In [83]:
```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

df = pd.read_csv("Desktop/acad_record.csv")
print(df)
```

```
    student_ID  gender   age  score  attendance  study_hours
0            1    male  20.0   90.0        99.0          2.0
1            2  female  22.0   99.0        90.0          1.0
2            3  female  21.0    NaN        75.0          3.0
3            4    male  19.0   89.0        89.0          5.0
4            5  female   NaN   80.0        77.0          2.0
5            6  female  32.0   94.0        88.0          1.0
6            7    male  21.0   90.0        90.0          NaN
7            8    male  20.0   29.0        20.0          2.0
8            9  female  19.0   88.0         NaN          1.0
9           10  female  20.0   91.0        94.0          NaN
10          11    male  19.0   93.0        95.0          3.0
11          12    male   NaN    NaN        90.0          2.0
12          13     NaN  22.0   85.0        97.0          1.0
13          14    male   NaN   93.0       100.0          NaN
14          15    male  21.0   99.0        76.0          3.0
```

In [167…
```
df.head()
df.info()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   student_ID       15 non-null     int64  
 1   gender           14 non-null     object 
 2   age              12 non-null     float64
 3   score            13 non-null     float64
 4   attendance       14 non-null     float64
 5   study_hours      12 non-null     float64
 6   study_hours_IQR  12 non-null     float64
dtypes: float64(5), int64(1), object(1)
memory usage: 972.0+ bytes
```

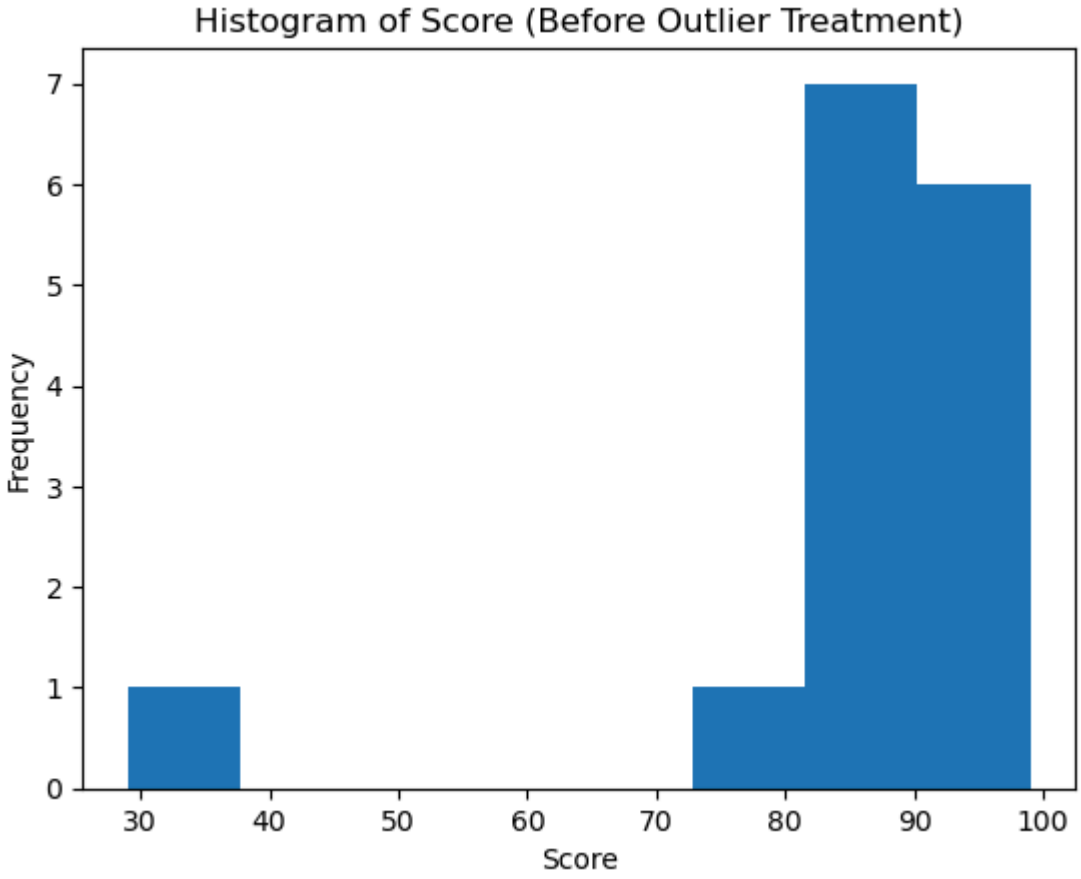
Out[167…
```
Index(['student_ID', 'gender', 'age', 'score', 'attendance', 'study_hour
s',
       'study_hours_IQR'],
      dtype='object')
```

