In [1]:
```
#Import Pandas
import pandas as pd

#Loading the data
df = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/ir
df.columns = ['sl','sw','pl','pw','flower_type']
```

In [2]:
```
iris = df.copy()
iris.head()
```

Out[2]:

	sl	sw	pl	pw	flower_type
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

In [3]:
```
#To get the column names use columns attribute

iris.columns
```

Out[3]:
```
Index(['sl', 'sw', 'pl', 'pw', 'flower_type'], dtype='object')
```

In [4]:
```
#To get the indexs of the dataframe, index attribute is used

iris.index
```

Out[4]:
```
RangeIndex(start=0, stop=150, step=1)
```

In [5]:
```
# drop() function is used to delete a row from the dataset
iris.drop(0)
iris.head()
```

Out[5]:

	sl	sw	pl	pw	flower_type
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

In [6]:
```
#No changes have been made because drop function does not delete the rows from the
#It creates a copy of the dataframe and returns the updated copy

#To make changes in the original dataframe we need to pass a inplace=True argument
#in the original dataset

iris.drop(0, inplace=True)
iris.head()
```

Out[6]:

	sl	sw	pl	pw	flower_type
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
5	5.4	3.9	1.7	0.4	Iris-setosa

In [7]:
```
#To delete more than one columns in one go, we can pass a list of row numbers

iris.drop([4,5,8], inplace=True)
iris.head(10)
```

Out[7]:

	sl	sw	pl	pw	flower_type
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
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa

In [8]:
```
# index[i] is used to access the ith row and not the row whose row no is i

print(iris.index[0])


#Therefore iris.index[i] can also be passed to drop function to delete the ith row
iris.drop(iris.index[0], inplace=True)
iris.head(10)
```

```
1
```

Out[8]:

	sl	sw	pl	pw	flower_type
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa

In [ ]:

In [9]:
```
#to get the ith row from starting we use iloc

print(iris.head())
iris.iloc[0]
```

Out[9]:
```
    sl   sw   pl   pw  flower_type
2  4.7  3.2  1.3  0.2  Iris-setosa
3  4.6  3.1  1.5  0.2  Iris-setosa
6  4.6  3.4  1.4  0.3  Iris-setosa
7  5.0  3.4  1.5  0.2  Iris-setosa
9  4.9  3.1  1.5  0.1  Iris-setosa
sl                   4.7
sw                   3.2
pl                   1.3
pw                   0.2
flower_type    Iris-setosa
Name: 2, dtype: object
```

In [10]:
```
#Similarly to access the ith label row, we use loc

print(iris.head())
iris.loc[2]
```

Out[10]:
```
    sl   sw   pl   pw  flower_type
2  4.7  3.2  1.3  0.2  Iris-setosa
3  4.6  3.1  1.5  0.2  Iris-setosa
6  4.6  3.4  1.4  0.3  Iris-setosa
7  5.0  3.4  1.5  0.2  Iris-setosa
9  4.9  3.1  1.5  0.1  Iris-setosa
sl                   4.7
sw                   3.2
pl                   1.3
pw                   0.2
flower_type    Iris-setosa
Name: 2, dtype: object
```

In [ ]:

In [11]:
```
#To add a row into a dataframe,

iris.loc[0] = [1, 2, 3, 4, "Iris-setosa"]
```

```
#The entry will be added to the end
iris.tail()
```

Out[11]:

	sl	sw	pl	pw	flower_type
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica
0	1.0	2.0	3.0	4.0	Iris-setosa

In [12]:

```
#To reset the indices, reset_index() function is used

print("Dataframe before reset index")
print(iris.head(10))
iris.reset_index(inplace=True)
print("\nDataframe after reset index")
print(iris.head(10))

#Note that in the above dataframe the indices before resetting index are added as c
```

```
Dataframe before reset index
     sl   sw   pl   pw  flower_type
2   4.7  3.2  1.3  0.2  Iris-setosa
3   4.6  3.1  1.5  0.2  Iris-setosa
6   4.6  3.4  1.4  0.3  Iris-setosa
7   5.0  3.4  1.5  0.2  Iris-setosa
9   4.9  3.1  1.5  0.1  Iris-setosa
10  5.4  3.7  1.5  0.2  Iris-setosa
11  4.8  3.4  1.6  0.2  Iris-setosa
12  4.8  3.0  1.4  0.1  Iris-setosa
13  4.3  3.0  1.1  0.1  Iris-setosa
14  5.8  4.0  1.2  0.2  Iris-setosa

Dataframe after reset index
   index   sl   sw   pl   pw  flower_type
0      2  4.7  3.2  1.3  0.2  Iris-setosa
1      3  4.6  3.1  1.5  0.2  Iris-setosa
2      6  4.6  3.4  1.4  0.3  Iris-setosa
3      7  5.0  3.4  1.5  0.2  Iris-setosa
4      9  4.9  3.1  1.5  0.1  Iris-setosa
5     10  5.4  3.7  1.5  0.2  Iris-setosa
6     11  4.8  3.4  1.6  0.2  Iris-setosa
7     12  4.8  3.0  1.4  0.1  Iris-setosa
8     13  4.3  3.0  1.1  0.1  Iris-setosa
9     14  5.8  4.0  1.2  0.2  Iris-setosa
```

