```python
import pandas as pd
import seaborn as sns
import statistics as stats
df = pd.read_csv("/content/Iris Dataset.csv")
df = df.drop(['Id'],axis = 1)
```

```python
df.info()
```

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

```python
m,n = df.shape
m,n
```

➡ (150, 5)

```python
df.describe()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
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
	0000		4.350000	1.300000
	0000		5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Saved successfully! ✕

```python
SepalLengthCm_mean = round(df['SepalLengthCm'].mean(),1)
print('Mean for SepalLengthCm:',SepalLengthCm_mean)

SepalWidthCm_mean = round(df['SepalWidthCm'].mean(),1)
print('Mean for SepalWidthCm:',SepalWidthCm_mean)

PetalLengthCm_mean = round(df['PetalLengthCm'].mean(),1)
print('Mean for PetalLengthCm:',PetalLengthCm_mean)

PetalWidthCm_mean = round(df['PetalWidthCm'].mean(),1)
print('Mean for PetalWidthCm:',PetalWidthCm_mean)
```

```
Mean for SepalLengthCm: 5.8
Mean for SepalWidthCm: 3.1
Mean for PetalLengthCm: 3.8
Mean for PetalWidthCm: 1.2
```

```python
print('Mode for SepalLengthCm:',df['SepalLengthCm'].mode())
print('Mode for SepalWidthCm:',df['SepalWidthCm'].mode())
print('Mode for PetalLengthCm:',df['PetalLengthCm'].mode())
print('Mode for PetalWidthCm:',df['PetalWidthCm'].mode())
```

```
Mode for SepalLengthCm: 0    5.0
Name: SepalLengthCm, dtype: float64
Mode for SepalWidthCm: 0    3.0
Name: SepalWidthCm, dtype: float64
Mode for PetalLengthCm: 0    1.5
Name: PetalLengthCm, dtype: float64
Mode for PetalWidthCm: 0    0.2
Name: PetalWidthCm, dtype: float64
```

```python
print('Median for SepalLengthCm:',df['SepalLengthCm'].median())
print('Median for SepalWidthCm:',df['SepalWidthCm'].median())
print('Median for PetalLengthCm:',df['PetalLengthCm'].median())
print('Median for PetalWidthCm:',df['PetalWidthCm'].median())
```

```
Median for SepalLengthCm: 5.8
Median for SepalWidthCm: 3.0
```

```
        Median for PetalLengthCm: 4.35
        Median for PetalWidthCm: 1.3
```

```
var1 = stats.variance(df['SepalLengthCm'])
print('Variance and Squared devaition for SepalLengthCm:',round(var1,2),round(var1**0.5,2))

var2 = stats.variance(df['SepalWidthCm'])
print('Variance and Squared devaition for SepalWidthCm:',round(var2,2),round((var2**0.5),2))

var3 = stats.variance(df['SepalWidthCm'])
print('Variance and Squared devaition for PetalWidthCm:',round(var3,2),round((var3**0.5),2))

var4 = stats.variance(df['SepalWidthCm'])
print('Variance and Squared devaition for PetalLengthCm:',round(var4,2),round((var4**0.5),2))
```

```
        Variance and Squared devaition for SepalLengthCm: 0.69 0.83
        Variance and Squared devaition for SepalWidthCm: 0.19 0.43
        Variance and Squared devaition for PetalWidthCm: 0.19 0.43
        Variance and Squared devaition for PetalLengthCm: 0.19 0.43
```

```
missing_values = ['?','-3.4','-2.19','-1']
fault_df = pd.read_csv("/content/Iris Dataset.csv",na_values = missing_values )
fault_df.head(10)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
			3.1	1.5	0.1	Iris-setosa

Saved successfully! ✕

```
#Let's check if any missing values?
print(fault_df.isnull().values.any())
```

```
        False
```

```
# Total number of missing values
print(fault_df.isnull().sum().sum())
```

```
        0
```

```
# Total missing values for each feature
print(fault_df.isnull().sum())
```

```
        Id               0
        SepalLengthCm    0
        SepalWidthCm     0
        PetalLengthCm    0
        PetalWidthCm     0
        Species          0
        dtype: int64
```

```
#Mean for SepalLengthCm: 5.8
#Mean for SepalWidthCm: 3.1
#Mean for PetalLengthCm: 3.8
#Mean for PetalWidthCm: 1.2
```

```
#Replacing missing values with Mean values of each attribute...
fault_df['SepalLengthCm'].fillna(SepalLengthCm_mean,inplace = True)
fault_df['SepalWidthCm'].fillna(SepalWidthCm_mean,inplace = True)
fault_df['PetalLengthCm'].fillna(PetalLengthCm_mean,inplace = True)
```

```
fault_df.head(10)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa

```python
#If you want to round whole column...
fault_df['SepalLengthCm'] = fault_df['SepalLengthCm'].round(1)
fault_df.head(5)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```python
fault_df = fault_df.round({'SepalWidthCm': 1})
```

```python
fault_df.head(10)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
			3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

