

Name : Tina Borundia

Batch : C1-16

Practical 6

```
In [ ]: import pandas as pd
df=pd.read_csv('Iris_dataset.csv')
print (df)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	\
0	1	5.1	3.5	1.4	0.2	
1	2	4.9	NaN	1.4	0.2	
2	3	4.7	3.2	1.3	0.2	
3	4	??	3.1	1.5	0.2	
4	5	5	3.6	###	0.2	
..	
145	146	6.7	3.0	5.2	2.3	
146	147	6.3	2.5	5	1.9	
147	148	6.5	3.0	5.2	2.0	
148	149	6.2	3.4	5.4	2.3	
149	150	5.9	3.0	5.1	1.8	

	Species
0	Iris-setosa
1	NaN
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
..	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica

[150 rows x 6 columns]

Task 1

```
In [ ]: df.head(4)
```

```
Out[3]:
```

	Unnamed: 0	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	NaN	1.4	0.2	NaN
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	??	3.1	1.5	0.2	Iris-setosa

Type *Markdown* and LaTeX: α^2

In []: `df.tail(4)`

Out[4]:

	Unnamed: 0	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
146	147	6.3	2.5	5	1.9	Iris-virgi
147	148	6.5	3.0	5.2	2.0	Iris-virgi
148	149	6.2	3.4	5.4	2.3	Iris-virgi
149	150	5.9	3.0	5.1	1.8	Iris-virgi

Task 2

In []: `import numpy as np`
`df.replace(['??', '###'], np.nan, inplace=True)`
`print(df)`

	Unnamed: 0	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	NaN	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	NaN	3.1	1.5	0.2
4	5	5	3.6	NaN	0.2
..
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

	Species
0	Iris-setosa
1	NaN
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa
..	...
145	Iris-virginica
146	Iris-virginica
147	Iris-virginica
148	Iris-virginica
149	Iris-virginica

[150 rows x 6 columns]

Task 3

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

```
Out[6]:
```

	Unnamed: 0	SepalWidthCm	PetalWidthCm
count	150.000000	149.000000	150.000000
mean	75.500000	3.054362	1.198667
std	43.445368	0.435034	0.763161
min	1.000000	2.000000	0.100000
25%	38.250000	2.800000	0.300000
50%	75.500000	3.000000	1.300000
75%	112.750000	3.300000	1.800000
max	150.000000	4.400000	2.500000

```
In [ ]: num_observations = df.shape[0]  
print(num_observations)
```

```
150
```

```
In [ ]: num_missing_values = df.isna().sum().sum()  
print(num_missing_values)
```

```
5
```

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

```
Out[9]:
```

	Unnamed: 0	SepalWidthCm	PetalWidthCm
count	150.000000	149.000000	150.000000
mean	75.500000	3.054362	1.198667
std	43.445368	0.435034	0.763161
min	1.000000	2.000000	0.100000
25%	38.250000	2.800000	0.300000
50%	75.500000	3.000000	1.300000
75%	112.750000	3.300000	1.800000
max	150.000000	4.400000	2.500000

Task 4

```
In [ ]: from sklearn.datasets import load_iris
iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
stats = df.describe()
print(stats)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	\
count	150.000000	150.000000	150.000000	
mean	5.843333	3.057333	3.758000	
std	0.828066	0.435866	1.765298	
min	4.300000	2.000000	1.000000	
25%	5.100000	2.800000	1.600000	
50%	5.800000	3.000000	4.350000	
75%	6.400000	3.300000	5.100000	
max	7.900000	4.400000	6.900000	

	petal width (cm)
count	150.000000
mean	1.199333
std	0.762238
min	0.100000
25%	0.300000
50%	1.300000
75%	1.800000
max	2.500000

Task 5

```
In [ ]: iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['species'] = iris.target_names[iris.target]
stats = df.groupby('species').describe()
print(stats)
```

	sepal length (cm)							
	count	mean	std	min	25%	50%	75%	max
species								
setosa	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8
versicolor	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0
virginica	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9

	sepal width (cm)		petal length (cm)				\	
	count	mean	...	75%	max			
species			...					
setosa	50.0	3.428	...	1.575	1.9			
versicolor	50.0	2.770	...	4.600	5.1			
virginica	50.0	2.974	...	5.875	6.9			

	petal width (cm)							
	count	mean	std	min	25%	50%	75%	max
species								
setosa	50.0	0.246	0.105386	0.1	0.2	0.2	0.3	0.6
versicolor	50.0	1.326	0.197753	1.0	1.2	1.3	1.5	1.8
virginica	50.0	2.026	0.274650	1.4	1.8	2.0	2.3	2.5

[3 rows x 32 columns]


Task 6

```
In [ ]: new = df.drop('Id', axis=1)
print(new.head())
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	NaN	1.4	0.2	NaN
2	4.7	3.2	1.3	0.2	Iris-setosa
3	??	3.1	1.5	0.2	Iris-setosa
4	5	3.6	###	0.2	Iris-setosa

Task 7

