Consider the “Academic performance” dataset of students (Academic_Performance_Dataset.csv) and perform the following operations using Python. a) Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them. b) Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them. c) Apply data transformations on categorical variables to convert it into numerical variables. Reason and document your approach properly.

```
import pandas as pd

df= pd.read_csv("/content/Academic_Performance.csv")

df
```

YPE	ACADEMIC_PROGRAM	COURSE 1 MARKS	COURSE 2 MARKS	COURSE 3 MARKS	COURSE 4 MARKS	COURSE 5 MARKS	PERCENTILE	OVEARLL_GRAD
MIC	INDUSTRIAL ENGINEERING	71.0	93.0	71.0	93.0	79.0	91	FIRST CLAS
MIC	INDUSTRIAL ENGINEERING	97.0	38.0	86.0	98.0	78.0	92	THIRD CLAS
MIC	ELECTRONIC ENGINEERING	17.0	1.0	18.0	43.0	22.0	7	DISTINCTIO
MIC	INDUSTRIAL ENGINEERING	65.0	35.0	76.0	80.0	48.0	67	FIRST CLAS
MIC	INDUSTRIAL ENGINEERING	94.0	94.0	98.0	100.0	71.0	98	FIRST CLAS
...	...	...	...	...	...	...	...	.
MIC	MECHATRONICS ENGINEERING	88.0	71.0	86.0	87.0	65.0	88	FIRST CLAS
MIC	INDUSTRIAL ENGINEERING	46.0	39.0	44.0	11.0	0.0	4	FIRST CLAS
MIC	INDUSTRIAL ENGINEERING	98.0	88.0	90.0	81.0	87.0	95	FIRST CLAS
MIC	NaN	60.0	80.0	51.0	8.0	42.0	50	FIRST CLAS
MIC	INDUSTRIAL ENGINEERING	83.0	95.0	91.0	79.0	47.0	89	THIRD CLAS

```
df.dtypes
```

```
STUDENT_ID             object
GENDER                 object
PLACEMENT              object
HONOR_OPTED_OR_NOT     object
EDUCATION_TYPE         object
ACADEMIC_PROGRAM       object
COURSE 1 MARKS        float64
COURSE 2 MARKS        float64
COURSE 3 MARKS        float64
COURSE 4 MARKS        float64
COURSE 5 MARKS        float64
PERCENTILE              int64
OVEARLL_GRADE          object
dtype: object
```

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

```
STUDENT_ID             0
GENDER                22
PLACEMENT             15
HONOR_OPTED_OR_NOT    14
EDUCATION_TYPE        15
ACADEMIC_PROGRAM      34
COURSE 1 MARKS        11
COURSE 2 MARKS         8
COURSE 3 MARKS        14
COURSE 4 MARKS        14
COURSE 5 MARKS        22
PERCENTILE             0
OVEARLL_GRADE          0
dtype: int64
```

```
cols = ['COURSE 1 MARKS', 'COURSE 2 MARKS', 'COURSE 3 MARKS', 'COURSE 4 MARKS', 'COURSE 5
df[cols] = df[cols].fillna(df[cols].mean())
```

```
df
```

EDUCATION_TYPE	ACADEMIC_PROGRAM	COURSE 1 MARKS	COURSE 2 MARKS	COURSE 3 MARKS	COURSE 4 MARKS	COURSE 5 MARKS	PERCENTILE	C
ACADEMIC	INDUSTRIAL ENGINEERING	71.0	93.0	71.0	93.0	79.0	91	
ACADEMIC	INDUSTRIAL ENGINEERING	97.0	38.0	86.0	98.0	78.0	92	
ACADEMIC	ELECTRONIC ENGINEERING	17.0	1.0	18.0	43.0	22.0	7	
ACADEMIC	INDUSTRIAL ENGINEERING	65.0	35.0	76.0	80.0	48.0	67	
ACADEMIC	INDUSTRIAL ENGINEERING	94.0	94.0	98.0	100.0	71.0	98	
...	...	...	...	...	...	...	...	
ACADEMIC	MECHATRONICS ENGINEERING	88.0	71.0	86.0	87.0	65.0	88	
ACADEMIC	INDUSTRIAL ENGINEERING	46.0	39.0	44.0	11.0	0.0	4	
ACADEMIC	INDUSTRIAL ENGINEERING	98.0	88.0	90.0	81.0	87.0	95	
ACADEMIC	NaN	60.0	80.0	51.0	8.0	42.0	50	
ACADEMIC	INDUSTRIAL ENGINEERING	83.0	95.0	91.0	79.0	47.0	89	