Saved successfully! ✕

```python
fault_df.to_csv('solved_fault_df',index = False)
```

```python
#unique values for SepalLengthCm
uqvalprob_SepalLengthCm = []
distinct_SepalLengthCm = df['SepalLengthCm'].unique()
print(f'Total distinct element in SepalLengthCm are {len(distinct_SepalLengthCm)} there max prob is:')
for elem in distinct_SepalLengthCm:
    c = (df['SepalLengthCm']==elem).sum()
    uqvalprob_SepalLengthCm.append(round((c/m),2))
    #print(f'P({elem}) = {round((c/m),2)}')
maxuqvalprob_SepalLengthCm = max(uqvalprob_SepalLengthCm)
print(maxuqvalprob_SepalLengthCm)
```

```
    Total distinct element in SepalLengthCm are 35 there max prob is:
    0.07
```

```python
#unique values for SepalWidthCm
uqvalprob_SepalWidthCm = []
distinct_SepalWidthCm = df['SepalWidthCm'].unique()
print(f'Total distinct element in SepalWidthCm are {len(distinct_SepalWidthCm)} there max prob is:')
for elem in distinct_SepalWidthCm:
    c = (df['SepalWidthCm']==elem).sum()
    uqvalprob_SepalWidthCm.append(round((c/m),2))
    #print(f'P({elem}) = {round((c/m),2)}')
maxuqvalprob_SepalWidthCm = max(uqvalprob_SepalWidthCm)
print(maxuqvalprob_SepalWidthCm)
```

```
    Total distinct element in SepalWidthCm are 23 there max prob is:
    0.17
```

```python
#unique values for PetalWidthCm
uqvalprob_PetalWidthCm = []
distinct_PetalWidthCm = df['PetalWidthCm'].unique()
print(f'Total distinct element in PetalWidthCm are {len(distinct_PetalWidthCm)} there max prob is:')
for elem in distinct_PetalWidthCm:
    c = (df['PetalWidthCm']==elem).sum()
    #print(f'P({elem}) = {round((c/m),2)}')
    uqvalprob_PetalWidthCm.append(round((c/m),2))
maxuqvalprob_PetalLengthCm = max(uqvalprob_PetalWidthCm)
print(maxuqvalprob_PetalLengthCm)
```

```
    Total distinct element in PetalWidthCm are 22 there max prob is:
    0.19
```

```python
#unique values for PetalLengthCm
uqvalprob_PetalLengthCm = []
distinct_PetalLengthCm = df['PetalLengthCm'].unique()
print(f'Total distinct element in PetalLengthCm are {len(distinct_PetalLengthCm)} there max prob is:')
for elem in distinct_PetalLengthCm:
    c = (df['PetalLengthCm']==elem).sum()
    uqvalprob_PetalWidthCm.append(round((c/m),2))
    #print(f'P({elem}) = {round((c/m),2)}')
maxuqvalprob_PetalLengthCm = max(uqvalprob_PetalWidthCm)
print(maxuqvalprob_PetalLengthCm)
```

```
    Total distinct element in PetalLengthCm are 43 there max prob is:
    0.19
```

```python
missing_values = ['?','-3.4','-2.19','-1']
fault_df_m2 = pd.read_csv("/content/Iris Dataset.csv",na_values = missing_values)
fault_df_m2
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
			3.1	1.5	0.2	Iris-setosa
			3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Saved successfully!

```python
#Replacing missing values with max value of distinct values probs...
fault_df_m2['SepalLengthCm'].fillna(maxuqvalprob_SepalLengthCm,inplace = True)
fault_df_m2['SepalWidthCm'].fillna(maxuqvalprob_SepalWidthCm,inplace = True)
fault_df_m2['PetalLengthCm'].fillna(maxuqvalprob_PetalLengthCm,inplace = True)
```

```python
fault_df_m2
```

```
fault_df_m2.to_csv('solved_fault_df_m2',index = False)
```

```
missing_values = ['?','-3.4','-2.19','-1']
fault_df_m3 = pd.read_csv("/content/Iris Dataset.csv",na_values = missing_values)
fault_df_m3
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

```
fault_df_m3['Species'].unique()
```

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

```
fault_df_m3['Species'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
```

```
sum_SepalLengthCm = (df['SepalLengthCm']).sum()
s1_SepalLengthCm = (df['SepalLengthCm'][:50]).sum()
s2_SepalLengthCm = (df['SepalLengthCm'][50:100]).sum()
s3_SepalLengthCm = (df['SepalLengthCm'][100:]).sum()

sum_SepalWidthCm = (df['SepalWidthCm']).sum()
s1_SepalWidthCm = (df['SepalWidthCm'][:50]).sum()
s2_SepalWidthCm = (df['SepalWidthCm'][50:100]).sum()
s3_SepalWidthCm = (df['SepalWidthCm'][100:]).sum()

sum_PetalLengthCm = (df['PetalLengthCm']).sum()
s1_PetalLengthCm = (df['PetalLengthCm'][:50]).sum()
s2_PetalLengthCm = (df['PetalLengthCm'][50:100]).sum()
s3_PetalLengthCm = (df['PetalLengthCm'][100:]).sum()


#Replacing missing values with mean of Iris-setosa, Iris-versicolor, and Iris-virginica
fault_df_m3['SepalLengthCm'].fillna(s1_SepalLengthCm/50,inplace = True)
fault_df_m3['SepalWidthCm'].fillna(s2_SepalWidthCm/50,inplace = True)
fault_df_m3['PetalLengthCm'].fillna(s3_PetalLengthCm/50,inplace = True)
```

```
fault_df_m3
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

✓ 0s   completed at 9:39 AM

Saved successfully!