

Name : Suraj Sawant

Roll No : TEB38

Practical No : 3 (B)

```
In [1]: import pandas as pd
```

```
In [2]: from sklearn import datasets
```

```
In [4]: iris=datasets.load_iris()
```

```
In [5]: iris
```

```
Out[5]: {'data': array([[5.1, 3.5, 1.4, 0.2],
```

```
[4.9, 3. , 1.4, 0.2],  
[4.7, 3.2, 1.3, 0.2],  
[4.6, 3.1, 1.5, 0.2],  
[5. , 3.6, 1.4, 0.2],  
[5.4, 3.9, 1.7, 0.4],  
[4.6, 3.4, 1.4, 0.3],  
[5. , 3.4, 1.5, 0.2],  
[4.4, 2.9, 1.4, 0.2],  
[4.9, 3.1, 1.5, 0.1],  
[5.4, 3.7, 1.5, 0.2],  
[4.8, 3.4, 1.6, 0.2],  
[4.8, 3. , 1.4, 0.1],  
[4.3, 3. , 1.1, 0.1],  
[5.8, 4. , 1.2, 0.2],  
[5.7, 4.4, 1.5, 0.4],  
[5.4, 3.9, 1.3, 0.4],  
[5.1, 3.5, 1.4, 0.3],  
[5.7, 3.8, 1.7, 0.3],  
[5.1, 3.8, 1.5, 0.3],  
[5.4, 3.4, 1.7, 0.2],  
[5.1, 3.7, 1.5, 0.4],  
[4.6, 3.6, 1. , 0.2],  
[5.1, 3.3, 1.7, 0.5],  
[4.8, 3.4, 1.9, 0.2],  
[5. , 3. , 1.6, 0.2],  
[5. , 3.4, 1.6, 0.4],  
[5.2, 3.5, 1.5, 0.2],  
[5.2, 3.4, 1.4, 0.2],  
[4.7, 3.2, 1.6, 0.2],  
[4.8, 3.1, 1.6, 0.2],  
[5.4, 3.4, 1.5, 0.4],  
[5.2, 4.1, 1.5, 0.1],  
[5.5, 4.2, 1.4, 0.2],  
[4.9, 3.1, 1.5, 0.2],  
[5. , 3.2, 1.2, 0.2],  
[5.5, 3.5, 1.3, 0.2],  
[4.9, 3.6, 1.4, 0.1],  
[4.4, 3. , 1.3, 0.2],  
[5.1, 3.4, 1.5, 0.2],  
[5. , 3.5, 1.3, 0.3],  
[4.5, 2.3, 1.3, 0.3],  
[4.4, 3.2, 1.3, 0.2],  
[5. , 3.5, 1.6, 0.6],  
[5.1, 3.8, 1.9, 0.4],  
[4.8, 3. , 1.4, 0.3],  
[5.1, 3.8, 1.6, 0.2],  
[4.6, 3.2, 1.4, 0.2],  
[5.3, 3.7, 1.5, 0.2],  
[5. , 3.3, 1.4, 0.2],
```

```
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5], [5.8,
2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
```

```
[5.8, 2.7, 5.1, 1.9],  
[7.1, 3. , 5.9, 2.1],  
[6.3, 2.9, 5.6, 1.8],  
[6.5, 3. , 5.8, 2.2],  
[7.6, 3. , 6.6, 2.1],  
[4.9, 2.5, 4.5, 1.7],  
[7.3, 2.9, 6.3, 1.8],  
[6.7, 2.5, 5.8, 1.8],  
[7.2, 3.6, 6.1, 2.5],  
[6.5, 3.2, 5.1, 2. ],  
[6.4, 2.7, 5.3, 1.9],  
[6.8, 3. , 5.5, 2.1], [5.7,  
2.5, 5. , 2. ],  
[5.8, 2.8, 5.1, 2.4],  
[6.4, 3.2, 5.3, 2.3], [6.5,  
3. , 5.5, 1.8],  
[7.7, 3.8, 6.7, 2.2],
```

```

[7.7, 2.6, 6.9, 2.3], [6. ,
2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2. ],
[7.7, 2.8, 6.7, 2. ],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2. ],
[6.4, 2.8, 5.6, 2.2],
[6.3, 2.8, 5.1, 1.5],
[6.1, 2.6, 5.6, 1.4],
[7.7, 3. , 6.1, 2.3],
[6.3, 3.4, 5.6, 2.4],
[6.4, 3.1, 5.5, 1.8],
[6. , 3. , 4.8, 1.8],
[6.9, 3.1, 5.4, 2.1],
[6.7, 3.1, 5.6, 2.4],
[6.9, 3.1, 5.1, 2.3],
[5.8, 2.7, 5.1, 1.9],
[6.8, 3.2, 5.9, 2.3],
[6.7, 3.3, 5.7, 2.5],
[6.7, 3. , 5.2, 2.3],
[6.3, 2.5, 5. , 1.9],
[6.5, 3. , 5.2, 2. ],
[6.2, 3.4, 5.4, 2.3],
[5.9, 3. , 5.1, 1.8]]),
'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
'frame': None,
'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n ----- \n\n**Data
Set Characteristics:**\n\n :Number of Instances: 150 (50 in each of three classe s)\n

```

```
:Number of Attributes: 4 numeric, predictive attributes and the class\n :Attribute
Information:\n - sepal length in cm\n - sepal width in cm
\n      - petal length in cm\n      - petal width in cm\n      - class:\n
- Iris-Setosa\n      - Iris-Versicolour\n      - Iris-Virginica
\n      \n      :Summary Statistics:\n\n      =====
= =====\n Min Max Mean SD Class Cor relation\n =====
===== \n sepal length: 4.3 7.9 5.84 0.83
0.7826\n sepal width: 2.0 4.4 3.05 0.43 -0.4194\n petal length: 1.0 6.9 3.76 1.76
0.9490 (high!)\n petal width: 0.1 2.5 1.20 0.76 0.9565 (high!)\n =====
===== \n\n :Missing Attribute Values: None\n :Class
Distribution: 33.3% for each of 3 classes.\n :Creator: R.A. Fisher\n : Donor:
Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n :Date: July, 1988\n\nTh e famous
Iris database, first used by Sir R.A. Fisher. The dataset is taken\nto Fisher's
paper. Note that it's the same as in R, but not as in the UCI\nMachine Learning
Repository, which has two wrong data points.\n\nThis is perhaps the best known
database to be found in the\npattern recognition literature. Fisher's paper is a
classic in the field and\nis referenced frequently to this day. (See Duda & Hart ,
for example.) The\ndata set contains 3 classes of 50 instances each, where each class
refers to a\ntype of iris plant. One class is linearly separable from the other 2;
the\nlatter are NOT linearly separable from each other.\n\n.. topic:: References\n\n
- Fisher, R.A. "The use of multiple measurements in taxonomic problems"\n Annual
Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to\n Mathematical
Statistics" (John Wiley, NY, 1950).\n - Duda, R.O., & Hart, P.E. (1973) Pattern
Classification and Scene Analysis.\n (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-
1. See page 218.\n - Dasarthy, B.V. (1980) "Nosing Around the Neighborhood: A New
System\n Structure and Classification Rule for Recognition in Partially Exposed\n
Environments". IEEE Transactions on Pattern Analysis and Machine\n Intelligence,
Vol. PAMI-2, No. 1, 67-71.\n - Gates, G.W. (1972) " The Reduced Nearest Neighbor
Rule". IEEE Transactions\n on Information Theory,
May 1972, 431-433.\n - See also: 1988 MLC Proceedings, 54-64. Cheeseman et al's
AUTOCLASS II\n conceptual clustering system finds 3 classes in the data.\n -
Many, many more ...',
'feature_names': ['sepal length (cm)',
'sepal width (cm)',
'petal length (cm)',
'petal width (cm)'],
'filename': 'iris.csv',
'data_module': 'sklearn.datasets.data'}
```

In [6]: `df=pd.DataFrame(iris['data'],columns=iris['feature_names'])`

In
[7]: `df`

Out[7]:

	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			

In [8]: `df.describe()`

Out[8]: **sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)**

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

In [10]:

`df['species']=iris['target']`

In [15]:

In [16]:

`df['species']=df['species'].map({0:'Setosa',1:'Versicolor',2:'Virginica'})`

Out[16]:

sepal
length (cm)

sepal width (cm)

petal length (cm) petal width (cm)

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				

In [17]: `df.describe()`

Out[17]:

		sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
mean	count	150.000000	150.000000	150.000000	150.000000
	std	5.843333	3.057333	3.758000	1.199333
	min	0.828066	0.435866	1.765298	0.762238
	max	7.900000	4.400000	6.900000	2.500000
25%	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

In [20]:

In [21]: `setosa_df=df[df['species']=='Setosa']`

Out[21]: `setosa_df.describe()`

sepal
length (cm) sepal width (cm) petal length (cm) petal width (cm)

mean	count	50.00000	50.000000	50.000000	50.000000
		5.00600	3.428000	1.462000	0.246000
	std	0.35249	0.379064	0.173664	0.105386
	min	4.30000	2.300000	1.000000	0.100000
	25%	4.80000	3.200000	1.400000	0.200000
	50%	5.00000	3.400000	1.500000	0.200000
	75%	5.20000	3.675000	1.575000	0.300000
	max	5.80000	4.400000	1.900000	0.600000

```
In [22]: versicolor_df=df[df['species']=='Versicolor']
```

```
In [24]: versicolor_df.describe()
```

Out[24]:

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)

mean	count	50.000000	50.000000	50.000000	50.000000
		5.936000	2.770000	4.260000	1.326000
	std	0.516171	0.313798	0.469911	0.197753
	min	4.900000	2.000000	3.000000	1.000000
	25%	5.600000	2.525000	4.000000	1.200000
	50%	5.900000	2.800000	4.350000	1.300000
	75%	6.300000	3.000000	4.600000	1.500000
	max	7.000000	3.400000	5.100000	1.800000

```
In [25]: virginica_df=df[df['species']=='Virginica']
```

```
In [26]: virginica_df.describe()
```

sepal

length (cm) sepal width (cm) petal length (cm) petal width (cm)

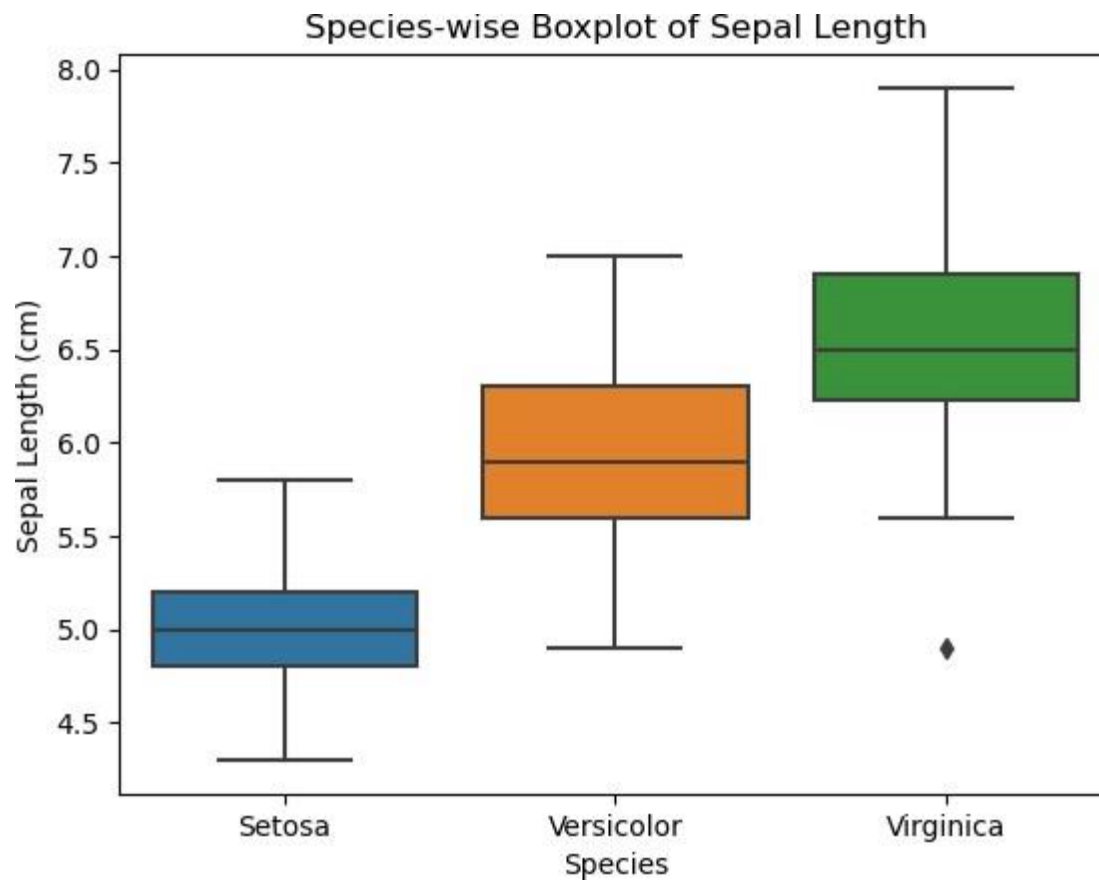
mean	count	50.00000	50.000000	50.000000	50.00000
		6.58800	2.974000	5.552000	2.02600

std		0.63588	0.322497	0.551895	0.27465
min		4.90000	2.200000	4.500000	1.40000
	25%	6.22500	2.800000	5.100000	1.80000
	50%	6.50000	3.000000	5.550000	2.00000
	75%	6.90000	3.175000	5.875000	2.30000
max		7.90000	3.800000	6.900000	2.50000

In
[27]:

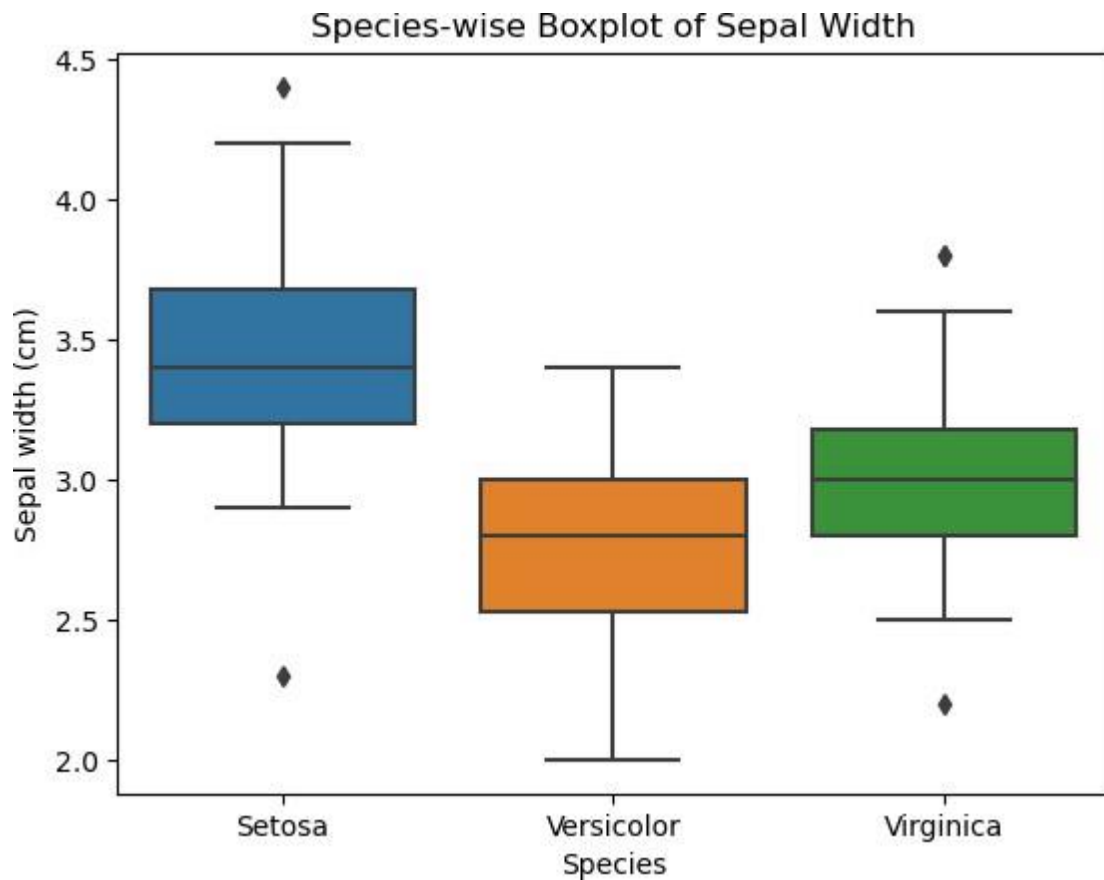
```
import seaborn as sns
import matplotlib.pyplot as plt
# Create the boxplot
sns.boxplot(x='species', y='sepal length (cm)',
data=df) plt.xlabel("Species") plt.ylabel("Sepal Length
(cm)") plt.title("Species-wise Boxplot of Sepal
Length")
```

