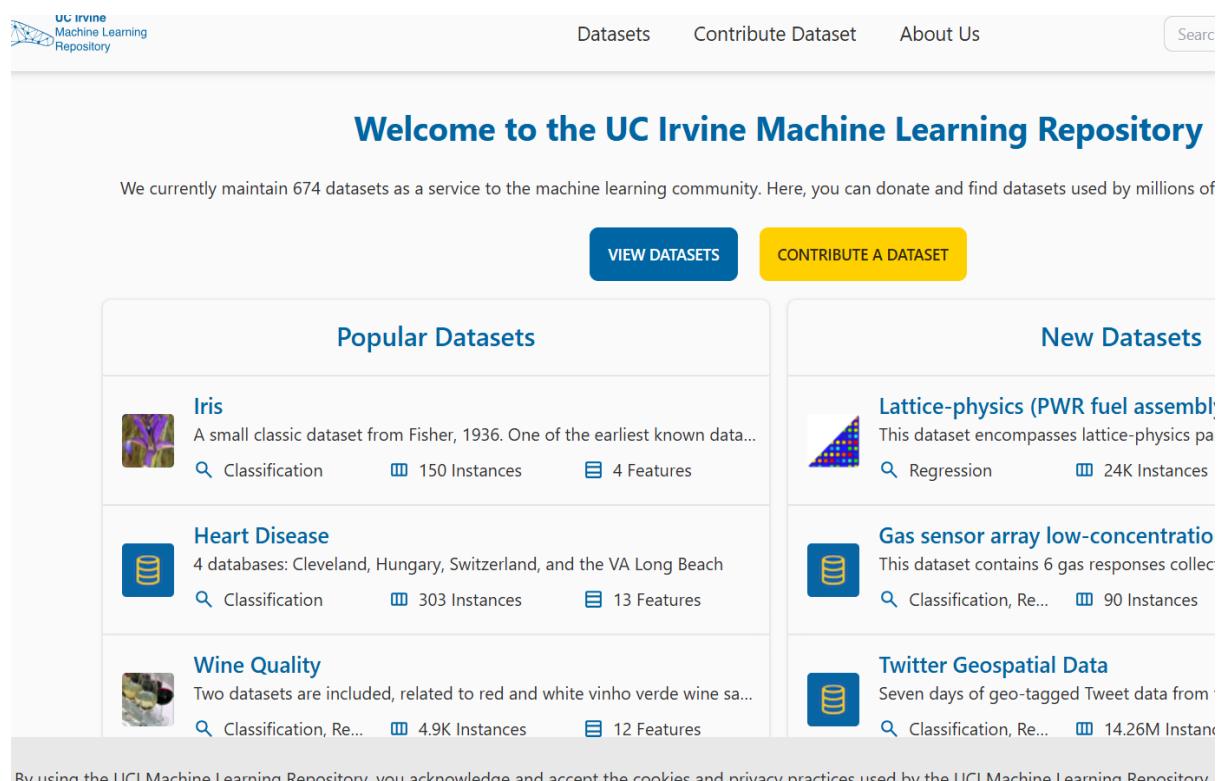


ASSIGNMENT 1

Extract the data from dataset/database using python

STEP 1: Open UCIML and select a dataset



The screenshot shows the homepage of the UC Irvine Machine Learning Repository. At the top, there is a navigation bar with links for 'Datasets', 'Contribute Dataset', and 'About Us'. A search bar is also present. The main heading is 'Welcome to the UC Irvine Machine Learning Repository'. Below this, a message states: 'We currently maintain 674 datasets as a service to the machine learning community. Here, you can donate and find datasets used by millions of...'. There are two prominent buttons: 'VIEW DATASETS' (blue) and 'CONTRIBUTE A DATASET' (yellow). The page is divided into sections: 'Popular Datasets' and 'New Datasets'. The 'Popular Datasets' section features three entries: 'Iris' (Classification, 150 Instances, 4 Features), 'Heart Disease' (Classification, 303 Instances, 13 Features), and 'Wine Quality' (Classification, 4.9K Instances, 12 Features). The 'New Datasets' section features three entries: 'Lattice-physics (PWR fuel assembly)' (Regression, 24K Instances), 'Gas sensor array low-concentration' (Classification, 90 Instances), and 'Twitter Geospatial Data' (Classification, 14.26M Instances). Each dataset entry includes a small thumbnail image, a title, a brief description, and icons for classification, instances, and features.

STEP 2: download data set. I have downloaded the Iris dataset

The screenshot shows the Iris dataset page on the UCI Machine Learning Repository. At the top, there's a blue header with the dataset name 'Iris' and a small image of an iris flower. Below the header, it says 'Donated on 6/30/1988'. To the right, there are buttons for 'DOWNLOAD (3.7 KB)', 'IMPORT IN PYTHON', and 'CITE'. Below these are statistics: '352 citations' and '801628 views'. A section titled 'Dataset Information' contains a question about what instances represent (a plant) and additional information about the dataset being one of the earliest used for classification and containing 150 instances with 4 features. On the right side, there are sections for 'Keywords' (ecology), 'Creators' (R. A. Fisher), and 'DOI' (10.24432/C56C76).

STEP 3: Open google colab and mount the dataset

```
from google.colab import drive  
drive.mount('/content/drive')
```

STEP 4

To read the data

The screenshot shows a Google Colab notebook cell. The title is 'Read the data'. The code cell contains:

```
import pandas as pd  
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data')  
iris_data.head()
```

When the cell is run, it displays the first five rows of the Iris dataset as a Pandas DataFrame:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

STEP 5: To describe the data

```
describe
```

```
▶ import pandas as pd  
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data')  
iris_data.describe()
```

	5.1	3.5	1.4	0.2
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

STEP 6 : To shape and info to data

```
shape
```

```
▶ import pandas as pd  
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data')  
iris_data.shape
```

```
→ (149, 5)
```

```
info
```

```
[ ] import pandas as pd  
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data')  
iris_data.info()
```

```
→ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 149 entries, 0 to 148  
Data columns (total 5 columns):  
 #   Column      Non-Null Count  Dtype    
---  --          --          --  
 0   5.1         149 non-null    float64  
 1   3.5         149 non-null    float64  
 2   1.4         149 non-null    float64  
 3   0.2         149 non-null    float64  
 4   Iris-setosa 149 non-null    object  
dtypes: float64(4), object(1)  
memory usage: 5.9+ KB
```

STEP 7: For size of the data

```
size
```

```
▶ import pandas as pd  
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data')  
iris_data.size
```

```
→ 745
```

STEP 8 : to check is null or not

```

# Check for missing values in each column
print("\nMissing values in each column:")
print(iris_data.isnull().sum())

# Check if there are any missing values in the entire dataset
print("\nAre there any missing values in the dataset?")
print(iris_data.isnull().any().any())

# Display rows with missing values, if any
missing_rows = iris_data[iris_data.isnull().any(axis=1)]
if missing_rows.empty:
    print("\nNo rows contain missing values.")
else:
    print("\nRows with missing values:")
    print(missing_rows)

# Optionally, fill missing values with 0
iris_data.fillna(value=0, inplace=True)
print("\nMissing values have been filled with 0.")

# Verify again after filling missing values
print("\nMissing values check after filling:")
print(iris_data.isnull().sum())

```

```

Missing values in each column:
5.1      0
3.5      0
1.4      0
0.2      0
Iris-setosa  0
dtype: int64

Are there any missing values in the dataset?
False

No rows contain missing values.

Missing values have been filled with 0.

Missing values check after filling:
5.1      0
3.5      0
1.4      0
0.2      0
Iris-setosa  0
dtype: int64

