

```
In [1]: from sklearn import datasets
from sklearn.datasets import load_iris
import pandas as pd
```

```
In [2]: iris_df = pd.DataFrame(load_iris().data, columns = load_iris().feature_names)
print("iris dataframe: ")
iris_df
```

iris dataframe:

	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 [3]: print("iris target dataframe:")
specimen_class= pd.DataFrame(load_iris().target,columns=['specimen_class'])
specimen_class
```

iris target dataframe:

	specimen_class
0	0
1	0
2	0
3	0
4	0
...	...
145	2
146	2
147	2
148	2
149	2

150 rows × 1 columns

```
In [4]: print("Concatenating both the dataframes - columnwise: ")
df_joined = pd.concat([iris_df, specimen_class],axis=1)
df_joined
```

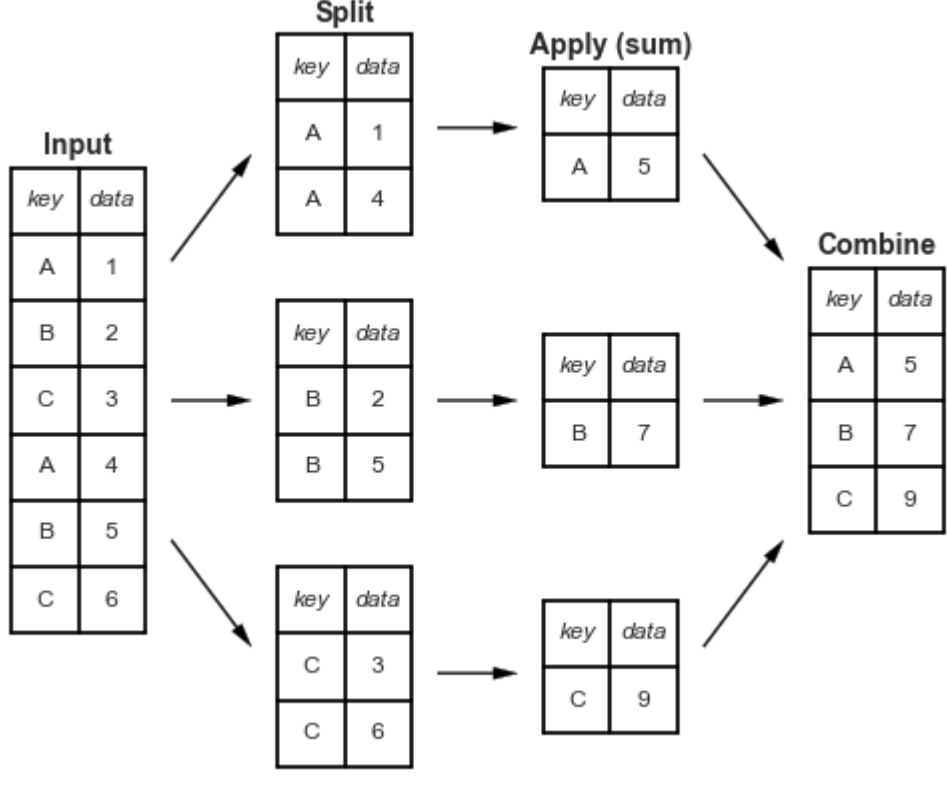
Concatenating both the dataframes - columnwise:

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

150 rows × 5 columns

## Groupwise Operations :

### HOW DOES GROUPBY WORK?



```
In [5]: # CHECKING HEADERS OF CURRENT DATAFRAME
df_joined.columns
```

```
Out[5]: Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
             'petal width (cm)', 'specimen_class'],
            dtype='object')
```

```
In [6]: # renaming the columns for ease of use
df_joined.columns=['sepal_len','sepal_wid','petal_len','petal_wid','class']
```

```
In [7]: df_joined
```

	sepal_len	sepal_wid	petal_len	petal_wid	class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

```
In [8]: #grouping by petal_wid,sepal_wid, petal_len,sepal_len --> mean
petal_width_mean = df_joined.groupby(['class'])['petal_wid'].mean()
print("petal_width_mean \n",petal_width_mean,"\n -----")
petal_length_mean = df_joined.groupby(['class'])['petal_len'].mean()
print("petal_length_mean \n",petal_length_mean, "\n -----")
sepal_length_mean = df_joined.groupby(['class'])['sepal_len'].mean()
print("sepal_length_mean \n",sepal_length_mean, "\n -----")
sepal_width_mean = df_joined.groupby(['class'])['sepal_wid'].mean()
print("sepal_width_mean \n",sepal_width_mean,"\n -----")
```

```
petal_width_mean
class
0    0.246
1    1.326
2    2.026
Name: petal_wid, dtype: float64
-----
petal_length_mean
class
0    1.462
1    4.260
2    5.552
Name: petal_len, dtype: float64
-----
sepal_length_mean
class
0    5.006
1    5.936
2    6.588
Name: sepal_len, dtype: float64
-----
sepal_width_mean
class
0    3.428
1    2.770
2    2.974
Name: sepal_wid, dtype: float64
-----
```

```
In [9]: print("count of all features in each class")
counts=df_joined.groupby(['class'])['petal_wid','petal_len','sepal_wid','sepal_len'].count()
counts
```

count of all features in each class

C:\Users\sowmiya\AppData\Local\Temp\ipykernel\_16308\70935550.py:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
counts=df_joined.groupby(['class'])['petal_wid','petal_len','sepal_wid','sepal_len'].count()
```

	petal_wid	petal_len	sepal_wid	sepal_len
class				
0	50	50	50	50
1	50	50	50	50
2	50	50	50	50

```
In [10]: max_grp=df_joined.groupby('class').aggregate({'petal_len':'max',
                                                    'petal_wid':'max',
                                                    'sepal_len':'max',
                                                    'sepal_wid':'max'})
max_grp
```

	petal_len	petal_wid	sepal_len	sepal_wid
class				
0	1.9	0.6	5.8	4.4
1	5.1	1.8	7.0	3.4
2	6.9	2.5	7.9	3.8

```
In [11]: min_grp=df_joined.groupby('class').aggregate({'petal_len':'min',
                                                    'petal_wid':'min',
                                                    'sepal_len':'min',
                                                    'sepal_wid':'min'})
min_grp
```

	petal_len	petal_wid	sepal_len	sepal_wid
class				
0	1.0	0.1	4.3	2.3
1	3.0	1.0	4.9	2.0
2	4.5	1.4	4.9	2.2

```
In [ ]:
```