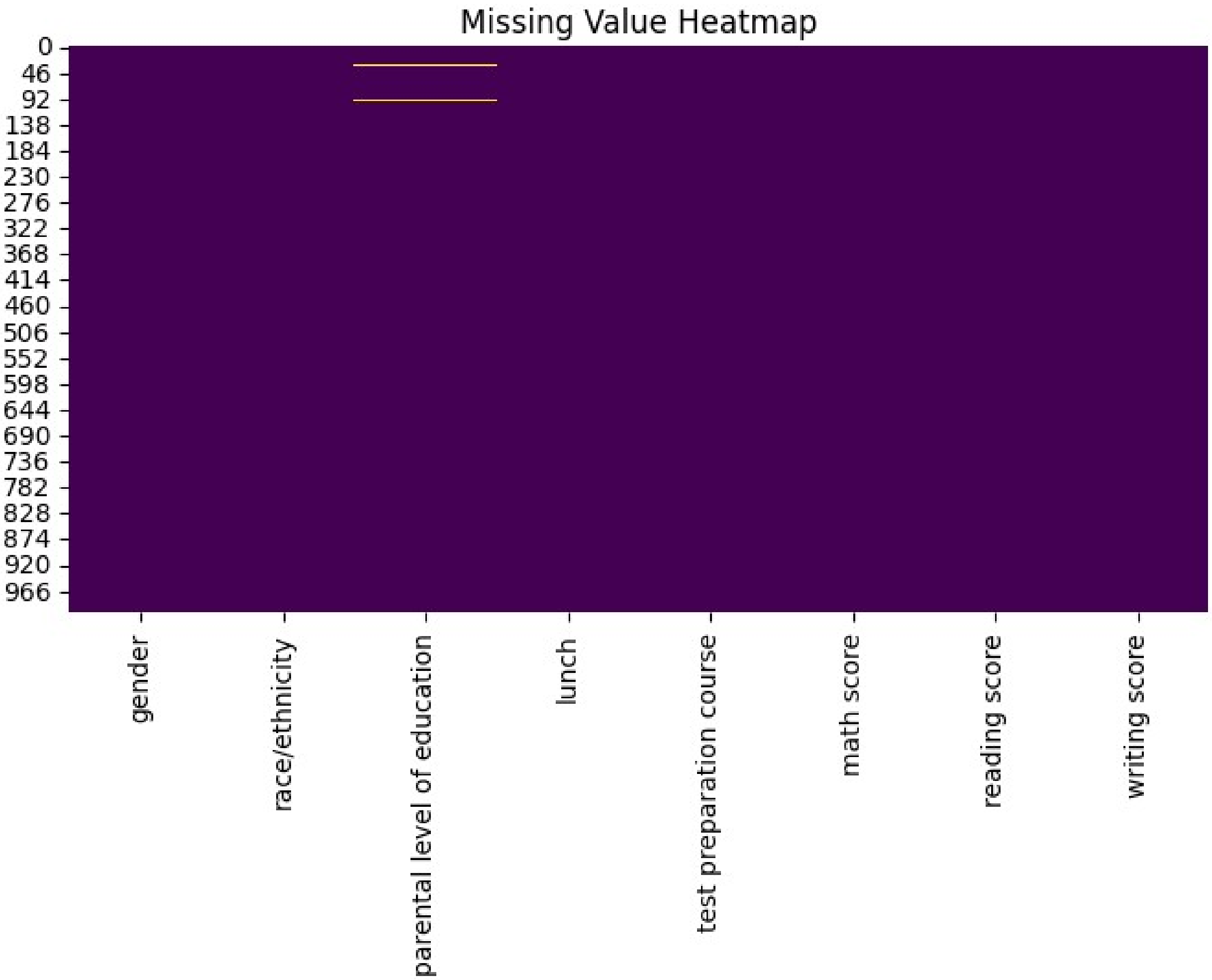
```
# Step 1: Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from google.colab import files

# Step 2: Upload dataset
print("□ Please choose your CSV file...")
uploaded = files.upload()

□ Please choose your CSV file...

<IPython.core.display.HTML object>

Saving StudentsPerformance.csv to StudentsPerformance.csv

# Get uploaded filename
filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

print("\n□ File loaded successfully!")
print(f"Shape of dataset: {df.shape}")
print(df.head())


□ File loaded successfully!
Shape of dataset: (1005, 8)
   gender race/ethnicity parental level of education         lunch  \
0  female        group B           bachelor's degree      standard   
1  female        group C                some college      standard   
2  female        group B             master's degree      standard   
3    male        group A          associate's degree  free/reduced   
4    male        group C                some college      standard   

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75

# ---------------------------------
# 1. Find Missing Values
# ---------------------------------
print("\nMissing Values in Each Column:")
print(df.isnull().sum())

# Heatmap for missing values
plt.figure(figsize=(8, 4))
```

```
sns.heatmap(df.isnull(), cbar=False, cmap="viridis")
plt.title("Missing Value Heatmap")
plt.show()


Missing Values in Each Column:
gender                         0
race/ethnicity                 0
parental level of education    7
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```



```
# --------------------------------
# 2. Imputation of Missing Values
# --------------------------------
# Numeric columns → fill with mean
num_cols = df.select_dtypes(include=['float64', 'int64']).columns
for col in num_cols:
    df[col] = df[col].fillna(df[col].mean())

cat_cols = df.select_dtypes(include=['object']).columns
for col in cat_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

print("\nMissing Values After Imputation:")
print(df.isnull().sum())
```

```
Missing Values After Imputation:
gender                         0
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     0
reading score                  0
writing score                  0
dtype: int64
```

```
# ------------------------------
# 3. Remove Duplicates
# ------------------------------
print(f"\nRows before removing duplicates: {len(df)}")
df.drop_duplicates(inplace=True)
print(f"Rows after removing duplicates: {len(df)}")
```

```
Rows before removing duplicates: 1005
Rows after removing duplicates: 1000
```

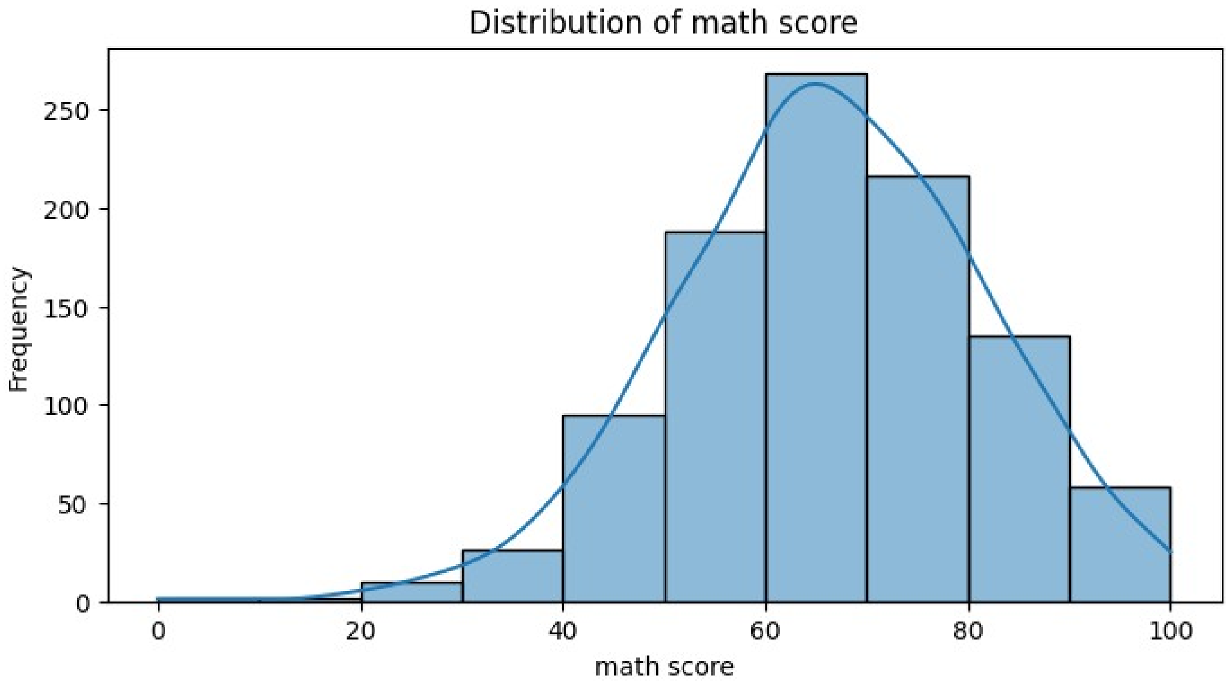
```
print("\nData Types After Conversion:")
print(df.dtypes)
```

```
Data Types After Conversion:
gender                         object
race/ethnicity                 object
parental level of education    object
lunch                          object
test preparation course        object
math score                      int64
reading score                   int64
writing score                   int64
dtype: object
```

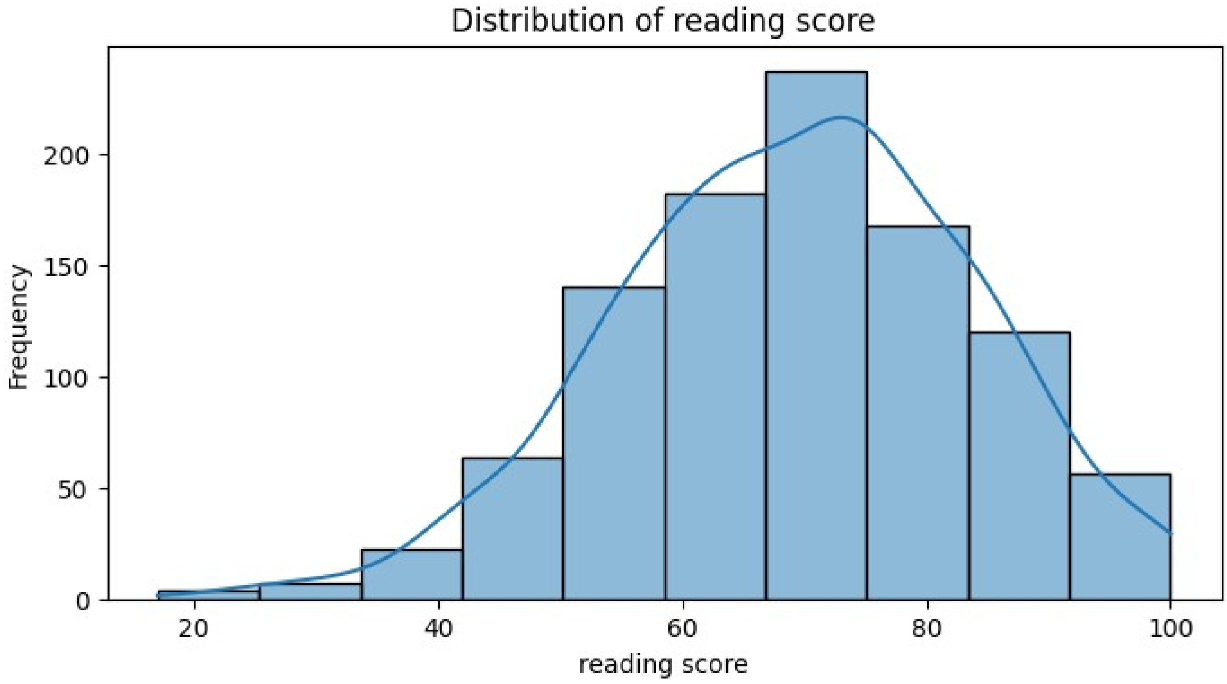
```
scaler_minmax = MinMaxScaler()
df_minmax = pd.DataFrame(scaler_minmax.fit_transform(df[num_cols]),
columns=num_cols)

# Z-score Standardization
scaler_zscore = StandardScaler()
df_zscore = pd.DataFrame(scaler_zscore.fit_transform(df[num_cols]),
columns=num_cols)

print("\nFirst 5 Rows After Min–Max Normalization:")
print(df_minmax.head())

print("\nFirst 5 Rows After Z-score Standardization:")
print(df_zscore.head())
```

```
First 5 Rows After Min–Max Normalization:
   math score  reading score  writing score
0        0.72       0.662651       0.711111
1        0.69       0.879518       0.866667
2        0.90       0.939759       0.922222
3        0.47       0.481928       0.377778
4        0.76       0.734940       0.722222

First 5 Rows After Z-score Standardization:
   math score  reading score  writing score
0    0.390024       0.193999       0.391492
1    0.192076       1.427476       1.313269
2    1.577711       1.770109       1.642475
3   -1.259543      -0.833899      -1.583744
4    0.653954       0.605158       0.457333
```

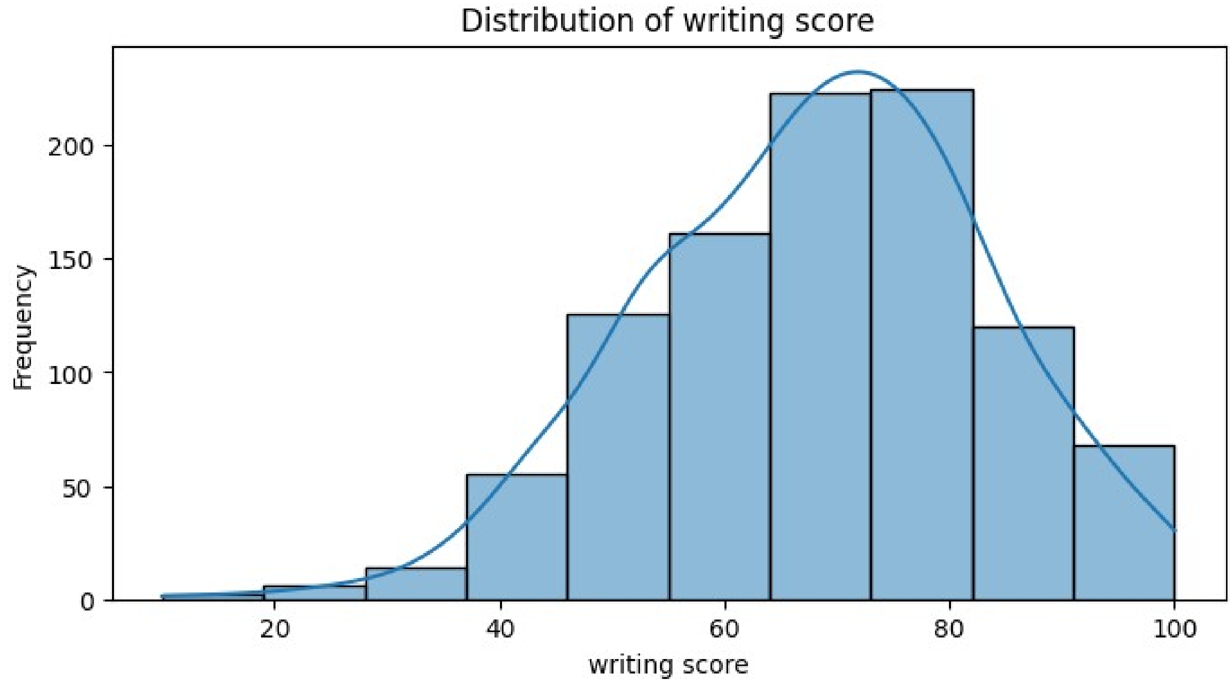
```
# --------------------------------
# 6. Visualization in Seaborn (Histograms Only)
# --------------------------------

# Plot histograms for all numeric columns
for col in num_cols:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], kde=True, bins=10)
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
    plt.show()

print("\n□ Histograms for numeric features displayed successfully!")
```



Distribution of math score



Distribution of reading score



```
□ Histograms for numeric features displayed successfully!
```