plt.scatter(X_test, y_test, color='blue', label='Actual Prices')

plt.plot(X_test, y_pred, color='red', label='Predicted Prices')

plt.title('Actual vs Predicted House Prices')

plt.xlabel('Average Number of Rooms (RM)')

plt.ylabel('House Price')

plt.legend()


# Saving the actual vs predicted price plot as an image

plt.savefig('actual_vs_predicted_prices.png')  # Save the result image

plt.show()
```

Explanation of Missing Parts:

1. **Step 1**: Students should explore how to load datasets using `pandas`. The **Pandas Documentation** will help them understand the basic methods for reading CSV files or other data formats.
2. **Step 2**: For EDA, students should:
   - Use **pandas** methods to display the first few rows of the dataset.
   - Learn how to generate summary statistics and check for missing data.
   - Relevant articles and resources have been provided to guide them through the process.
3. **Step 3**: Data visualization will be partially implemented, but students must complete the scatter plot between `RM` and `PRICE` after reading the **Seaborn Tutorial**.
4. **Step 4**: Students need to split the dataset into training and testing sets by referring to the **train_test_split** documentation.
5. **Step 5**: They will read and implement **Linear Regression** using the **Scikit-Learn** guide.
6. **Step 6**: The process of making predictions is left as an assignment.
7. **Step 8**: Visualization of the results (Actual vs. Predicted prices) is to be implemented by students.

8. Assuming predictions are made (y_pred) and actual data is y_test, plot and save actual vs predicted price as an image.

Assignment Structure:

- **Reference 1**: Explore data loading techniques using pandas.
- **Reference 2**: Understand and use pandas DataFrame methods for inspecting datasets.
- **Reference 3**: Complete EDA by writing code for summary statistics and missing data.
- **Reference 5**: Create a scatter plot between features and target variables using seaborn.
- **Reference 6**: Split the dataset using `train_test_split`.
- **Reference 7**: Train the Linear Regression model.
- **Reference 8**: Implement code to make predictions using the model.
- **Reference 10**: Visualize the comparison of actual and predicted values using matplotlib.