

```
In [1]: pip install ucimlrepo
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: ucimlrepo in c:\users\arooj\appdata\roaming\python\python311\site-packages (0.0.3)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: from ucimlrepo import fetch_ucirepo
```

```
# fetch dataset  
iris = fetch_ucirepo(id=53)  
  
# data (as pandas dataframes)  
X = iris.data.features  
y = iris.data.targets  
  
# metadata  
print(iris.metadata)  
  
# variable information  
print(iris.variables)
```

```
{'uci_id': 53, 'name': 'Iris', 'repository_url': 'https://archive.ics.uci.edu/dataset/53/iris', 'data_url': 'https://archive.ics.uci.edu/static/public/53/data.csv', 'abstract': 'A small classic dataset from Fisher, 1936. One of the earliest known datasets used for evaluating classification methods.\n', 'area': 'Biology', 'tasks': ['Classification'], 'characteristics': ['Tabular'], 'num_instances': 150, 'num_features': 4, 'feature_types': ['Real'], 'demographics': [], 'target_col': ['class'], 'index_col': None, 'has_missing_values': 'no', 'missing_values_symbol': None, 'year_of_dataset_creation': 1936, 'last_updated': 'Tue Sep 12 2023', 'dataset_doi': '10.24432/C56C76', 'creators': ['R. A. Fisher'], 'intro_paper': {'title': 'The Iris data set: In search of the source of virginica', 'authors': 'A. Unwin, K. Kleinman', 'published_in': 'Significance, 2021', 'year': 2021, 'url': 'https://www.semanticscholar.org/paper/4599862ea877863669a6a8e63a3c707a787d5d7e', 'doi': '1740-9713.01589'}, 'additional_info': {'summary': 'This is one of the earliest datasets used in the literature on classification methods and widely used in statistics and machine learning. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other.\n\nPredicted attribute: class of iris plant.\n\nThis is an exceedingly simple domain.\n\nThis data differs from the data presented in Fishers article (identified by Steve Chadwick, spchadwick@espeedaz.net ). The 35th sample should be: 4.9,3.1,1.5,0.2,"Iris-setosa" where the error is in the fourth feature. The 38th sample: 4.9,3.6,1.4,0.1,"Iris-setosa" where the errors are in the second and third features. ', 'purpose': 'N/A', 'funded_by': None, 'instances_represent': 'Each instance is a plant', 'recommended_data_splits': None, 'sensitive_data': None, 'preprocessing_description': None, 'variable_info': None, 'citation': None}}
```

```
name      role      type demographic \
```

0	sepal length	Feature	Continuous	None
1	sepal width	Feature	Continuous	None
2	petal length	Feature	Continuous	None
3	petal width	Feature	Continuous	None
4	class	Target	Categorical	None

		description	units	missing_values
0		None	cm	no
1		None	cm	no
2		None	cm	no
3		None	cm	no
4	class of iris plant: Iris Setosa, Iris Versico...	None		no

In []:

In [3]:

```
print(type(iris))
```

```
<class 'ucimlrepo.dotdict.dotdict'>
```

```
In [4]: import seaborn as sns
import pandas as pd

# Load the Iris dataset
iris = sns.load_dataset('iris')

# Check the first few rows of the DataFrame
print(iris.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

```
In [5]: iris.head(10)
```

```
Out[5]:
```

	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
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

```
In [6]: iris.sepal_length
```

```
Out[6]: 0      5.1
        1      4.9
        2      4.7
        3      4.6
        4      5.0
        ...
        145    6.7
        146    6.3
        147    6.5
        148    6.2
        149    5.9
        Name: sepal_length, Length: 150, dtype: float64
```

```
In [7]: iris.tail()
```

```
Out[7]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
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

```
In [8]: iris.species
```

```
Out[8]: 0      setosa
        1      setosa
        2      setosa
        3      setosa
        4      setosa
        ...
        145    virginica
        146    virginica
        147    virginica
        148    virginica
        149    virginica
        Name: species, Length: 150, dtype: object
```

```
In [9]: a = iris[iris["species"] == "setosa"]
```

```
In [10]: print (a)
```

	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
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa
10	5.4	3.7	1.5	0.2	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
17	5.1	3.5	1.4	0.3	setosa
18	5.7	3.8	1.7	0.3	setosa
19	5.1	3.8	1.5	0.3	setosa
20	5.4	3.4	1.7	0.2	setosa
21	5.1	3.7	1.5	0.4	setosa
22	4.6	3.6	1.0	0.2	setosa
23	5.1	3.3	1.7	0.5	setosa
24	4.8	3.4	1.9	0.2	setosa

25	5.0	3.0	1.6	0.2	setosa
26	5.0	3.4	1.6	0.4	setosa
27	5.2	3.5	1.5	0.2	setosa
28	5.2	3.4	1.4	0.2	setosa
29	4.7	3.2	1.6	0.2	setosa
30	4.8	3.1	1.6	0.2	setosa
31	5.4	3.4	1.5	0.4	setosa
32	5.2	4.1	1.5	0.1	setosa
33	5.5	4.2	1.4	0.2	setosa
34	4.9	3.1	1.5	0.2	setosa
35	5.0	3.2	1.2	0.2	setosa
36	5.5	3.5	1.3	0.2	setosa
37	4.9	3.6	1.4	0.1	setosa
38	4.4	3.0	1.3	0.2	setosa
39	5.1	3.4	1.5	0.2	setosa
40	5.0	3.5	1.3	0.3	setosa
41	4.5	2.3	1.3	0.3	setosa
42	4.4	3.2	1.3	0.2	setosa
43	5.0	3.5	1.6	0.6	setosa
44	5.1	3.8	1.9	0.4	setosa
45	4.8	3.0	1.4	0.3	setosa
46	5.1	3.8	1.6	0.2	setosa
47	4.6	3.2	1.4	0.2	setosa
48	5.3	3.7	1.5	0.2	setosa
49	5.0	3.3	1.4	0.2	setosa

```
In [11]: file1=open("myfile_iris.csv", "a")
```

```
In [12]: file1.write(str(a))
```

```
Out[12]: 3314
```

```
In [13]: file1.close()
```

```
In [14]: iris.shape
```

```
Out[14]: (150, 5)
```

```
In [15]: a.shape
```

```
Out[15]: (50, 5)
```

```
In [16]: # Load the Iris dataset
iris = sns.load_dataset('iris')
iris_class=["setosa", "virginica"]
for i in iris_class:
    variety=iris[(iris["species"]==i)]
    iris1=open(i+'myfile.csv','a')
    iris1.write (str(variety))
    iris1.close()
```

```
In [17]: # Create empty Lists for each class
setosa_data = []
virginica_data = []
# Loop through each row in the dataset
for index, row in iris.iterrows():
    if row['species'] == 'setosa':
        setosa_data.append(row)
    elif row['species'] == 'virginica':
        virginica_data.append(row)
iris1=open(i+'myfile.csv','a')
iris1.write (str(variety))
iris1.close()
```

```
In [31]: # Load the Iris dataset
import seaborn as sns

iris = sns.load_dataset('iris')
unique_classes = iris['species'].unique()

for class_label in unique_classes:
    subset_df = iris[iris['species'] == class_label]

    # Define the filename for the subset file
    filename = class_label + '_myfile.csv'

    # Open the file and write the subset to it
    with open(filename, 'a') as file:
        subset_df.to_csv(file, index=False)

print("Subset files created successfully.")
```

Subset files created successfully.

```
In [ ]:
```