

✓ Lab 2 (sample.csv) questions solutions

```
import pandas as pd
#from google.colab import files
#uploaded = files.upload()
df = pd.read_csv('sample.csv')
print(df)
```

```
↗
```

	id	first	last	gender	Marks	selected
0	1	Leone	Debrick	Female	50	True
1	2	Romola	Phinnessy	Female	60	False
2	3	Geri	Prium	Male	65	False
3	4	Sandy	Doveston	Female	95	False
4	5	Jacenta	Jansik	Female	31	True
5	6	Diane-marie	Medhurst	Female	45	True
6	7	Austen	Pool	Male	45	True
7	8	Vanya	Teffrey	Male	70	False
8	9	Giordano	Elloy	Male	36	False
9	10	Rozele	Fawcett	Female	50	False

```
#Statistical Operations
print("Mean Marks:", df['Marks'].mean())
print("Median Marks:", df['Marks'].median())
print("Mode Marks:", df['Marks'].mode()[0])
print("Variance Marks:", df['Marks'].var())
print("Standard Deviation Marks:", df['Marks'].std())
print("Minimum Marks:", df['Marks'].min())
print("Maximum Marks:", df['Marks'].max())
print("Count of Marks:", df['Marks'].count())
```

```
↗
```

Mean Marks: 54.7
Median Marks: 50.0
Mode Marks: 45
Variance Marks: 348.4555555555555
Standard Deviation Marks: 18.666964283341724
Minimum Marks: 31
Maximum Marks: 95
Count of Marks: 10

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

```
↗
```

	id	Marks
count	10.00000	10.000000
mean	5.50000	54.700000
std	3.02765	18.666964
min	1.00000	31.000000
25%	3.25000	45.000000
50%	5.50000	50.000000
75%	7.75000	63.750000
max	10.00000	95.000000

```
#Univariate Data - Analyze one variable at a time
print(df['Marks'].describe())
```

```
↗
```

count	10.000000
mean	54.700000
std	18.666964
min	31.000000
25%	45.000000
50%	50.000000
75%	63.750000
max	95.000000

Name: Marks, dtype: float64

```
#Bivariate Data - Analyze two variables at a time
print(df.groupby('gender')['Marks'].mean())
```

```
↗
```

gender	Marks
Female	55.166667
Male	54.000000

Name: Marks, dtype: float64

```
#Performing Label Encoding Operation on Gender attribute
from sklearn.preprocessing import LabelEncoder
print("Before Label Encoding\n",df['gender'].unique())
```

```
label_enc = LabelEncoder()
df['gender']=label_enc.fit_transform(df['gender'])
print("After Label Encoding\n",df['gender'].unique())
```

```
Before Label Encoding
['Female' 'Male']
After Label Encoding
[0 1]
```

```
#Performing Label Encoding Operation of Selected attribute
print("Before Label Encoding\n",df['selected'].unique())
df['selected']=label_enc.fit_transform(df['selected'])
print("After Label Encoding\n",df['selected'].unique())
```

```
Before Label Encoding
[ True False]
After Label Encoding
[1 0]
```

```
from sklearn.preprocessing import StandardScaler
std_scaler=StandardScaler()
df['Marks_std_scaled'] = std_scaler.fit_transform(df[['Marks']])
print(df[['Marks', 'Marks_std_scaled']])
```

```
Marks Marks_std_scaled
0      50      -0.265401
1      60       0.299282
2      65       0.581624
3      95       2.275674
4      31      -1.338300
5      45      -0.547743
6      45      -0.547743
7      70       0.863966
8      36      -1.055958
9      50      -0.265401
```

```
#Still negative values after scaling so we go for the MinMax Scaling
from sklearn.preprocessing import MinMaxScaler
std_scaler=MinMaxScaler()
df['Marks_mm_scaled'] = std_scaler.fit_transform(df[['Marks']])
print(df[['Marks', 'Marks_mm_scaled']])
```

```
Marks Marks_mm_scaled
0      50      0.296875
1      60      0.453125
2      65      0.531250
3      95      1.000000
4      31      0.000000
5      45      0.218750
6      45      0.218750
7      70      0.609375
8      36      0.078125
9      50      0.296875
```

```
print(df[['Marks', 'Marks_std_scaled', 'Marks_mm_scaled']])
```

```
Marks Marks_std_scaled Marks_mm_scaled
0      50      -0.265401      0.296875
1      60       0.299282      0.453125
2      65       0.581624      0.531250
3      95       2.275674      1.000000
4      31      -1.338300      0.000000
5      45      -0.547743      0.218750
6      45      -0.547743      0.218750
7      70       0.863966      0.609375
8      36      -1.055958      0.078125
9      50      -0.265401      0.296875
```

```
from sklearn.preprocessing import Binarizer
binarizer = Binarizer(threshold=40)
df['Marks_binary'] = binarizer.fit_transform(df[['Marks']])
print(df[['Marks', 'Marks_binary']])
```

```
Marks Marks_binary
0      50           1
1      60           1
2      65           1
3      95           1
```

4	31	0
5	45	1
6	45	1
7	70	1
8	36	0
9	50	1