Next steps: Generate code with df View recommended plots

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

```
STUDENT_ID             0
GENDER                22
PLACEMENT             15
HONOR_OPTED_OR_NOT    14
EDUCATION_TYPE        15
ACADEMIC_PROGRAM      34
COURSE 1 MARKS         0
COURSE 2 MARKS         0
COURSE 3 MARKS         0
COURSE 4 MARKS         0
COURSE 5 MARKS         0
PERCENTILE             0
OVEARLL_GRADE          0
dtype: int64
```

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy = 'most_frequent')
data_new = pd.DataFrame(imputer.fit_transform(df),columns=df.columns,index=df.index)
data_new = data_new.astype(df.dtypes)
df=data_new
```

```
df
```

	STUDENT_ID	GENDER	PLACEMENT	HONOR_OPTED_OR_NOT	EDUCATION_TYPE	ACADI
0	SB11201210000129	F	Yes	Yes	ACADEMIC	E
1	SB11201210000137	F	Yes	Yes	ACADEMIC	E
2	SB11201210005154	M	No	Yes	ACADEMIC	E
3	SB11201210007504	F	Yes	Yes	ACADEMIC	E
4	SB11201210007548	M	Yes	Yes	ACADEMIC	E
...	...	...	...	...	...	
12406	SB11201420568705	M	Yes	Yes	ACADEMIC	ME E
12407	SB11201420573045	M	Yes	Yes	ACADEMIC	E
12408	SB11201420578809	M	Yes	No	ACADEMIC	E
12409	SB11201420578812	F	Yes	Yes	ACADEMIC	E
12410	SB11201420583232	M	No	No	ACADEMIC	E

12411 rows × 13 columns

Next steps: Generate code with `df` View recommended plots

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

```
STUDENT_ID            0
GENDER                0
PLACEMENT             0
HONOR_OPTED_OR_NOT    0
EDUCATION_TYPE        0
ACADEMIC_PROGRAM      0
COURSE 1 MARKS        0
```

```
COURSE 2 MARKS        0
COURSE 3 MARKS        0
COURSE 4 MARKS        0
COURSE 5 MARKS        0
PERCENTILE            0
OVEARLL_GRADE         0
dtype: int64
```

```
df.dtypes
```

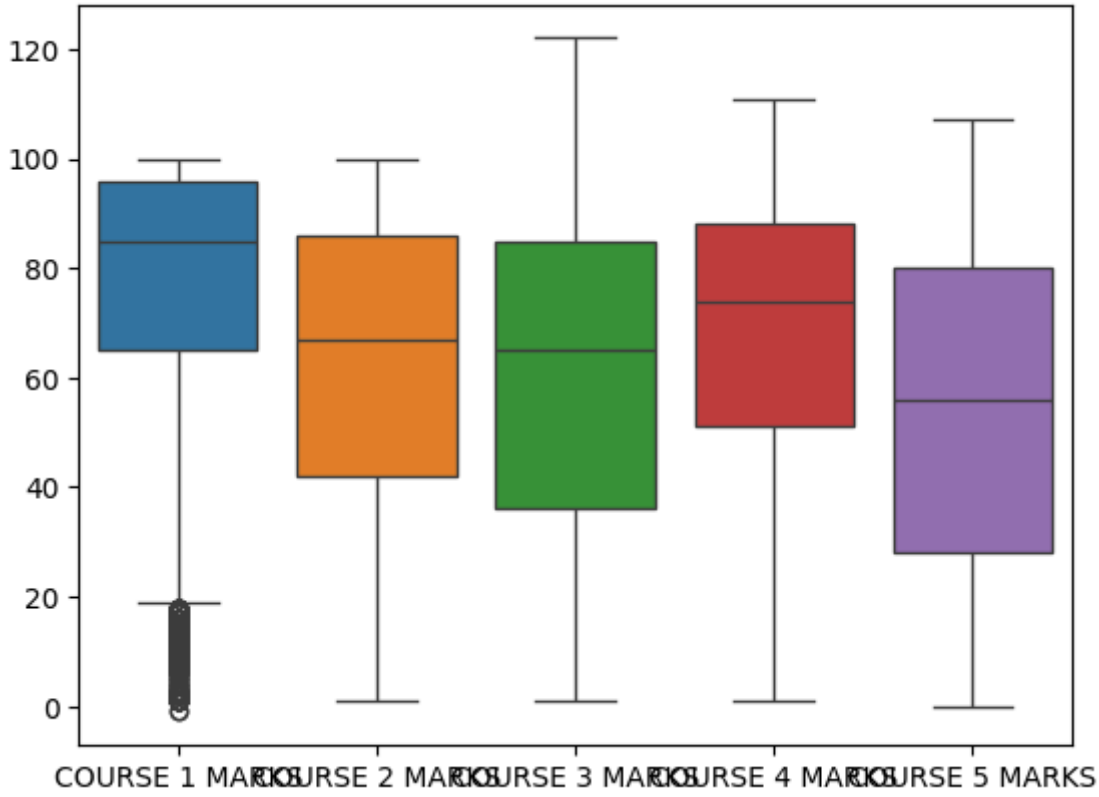
```
STUDENT_ID             object
GENDER                 object
PLACEMENT              object
HONOR_OPTED_OR_NOT     object
EDUCATION_TYPE         object
ACADEMIC_PROGRAM       object
COURSE 1 MARKS        float64
COURSE 2 MARKS        float64
COURSE 3 MARKS        float64
COURSE 4 MARKS        float64
COURSE 5 MARKS        float64
PERCENTILE              int64
OVEARLL_GRADE          object
dtype: object
```

OUTLIERS