In [32]:
```
print(df.isnull().sum())
```

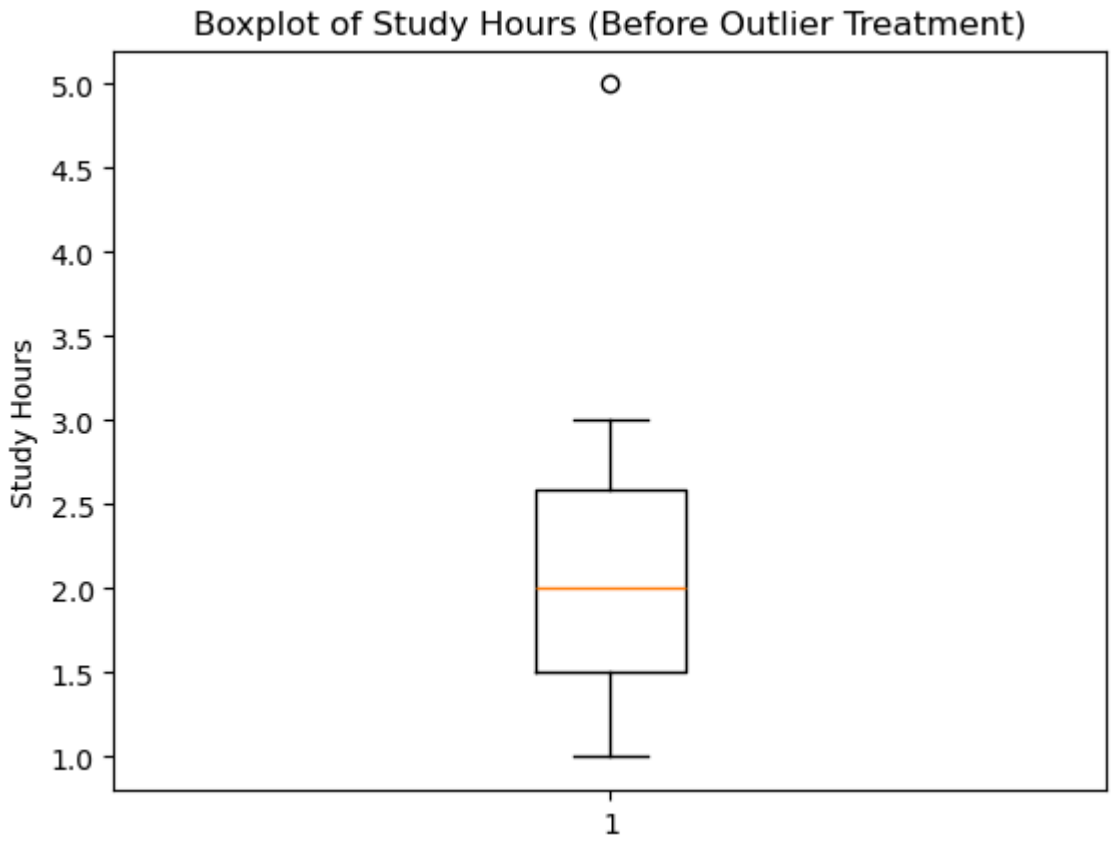
```
student_ID     0
gender         1
age            3
score          2
attendance     1
study_hours    3
dtype: int64
```