In [13]:

```
# To not add the previous index values in the dataframe, drop=True argument in pass

print("Dataframe before reset index")
print(iris.head(10))
iris.reset_index(drop=True, inplace=True)
print("\nDataframe after reset index")
print(iris.head(10))
```

```
Dataframe before reset index
   index   sl   sw   pl   pw  flower_type
0      2  4.7  3.2  1.3  0.2  Iris-setosa
1      3  4.6  3.1  1.5  0.2  Iris-setosa
2      6  4.6  3.4  1.4  0.3  Iris-setosa
3      7  5.0  3.4  1.5  0.2  Iris-setosa
4      9  4.9  3.1  1.5  0.1  Iris-setosa
5     10  5.4  3.7  1.5  0.2  Iris-setosa
6     11  4.8  3.4  1.6  0.2  Iris-setosa
7     12  4.8  3.0  1.4  0.1  Iris-setosa
8     13  4.3  3.0  1.1  0.1  Iris-setosa
9     14  5.8  4.0  1.2  0.2  Iris-setosa

Dataframe after reset index
   index   sl   sw   pl   pw  flower_type
0      2  4.7  3.2  1.3  0.2  Iris-setosa
1      3  4.6  3.1  1.5  0.2  Iris-setosa
2      6  4.6  3.4  1.4  0.3  Iris-setosa
3      7  5.0  3.4  1.5  0.2  Iris-setosa
4      9  4.9  3.1  1.5  0.1  Iris-setosa
5     10  5.4  3.7  1.5  0.2  Iris-setosa
6     11  4.8  3.4  1.6  0.2  Iris-setosa
7     12  4.8  3.0  1.4  0.1  Iris-setosa
8     13  4.3  3.0  1.1  0.1  Iris-setosa
9     14  5.8  4.0  1.2  0.2  Iris-setosa
```

In [14]:
```
#Now check the indices of the updated dataframe
iris.index
```

Out[14]:
```
RangeIndex(start=0, stop=146, step=1)
```

In [15]:
```
#Deleting columns is similar to deleting rows.
#We use drop function to delete whole columns with an extra argument axis=1
#By default value of axis is 0. That means, rows will be deleted by default

iris.drop('sl', axis=1, inplace=True)
iris.head()
```

Out[15]:

	index	sw	pl	pw	flower_type
0	2	3.2	1.3	0.2	Iris-setosa
1	3	3.1	1.5	0.2	Iris-setosa
2	6	3.4	1.4	0.3	Iris-setosa
3	7	3.4	1.5	0.2	Iris-setosa
4	9	3.1	1.5	0.1	Iris-setosa

In [16]:
```
iris = df.copy()
```

In [17]:
```
#To add new columns, we use
#df["column_name"] = value_to_be_added

iris['diff_pl_pw'] = iris['pl']-iris['pw']
iris.head()
```

Out[17]:

	sl	sw	pl	pw	flower_type	diff_pl_pw
0	5.1	3.5	1.4	0.2	Iris-setosa	1.2
1	4.9	3.0	1.4	0.2	Iris-setosa	1.2
2	4.7	3.2	1.3	0.2	Iris-setosa	1.1
3	4.6	3.1	1.5	0.2	Iris-setosa	1.3
4	5.0	3.6	1.4	0.2	Iris-setosa	1.2

In [18]:
```
iris = df.copy()
```

In [19]:
```
#Handling nan values

#First we will add some nan values in our iris dataset using a constant from numpy

import numpy as np

iris.iloc[2:4, 1:3] = np.nan
iris.head()
```

Out[19]:

	sl	sw	pl	pw	flower_type
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	NaN	NaN	0.2	Iris-setosa
3	4.6	NaN	NaN	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [20]:
```
#dropna function is used to drop the rows which have nan entries

df.dropna(inplace=True)
df.reset_index(drop=True, inplace=True)
df.head()
```

Out[20]:

	sl	sw	pl	pw	flower_type
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

In [21]:
```
iris.iloc[2:4, 1:3] = np.nan
iris.head()
```

Out[21]:

	sl	sw	pl	pw	flower_type
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	NaN	NaN	0.2	Iris-setosa
3	4.6	NaN	NaN	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [22]:

```
#To fill the nan entries, fillna function is used
#There are several ways to fill the nan entries

iris.sw.fillna(iris.sw.mean(), inplace=True)
print(iris.head())
print("\n")

iris.pl.fillna(iris.pl.mean(), inplace=True)
print(iris.head())
```

```
    sl        sw   pl   pw  flower_type
0  5.1  3.500000  1.4  0.2  Iris-setosa
1  4.9  3.000000  1.4  0.2  Iris-setosa
2  4.7  3.052703  NaN  0.2  Iris-setosa
3  4.6  3.052703  NaN  0.2  Iris-setosa
4  5.0  3.600000  1.4  0.2  Iris-setosa


    sl        sw        pl   pw  flower_type
0  5.1  3.500000  1.400000  0.2  Iris-setosa
1  4.9  3.000000  1.400000  0.2  Iris-setosa
2  4.7  3.052703  3.790541  0.2  Iris-setosa
3  4.6  3.052703  3.790541  0.2  Iris-setosa
4  5.0  3.600000  1.400000  0.2  Iris-setosa
```