```
In [ ]: iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['species'] = iris.target_names[iris.target]
max_row = df.loc[df['sepal length (cm)'].idxmax()]
print(max_row)
print(f"The species with maximum Sepal Length is {max_row['species']} and its row number is {max_row.name}")
```




sepal length (cm)	7.9
sepal width (cm)	3.8
petal length (cm)	6.4
petal width (cm)	2.0
species	virginica

Name: 131, dtype: object
The species with maximum Sepal Length is virginica and its row number is 131

Task 8

```
In [ ]: iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['species'] = iris.target_names[iris.target]
min_row = df.loc[df['petal length (cm)'].idxmin()]
print(min_row)
print(f"The species with minimum Petal Length is {min_row['species']} and its row number is {min_row.name}")
```



sepal length (cm)	4.6
sepal width (cm)	3.6
petal length (cm)	1.0
petal width (cm)	0.2
species	setosa

Name: 22, dtype: object
The species with minimum Petal Length is setosa and its row number is 23

```
In [ ]: grouped_data = df.groupby('Species')
```

```
In [ ]: print(grouped_data.describe())
```

```

                Unnamed: 0
                count      mean      std    min    25%    50%
Species
Iris-setosa      49.0    25.979592  14.324353    1.0   14.00   26.0
Iris-versicolor  50.0    75.500000  14.577380   51.0   63.25   75.5
Iris-virginica   50.0   125.500000  14.577380  101.0  113.25  125.5

```

```

                SepalWidthCm
                75%    max      count      mean    ...    75%    max
Species
Iris-setosa      38.00   50.0         49.0  3.426531  ...   3.700   4.4
Iris-versicolor  87.75  100.0         50.0  2.770000  ...   3.000   3.4
Iris-virginica  137.75  150.0         50.0  2.974000  ...   3.175   3.8

```

```

                PetalWidthCm
                count      mean      std    min  25%  50%  75%  ma
Species
Iris-setosa      49.0  0.244898  0.108130  0.1  0.2  0.2  0.3  0.
Iris-versicolor  50.0  1.326000  0.197753  1.0  1.2  1.3  1.5  1.
Iris-virginica   50.0  2.026000  0.274650  1.4  1.8  2.0  2.3  2.

```

```
[3 rows x 24 columns]
```

Task 9

```
In [ ]: df = df.sort_values(by=["SepalLengthCm"], ascending=True)
print(df)
```

	Unnamed: 0	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
13	14	4.3	3.0	1.1	0.1
42	43	4.4	3.2	1.3	0.2
38	39	4.4	3.0	1.3	0.2
8	9	4.4	2.9	1.4	0.2
41	42	4.5	2.3	1.3	0.3
..
117	118	7.7	3.8	6.7	2.2
122	123	7.7	2.8	6.7	2.0
135	136	7.7	3.0	6.1	2.3
131	132	7.9	3.8	6.4	2.0
3	4	NaN	3.1	1.5	0.2
Species					
13	Iris-setosa				
42	Iris-setosa				
38	Iris-setosa				
8	Iris-setosa				
41	Iris-setosa				
..	...				
117	Iris-virginica				
122	Iris-virginica				
135	Iris-virginica				
131	Iris-virginica				
3	Iris-setosa				

[150 rows x 6 columns]

Task 10

```
In [ ]: df["PetalWidthCm"] = df["PetalWidthCm"].fillna(df["PetalWidthCm"].mean())

df["SepalLengthCm"] = df["SepalLengthCm"].fillna(df["SepalLengthCm"].mod
print(df)
```

	Unnamed: 0	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
13	14	4.3	3.0	1.1	0.1
42	43	4.4	3.2	1.3	0.2
38	39	4.4	3.0	1.3	0.2
8	9	4.4	2.9	1.4	0.2
41	42	4.5	2.3	1.3	0.3
..
117	118	7.7	3.8	6.7	2.2
122	123	7.7	2.8	6.7	2.0
135	136	7.7	3.0	6.1	2.3
131	132	7.9	3.8	6.4	2.0
3	4	5	3.1	1.5	0.2

	Species
13	Iris-setosa
42	Iris-setosa
38	Iris-setosa
8	Iris-setosa
41	Iris-setosa
..	...
117	Iris-virginica
122	Iris-virginica
135	Iris-virginica
131	Iris-virginica
3	Iris-setosa

[150 rows x 6 columns]

Task 11

In []: `print(df.corr())`

	sepal length (cm)	sepal width (cm)	petal length (cm)
sepal length (cm)	1.000000	-0.117570	0.87175
sepal width (cm)	-0.117570	1.000000	-0.42844
petal length (cm)	0.871754	-0.428440	1.00000
petal width (cm)	0.817941	-0.366126	0.96286

	petal width (cm)
sepal length (cm)	0.817941
sepal width (cm)	-0.366126
petal length (cm)	0.962865
petal width (cm)	1.000000

<ipython-input-37-23236a4e6045>:1: FutureWarning: The default value of n_c_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
print(df.corr())
```

Task 12

In []:

```
from scipy.stats import pearsonr
iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
c, cd = pearsonr(df['petal width (cm)'], df['petal length (cm)'])

print("The Pearson correlation coefficient between Petal width and Petal
```

The Pearson correlation coefficient between Petal width and Petal Length
0.962865431402796

In []: