Assignment No. 1

Aim : Data Wrangling I Perform the following operations using Python on any open source dataset (e.g., data.csv)

1. Import all the required Python Libraries.
2. Locate an open source data from the web (e.g., https://www.kaggle.com (https://www.kaggle.com)). Provide a clear 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 isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
6. Turn categorical variables into quantitative variables in Python. In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set.

Code :

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from pandas import DataFrame, Series
```

```
import seaborn as ans
```

In [6]:
```
data = ans.load_dataset("iris")
```

In [12]:
```
print(data)
```

In [14]:

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

In [18]:
```
print(data)
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica
```

```
1
```

```
[150 rows x 5 columns]
```

In [19]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
  0   sepal_length  150 non-null    float64
  1   sepal_width   150 non-null    float64
  2   petal_length  150 non-null    float64
  3   petal_width   150 non-null    float64
  4   species       150 non-null    object dtypes: float64(4), object(1) memory usage: 6.0+ KB
```

In [21]:
```
1 data.head()
```

Out[21]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
1 data.tail()
```

In [22]:

Out[22]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
1 data.describe()
```

In [23]:

Out[23]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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

In

```
1
```

max	7.900000	4.400000	6.900000	2.500000

In [24]:

```
1 top_left_corner_df = data.iloc[:4, :4]
```

In [25]:

```
1 print(top_left_corner_df)
```

```
   sepal_length  sepal_width  petal_length  petal_width
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
```

[27]:

```
data.to_csv()
```

Out[27]:

```
',sepal_length,sepal_width,petal_length,petal_width,species\r\n0,5.1,3.5,1.4,0.2,setosa\r\n1,4.9,
3.0,1.4,0.2,setosa\r\n2,4.7,3.2,1.3,0.2,setosa\r\n3,4.6,3.1,1.5,0.2,setosa\r\n4,5.0,3.6,1.4,0.2,
se
tosa\r\n5,5.4,3.9,1.7,0.4,setosa\r\n6,4.6,3.4,1.4,0.3,setosa\r\n7,5.0,3.4,1.5,0.2,setosa\r\n8,4.
4,
2.9,1.4,0.2,setosa\r\n9,4.9,3.1,1.5,0.1,setosa\r\n10,5.4,3.7,1.5,0.2,setosa\r\n11,4.8,3.4,1.6,0.
2,
setosa\r\n12,4.8,3.0,1.4,0.1,setosa\r\n13,4.3,3.0,1.1,0.1,setosa\r\n14,5.8,4.0,1.2,0.2,setosa\r\
n1
5,5.7,4.4,1.5,0.4,setosa\r\n16,5.4,3.9,1.3,0.4,setosa\r\n17,5.1,3.5,1.4,0.3,setosa\r\n18,5.7,3.8
,
1.7,0.3,setosa\r\n19,5.1,3.8,1.5,0.3,setosa\r\n20,5.4,3.4,1.7,0.2,setosa\r\n21,5.1,3.7,1.5,0.4,se
t
osa\r\n22,4.6,3.6,1.0,0.2,setosa\r\n23,5.1,3.3,1.7,0.5,setosa\r\n24,4.8,3.4,1.9,0.2,setosa\r\n25
,
5.0,3.0,1.6,0.2,setosa\r\n26,5.0,3.4,1.6,0.4,setosa\r\n27,5.2,3.5,1.5,0.2,setosa\r\n28,5.2,3.4,1
.
4,0.2,setosa\r\n29,4.7,3.2,1.6,0.2,setosa\r\n30,4.8,3.1,1.6,0.2,setosa\r\n31,5.4,3.4,1.5,0.4,set
os
a\r\n32,5.2,4.1,1.5,0.1,setosa\r\n33,5.5,4.2,1.4,0.2,setosa\r\n34,4.9,3.1,1.5,0.2,setosa\r\n35,5
.
0,3.2,1.2,0.2,setosa\r\n36,5.5,3.5,1.3,0.2,setosa\r\n37,4.9,3.6,1.4,0.1,setosa\r\n38,4.4,3.0,1.3
,
0.2,setosa\r\n39,5.1,3.4,1.5,0.2,setosa\r\n40,5.0,3.5,1.3,0.3,setosa\r\n41,4.5,2.3,1.3,0.3,setosa
\r\n42,4.4,3.2,1.3,0.2,setosa\r\n43,5.0,3.5,1.6,0.6,setosa\r\n44,5.1,3.8,1.9,0.4,setosa\r\n45,4.
8,
3.0,1.4,0.3,setosa\r\n46,5.1,3.8,1.6,0.2,setosa\r\n47,4.6,3.2,1.4,0.2,setosa\r\n48,5.3,3.7,1.5,0
.
2,setosa\r\n49,5.0,3.3,1.4,0.2,setosa\r\n50,7.0,3.2,4.7,1.4,versicolor\r\n51,6.4,3.2,4.5,1.5,ver
si
color\r\n52,6.9,3.1,4.9,1.5,versicolor\r\n53,5.5,2.3,4.0,1.3,versicolor\r\n54,6.5,2.8,4.6,1.5,ve
rs
icolor\r\n55,5.7,2.8,4.5,1.3,versicolor\r\n56,6.3,3.3,4.7,1.6,versicolor\r\n57,4.9,2.4,3.3,1.0,v
er
sicolor\r\n58,6.6,2.9,4.6,1.3,versicolor\r\n59,5.2,2.7,3.9,1.4,versicolor\r\n60,5.0,2.0,3.5,1.0,
ve
rsicolor\r\n61,5.9,3.0,4.2,1.5,versicolor\r\n62,6.0,2.2,4.0,1.0,versicolor\r\n63,6.1,2.9,4.7,1.4
,v
ersicolor\r\n64,5.6,2.9,3.6,1.3,versicolor\r\n65,6.7,3.1,4.4,1.4,versicolor\r\n66,5.6,3.0,4.5,1.
5,
versicolor\r\n67,5.8,2.7,4.1,1.0,versicolor\r\n68,6.2,2.2,4.5,1.5,versicolor\r\n69,5.6,2.5,3.9,1
.
1,versicolor\r\n70,5.9,3.2,4.8,1.8,versicolor\r\n71,6.1,2.8,4.0,1.3,versicolor\r\n72,6.3,2.5,4.9
,
1.5,versicolor\r\n73,6.1,2.8,4.7,1.2,versicolor\r\n74,6.4,2.9,4.3,1.3,versicolor\r\n75,6.6,3.0,4
.
4,1.4,versicolor\r\n76,6.8,2.8,4.8,1.4,versicolor\r\n77,6.7,3.0,5.0,1.7,versicolor\r\n78,6.0,2.9
,
4.5,1.5,versicolor\r\n79,5.7,2.6,3.5,1.0,versicolor\r\n80,5.5,2.4,3.8,1.1,versicolor\r\n81,5.5,2
.
4,3.7,1.0,versicolor\r\n82,5.8,2.7,3.9,1.2,versicolor\r\n83,6.0,2.7,5.1,1.6,versicolor\r\n84,5.4
,
```

```
1
```

```
3.0,4.5,1.5,versicolor\r\n85,6.0,3.4,4.5,1.6,versicolor\r\n86,6.7,3.1,4.7,1.5,versicolor\r\n87,6
.
3,2.3,4.4,1.3,versicolor\r\n88,5.6,3.0,4.1,1.3,versicolor\r\n89,5.5,2.5,4.0,1.3,versicolor\r\n90
,
5.5,2.6,4.4,1.2,versicolor\r\n91,6.1,3.0,4.6,1.4,versicolor\r\n92,5.8,2.6,4.0,1.2,versicolor\r\n9
3,5.0,2.3,3.3,1.0,versicolor\r\n94,5.6,2.7,4.2,1.3,versicolor\r\n95,5.7,3.0,4.2,1.2,versicolor\r
\n
96,5.7,2.9,4.2,1.3,versicolor\r\n97,6.2,2.9,4.3,1.3,versicolor\r\n98,5.1,2.5,3.0,1.1,versicolor\r
\n99,5.7,2.8,4.1,1.3,versicolor\r\n100,6.3,3.3,6.0,2.5,virginica\r\n101,5.8,2.7,5.1,1.9,virginica
\r\n102,7.1,3.0,5.9,2.1,virginica\r\n103,6.3,2.9,5.6,1.8,virginica\r\n104,6.5,3.0,5.8,2.2,virgini
c
a\r\n105,7.6,3.0,6.6,2.1,virginica\r\n106,4.9,2.5,4.5,1.7,virginica\r\n107,7.3,2.9,6.3,1.8,virgi
ni
ca\r\n108,6.7,2.5,5.8,1.8,virginica\r\n109,7.2,3.6,6.1,2.5,virginica\r\n110,6.5,3.2,5.1,2.0,virg
in
ica\r\n111,6.4,2.7,5.3,1.9,virginica\r\n112,6.8,3.0,5.5,2.1,virginica\r\n113,5.7,2.5,5.0,2.0,vir
gi
nica\r\n114,5.8,2.8,5.1,2.4,virginica\r\n115,6.4,3.2,5.3,2.3,virginica\r\n116,6.5,3.0,5.5,1.8,vi
rg
inica\r\n117,7.7,3.8,6.7,2.2,virginica\r\n118,7.7,2.6,6.9,2.3,virginica\r\n119,6.0,2.2,5.0,1.5,v
ir
ginica\r\n120,6.9,3.2,5.7,2.3,virginica\r\n121,5.6,2.8,4.9,2.0,virginica\r\n122,7.7,2.8,6.7,2.0,
vi
rginica\r\n123,6.3,2.7,4.9,1.8,virginica\r\n124,6.7,3.3,5.7,2.1,virginica\r\n125,7.2,3.2,6.0,1.8
,v
irginica\r\n126,6.2,2.8,4.8,1.8,virginica\r\n127,6.1,3.0,4.9,1.8,virginica\r\n128,6.4,2.8,5.6,2.
1,
virginica\r\n129,7.2,3.0,5.8,1.6,virginica\r\n130,7.4,2.8,6.1,1.9,virginica\r\n131,7.9,3.8,6.4,2
.
0,virginica\r\n132,6.4,2.8,5.6,2.2,virginica\r\n133,6.3,2.8,5.1,1.5,virginica\r\n134,6.1,2.6,5.6
,
1.4,virginica\r\n135,7.7,3.0,6.1,2.3,virginica\r\n136,6.3,3.4,5.6,2.4,virginica\r\n137,6.4,3.1,5
.
5,1.8,virginica\r\n138,6.0,3.0,4.8,1.8,virginica\r\n139,6.9,3.1,5.4,2.1,virginica\r\n140,6.7,3.1
,
5.6,2.4,virginica\r\n141,6.9,3.1,5.1,2.3,virginica\r\n142,5.8,2.7,5.1,1.9,virginica\r\n143,6.8,3
.
2,5.9,2.3,virginica\r\n144,6.7,3.3,5.7,2.5,virginica\r\n145,6.7,3.0,5.2,2.3,virginica\r\n146,6.3
,
2.5,5.0,1.9,virginica\r\n147,6.5,3.0,5.2,2.0,virginica\r\n148,6.2,3.4,5.4,2.3,virginica\r\n149,5
.
9,3.0,5.1,1.8,virginica\r\n'
```

In [28]:

```
1 ash = data.copy()
```

In [29]:

```
1 print(ash)
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

In

```
1
```

In [ ]:

In [34]:

In [31]:

Out[34]:

Out[31]:

```
sepal_length    150
sepal_width     150
petal_length    150
petal_width     150
species         150
dtype: int64
```

```
1 data.cummax()
```

	sepal_length	sepal_width	petal_length	petal_width	species

150 rows × 5 columns

```
1 data.cummin()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
0	5.1	3.5	1.4	0.2	setosa
1	5.1	3.5	1.4	0.2	setosa
2	5.1	3.5	1.4	0.2	setosa
3	5.1	3.5	1.5	0.2	setosa
4	5.1	3.6	1.5	0.2	setosa
...	...	...	...	...	...
145	7.9	4.4	6.9	2.5	virginica
146	7.9	4.4	6.9	2.5	virginica
147	7.9	4.4	6.9	2.5	virginica
148	7.9	4.4	6.9	2.5	virginica
149	7.9	4.4	6.9	2.5	virginica

In [35]:

Out[35]:

1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.0	1.3	0.2	setosa
3	4.6	3.0	1.3	0.2	setosa
4	4.6	3.0	1.3	0.2	setosa
...	...	...	...	...	...
145	4.3	2.0	1.0	0.1	setosa
146	4.3	2.0	1.0	0.1	setosa
147	4.3	2.0	1.0	0.1	setosa
148	4.3	2.0	1.0	0.1	setosa
149	4.3	2.0	1.0	0.1	setosa

```
1
```

150 rows × 5 columns

In [36]:
```
data.dropna()
```

Out[36]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
1 data.any()
```

```
sepal_length    True
sepal_width     True
petal_length    True
```

In [37]:
```
petal_width     True
```

Out[37]:
```
species         True
```

In

```
1
```

In

```
dtype: bool
```

In [39]:

```
1 data.get(40)
```

In [40]:

```
1 mr = data.get(40)
```

In [41]:

```
1 print(mr)
```

```
None
```

In [4]:

```
1 import seaborn as sea
2
```

```
1 data = sea.get_dataset_names()
```

In [5]:

```
1 print(data)
```

In [6]:

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamonds', 'dots',
'dowjon es', 'exercise', 'flights', 'fmri', 'geyser', 'glue', 'healthexp', 'iris', 'mpg',
'penguins', 'pla nets', 'seaice', 'taxis', 'tips', 'titanic', 'anagrams', 'anagrams',
'anscombe', 'anscombe', 'atte ntion', 'attention', 'brain_networks', 'brain_networks',
'car_crashes', 'car_crashes', 'diamonds', 'diamonds', 'dots', 'dots', 'dowjones', 'dowjones',
'exercise', 'exercise', 'flights', 'flights', 'fmri', 'fmri', 'geyser', 'geyser', 'glue',
'glue', 'healthexp', 'healthexp', 'iris', 'iris', 'mp g', 'mpg', 'penguins', 'penguins',
'planets', 'planets', 'seaice', 'seaice', 'taxis', 'taxis', 'ti ps', 'tips', 'titanic',
'titanic', 'anagrams', 'anscombe', 'attention', 'brain_networks', 'car_cra shes', 'diamonds',
'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glue', 'healthex p', 'iris',
'mpg', 'penguins', 'planets', 'seaice', 'taxis', 'tips', 'titanic']
```

In [7]:

```
1 data = sea.load_dataset("iris")
```

In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
  0   sepal_length  150 non-null    float64
  1   sepal_width   150 non-null    float64
  2   petal_length  150 non-null    float64
  3   petal_width   150 non-null    float64
  4   species       150 non-null    object dtypes: float64(4), object(1) memory usage: 6.0+ KB
```

In [9]:

```
1 data.describe()
```

Out[9]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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 [10]:
1
1 data.head()
```

Out[10]:

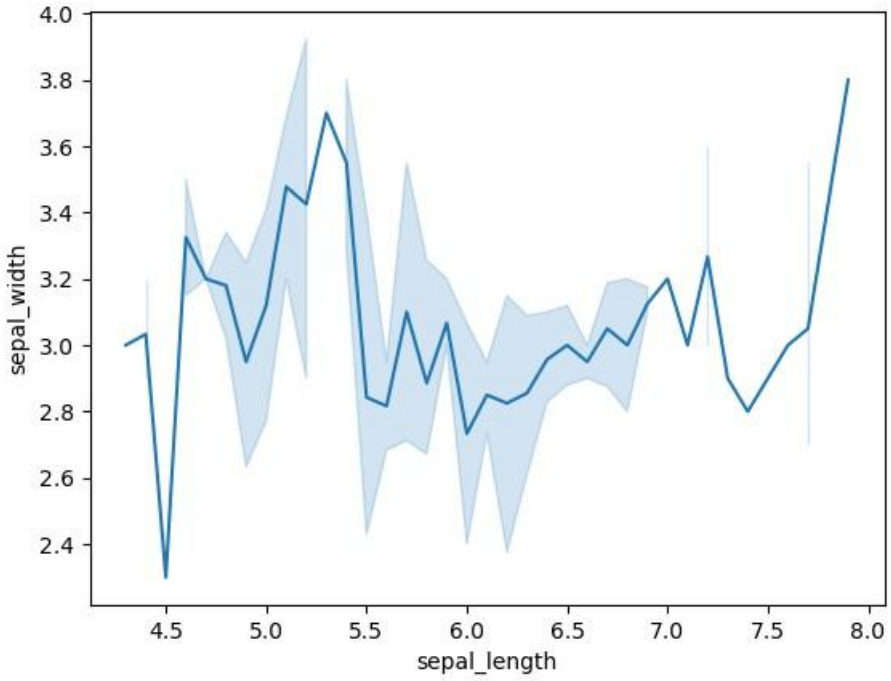
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
[11]: sea.lineplot(x="sepal_length", y="sepal_width", data=data)
```

```
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_n a option is deprecated and will be removed in a future version. Convert inf values
to NaN before o perating instead.  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Welcome\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_n a option is deprecated and will be removed in a future version. Convert inf values
to NaN before o perating instead.  with pd.option_context('mode.use_inf_as_na', True):
```

Out[11]: <Axes: xlabel='sepal_length', ylabel='sepal_width'>

In




```
1 data.min()
```

Out[12]:
```
sepal_length       4.3
sepal_width        2.0
[12]:
petal_length       1.0
petal_width        0.1
species         setosa
dtype: object
```

In [13]:

```
1 data.max()
```

Out[13]:
```
sepal_length          7.9
sepal_width           4.4
petal_length          6.9
petal_width           2.5

species         virginica dtype: object
```

In

```
1
```

In [15]:

```
1 data.mode()
```

Out[15]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.0	3.0	1.4	0.2	setosa
1	NaN	NaN	1.5	NaN	versicolor
2	NaN	NaN	NaN	NaN	virginica

In [7]:

```
import seaborn as san
data = san.get_dataset_names()
data
```

```
Out[7]: ['anagrams',
         'anscombe',
         'attention',
         'brain_networks'
         ,
         'car_crashes',
         'diamonds',
         'dots',
         'dowjones',
         'exercise',
         'flights',
         'fmri',
         'geyser',
         'glue',
         'healthexp',
         'iris',
         'mpg',
         'penguins',
         'planets',
         'seaice',
         'taxis',
         'tips',
         'titanic',
         'anagrams',
         'anagrams',
         'anscombe',
         'anscombe',
         'attention',
         'attention',
         'brain_networks'
         ,
         'brain_networks'
         ,
         'car_crashes',
         'car_crashes',
         'diamonds',
         'diamonds',
         'dots',
         'dots',
         'dowjones',
         'dowjones',
         'exercise',
         'exercise',
         'flights',
         'flights',
         'fmri',
         'fmri',
         'geyser',
         'geyser',
         'glue',
         'glue',
         'healthexp',
         'healthexp',
         'iris',
         'iris',
         'mpg',
         'mpg',
         'penguins',
         'penguins',
         'planets',
         'planets',
         'seaice',
         'seaice',
         'taxis',
         'taxis',
         'tips',
         'tips',
         'titanic',
         'titanic',
         'anagrams',
```

```
'anscombe',
'attention',
'brain_networks'
,
'car_crashes',
'diamonds',
'dots',
'dowjones',
'exercise',
'flights',
'fmri',
'geyser',
'glue',
'healthexp',
'iris',
'mpg',
'penguins',
'planets',
'seaice',
'taxis',
'tips',
'titanic']
```

In [16]:
```
df.describe(include='all')
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width	species
count	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	setosa
freq	NaN	NaN	NaN	NaN	50
mean	5.843333	3.057333	3.758000	1.199333	NaN
std	0.828066	0.435866	1.765298	0.762238	NaN
min	4.300000	2.000000	1.000000	0.100000	NaN
25%	5.100000	2.800000	1.600000	0.300000	NaN
50%	5.800000	3.000000	4.350000	1.300000	NaN
75%	6.400000	3.300000	5.100000	1.800000	NaN
max	7.900000	4.400000	6.900000	2.500000	NaN

In [20]:
```
df.sort_index(axis=1, ascending=False)
```

Out[20]:

	species	sepal_width	sepal_length	petal_width	petal_length
0	setosa	3.5	5.1	0.2	1.4

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

150 rows × 5 columns

60	5.0	2.0	3.5	1.0	versicolor
62	6.0	2.2	4.0	1.0	versicolor
119	6.0	2.2	5.0	1.5	virginica
68	6.2	2.2	4.5	1.5	versicolor
41	4.5	2.3	1.3	0.3	setosa
...	...	...	...	...	...
16	5.4	3.9	1.3	0.4	setosa
14	5.8	4.0	1.2	0.2	setosa
32	5.2	4.1	1.5	0.1	setosa
33	5.5	4.2	1.4	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa

150 rows × 5 columns

In

In [25]:

Out[25]:
```
sepal_length       5.4
sepal_width        3.9
petal_length       1.7
petal_width        0.4
species         setosa
```

In [24]:
```
df.iloc[5]
```

```
df.sort_values(by="sepal_width")
```

Out[24]:
```
sepal_length    sepal_width
petal_length    petal_width
species
Name: 5, dtype: object
```

```
df[0:3]
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa

In [26]:

Out[26]:

2	4.7	3.2	1.3	0.2	setosa

```
df.loc[:, ["sepal_width","petal_length"]]
```

	sepal_width	petal_length
0	3.5	1.4
1	4.9	3.0
1.4	0.2	setosa

In [27]:

Out[27]:

1	3.0	1.4
2	3.2	1.3
3	3.1	1.5
4	3.6	1.4
...	...	...
145	3.0	5.2
146	2.5	5.0
147	3.0	5.2

In

```
1
```

148	3.4	5.4
149	3.0	5.1

rows × 2 columns

In [29]:

```
1 df.iloc[:30, :]
```

Out[29]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa
10	5.4	3.7	1.5	0.2	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
17	5.1	3.5	1.4	0.3	setosa
18	5.7	3.8	1.7	0.3	setosa
19	5.1	3.8	1.5	0.3	setosa
20	5.4	3.4	1.7	0.2	setosa
21	5.1	3.7	1.5	0.4	setosa
22	4.6	3.6	1.0	0.2	setosa
23	5.1	3.3	1.7	0.5	setosa
24	4.8	3.4	1.9	0.2	setosa
25	5.0	3.0	1.6	0.2	setosa
26	5.0	3.4	1.6	0.4	setosa
27	5.2	3.5	1.5	0.2	setosa
28	5.2	3.4	1.4	0.2	setosa
29	4.7	3.2	1.6	0.2	setosa

In

In [30]:

```
df.iloc[:, :17]
```

Out[30]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

[31]:

```
df.iloc[:6,
:12]
```

Out[31]:

	sepal_length	sepal_width	petal_length	petal_width	species
5	5.4	3.9	1.7	0.4	setosa

```
df.iloc[3:5, 0:2]
```

	sepal_length	sepal_width
3	4.6	3.1
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

In [32]:

Out[32]:

```
df.iloc[[1, 2,4], [0, 2]]
```

	sepal_length	petal_length
1	4.9	1.4
4	5.0	3.6

In [33]:

Out[33]:

2	4.7	1.3

In

```
1
```

4	5.0	1.4	

In [34]:

```
1 df.iloc[1:3, :]
```

Out[34]:

	sepal_length	sepal_width	petal_length	petal_width	species
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2			
1.3	0.2	setosa			

In [35]:

```
1 df.iloc[:, 1:3]
```

Out[35]:

	sepal_width	petal_length
0	3.5	1.4
1	3.0	1.4
2	3.2	1.3
3	3.1	1.5
4	3.6	1.4
...	...	...
145	3.0	5.2
146	2.5	5.0
147	3.0	5.2
148	3.4	5.4
149	3.0	5.1

150 rows × 2 columns

In [36]:

```
1 df.iloc[1, 1]
```

Out[36]:

```
1 df['sepal_length'].iloc[5]
```

5.4
3.0

In [38]:

```
1 cols_2_4 = df.columns[2:4]
2 df[cols_2_4]
```

Out[38]:

	petal_length	petal_width
0	1.4	0.2

[41]:

Out[41]:

1	1.4	0.2
2	1.3	0.2
3	1.5	0.2
4	1.4	0.2
...	...	...
145	5.2	2.3

In

146	5.0	1.9
147	5.2	2.0
148	5.4	2.3

150 rows × 2 columns

```
1 df[df.columns[2:4]].iloc[5:10]
```

	petal_length	petal_width
5	1.7	0.4
149	5.1	1.8

In [42]:

Out[42]:

6	1.4	0.3
7	1.5	0.2
8	1.4	0.2
9	1.5	0.1

In [43]:

```
1 df.isnull()
```

Out[43]:

	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 [44]:

Out[44]:

```
sepal_length    False
sepal_width     False
petal_length    False
petal_width     False
species         False
dtype: bool
```

In [46]:

Out[46]:

```
0
```

[47]:

```
1 df.isnull().sum()
```

df.isnull().sum()

Out[47]:

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

In

```
1
```

In [49]:

```
1 df.isnull().sum(axis=1)
```

Out[49]:

```
0      0
1      0
2      0
3      0
4      0
      ..
145    0
146    0
147    0
148    0
149    0
Length: 150, dtype:
int64
```

In [50]:

```
1 df.isna().sum()
```

Out[50]:

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

In [51]:

```
1 df.petal_length.isnull().sum()
```

Out[51]:

```
0
```

In [53]:
```
df.groupby(['sepal_length'])['petal_width'].apply(lambda x:x.isnull().sum())
```

Out[53]:
```
sepal_length
4.3    0
4.4    0
4.5    0
4.6    0
4.7    0
4.8    0
4.9    0
5.0    0
5.1    0
5.2    0
5.3    0
5.4    0
5.5    0
5.6    0
5.7    0
5.8    0
5.9    0
6.0    0
6.1    0
6.2    0
6.3    0
6.4    0
6.5    0
6.6    0
6.7    0
6.8    0
6.9    0
7.0    0
7.1    0
7.2    0
7.3    0
7.4    0
7.6    0
7.7    0
7.9    0 Name: petal_width,
dtype: int64
```

In [55]:
```
df.dtypes
```

Out[55]:
```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

In [61]:
```
df['petal_length']= df['petal_length'].astype("int")
df['petal_length']
```

Out[61]:
```
0      1
1      1
2    1
3    1
4    1      .. 145    5
146    5
147    5
148    5
149    5
Name: petal_length, Length: 150, dtype: int32
```

In [68]:
```
import pandas as pd
```

```
from sklearn import preprocessing
```

In [69]:
```
1
```

In [70]:
```
df.head()
```

Out[70]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1	0.2	setosa
1	4.9	3.0	1	0.2	setosa
4	5.0	3.6	1	0.2	setosa

```
1 min_max_scaler = preprocessing.MinMaxScaler()
2 print(min_max_scaler)
```

```
MinMaxScaler()
```

```
1 x=df.iloc[:,:4]
```

```
1 x_scaled = min_max_scaler.fit_transform(x)
```

```
1 df_normalized = pd.DataFrame(x_scaled)
```

```
1 df_normalized
```

	0	1	2	3
0	0.222222	0.625000	0.0	0.041667
2	4.7	3.2	1	0.2	setosa
3	4.6	3.1	1	0.2	setosa

In [72]:

In [75]:

In [76]:

In [77]:

In [78]:

Out[78]:

