Data Wrangling II

Create an “Academic performance” dataset of students and perform the following operations using Python.

1. 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.
2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.
3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution. Reason and document your approach properly.

Import all Python Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("datasets/ass2/student.csv")
df
```

```
    mathScore  writingScore  readingScore  placementScore  clubJoining
\
0        68.0          85.0          72.0              85         20.0

1        63.0          94.0          75.0              79         18.0

2        75.0          75.0          73.0              87         19.0

3         NaN          83.0          60.0              87         19.0

4        77.0          91.0          60.0              80         20.0

5        74.0          86.0           NaN              75         18.0

6        64.0          93.0          68.0              99         20.0

7        68.0          75.0          73.0              98         18.0

8        40.0          85.0          66.0              94          NaN

9         2.0          82.0          74.0              92         19.0
```

10	62.0	79.0	70.0	10	18.0
11	64.0	84.0	67.0	91	19.0
12	68.0	87.0	75.0	92	20.0
13	67.0	94.0	76.0	95	19.0
14	73.0	85.0	62.0	76	20.0
15	60.0	85.0	64.0	84	NaN
16	71.0	95.0	78.0	88	19.0
17	73.0	89.0	80.0	85	19.0
18	68.0	80.0	71.0	92	18.0
19	63.0	150.0	71.0	77	18.0
20	67.0	89.0	65.0	95	19.0
21	69.0	93.0	71.0	82	NaN
22	67.0	75.0	60.0	80	19.0
23	63.0	92.0	68.0	76	19.0
24	70.0	NaN	63.0	96	50.0
25	69.0	76.0	74.0	93	18.0
26	67.0	83.0	68.0	96	19.0
27	74.0	80.0	76.0	100	18.0
28	77.0	91.0	66.0	79	18.0

	placementOfferrCount
0	3.0
1	2.0
2	3.0
3	NaN
4	2.0
5	2.0
6	3.0
7	3.0
8	3.0
9	3.0
10	1.0

```
11                    3.0
12                    3.0
13                    3.0
14                    2.0
15                    2.0
16                    3.0
17                    3.0
18                    3.0
19                    2.0
20                    3.0
21                    2.0
22                    2.0
23                    2.0
24                    3.0
25                    3.0
26                    3.0
27                    3.0
28                    2.0
```

```
df.isna().sum()
```

```
mathScore                1
writingScore             1
readingScore             1
placementScore           0
clubJoining              3
placementOfferrCount     1
dtype: int64
```

Handling missing values

```
df.ffill(inplace=True)
df.isna().sum()
```

```
mathScore                0
writingScore             0
readingScore             0
placementScore           0
clubJoining              0
placementOfferrCount     0
dtype: int64
```

```
df
```

```
    mathScore  writingScore  readingScore  placementScore  clubJoining
\
0        68.0          85.0          72.0              85         20.0

1        63.0          94.0          75.0              79         18.0
```

2	75.0	75.0	73.0	87	19.0
3	75.0	83.0	60.0	87	19.0
4	77.0	91.0	60.0	80	20.0
5	74.0	86.0	60.0	75	18.0
6	64.0	93.0	68.0	99	20.0
7	68.0	75.0	73.0	98	18.0
8	40.0	85.0	66.0	94	18.0
9	2.0	82.0	74.0	92	19.0
10	62.0	79.0	70.0	10	18.0
11	64.0	84.0	67.0	91	19.0
12	68.0	87.0	75.0	92	20.0
13	67.0	94.0	76.0	95	19.0
14	73.0	85.0	62.0	76	20.0
15	60.0	85.0	64.0	84	20.0
16	71.0	95.0	78.0	88	19.0
17	73.0	89.0	80.0	85	19.0
18	68.0	80.0	71.0	92	18.0
19	63.0	150.0	71.0	77	18.0
20	67.0	89.0	65.0	95	19.0
21	69.0	93.0	71.0	82	19.0
22	67.0	75.0	60.0	80	19.0
23	63.0	92.0	68.0	76	19.0
24	70.0	92.0	63.0	96	50.0
25	69.0	76.0	74.0	93	18.0
26	67.0	83.0	68.0	96	19.0
27	74.0	80.0	76.0	100	18.0

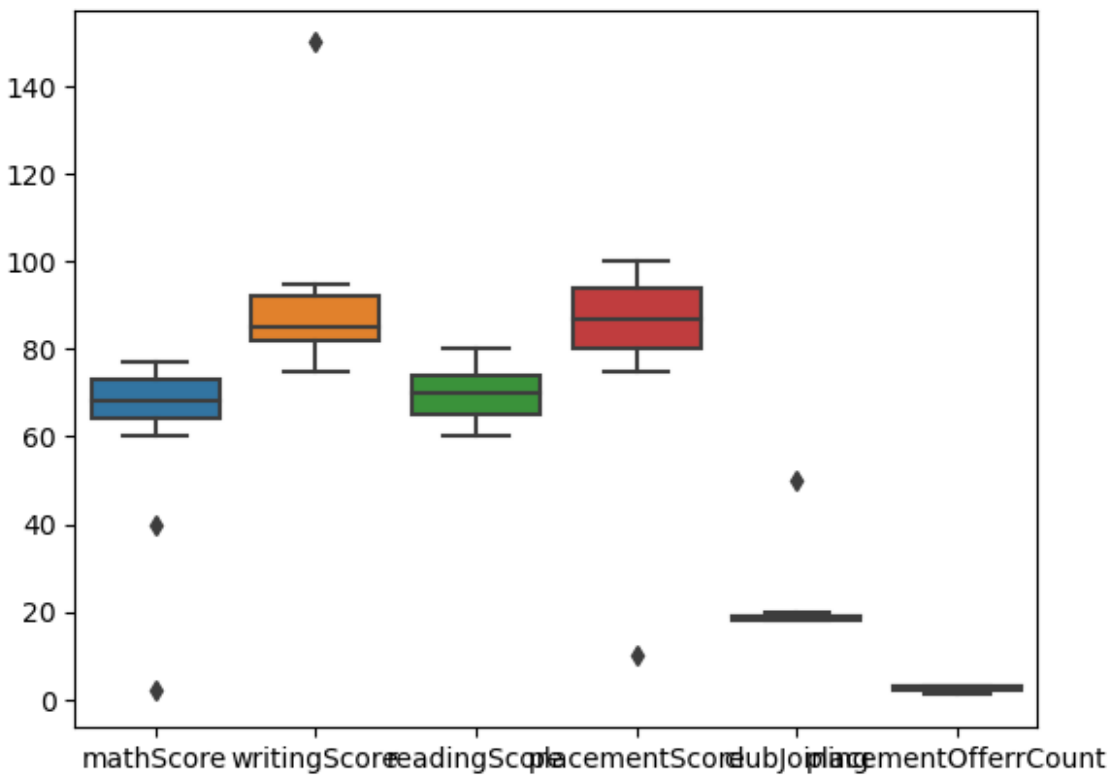
```
28        77.0           91.0           66.0                79          18.0
```

```
    placementOfferrCount
0                    3.0
1                    2.0
2                    3.0
3                    3.0
4                    2.0
5                    2.0
6                    3.0
7                    3.0
8                    3.0
9                    3.0
10                   1.0
11                   3.0
12                   3.0
13                   3.0
14                   2.0
15                   2.0
16                   3.0
17                   3.0
18                   3.0
19                   2.0
20                   3.0
21                   2.0
22                   2.0
23                   2.0
24                   3.0
25                   3.0
26                   3.0
27                   3.0
28                   2.0
```

2. Outliers

```
sns.boxplot(df)
```

```
<Axes: >
```




```
q1 = df.quantile(0.25)
q3 = df.quantile(0.75)
iqr = q3 - q1
ub = q3 + 1.5 * iqr
lb = q1 - 1.5 * iqr

ub,lb
```

```
(mathScore                 86.5
 writingScore             107.0
 readingScore              87.5
 placementScore           115.0
 clubJoining               20.5
 placementOfferrCount       4.5
 dtype: float64,
 mathScore                50.5
 writingScore             67.0
 readingScore             51.5
 placementScore           59.0
 clubJoining              16.5
 placementOfferrCount      0.5
 dtype: float64)
```

```
filtered_data = df[~((df > ub) | (df < lb))]
```

```
filtered_data.dropna(inplace = True)
filtered_data
```

	mathScore	writingScore	readingScore	placementScore	clubJoining
0	68.0	85.0	72.0	85.0	20.0
1	63.0	94.0	75.0	79.0	18.0
2	75.0	75.0	73.0	87.0	19.0
3	75.0	83.0	60.0	87.0	19.0
4	77.0	91.0	60.0	80.0	20.0
5	74.0	86.0	60.0	75.0	18.0
6	64.0	93.0	68.0	99.0	20.0
7	68.0	75.0	73.0	98.0	18.0
11	64.0	84.0	67.0	91.0	19.0
12	68.0	87.0	75.0	92.0	20.0
13	67.0	94.0	76.0	95.0	19.0
14	73.0	85.0	62.0	76.0	20.0
15	60.0	85.0	64.0	84.0	20.0
16	71.0	95.0	78.0	88.0	19.0
17	73.0	89.0	80.0	85.0	19.0
18	68.0	80.0	71.0	92.0	18.0
20	67.0	89.0	65.0	95.0	19.0
21	69.0	93.0	71.0	82.0	19.0
22	67.0	75.0	60.0	80.0	19.0
23	63.0	92.0	68.0	76.0	19.0
25	69.0	76.0	74.0	93.0	18.0
26	67.0	83.0	68.0	96.0	19.0
27	74.0	80.0	76.0	100.0	18.0
28	77.0	91.0	66.0	79.0	18.0

```
    placementOfferrCount
0                    3.0
1                    2.0
2                    3.0
3                    3.0
4                    2.0
5                    2.0
6                    3.0
7                    3.0
11                   3.0
12                   3.0
13                   3.0
14                   2.0
15                   2.0
16                   3.0
17                   3.0
18                   3.0
20                   3.0
21                   2.0
22                   2.0
23                   2.0
25                   3.0
26                   3.0
27                   3.0
28                   2.0
```

outliers using another method z-score

```
zscore = (df - df.mean())/df.std()
newdf = df[(zscore<3) & (zscore> -3)]
newdf
```

```
    mathScore  writingScore  readingScore  placementScore  clubJoining
\
0        68.0          85.0          72.0            85.0         20.0

1        63.0          94.0          75.0            79.0         18.0

2        75.0          75.0          73.0            87.0         19.0

3        75.0          83.0          60.0            87.0         19.0

4        77.0          91.0          60.0            80.0         20.0

5        74.0          86.0          60.0            75.0         18.0
```

6	64.0	93.0	68.0	99.0	20.0
7	68.0	75.0	73.0	98.0	18.0
8	40.0	85.0	66.0	94.0	18.0
9	NaN	82.0	74.0	92.0	19.0
10	62.0	79.0	70.0	NaN	18.0
11	64.0	84.0	67.0	91.0	19.0
12	68.0	87.0	75.0	92.0	20.0
13	67.0	94.0	76.0	95.0	19.0
14	73.0	85.0	62.0	76.0	20.0
15	60.0	85.0	64.0	84.0	20.0
16	71.0	95.0	78.0	88.0	19.0
17	73.0	89.0	80.0	85.0	19.0
18	68.0	80.0	71.0	92.0	18.0
19	63.0	NaN	71.0	77.0	18.0
20	67.0	89.0	65.0	95.0	19.0
21	69.0	93.0	71.0	82.0	19.0
22	67.0	75.0	60.0	80.0	19.0
23	63.0	92.0	68.0	76.0	19.0
24	70.0	92.0	63.0	96.0	NaN
25	69.0	76.0	74.0	93.0	18.0
26	67.0	83.0	68.0	96.0	19.0
27	74.0	80.0	76.0	100.0	18.0
28	77.0	91.0	66.0	79.0	18.0

	placementOfferrCount
0	3.0
1	2.0
2	3.0

```
3                    3.0
4                    2.0
5                    2.0
6                    3.0
7                    3.0
8                    3.0
9                    3.0
10                   1.0
11                   3.0
12                   3.0
13                   3.0
14                   2.0
15                   2.0
16                   3.0
17                   3.0
18                   3.0
19                   2.0
20                   3.0
21                   2.0
22                   2.0
23                   2.0
24                   3.0
25                   3.0
26                   3.0
27                   3.0
28                   2.0
```

3. Transformation using MinMax Scaler

```
from sklearn.preprocessing import MinMaxScaler , StandardScaler

minmax = MinMaxScaler()

minmaxScaled = minmax.fit_transform(filtered_data)

minmaxScaledDf = pd.DataFrame(minmaxScaled)
minmaxScaledDf.head()
```

```
          0     1     2     3    4    5
0  0.470588  0.50  0.60  0.40  1.0  1.0
1  0.176471  0.95  0.75  0.16  0.0  0.0
2  0.882353  0.00  0.65  0.48  0.5  1.0
3  0.882353  0.40  0.00  0.48  0.5  1.0
4  1.000000  0.80  0.00  0.20  1.0  0.0
```