**Q) Create an “Academic performance” dataset of students and perform the following operations using Python. transformation should be one**

**of the following reasons: to change the scale for better**

**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**

**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.**

**Import all the required Python Libraries.**

In [1]: **import** pandas **as** pd

In [2]: **import** numpy **as** np

**Reading the dataset and loading into pandas dataframe .**

In [3]: df **=** pd**.**read\_csv("StudentPerformance.csv")

In [4]: df**.**head()

Out[4]: **No math score writing score Ready Score placement score club joining placement offer count**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** 1 67.0 76.0 70.0 | | | | | | | 86.0 2020 3.0 | | |
| **1** | | 2 |  | 60.0 | 84.0 | 75.0 |  | 81.0 | 2018 | 2.0 | |
| **2** | | 3 |  | 69.0 | 86.0 | 70.0 |  | 0.0 | 2021 | 1.0 | |
| **3** | | 4 |  | 62.0 | NaN | 66.0 |  | 100.0 | 2019 | 3.0 | |
| **4** | | 5 |  | 65.0 | 95.0 | 61.0 |  | 77.0 | 2019 | 2.0 | |

In [5]: df**.**shape

Out[5]: (28, 7)

In [6]: df**.**dtypes

Out[6]: No int64

math score float64

writing score float64

Ready Score float64

placement score float64

club joining int64

placement offer count float64

dtype: object

**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.**

**Handle the Missing value**

In [7]: df**.**isna()**.**sum()

Out[7]: No 0

math score 1

writing score 2

Ready Score 1

placement score 2

club joining 0 placement offer count 1

dtype: int64

**Make a list of column having missing value**

|  |
| --- |
| cols\_with\_na **=** [] **for** col **in** df**.**columns: **if** df[col]**.**isna()**.**any():  cols\_with\_na**.**append(col) cols\_with\_na |

In [8]:

Out[8]: ['math score',

'writing score',

'Ready Score',

'placement score',

'placement offer count']

**Fill the missing value using mean for float and int datatypes and for other forword fill.**

In [9]: **for** col **in** cols\_with\_na: col\_dt **=** df[col]**.**dtypes **if** (col\_dt **==** 'int64' **or** col\_dt **==** 'float64'): outliers **=** (df[col] **<** 0) **|** (100 **<** df[col]) df**.**loc[outliers, col] **=** np**.**nan df[col] **=** df[col]**.**fillna(df[col]**.**mean()) **else**: df[col] **=** df[col]**.**fillna(method**=**'ffill') df**.**head()

Out[9]: **No math score writing score Ready Score placement score club joining placement offer count**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** 1 67.0 76.000000 | | | | | | 70.0 | | 86.0 2020 3.0 | | |
| **1** | | 2 |  | 60.0 | 84.000000 |  | 75.0 |  | 81.0 | 2018 | 2.0 | |
| **2** | | 3 |  | 69.0 | 86.000000 |  | 70.0 |  | 0.0 | 2021 | 1.0 | |
| **3** | | 4 |  | 62.0 | 82.692308 |  | 66.0 |  | 100.0 | 2019 | 3.0 | |
| **4** | | 5 |  | 65.0 | 95.000000 |  | 61.0 |  | 77.0 | 2019 | 2.0 | |