```
import seaborn as sns
import numpy as np
```

```
cols = ['COURSE 1 MARKS', 'COURSE 2 MARKS', 'COURSE 3 MARKS', 'COURSE 4 MARKS', 'COURSE 5
sns.boxplot(df[cols])
```

```
<Axes: >
```




```
def outliers(data_item):
    outliers=[]
    data_item=sorted(data_item)
    q1 = np.percentile(data_item,25)
    q3 = np.percentile(data_item,75)
    iqr = q3-q1
    lower_bound = q1-(1.5*iqr)
    upper_bound = q3+(1.5*iqr)
    print(lower_bound,upper_bound)
    for i in data_item:
        if(i<lower_bound or i>upper_bound):
            outliers.append(i)
    print(outliers)
    return lower_bound,upper_bound


lower,upper = outliers(df['COURSE 1 MARKS'])
```

```
 23.5 139.5
[19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0, 19.0,
```

```
df = df[df['COURSE 1 MARKS']>lower]
df = df[df['COURSE 1 MARKS']<upper]


df
```

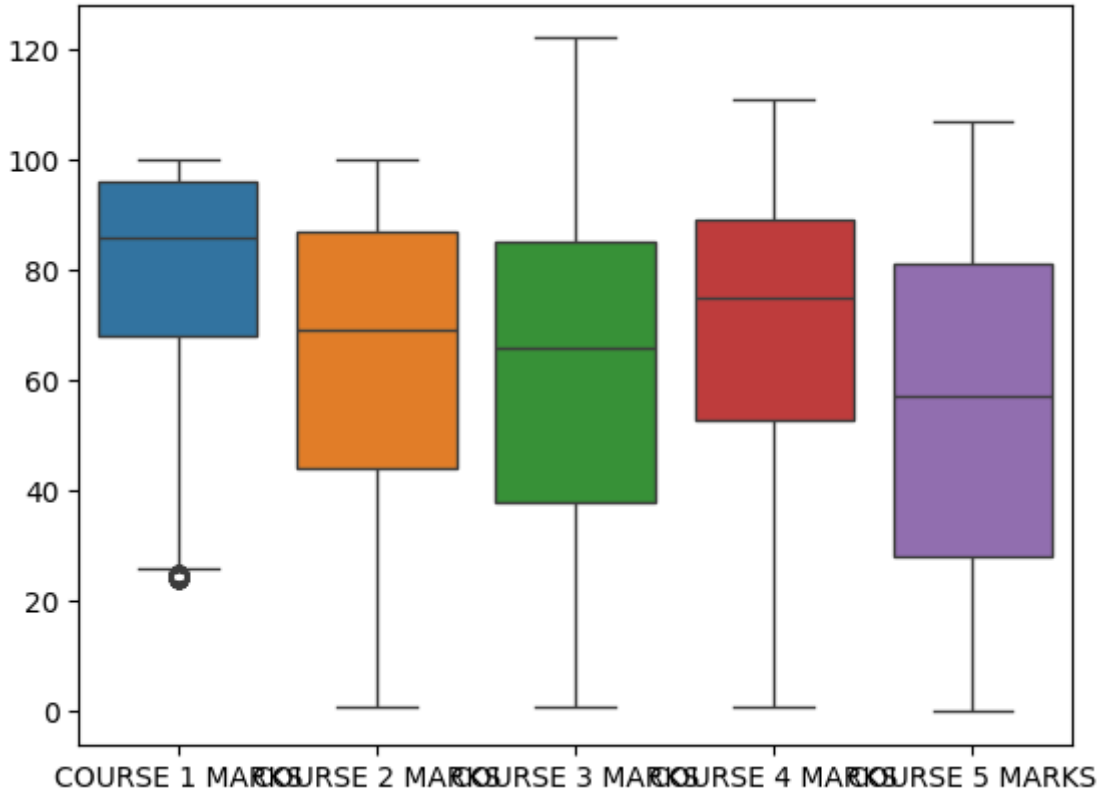
	STUDENT_ID	GENDER	PLACEMENT	HONOR_OPTED_OR_NOT	EDUCATION_TYPE	ACADI
0	SB11201210000129	F	Yes	Yes	ACADEMIC	E
1	SB11201210000137	F	Yes	Yes	ACADEMIC	E
3	SB11201210007504	F	Yes	Yes	ACADEMIC	E
4	SB11201210007548	M	Yes	Yes	ACADEMIC	E
5	SB11201210007568	F	Yes	Yes	ACADEMIC	E
...	...	...	...	...	...	
12406	SB11201420568705	M	Yes	Yes	ACADEMIC	ME E
12407	SB11201420573045	M	Yes	Yes	ACADEMIC	E
12408	SB11201420578809	M	Yes	No	ACADEMIC	E
12409	SB11201420578812	F	Yes	Yes	ACADEMIC	E
12410	SB11201420583232	M	No	No	ACADEMIC	E

11968 rows × 13 columns

Next steps: Generate code with `df` View recommended plots

