# WEEK 2 LAB A Activity: Heart Disease Data Cleaning & EDA Lab Activity

## 1. Import Libraries

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

## 2. Load the Dataset

```
# Load dataset

df = pd.read_csv("heart_disease.csv ")  # from Kaggle



# Preview data

df.head()
```

## 3. Basic Dataset Info

```
print("Shape:", df.shape)

print("\nData Info:")

df.info()



print("\nMissing values per column:")

print(df.isnull().sum())
```

## 4. Remove Missing Data

```
# Remove rows with missing values

df = df.dropna()

print("Shape after removing missing values:", df.shape)
```

## 5. Check & Remove Duplicates

```
# Check duplicates

duplicate_count = df.duplicated().sum()
```

```
print("Number of duplicate rows:", duplicate_count)


# Remove duplicates

df = df.drop_duplicates()

print("Shape after removing duplicates:", df.shape)
```

### 6. Column Cleanup

```
# Standardize column names (lowercase, no spaces)

df.columns = df.columns.str.lower().str.strip()

print("Cleaned column names:")

print(df.columns)
```

### 7. Basic Statistical Summary

```
df.describe()
```

### 8. Target Variable Distribution

```
df['target'].value_counts().plot(kind='bar')

plt.title("Heart Disease Distribution")

plt.xlabel("Target (0 = No Disease, 1 = Disease)")

plt.ylabel("Count")

plt.show()
```

### 9. Age Distribution

```
plt.hist(df['age'], bins=20)

plt.title("Age Distribution")

plt.xlabel("Age")

plt.ylabel("Frequency")

plt.show()
```

### 10. Cholesterol vs Heart Disease

```
plt.figure()
```

```
plt.scatter(df['chol'], df['target'])

plt.title("Cholesterol vs Heart Disease")

plt.xlabel("Cholesterol")

plt.ylabel("Heart Disease")

plt.show()
```

### 11. Chest Pain Type vs Target

```
df.groupby('cp')['target'].mean().plot(kind='bar')

plt.title("Chest Pain Type vs Heart Disease Rate")

plt.xlabel("Chest Pain Type")

plt.ylabel("Heart Disease Probability")

plt.show()
```

### 12. Correlation Heat-style Plot

```
corr = df.corr()

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

plt.imshow(corr)

plt.colorbar()

plt.title("Feature Correlation Matrix")

plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)

plt.yticks(range(len(corr.columns)), corr.columns)

plt.show()
```

### 13. Save Cleaned Dataset

```
df.to_csv("cleaned_heart_disease.csv", index=False)

print("Cleaned dataset saved successfully.")
```

### Summary of Achievements

- Handles missing data

- Removes duplicates
- Cleans column names
- Performs EDA
- Generates publication-ready graphs
- Saves cleaned dataset