Out[27]: Text(0.5, 1.0, 'Species-wise Boxplot of Sepal Length')



```
In [28]: sns.boxplot(x='species', y='sepal width (cm)', data=df)
plt.xlabel("Species") plt.ylabel("Sepal width (cm)")
plt.title("Species-wise Boxplot of Sepal Width")
```

Out[28]: Text(0.5, 1.0, 'Species-wise Boxplot of Sepal Width')



In [29]: `df.shape`

Out[29]: (150, 5)

In [30]: `df.mean()`

C:\Users\Admin\AppData\Local\Temp\ipykernel_7276\3698961737.py:1: FutureWarning: The default value of `numeric_only` in `DataFrame.mean` is deprecated. In a future version, it will default to `False`. In addition, specifying `'numeric_only=None'` is deprecated.

Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
df.mean()
sepal length (cm)    5.843333
                    3.057333
Out[30]: sepal width (cm)    3.758000
         petal length (cm)    petal 1.199333
         width (cm)         dtype: float64
```

In [31]: `df.groupby('species').mean()`

Out[31]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
Setosa	5.006	3.429	1.462	0.435
Versicolor	5.936	2.770	4.353	1.309
Virginica	6.595	2.973	5.552	1.549

species				
Setosa	5.006	3.428	1.462	0.246
Versicolor	5.936	2.770	4.260	1.326
Virginica	6.588	2.974	5.552	2.026

```
In [32]: df.median()
```

C:\Users\Admin\AppData\Local\Temp\ipykernel_7276\530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated.

Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.median()
```

```
sepal length (cm)    5.80
```

```
Out[32]: sepal width (cm)    3.00
petal length (cm)  4.35  petal width
(cm)  1.30
```

```
dtype: float64
```

```
df.groupby('species').median()
```

```
In [33]:
```

```
Out[33]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
--	-------------------	------------------	-------------------	------------------

species				
Setosa	5.0	3.4	1.50	0.2
Versicolor	5.9	2.8	4.35	1.3
Virginica	6.5	3.0	5.55	2.0

```
In [34]: df.groupby('species').count()
```

```
Out[34]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
--	-------------------	------------------	-------------------	------------------

species				
Setosa	50	50	50	50
Versicolor	50	50	50	50

Virginica

50

50

50

50

```
In [35]: df.species.mode()
```

```
Out[35]: 0          Setosa
1    Versicolor
2    Virginica
         Name: species, dtype: object
```

```
In [38]: df.quantile(0.25)
```

C:\Users\Admin\AppData\Local\Temp\ipykernel_7276\3656653379.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of

```
numeric_only to silence this warning.
df.quantile(0.25)
```

```
Out[38]: sepal length (cm) 5.1 sepal
         width (cm) 2.8 petal
         length (cm) 1.6 petal
         width (cm) 0.3 Name: 0.25,
         dtype: float64
```

```
In [39]: df.quantile(0.5)
```

C:\Users\Admin\AppData\Local\Temp\ipykernel_7276\1793688606.py:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of

```
numeric_only to silence this warning.
df.quantile(0.5)
```

```
Out[39]: sepal length (cm) 5.80
sepal width (cm) 3.00 petal length
(cm) 4.35 petal width (cm) 1.30
Name: 0.5, dtype: float64
```

```
In [40]: df.std()
```

C:\Users\Admin\AppData\Local\Temp\ipykernel_7276\3390915376.py:1: FutureWarning: The default value of numeric_only in DataFrame.std is deprecated. In a future version

, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.std()
```

```
Out[40]: sepal length (cm)    0.828066
         sepal width (cm)    petal 0.435866
length (cm)  petal width (cm) 1.765298
dtype: float64              0.762238
```

In []:
]: