In [ ]:
```
#Data Wrangling, I
Perform the following operations using Python on any open source dataset (e.g.
1. Import all the required Python Libraries.
2. Locate an open source data from the web (e.g., https://www.kaggle.com). Pro
 description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into pandas dataframe.
4. Data Preprocessing: check for missing values in the data using pandas isnul
function to get some initial statistics. Provide variable descriptions. Types
Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by
the data types (i.e., character, numeric, integer, factor, and logical) of the
data set. If variables are not in the correct data type, apply proper type cor
6. Turn categorical variables into quantitative variables in Python.
In addition to the codes and outputs, explain every operation that you do in t
explain everything that you do to import/read/scrape the data set.
```

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

In [2]:
```
csv_url = 'https://archive.ics.uci.edu/ml/
machine-learning-databases/iris/iris.data'
```

In [3]:
```
iris = pd.read_csv(csv_url, header = None)
```

In [4]:
```
col_names = ['Sepal_Length','Sepal_Width',
             'Petal_Length','Petal_Width','Species']
```

In [5]:
```
iris = pd.read_csv(csv_url, names = col_names)
```

In [6]:
```
df=pd.DataFrame(iris)
```

In [7]:
```
df
```

Out[7]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
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

150 rows × 5 columns

Data preprocessing

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

Out[8]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
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 [9]:
```
df.head(n=7)
```

Out[9]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
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
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa

In [10]:
```
df.tail()
```

Out[10]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
145	6.7	3.0	5.2	2.3	Iris-virginica
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

In [11]:
```
df.index
```

Out[11]:
```
RangeIndex(start=0, stop=150, step=1)
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width',
       'Species'],
      dtype='object')
```

In [13]:
```
df.shape
```

Out[13]:
```
(150, 5)
```

In [14]:
```
df.dtypes
```

Out[14]:
```
Sepal_Length    float64
Sepal_Width     float64
Petal_Length    float64
Petal_Width     float64
Species          object
dtype: object
```

In [15]:
```
df.describe()
```

Out[15]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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 [16]:
```
df['Sepal_Length']
```

Out[16]:
```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
      ...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: Sepal_Length, Length: 150, dtype: float64
```

In [17]:
```
df.sort_values(by="Sepal_Length")
```

Out[17]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
13	4.3	3.0	1.1	0.1	Iris-setosa
42	4.4	3.2	1.3	0.2	Iris-setosa
38	4.4	3.0	1.3	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
41	4.5	2.3	1.3	0.3	Iris-setosa
...	...	...	...	...	...
122	7.7	2.8	6.7	2.0	Iris-virginica
118	7.7	2.6	6.9	2.3	Iris-virginica
117	7.7	3.8	6.7	2.2	Iris-virginica
135	7.7	3.0	6.1	2.3	Iris-virginica
131	7.9	3.8	6.4	2.0	Iris-virginica

150 rows × 5 columns

iloc is used to select first n rows from dataframe

In [18]:
```
df.iloc[5]
```

Out[18]:
```
Sepal_Length            5.4
Sepal_Width             3.9
Petal_Length            1.7
Petal_Width             0.4
Species         Iris-setosa
Name: 5, dtype: object
```

In [19]:
```
df[0:3]
```

Out[19]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
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

checking is there any null value

In [20]:
```
df.isnull()
```

Out[20]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [21]:
```
df.isna()
```

Out[21]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

data formatting

In [22]:
```
df.dtypes
```

Out[22]:
```
Sepal_Length    float64
Sepal_Width     float64
Petal_Length    float64
Petal_Width     float64
Species          object
dtype: object
```

In [23]:
```
df['Petal_Length']=df['Petal_Length'].astype("int")
```

In [24]:
```
df.dtypes
```

Out[24]:
```
Sepal_Length    float64
Sepal_Width     float64
Petal_Length      int32
Petal_Width     float64
Species          object
dtype: object
```

data normalization

In [25]:
```
from sklearn import preprocessing
```

In [26]:
```
df.head()
```

Out[26]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1	0.2	Iris-setosa
1	4.9	3.0	1	0.2	Iris-setosa
2	4.7	3.2	1	0.2	Iris-setosa
3	4.6	3.1	1	0.2	Iris-setosa
4	5.0	3.6	1	0.2	Iris-setosa

In [27]:
```
min_max_scaler = preprocessing.MinMaxScaler()
```

In [28]:
```
x=df.iloc[:,:4]
```

In [29]:
```
x_scaled = min_max_scaler.fit_transform(x)
```

In [30]:
```
df_normalized = pd.DataFrame(x_scaled)
```

In [31]:
```
df_normalized
```

Out[31]:

	0	1	2	3
0	0.222222	0.625000	0.0	0.041667
1	0.166667	0.416667	0.0	0.041667
2	0.111111	0.500000	0.0	0.041667
3	0.083333	0.458333	0.0	0.041667
4	0.194444	0.666667	0.0	0.041667
...	...	...	...	...
145	0.666667	0.416667	0.8	0.916667
146	0.555556	0.208333	0.8	0.750000
147	0.611111	0.416667	0.8	0.791667
148	0.527778	0.583333	0.8	0.916667
149	0.444444	0.416667	0.8	0.708333

150 rows × 4 columns

Turn categorical variables into quantitative variables

In [32]:
```
df['Species'].unique()
```

Out[32]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [33]:
```
label_encoder = preprocessing.LabelEncoder()
```

In [34]:
```
df['Species']= label_encoder.fit_transform(df['Species'])
```

In [35]:
```
df['Species'].unique()
```

Out[35]:
```
array([0, 1, 2])
```

In [36]:
```
df
```

Out[36]:

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
0	5.1	3.5	1	0.2	0
1	4.9	3.0	1	0.2	0
2	4.7	3.2	1	0.2	0
3	4.6	3.1	1	0.2	0
4	5.0	3.6	1	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5	2.3	2
146	6.3	2.5	5	1.9	2
147	6.5	3.0	5	2.0	2
148	6.2	3.4	5	2.3	2
149	5.9	3.0	5	1.8	2

150 rows × 5 columns

In [ ]:
```

```