In [169…
```
df["gender"] = df["gender"].fillna(df["gender"].mode()[0])

num_cols = ["age", "score", "attendance", "study_hours"]
df[num_cols] = df[num_cols].fillna(df[num_cols].mean())
```

In [171…
```
plt.figure()
plt.hist(df["score"], bins=8)
plt.title("Histogram of Score (Before Outlier Treatment)")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.show()
```



In [173…
```
plt.figure()
plt.boxplot(df["study_hours"])
plt.title("Boxplot of Study Hours (Before Outlier Treatment)")
plt.ylabel("Study Hours")
plt.show()
```

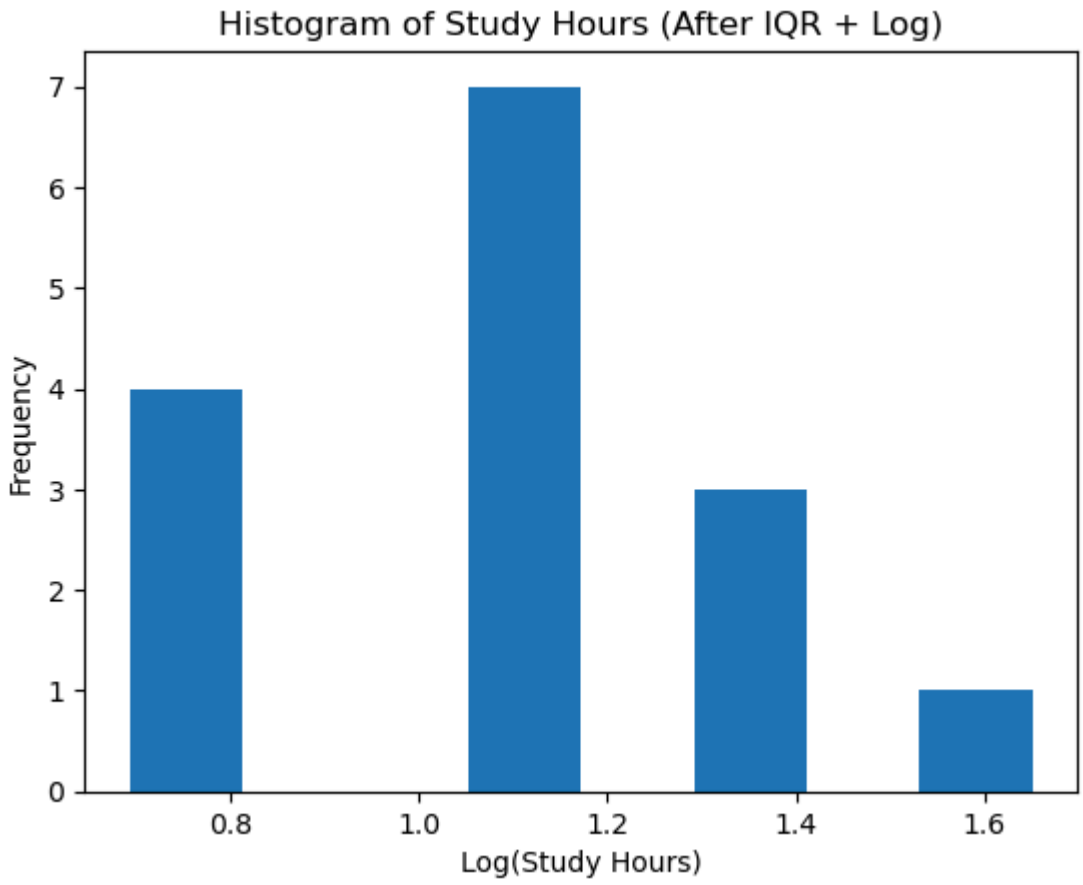


In [183…
```
#Q1 = df["study_hours"].quantile(0.25)
#Q3 = df["study_hours"].quantile(0.75)
#IQR = Q3 - Q1

#lower = Q1 - 1.5 * IQR
#upper = Q3 + 1.5 * IQR

#df["study_hours_IQR"] = df["study_hours"].clip(lower, upper)

Q1 = df["study_hours"].quantile(0.25)
Q3 = df["study_hours"].quantile(0.75)
IQR = Q3 - Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

df["study_hours_iqr"] = df["study_hours"].clip(lower, upper)
```