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

In

```
1
```

In [82]:
```
df['species'].unique()
```

Out[82]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

In [83]:
```
label_encoder = preprocessing.LabelEncoder()
```

In [84]:
```
df['species']= label_encoder.fit_transform(df['species'])
```

In [85]:
```
df['species'].unique()
```

Out[85]:
```
array([0, 1, 2])
```

In [86]:
```
features_df=df.drop(columns=['species'])
```

In [87]:
```
enc = preprocessing.OneHotEncoder()
```

In [93]:
```
enc_df=pd.DataFrame(enc.fit_transform(df[['species']]))
```

In [95]:
```
df_encode = features_df.join(enc_df)
```

[96]:
```
df_encode
```

Out[96]:

	sepal_length	sepal_width	petal_length	petal_width	0

150 rows × 5 columns

```
df_encode.rename(columns = {0:'Iris-Setosa',1:'Iris-Versicolor',2:'Iris-virginica'}, inplace =
```

```
df_encode
```

	sepal_length	sepal_width	petal_length	petal_width	Iris-Setosa
0	5.1	3.5	1	0.2	(0, 0)\t1.0
0	5.1	3.5	1	0.2	(0, 0)\t1.0
1	4.9	3.0	1	0.2	(0, 0)\t1.0
2	4.7	3.2	1	0.2	(0, 0)\t1.0
3	4.6	3.1	1	0.2	(0, 0)\t1.0
4	5.0	3.6	1	0.2	(0, 0)\t1.0
...	...	...	...	...	...
145	6.7	3.0	5	2.3	(0, 2)\t1.0
146	6.3	2.5	5	1.9	(0, 2)\t1.0
147	6.5	3.0	5	2.0	(0, 2)\t1.0
148	6.2	3.4	5	2.3	(0, 2)\t1.0
149	5.9	3.0	5	1.8	(0, 2)\t1.0

In [97]:

In [98]: Out[98]:

1	4.9	3.0	1	0.2	(0, 0)\t1.0

In

```
1
```

150 rows × 5 columns

```
1 one_hot_df = pd.get_dummies(df, prefix="species",columns=['species'], drop_first=True)
```

```
1 one_hot_df
```

	sepal_length	sepal_width	petal_length	petal_width	species_1	species_2
0	5.1	3.5	1	0.2	False	False
2	4.7	3.2	1	0.2	(0, 0)\t1.0	
3	4.6	3.1	1	0.2	(0, 0)\t1.0	
4	5.0	3.6	1	0.2	(0, 0)\t1.0	
...	...	...	...	...	...	
145	6.7	3.0	5	2.3	(0, 2)\t1.0	
146	6.3	2.5	5	1.9	(0, 2)\t1.0	
147	6.5	3.0	5	2.0	(0, 2)\t1.0	
148	6.2	3.4	5	2.3	(0, 2)\t1.0	
149	5.9	3.0	5	1.8	(0, 2)\t1.0	

In [100]:

In [101]:

Out[101]:

1	4.9	3.0	1	0.2	False	False
2	4.7	3.2	1	0.2	False	False
3	4.6	3.1	1	0.2	False	False
4	5.0	3.6	1	0.2	False	False
...	...	...	...	...	...	...
145	6.7	3.0	5	2.3	False	True
146	6.3	2.5	5	1.9	False	True
147	6.5	3.0	5	2.0	False	True
148	6.2	3.4	5	2.3	False	True
149	5.9	3.0	5	1.8	False	True

150 rows × 6 columns

In

```
class Solution:
   def solve(str, s):
      output = ""
      num=""
      for i in s:
         if i.isalpha():
            output+=i*int(num)
            num=""
         else:
            num+=i
      return output
print("Enter a string : ")
str = input()
ob = Solution()
print(ob.solve(str))
```

```
Enter a string :
4B3A
BBBBAAA
```

[17]:

In [28]:

```
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]


common = list(set(list1) & set(list2))

print(common)
```

```
[4, 5]
```

In [30]:

```
L1= ['Sohan', 'Mohan', 'Rohan']
for string in L1:
    print (string[0])
```

```
S
M
```

In

In R

```
a = ['pandas', 'numpy','flask', 'python', 'python']

s = set()

dup = []

for n in a:
    if n in s:
        dup.append(n)
    else:
        s.add(n)

print(dup)
```

[31]: ['python']

```
a = [1,2,5,3,4,8,9,"lis","a"]
length = len(a)
print(length)
```

9

In [4]:

Name : Avinash Kurandale

Roll no. : 13223 (TECO-b2)