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.

```
# Importing libraries
import pandas as pd
import numpy as np


# Read csv file into a pandas dataframe
df = pd.read_csv("academic_data.csv")

# Prints out the first few rows
print(df.head())

missing_value_formats = ["n.a.","?","NA","n/a", "na", "--"]
df = pd.read_csv("academic_data.csv", na_values = missing_value_formats)

#print gender again
print(df['Gender'].head(10))


# NaN values are marked True
```

```
print(df[‘Gender’].isnull().head(10))
# NaN values are marked False
print(df[‘Gender’].notnull().head(10))



# notnull will return False for all NaN values
null_filter = df['Gender'].notnull()
# prints only those rows where null_filter is True
print(df[null_filter])


print(df.isnull().values.any())
# Output
True
```

How to remove rows with missing values

```
# drop all rows with NaN values
df.dropna(axis=0,inplace=True)



# drop all rows with atleast one NaN
new_df = df.dropna(axis = 0, how ='any')

# drop all rows with all NaN
new_df = df.dropna(axis = 0, how ='all')

# drop all columns with atleast one NaN
new_df = df.dropna(axis = 1, how ='any')
```

```
# drop all columns with all NaN
new_df = df.dropna(axis = 1, how ='all')
```

Imputing Missing Values in our Dataset

There are many options to pick from when replacing a missing value:

A single pre-decided constant value, such as 0.

Taking value from another randomly selected sample.

Mean, median, or mode for the column.

Interpolate value using a predictive model.

Replacing NaNs with a single constant value

```
df['SPOS'].fillna(0, inplace=True)
# To check changes call
# print(df['SPOS'].head(10))
0    85.0
1    90.0
3    95.0
6    81.0
7    40.0
8    10.0
Name: SPOS, dtype: float64
```

Replacing NaNs with the value from the previous row or the next row

method = 'pad' for taking values from the previous row

```
df['DSBDA'].fillna(method='pad', inplace=True)
# print(df['SPOS'].head(10))
0    85.0
1    90.0
3    95.0
6    81.0
```

```
7    40.0
8    10.0
Name: SPOS, dtype: float64
```

We use method = 'bfill' for taking values from the next row.

```
df['SPOS'].fillna(method='bfill', inplace=True)
print(df['SPOS'].head(10))
0    85.0
1    90.0
3    95.0
6    81.0
7    40.0
8    10.0
Name: SPOS, dtype: float64
```

2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.

```
import matplotlib.pyplot as plt
```

```
df.head()
```

	Firstname	Lastname	Gender		SPOS	DSBDA		WT	DA
0	LA	Roy	M	85.0	88.0	90.0	92		
1	SA	Dixit	F	90.0	90.0	94.0	95		
3	DA	Kapoor	M	95.0	86.0	500.0	82		
6	DY	Kapoor	M	81.0	80.0	96.0	89		
7	JK	Khanna	F	40.0	88.0	95.0	87		

```
df.shape
(6, 7)
```

```
plt.boxplot(x=df['SPOS'])
```