In [187…
```
z_scores = np.abs(stats.zscore(df["score"]))
acad_record_z = df[z_scores < 3]
```

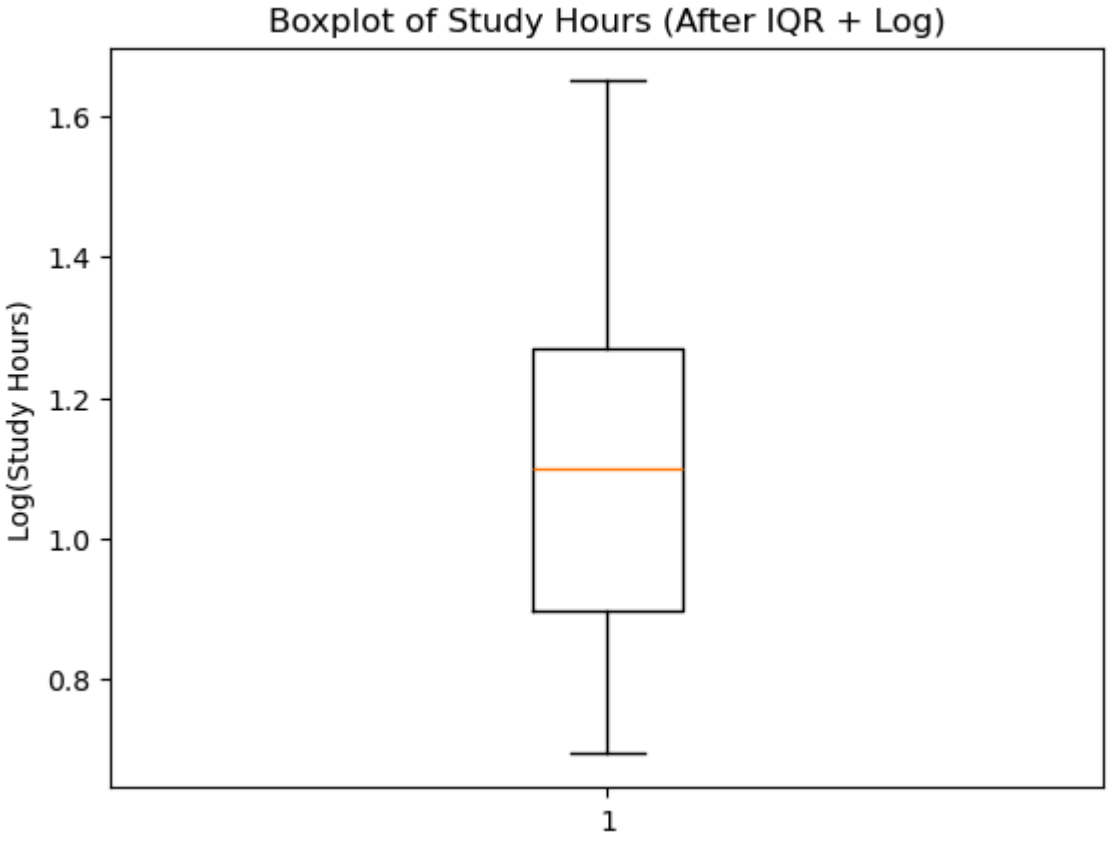
In [189…
```
df["study_hours_log"] = np.log1p(df["study_hours_iqr"])
```

