


Write a program to implement how to read and display a dataset in Python.

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
```

1. From Google Drive

```
from google.colab import drive
```


```
drive.mount('/content/drive')
```




 Mounted at /content/drive

```
import pandas as pd
```

```
df=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/pima-indians-diabetes.csv")
```

df



	preg	plas	pres	skin	test	mass	pedi	age	class	
0	6	148	72	35	0	33.6	0.627	50	1	
1	1	85	66	29	0	26.6	0.351	31	0	
2	8	183	64	0	0	23.3	0.672	32	1	
3	1	89	66	23	94	28.1	0.167	21	0	
4	0	137	40	35	168	43.1	2.288	33	1	
...	
763	10	101	76	48	180	32.9	0.171	63	0	
764	2	122	70	27	0	36.8	0.340	27	0	
765	5	121	72	23	112	26.2	0.245	30	0	
766	1	126	60	0	0	30.1	0.349	47	1	
767	1	93	70	31	0	30.4	0.315	23	0	

768 rows × 9 columns

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

2. From sklearn datasets

```
from sklearn.datasets import load_iris
iris = load_iris()# creating an instance of the data
df=pd.DataFrame(iris.data, columns= iris.feature_names) # storing the data in a data fram
df # displaying the data
```



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

Next steps:


[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

3. From URL

```
url = 'https://raw.githubusercontent.com/mwaskom/seaborn-data/master/iris.csv'
df = pd.read_csv(url)
df
```



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa



2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Next steps:




[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

✓ 4. Read from the local CSV file (this code works in local editor (Jupyter notebook, VScode etc))

```
print("\nReading dataset from an CSV file:")
try:
    df_CSV = pd.read_csv(r"D://Datasets//Mall_Customers.csv") # Replace with your file n
    print(df_CSV.head())
except FileNotFoundError:
    print("File not found. Make sure the file exists.")
```

```
Reading dataset from an CSV file:
File not found. Make sure the file exists.
```

```
df_CSV = pd.read_csv('/content/Salary_Data.csv')
df_CSV
```

	YearsExperience	Salary	
0	1.1	39343	
1	1.3	46205	
2	1.5	37731	
3	2.0	43525	
4	2.2	39891	

5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872

Next steps:

[Generate code with df_CSV](#)[View recommended plots](#)[New interactive sheet](#)

```
# Display basic properties of the dataset
print("Dataset Head:")
print(df.head()) # Display the first 5 rows of the dataset
```

Dataset Head:

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

```
print("\nDataset Info:")
```

```
print(df.info()) # Display summary of the dataset including column data types and non-nul
```

Dataset Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150 entries, 0 to 149

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	sepal_length	150 non-null	float64
1	sepal_width	150 non-null	float64
2	petal_length	150 non-null	float64
3	petal_width	150 non-null	float64
4	species	150 non-null	object

dtypes: float64(4), object(1)

memory usage: 6.0+ KB

None

```
print("\nDataset Description:")
```

```
print(df.describe()) # Display statistical summary of numeric columns
```

Dataset Description:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
print("\nDataset Shape:")
```

```
print(df.shape) # Display the dimensions of the dataset (rows, columns)
```

Dataset Shape:

(150, 5)

```
print("\nColumn Names:")
```

```
print(df.columns) # Display column names
```

Column Names:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',  
      'species'],  
      dtype='object')
```

```
print("\nMissing Values:")
```

```
print(df.isnull().sum()) # Display the number of missing values per column
```

Missing Values:

```
sepal_length    0  
sepal_width     0  
petal_length    0  
petal_width     0  
species         0  
dtype: int64
```

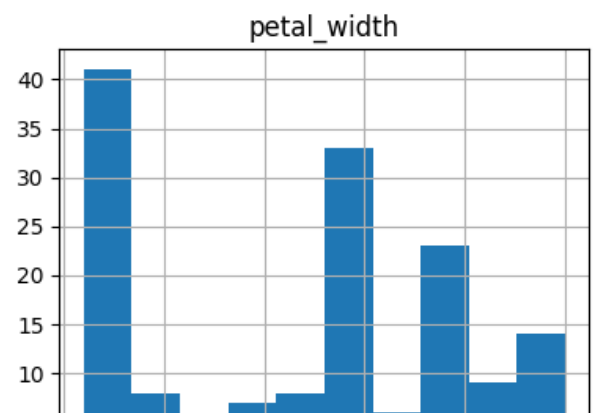
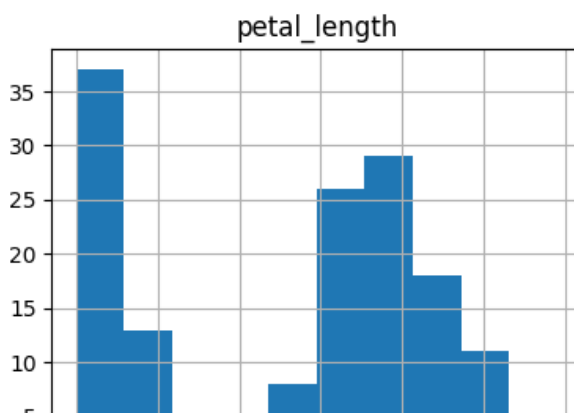
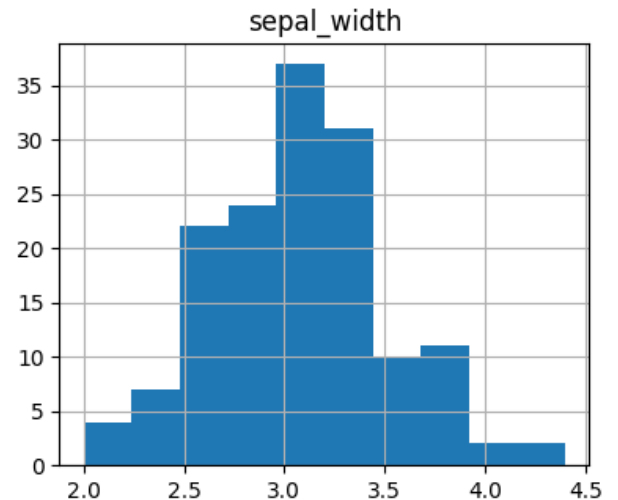
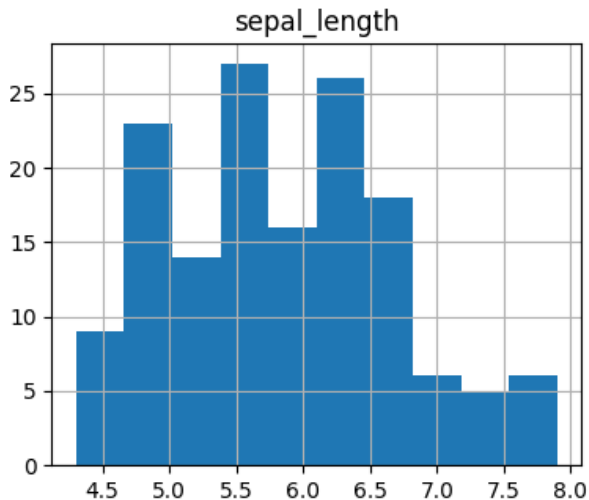
Optional: Plotting some basic visualizations

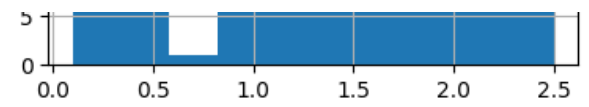
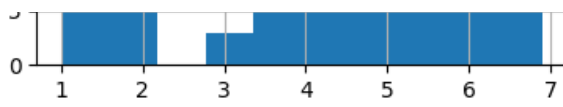
```
print("\nBasic Data Visualization:")
```

```
df.hist(figsize=(10, 8)) # Plot histograms for numeric columns
```

```
plt.show()
```

Basic Data Visualization:





Start coding or [generate](#) with AI.