```

STEP 9: to plot a graph we use matplotlib / seaborn library. Here, I have plotted a box plot/hostoplot

```
import pandas as pd
import matplotlib.pyplot as plt

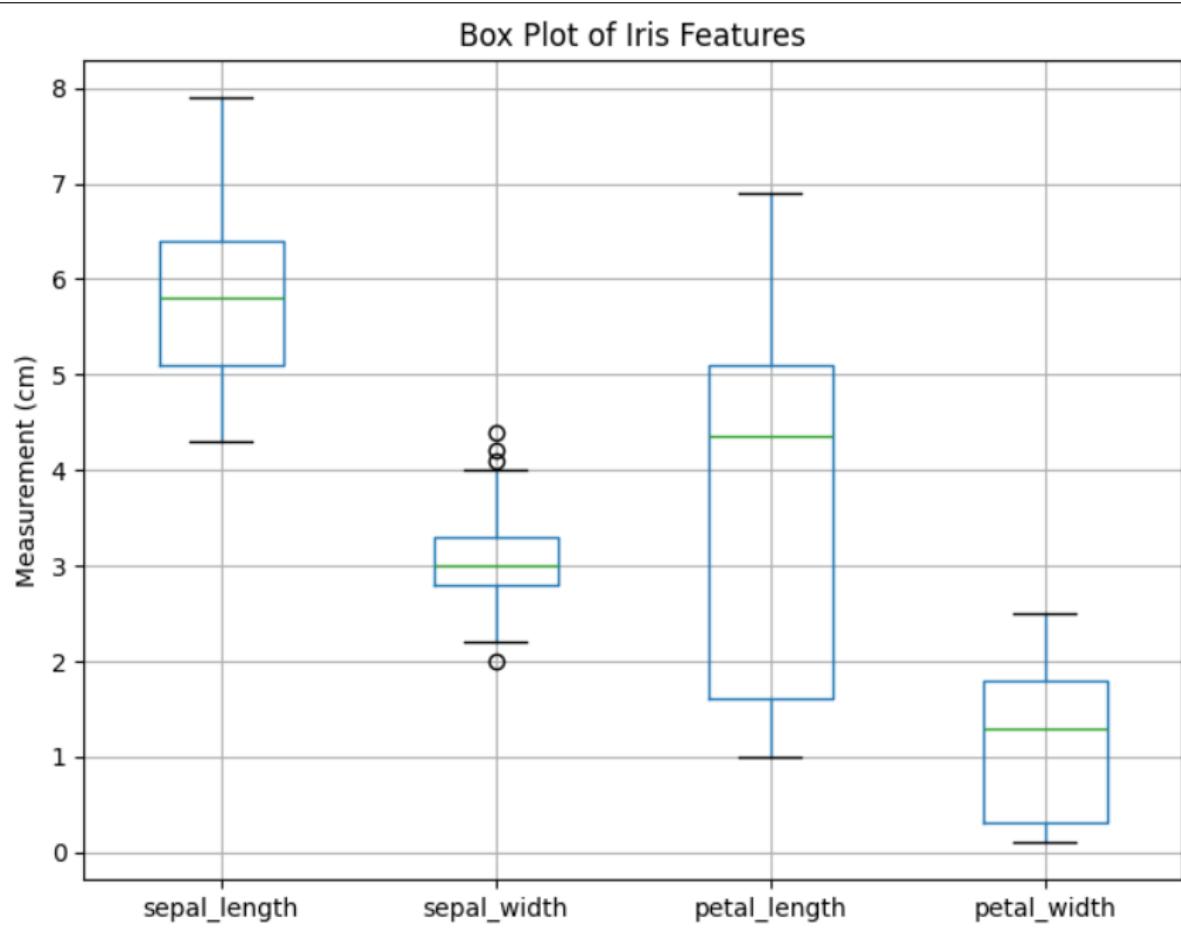
iris_data = pd.read_csv('/content/drive/MyDrive/iris.data', header=None)

# Assign column names if they are missing
iris_data.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species']

# Display the first few rows
print(iris_data.head())

plt.figure(figsize=(8, 6))
iris_data[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']].boxplot()
plt.title('Box Plot of Iris Features')
plt.ylabel('Measurement (cm)')
plt.grid(True)
plt.show()
```

```
sepal_length  sepal_width  petal_length  petal_width  species
0            5.1         3.5          1.4         0.2  Iris-setosa
1            4.9         3.0          1.4         0.2  Iris-setosa
2            4.7         3.2          1.3         0.2  Iris-setosa
3            4.6         3.1          1.5         0.2  Iris-setosa
4            5.0         3.6          1.4         0.2  Iris-setosa
```



STEP 10: Here's the complete code to achieve all the mentioned tasks:

```
▶ import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

iris_data=pd.read_csv("/content/drive/MyDrive/iris.data")
iris_data.columns = ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)', 'species'] # Adding column names
iris_data

#DATA CLEANING
iris_data.describe()

#TO CHECK NULL VALUES
iris_data.isnull()
print("Null values")
iris_data.isnull().sum()#---if present return sum

#to check nan values
print("Nan values")
iris_data.isna()
iris_data.isna().sum()
iris_data.dropna(inplace=True)
print("After removing null values")
iris_data.isna().sum()

#TO CHECK DUPLICATES
print("Duplicate values")
iris_data.duplicated()
print("Sum of Duplicate values")
iris_data.duplicated().sum()

iris_data.drop_duplicates(inplace=True)
print("After removing duplicates")
iris_data.duplicated().sum()

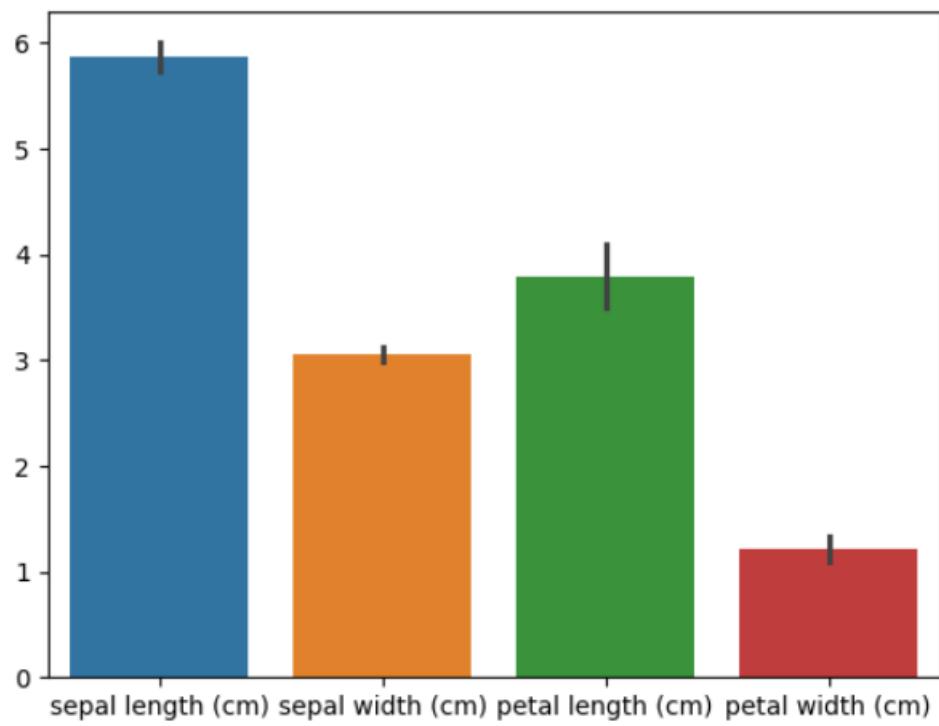
#Data preprocessing

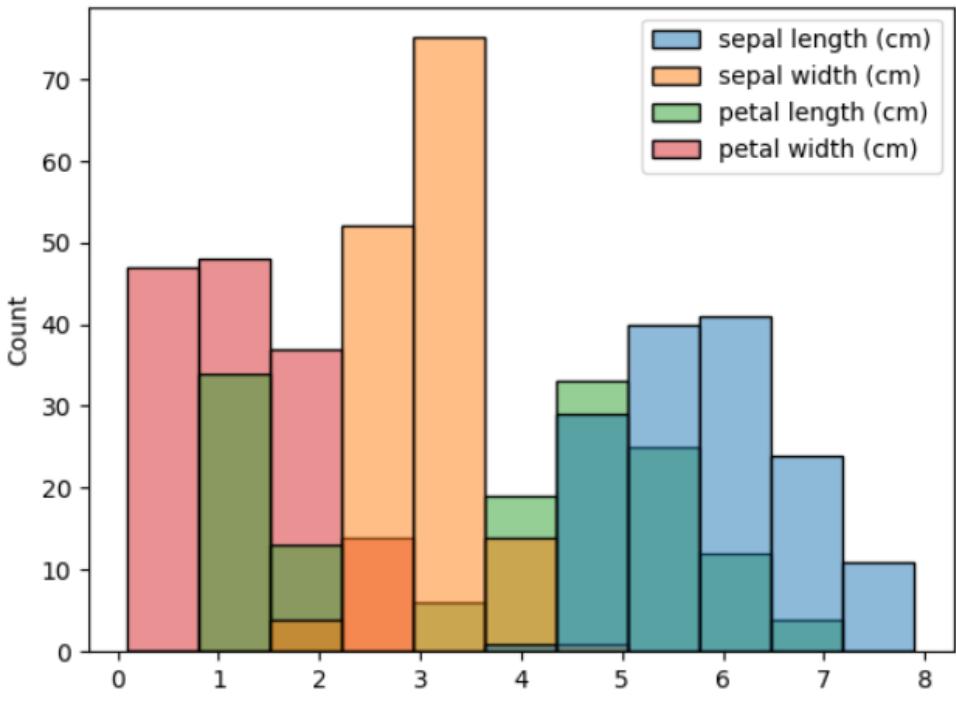
#1.Barplot using seaborn
sns.barplot(data=iris_data)
plt.show()
#2.Boxplot using seaborn

sns.histplot(data=iris_data)
plt.show()

print("Dataset: ")
print(iris_data)
```

```
Null values
Nan values
After removing null values
Duplicate values
Sum of Duplicate values
After removing duplicates
```





Dataset:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
0	4.9	3.0		1.4	0.2
1	4.7	3.2		1.3	0.2
2	4.6	3.1		1.5	0.2
3	5.0	3.6		1.4	0.2
4	5.4	3.9		1.7	0.4
..
144	6.7	3.0		5.2	2.3
145	6.3	2.5		5.0	1.9
146	6.5	3.0		5.2	2.0
147	6.2	3.4		5.4	2.3
148	5.9	3.0		5.1	1.8

```

      species
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
..    ...
144   Iris-virginica
145   Iris-virginica
146   Iris-virginica
147   Iris-virginica
148   Iris-virginica

[146 rows x 5 columns]

```

THANK YOU 😊