In [197…
```
plt.figure()
plt.hist(df["study_hours_log"], bins=8)
plt.title("Histogram of Study Hours (After IQR + Log)")
plt.xlabel("Log(Study Hours)")
plt.ylabel("Frequency")
plt.show()
```

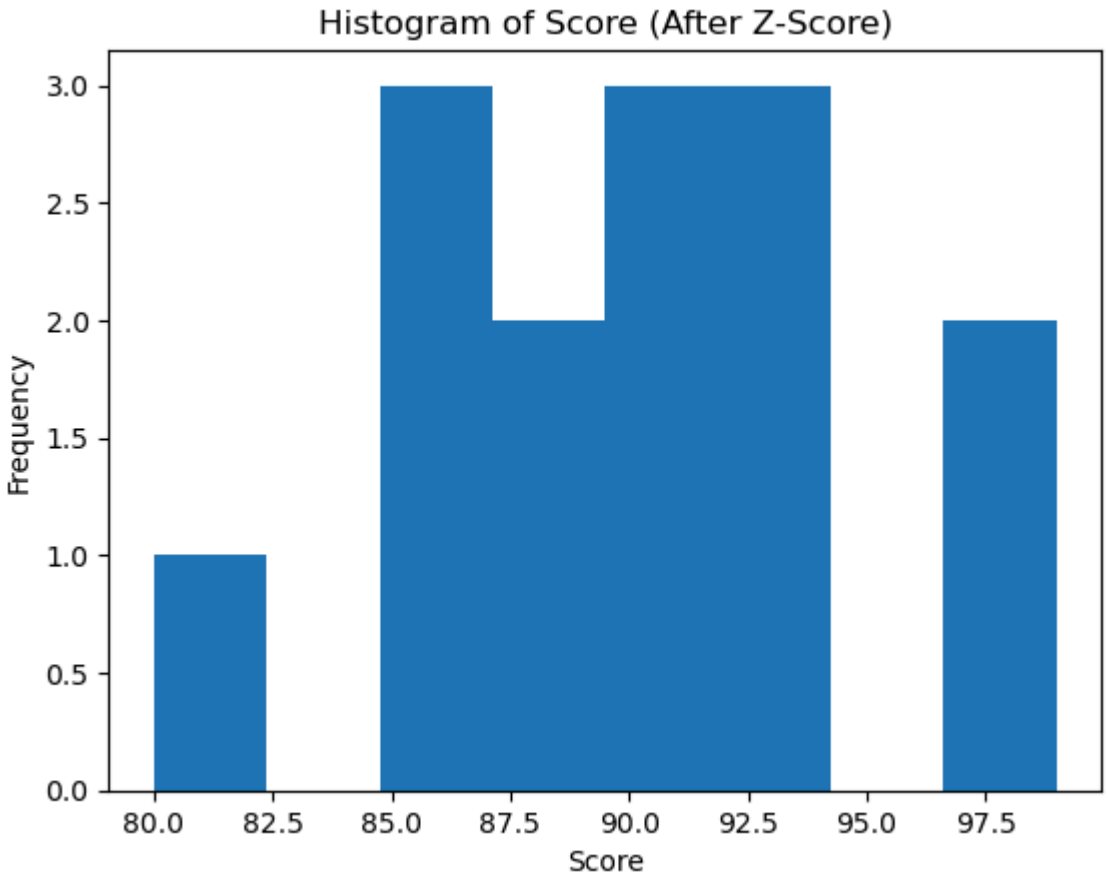


In [199…

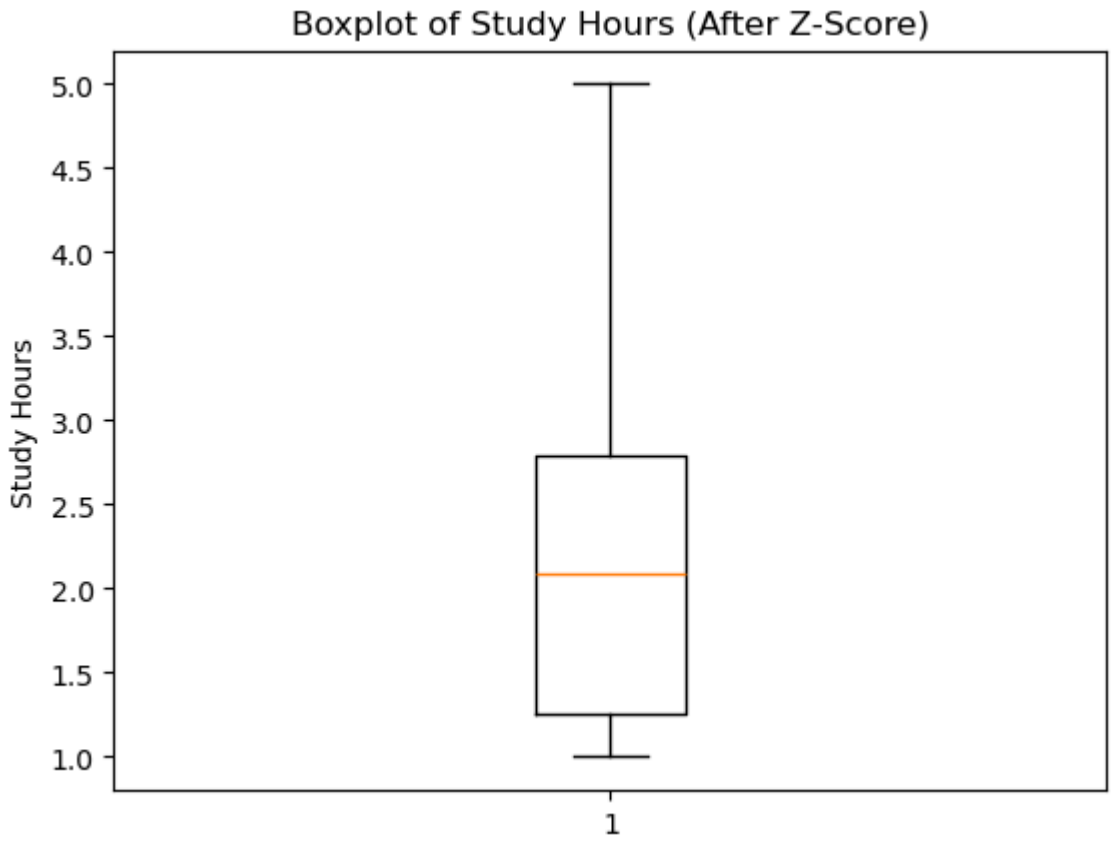
```
plt.figure()
plt.boxplot(df["study_hours_log"])
plt.title("Boxplot of Study Hours (After IQR + Log)")
plt.ylabel("Log(Study Hours)")
plt.show()
```



In [201…
```
plt.figure()
plt.hist(acad_record_z["score"], bins=8)
plt.title("Histogram of Score (After Z-Score)")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.show()
```



In [203…
```
plt.figure()
plt.boxplot(acad_record_z["study_hours"])
plt.title("Boxplot of Study Hours (After Z-Score)")
plt.ylabel("Study Hours")
plt.show()
```



In [207…
```
#last
df.to_csv("acad_record_cleaned.csv", index=False)
```

In [213…
```
#####fff
```

In [34]:
```
df["gender"] = df["gender"].fillna(df["gender"].mode()[0])
```

In [36]:
```
numeric_cols = ["age", "study_hours", "attendance", "score"]
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())
```

In [38]:
```
df.loc[df["age"] > 30, "age"] = df["age"].mean()
df.loc[df["attendance"] < 40, "attendance"] = df["attendance"].mean()
df.loc[df["score"] < 30, "score"] = df["score"].mean()
```

In [40]:
```
def handle_outliers_iqr(col):
    Q1 = col.quantile(0.25)
    Q3 = col.quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    return np.clip(col, lower, upper)
```

