Step 1. Import the pandas and numpy libraries

Answer1: (This one has been done for you)

```
In [140]: import pandas as pd
     ...: import numpy as np
```

Step 2. Import the popular ‘iris’ dataset from the below address. And then check the header of the dataset.

https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Answer2: (This one has also been done for you)

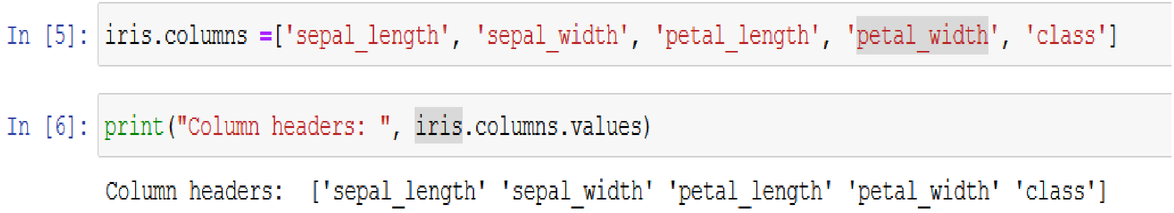
```
In [141]: url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'

In [142]: iris = pd.read_csv(url)

In [143]: iris.head()
Out[143]:
5.1 3.5 1.4 0.2 Iris-setosa
0 4.9 3.0 1.4 0.2 Iris-setosa
1 4.7 3.2 1.3 0.2 Iris-setosa
2 4.6 3.1 1.5 0.2 Iris-setosa
3 5.0 3.6 1.4 0.2 Iris-setosa
4 5.4 3.9 1.7 0.4 Iris-setosa
```

Step 3. You can see that the column headers are missing in the above case. Therefore this step is related to the creation of column heads for the dataset. Write code to create 5 column heads. Next write a code to display or show the headers.

1. sepal_length
2. sepal_width
3. petal_length
4. petal_width
5. class

Answer3: (write your code)

```
iris.columns =['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']
print("Column headers: ", iris.columns.values)
```

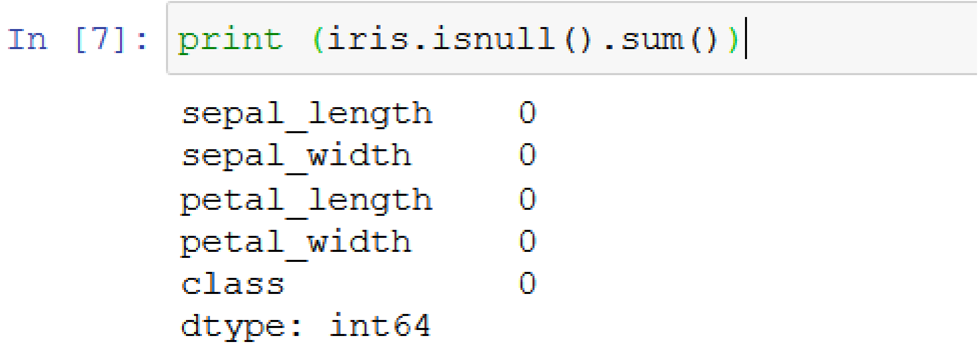


```
In [5]: iris.columns =['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']

In [6]: print("Column headers: ", iris.columns.values)

        Column headers:  ['sepal_length' 'sepal_width' 'petal_length' 'petal_width' 'class']
```

Step 4. Write a code to check if there are any missing values in the dataframe?

Answer4: (write your code)

```
print (iris.isnull().sum())
```



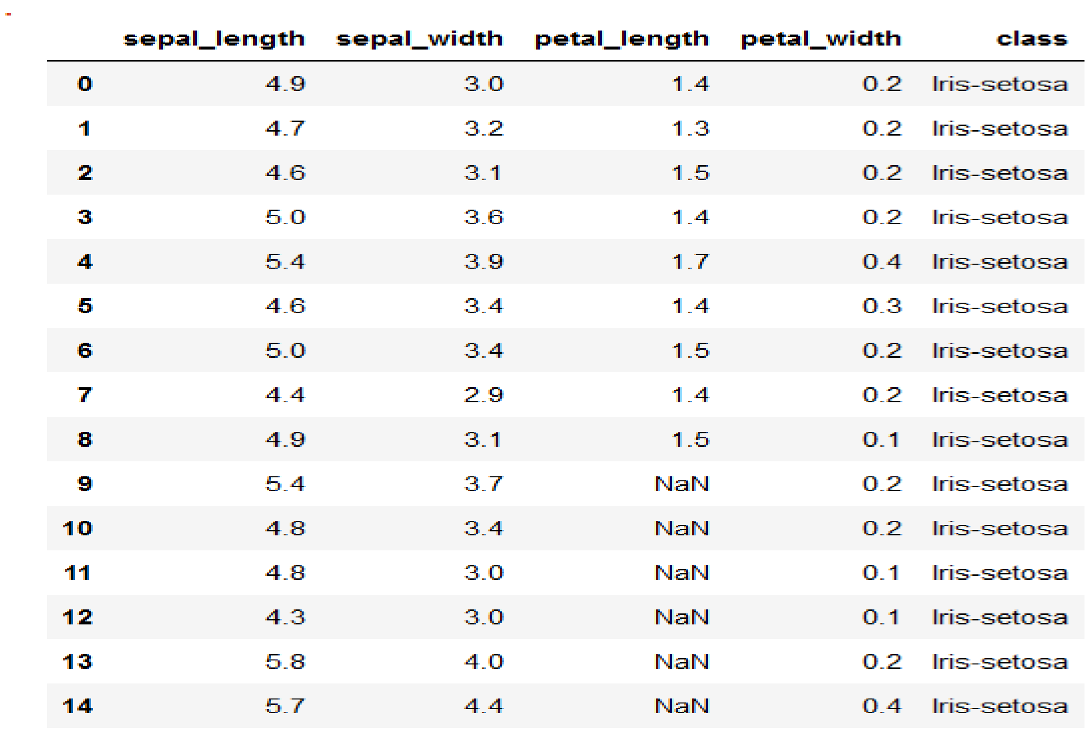
```
In [7]: print (iris.isnull().sum())

        sepal_length    0
        sepal_width     0
        petal_length    0
        petal_width     0
        class           0
        dtype: int64
```

Hints: there is no missing values but check it thorough the code

Step 5. Write a code to set the values of the rows 10 to 29 of the column 'petal_length' to NaN.

Answer5: (write your code)

```
iris.loc[iris.index[9:29], 'petal_length'] = np.nan
#Displaying some values set to NaN
```

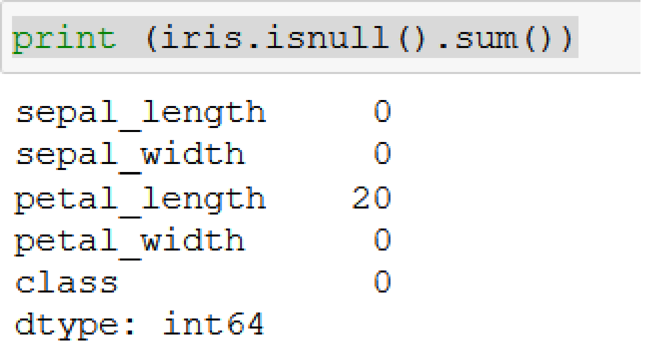


	sepal_length	sepal_width	petal_length	petal_width	class
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
5	4.6	3.4	1.4	0.3	Iris-setosa
6	5.0	3.4	1.5	0.2	Iris-setosa
7	4.4	2.9	1.4	0.2	Iris-setosa
8	4.9	3.1	1.5	0.1	Iris-setosa
9	5.4	3.7	NaN	0.2	Iris-setosa
10	4.8	3.4	NaN	0.2	Iris-setosa
11	4.8	3.0	NaN	0.1	Iris-setosa
12	4.3	3.0	NaN	0.1	Iris-setosa
13	5.8	4.0	NaN	0.2	Iris-setosa
14	5.7	4.4	NaN	0.4	Iris-setosa

Step 6. Now again, check if there is any missing values (NaN) in the dataframe? Count, how many missing values.

Answer6: (write your code)

```
print (iris.isnull().sum())
```



```
print (iris.isnull().sum())
```

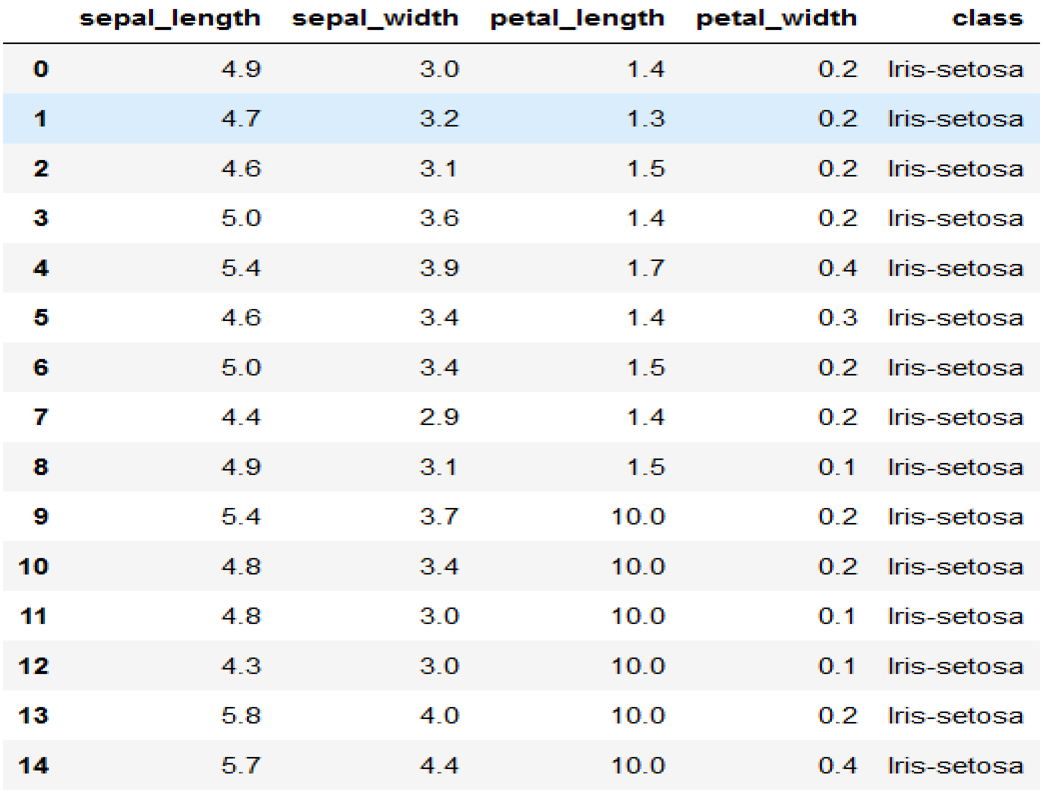
```
sepal_length     0
sepal_width      0
petal_length    20
petal_width      0
class            0
dtype: int64
```

Hints: this time you will have missing values.

Step 7. Substitute the NaN values to 10.0

Answer7: (write your code)

```
iris.fillna(10.0, inplace=True)
#displaying some values set to 10.0
```



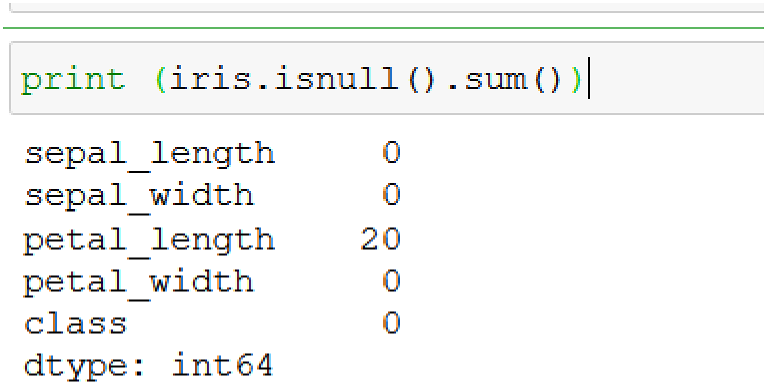
	sepal_length	sepal_width	petal_length	petal_width	class
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
5	4.6	3.4	1.4	0.3	Iris-setosa
6	5.0	3.4	1.5	0.2	Iris-setosa
7	4.4	2.9	1.4	0.2	Iris-setosa
8	4.9	3.1	1.5	0.1	Iris-setosa
9	5.4	3.7	10.0	0.2	Iris-setosa
10	4.8	3.4	10.0	0.2	Iris-setosa
11	4.8	3.0	10.0	0.1	Iris-setosa
12	4.3	3.0	10.0	0.1	Iris-setosa
13	5.8	4.0	10.0	0.2	Iris-setosa
14	5.7	4.4	10.0	0.4	Iris-setosa

Section 2

Python has support for multiple libraries like Pandas, Numpy, Matplotlib etc. These libraries come packed with functions which support large scale data loading, manipulation, processing, and visualization.
Some common libraries are:

- **Pandas**: This library is used to load datasets and provides operations such as joins, slicing etc. There is support for loading and preprocessing data from various sources like csv, excel sheets and from other sources into memory.

Null check example on iris data from section 1: isnull() and **isna**() are two functions which provide null check functionality in pandas. We can use this to find and handle missing data.

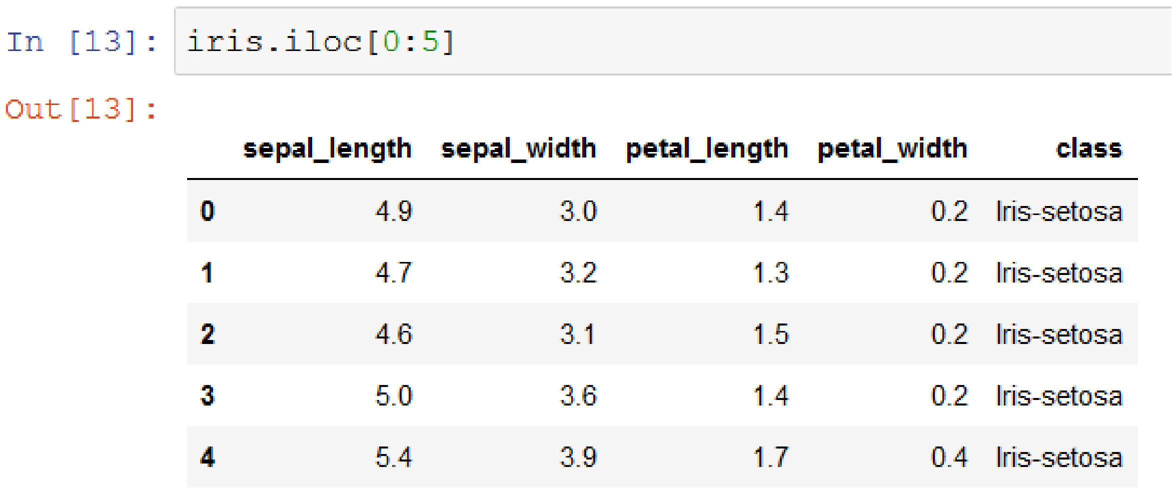


```
print (iris.isnull().sum())
```

```
sepal_length     0
sepal_width      0
petal_length    20
petal_width      0
class            0
dtype: int64
```

Slicing example on iris dataset:

We can use slicing to select a subset of data which has rows which are interesting to us. This is especially useful when we must select small portion of a large data for processing. In this example we select top 5 rows from iris datasets.



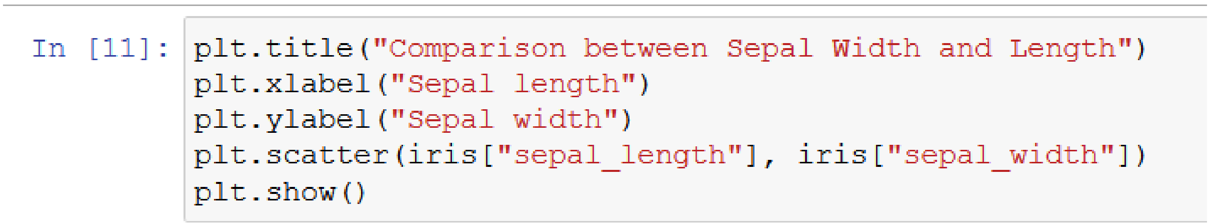
```
In [13]: iris.iloc[0:5]
```

Out[13]:

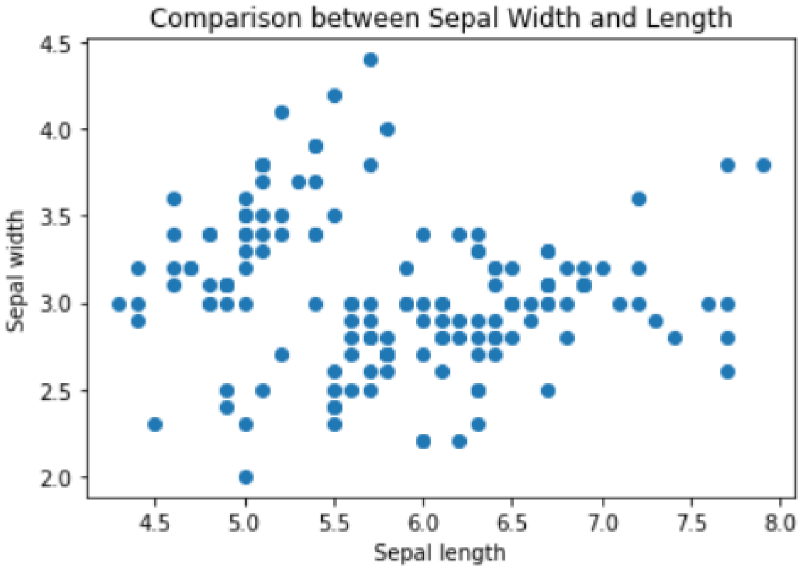
	sepal_length	sepal_width	petal_length	petal_width	class
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

- **Numpy**: This library is used to perform fast and optimized operations on big datasets, generally in form of n dimensional arrays.

- **Matplotlib:** This example shows how we can use matplotlib to visualize sepal length and width from iris dataset. This library is generally used to visualize data.



```
In [11]: plt.title("Comparison between Sepal Width and Length")
         plt.xlabel("Sepal length")
         plt.ylabel("Sepal width")
         plt.scatter(iris["sepal_length"], iris["sepal_width"])
         plt.show()
```


Comparison between Sepal Width and Length
Sepal width
Sepal length