**2) Scan all numeric variables for outliers. If there are outliers, use any of the**

**suitable techniques to deal with them.**

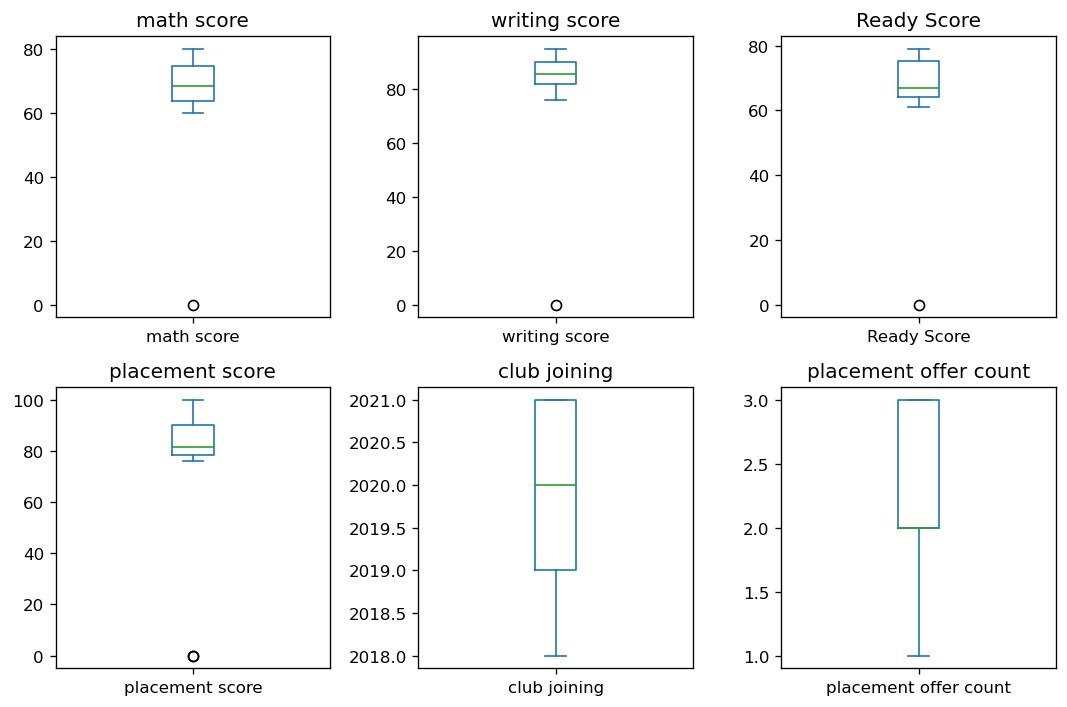
**Import all the required visualization Python Libraries.**

In [10]: **import** matplotlib.pyplot **as** plt

In [11]: **import** seaborn **as** sns

|  |
| --- |
| df\_list **=** ['math score', 'writing score', 'Ready Score', 'placement score', 'club joining', 'placement offer count'] fig, axes **=** plt**.**subplots(2, 3, figsize**=**(9, 6), dpi**=**120) **for** i, ax **in** enumerate(axes**.**flat): **if** i **<** len(df\_list):  df[df\_list[i]]**.**plot(kind**=**'box', ax**=**ax) ax**.**set\_title(df\_list[i]) **else**:  ax**.**axis('off') plt**.**tight\_layout() plt**.**show() |

In [15]:



|  |
| --- |
| Q1 **=** df['writing score']**.**quantile(0.25)  Q3 **=** df['writing score']**.**quantile(0.75)  IQR **=** Q3 **-** Q1  Lower\_limit **=** Q1 **-** 1.5 **\*** IQR Upper\_limit **=** Q3 **+** 1.5 **\*** IQR print(f'Q1 = {Q1}, Q3 = {Q3}, IQR = {IQR}, Lower\_limit = {Lower\_limit}, Upper\_limit = {U |

In [16]:

Q1 = 82.0, Q3 = 90.25, IQR = 8.25, Lower\_limit = 69.625, Upper\_limit = 102.625

|  |
| --- |
| df[(df['writing score'] **<** Lower\_limit) **|** (df['writing score'] **>** Upper\_limit)] |

In [17]:

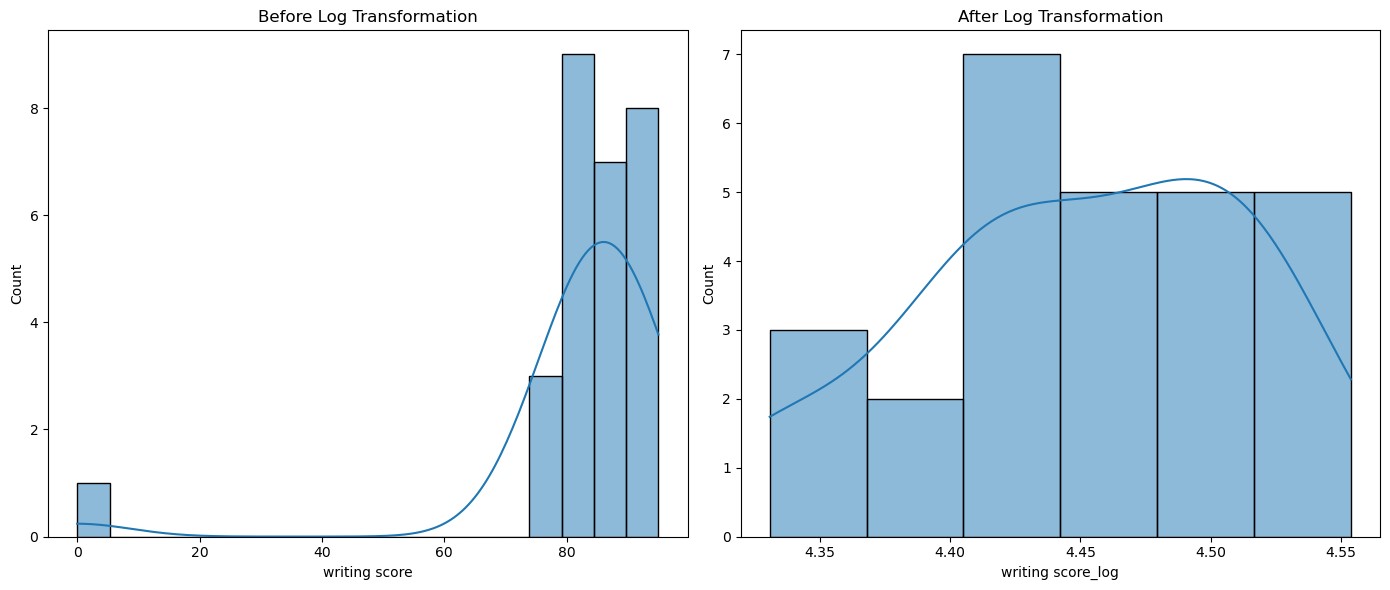
Out[17]: **No math score writing score Ready Score placement score club joining placement offer count**

5 6 60.0 0.0 65.0 100.0 2021 3.0

**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**

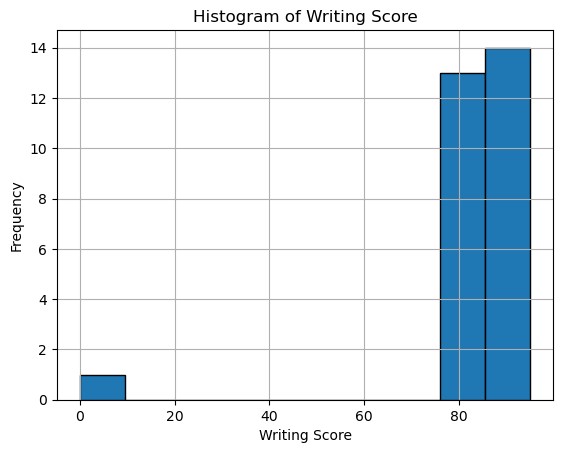
**linear one, or to decrease the skewness and convert the distribution into a normal distribution.**

|  |
| --- |
| df['writing score\_log'] **=** np**.**log(df['writing score']) fig, axes **=** plt**.**subplots(1, 2, figsize**=**(14, 6))  sns**.**histplot(data**=**df, x**=**"writing score", ax**=**axes[0], kde**=True**) sns**.**histplot(data**=**df, x**=**"writing score\_log", ax**=**axes[1], kde**=True**)  axes[0]**.**set\_title('Before Log Transformation') axes[1]**.**set\_title('After Log Transformation')  plt**.**tight\_layout() plt**.**show() |

In [18]: 

|  |
| --- |
| df['writing score']**.**plot**.**hist(bins**=**10, edgecolor**=**'black') plt**.**title('Histogram of Writing Score') plt**.**xlabel('Writing Score') plt**.**ylabel('Frequency') plt**.**grid(**True**) plt**.**show() |

In [19]:

In [ ]: