Practical 3 Find the below data set and perform the following operations:

Dataset name: -Diamond_DataDescription

1. Plot a boxplot for "price" vs "cut" from the dataset "diamond.csv". Which of the categories under "cut" have the highest median price?

2. Create a frequency table (one-way table) for the variable "cut" from the dataset "diamond.csv". What is the frequency for the cut type "Ideal"?

3. Show the subplot of the diamond carat weight distribution.

4. Show the subplot of diamond depth distribution.

5. Build the Model using linear regression and find the accuracy.

Step 1 Load data set

```
import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset

diamonds = pd.read_csv('Diamonds_DataDescription.csv')
```

1. Plot a boxplot for "price" vs "cut" from the dataset "diamond.csv". Which of the categories under "cut" have the highest median price?

```
# Boxplot for price vs cut

import seaborn as sns

plt.figure(figsize=(10, 6))

sns.boxplot(x='cut', y='price', data=diamonds, order=['Fair', 'Good', 'Very Good', 'Premium', 'Ideal'])

plt.title('Boxplot of Price vs Cut')

plt.xlabel('Cut')

plt.ylabel('Price')

plt.show()
```

2. Create a frequency table (one-way table) for the variable "cut" from the dataset "diamond.csv". What is the frequency for the cut type "Ideal"?

```
# Frequency table for cut

cut_frequency = diamonds['cut'].value_counts()

print(cut_frequency)

# Frequency for cut type 'Ideal'

frequency_ideal = cut_frequency['Ideal']

print(frequency_ideal)
```

3. Show the subplot of the diamond carat weight distribution.

```
# Subplot of diamond carat weight distribution
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.hist(diamonds['carat'], bins=30, color='skyblue', edgecolor='black')
plt.title('Diamond Carat Weight Distribution')
plt.xlabel('Carat')
plt.ylabel('Frequency')
plt.show()
```

4. Show the subplot of diamond depth distribution.

```
# Subplot of diamond depth distribution
plt.subplot(1, 2, 2)
plt.hist(diamonds['depth'], bins=30, color='lightgreen', edgecolor='black')
plt.title('Diamond Depth Distribution')
plt.xlabel('Depth')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

5. Build the Model using linear regression and find the accuracy.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
# Extract features and target
X = diamonds[['carat', 'depth']]
y = diamonds['price']
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize Linear Regression model
model = LinearRegression()
# Fit the model
model.fit(X_train, y_train)
# Predict on the test set
y_pred = model.predict(X_test)
# Calculate R-squared
accuracy = r2_score(y_test, y_pred)
print(f'Accuracy (R-squared): {accuracy}')plt.tight_layout()
plt.show()
```