



**Figure 11-1.** Pandas data structure

Let's check the data type of each column in the DataFrame.

```
my_DF.dtypes
'Output':
Capital      object
Population   int64
State        object
dtype: object
```

An **object** data type in Pandas represents **Strings**.

## Data Indexing (Selection/Subsets)

Similar to NumPy, Pandas objects can index or subset the dataset to retrieve a specific sub-record of the larger dataset. Note that data indexing returns a new **DataFrame** or **Series** if a 2-D or 1-D array is retrieved. They do not, however, alter the original dataset. Let's go through some examples of indexing a Pandas DataFrame.

First let's create a dataframe. Observe the default integer indices assigned.

```
# create the dataframe
my_DF = pd.DataFrame({'age': [15,17,21,29,25], \
                       'state_of_origin':['Lagos', 'Cross River', 'Kano', 'Abia', \
                                           'Benue']})
```

```
my_DF
'Output':
   age state_of_origin
0   15             Lagos
1   17      Cross River
2   21             Kano
3   29             Abia
4   25             Benue
```

## Selecting a Column from a DataFrame

Remember that the data type of a DataFrame column is a **Series** because it is a vector or 1-D array.

```
my_DF['age']
'Output':
0    15
1    17
2    21
3    29
4    25
Name: age, dtype: int64
# check data type
type(my_DF['age'])
'Output': pandas.core.series.Series
```

To select multiple columns, enclose the column names as **strings** with the double square brackets `[[ ]]`. The following code is an example:

```
my_DF[['age', 'state_of_origin']]
'Output':
   age state_of_origin
0   15             Lagos
1   17      Cross River
2   21             Kano
3   29             Abia
4   25             Benue
```