

Lab cycle-2

1. Print the dataset Iris

```
import pandas as pd
df=pd.read_csv("/content/IRIS.csv")
print(df)
```

Output:

	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 x 5 columns]

2. Print the structure of the dataset iris

```
Print("Structure of dataset iris")
print(df.info())
```

Output:

```
structure of dataset iris
<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
None
```

3. Print the summary of all the variables of the dataset iris (**Hint:** Use function `summary()`)

```
print(df.describe())
```

Output:

	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

4. How many of the variables (columns) are in the dataset *iris*

```
Print("Number of variables in the dataset iris:")  
print(len(df.columns))
```

Output:

Number of variables in the dataset iris:
5

5. How many observations (rows) are in the dataset *iris*

```
Print("Number of observations in the dataset iris:",len(df))
```

Output:

Number of observations in the dataset iris:150

6. Use *duplicated()* function to print the logical vector indicating the duplicate values present in the dataset *iris*

```
print(df.duplicated())
```

Output:

```
0    False  
1    False  
2    False  
3    False  
4    False
```

...

```
145  False  
146  False  
147  False  
148  False  
149  False
```

Length: 150, dtype: bool

7. Extract duplicate elements from the dataset *iris*

```
print(df[df.duplicated()==True])
```

Output:

	sepal_length	sepal_width	petal_length	petal_width	species
34	4.9	3.1	1.5	0.1	Iris-setosa
37	4.9	3.1	1.5	0.1	Iris-setosa

142 5.8 2.7 5.1 1.9 Iris-virginica

8. Extract unique elements from the dataset *iris*

```
print(df[df.duplicated()==False])
```

Output:

	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

[147 rows x 5 columns]

9. Print the indices of duplicate elements in the dataset *iris*

```
print(df[df.duplicated()==True].index)
```

Output:

```
Int64Index([34, 37, 142], dtype='int64')
```

10. Print the indices of unique elements in the dataset *iris*

```
print(df[df.duplicated()==False].index)
```

Output:

```
Int64Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9,  
            ...  
            139, 140, 141, 143, 144, 145, 146, 147, 148, 149],  
           dtype='int64', length=147)
```

11. How many unique elements are in the dataset *iris*

```
Print("Number of unique elements:")  
print(len(df[df.duplicated()==False]))
```

Output:

```
Number of unique elements:  
147
```

12. How many duplicate elements are in the dataset *iris*

```
Print("number of duplicate elements:")  
print(len(df[df.duplicated()==True]))
```

Output:

Number of duplicate elements:

3

13. Print the sorted elements in the dataset iris(Ascending order based on index)

df.sort_values(by="sepal_length")

Output:

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

14. find whether any missing values are in the dataset iris

df.isnull()

Output:

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

sepal_length	sepal_width	petal_length	petal_width	species	
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 \times 5 columns

15. display how many missing values are present in each column
df.isnull().sum()

Output:

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

16. replace all missing values with zero
df.fillna(0)

Output:

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

sepal_length	sepal_width	petal_length	petal_width	species
149	5.9	3.0	5.1	1.8 Iris-virginica

150 rows \times 5 columns

17. Calculate Petal width mean ,median ,SD,Variance for the species setosa

```
mean=df[df['species']=="Iris-setosa"]['petal_width'].mean()
print("mean:",mean)
median=df[df['species']=="Iris-setosa"]['petal_width'].median()
print("median:",median)
standard_deviation=df[df['species']=="Iris-setosa"]['petal_width'].std()
print("standard deviation:",standard_deviation)
variance=df[df['species']=="Iris-setosa"]['petal_width'].var()
print("variance:",variance)
```

Output:

```
mean: 0.244
median: 0.2
standard deviation: 0.1072095030816784
variance: 0.011493877551020411
```

18. Print from 10th row to 20th row of iris dataset.

```
print(df.iloc[10:21,:])
```

Output:

sepal_length	sepal_width	petal_length	petal_width	species
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
15	5.7	4.4	1.5	0.4 Iris-setosa
16	5.4	3.9	1.3	0.4 Iris-setosa
17	5.1	3.5	1.4	0.3 Iris-setosa
18	5.7	3.8	1.7	0.3 Iris-setosa
19	5.1	3.8	1.5	0.3 Iris-setosa
20	5.4	3.4	1.7	0.2 Iris-setosa

19. Print Species and its corresponding Petal length and Width

```
df[['species','petal_width','petal_length']]
```

Output:

sepal_length	sepal_width	petal_length	petal_width	species
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
15	5.7	4.4	1.5	0.4	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa
17	5.1	3.5	1.4	0.3	Iris-setosa
18	5.7	3.8	1.7	0.3	Iris-setosa
19	5.1	3.8	1.5	0.3	Iris-setosa
20	5.4	3.4	1.7	0.2	Iris-setosa

20. display records only with species "Iris-setosa"
`df[df['species']=="Iris-setosa"]`

Output:

	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

21. count number of times a particular species has occurred.
`df['species'].value_counts()`

Output:

Iris-setosa 50 Iris-versicolor 50 Iris-virginica 50 Name: species, dtype: int64

22. Identifying minimum and maximum Value of Sepal width
`print(df['sepal_width'].min())`
`print(df['sepal_width'].max())`

Output:

2.0
4.4

23. Add new column to store sum of first column values
`l=list(df)`
`l.remove('species')`
`df['new_col']=df[l].sum(axis=1)`

Output:

	sepal_length	sepal_width	petal_length	petal_width	species	new_col
0		5.1	3.5	1.4	0.2	Iris-setosa 10.2
1		4.9	3.0	1.4	0.2	Iris-setosa 9.5
2		4.7	3.2	1.3	0.2	Iris-setosa 9.4
3		4.6	3.1	1.5	0.2	Iris-setosa 9.4

sepal_length	sepal_width	petal_length	petal_width	species	new_col	
4	5.0	3.6	1.4	0.2	Iris-setosa	10.2
...
145	6.7	3.0	5.2	2.3	Iris- virginica	17.2
146	6.3	2.5	5.0	1.9	Iris- virginica	15.7
147	6.5	3.0	5.2	2.0	Iris- virginica	16.7
148	6.2	3.4	5.4	2.3	Iris- virginica	17.3
149	5.9	3.0	5.1	1.8	Iris- virginica	15.8

150 rows × 6 columns