```
cols = ['COURSE 1 MARKS', 'COURSE 2 MARKS', 'COURSE 3 MARKS', 'COURSE 4 MARKS', 'COURSE 5
sns.boxplot(df[cols])
```

<Axes: >




```
performance= df.select_dtypes(exclude=[np.number])
```

```
performance
```

	STUDENT_ID	GENDER	PLACEMENT	HONOR_OPTED_OR_NOT	EDUCATION_TYPE	ACADI
0	SB11201210000129	F	Yes	Yes	ACADEMIC	E
1	SB11201210000137	F	Yes	Yes	ACADEMIC	E
3	SB11201210007504	F	Yes	Yes	ACADEMIC	E
4	SB11201210007548	M	Yes	Yes	ACADEMIC	E
5	SB11201210007568	F	Yes	Yes	ACADEMIC	E
...	...	...	...	...	...	
12406	SB11201420568705	M	Yes	Yes	ACADEMIC	ME E
12407	SB11201420573045	M	Yes	Yes	ACADEMIC	E
12408	SB11201420578809	M	Yes	No	ACADEMIC	E
12409	SB11201420578812	F	Yes	Yes	ACADEMIC	E
12410	SB11201420583232	M	No	No	ACADEMIC	E

11968 rows × 7 columns

Next steps: Generate code with performance | View recommended plots

```
performance['PLACEMENT'].replace({"Yes": 1,"No":0},inplace=True)
performance
```

	STUDENT_ID	GENDER	PLACEMENT	HONOR_OPTED_OR_NOT	EDUCATION_TYPE	ACADI
0	SB11201210000129	F	1	Yes	ACADEMIC	E
1	SB11201210000137	F	1	Yes	ACADEMIC	E
3	SB11201210007504	F	1	Yes	ACADEMIC	E
4	SB11201210007548	M	1	Yes	ACADEMIC	E
5	SB11201210007568	F	1	Yes	ACADEMIC	E
...	...	...	...	...	...	
12406	SB11201420568705	M	1	Yes	ACADEMIC	ME E
12407	SB11201420573045	M	1	Yes	ACADEMIC	E
12408	SB11201420578809	M	1	No	ACADEMIC	E
12409	SB11201420578812	F	1	Yes	ACADEMIC	E
12410	SB11201420583232	M	0	No	ACADEMIC	E

11968 rows × 7 columns

Next steps: Generate code with performance | View recommended plots

```
performance['GENDER'].replace({"F":1, "M":0},inplace=True)
performance
```

PLACEMENT	HONOR_OPTED_OR_NOT	EDUCATION_TYPE	ACADEMIC_PROGRAM	OVEARLL_GRADE
1	Yes	ACADEMIC	INDUSTRIAL ENGINEERING	FIRST CLASS
1	Yes	ACADEMIC	INDUSTRIAL ENGINEERING	THIRD CLASS