In [42]:
```
for col in numeric_cols:
    df[col] = handle_outliers_iqr(df[col])
```

In [44]:
```
df["Study_Hours_Log"] = np.log1p(df["study_hours"])
```

In [115…
```
print(df)
#df.to_csv("acad_performance_cleaned.csv", index=False)
```

	student_ID	gender	age	score	attendance	study_hours
0	1	male	20.0	90.0	99.0	2.0
1	2	female	22.0	99.0	90.0	1.0
2	3	female	21.0	NaN	75.0	3.0
3	4	male	19.0	89.0	89.0	5.0
4	5	female	NaN	80.0	77.0	2.0
5	6	female	32.0	94.0	88.0	1.0
6	7	male	21.0	90.0	90.0	NaN
7	8	male	20.0	29.0	20.0	2.0
8	9	female	19.0	88.0	NaN	1.0
9	10	female	20.0	91.0	94.0	NaN
10	11	male	19.0	93.0	95.0	3.0
11	12	male	NaN	NaN	90.0	2.0
12	13	NaN	22.0	85.0	97.0	1.0
13	14	male	NaN	93.0	100.0	NaN
14	15	male	21.0	99.0	76.0	3.0

In [62]:
```
from scipy import stats
import matplotlib.pyplot as plt
```

In [50]:
```
numeric_cols = ["age", "study_hours", "attendance", "score"]
```

In [149…
```
#z_scores = np.abs(stats.zscore(df[numeric_cols]))
#print(z_scores)

z_scores = np.abs(stats.zscore(df[numeric_cols]))
print(z_scores)
#df_zscore_cleaned = df[(z_scores < 3).all(axis=1)]
#print(df_zscore_cleaned)
```

	age	study_hours	attendance	score
0	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN
9	NaN	NaN	NaN	NaN
10	NaN	NaN	NaN	NaN
11	NaN	NaN	NaN	NaN
12	NaN	NaN	NaN	NaN
13	NaN	NaN	NaN	NaN
14	NaN	NaN	NaN	NaN

In [141…
```
df_zscore_capped = df.copy()

for col in numeric_cols:
    mean = df[col].mean()
    std = df[col].std()
    upper = mean + 3 * std
    lower = mean - 3 * std
    df_zscore_capped[col] = df[col].clip(lower, upper)

print(df_zscore_capped)
```

	student_ID	gender	age	score	attendance	study_hours
0	1	male	20.000000	90.000000	99.000000	2.0
1	2	female	22.000000	99.000000	90.000000	1.0
2	3	female	21.000000	NaN	75.000000	3.0
3	4	male	19.000000	89.000000	89.000000	5.0
4	5	female	NaN	80.000000	77.000000	2.0
5	6	female	31.907745	94.000000	88.000000	1.0
6	7	male	21.000000	90.000000	90.000000	NaN
7	8	male	20.000000	32.378673	23.660945	2.0
8	9	female	19.000000	88.000000	NaN	1.0
9	10	female	20.000000	91.000000	94.000000	NaN
10	11	male	19.000000	93.000000	95.000000	3.0
11	12	male	NaN	NaN	90.000000	2.0
12	13	NaN	22.000000	85.000000	97.000000	1.0
13	14	male	NaN	93.000000	100.000000	NaN
14	15	male	21.000000	99.000000	76.000000	3.0

In [143…
```
df_zscore_capped = df.copy()

for col in numeric_cols:
    mean = df[col].mean()
    std = df[col].std()
    upper = mean + 3 * std
    lower = mean - 3 * std
    df_zscore_capped[col] = df[col].clip(lower, upper)

print(df_zscore_capped)
```

	student_ID	gender	age	score	attendance	study_hours
0	1	male	20.000000	90.000000	99.000000	2.0
1	2	female	22.000000	99.000000	90.000000	1.0
2	3	female	21.000000	NaN	75.000000	3.0
3	4	male	19.000000	89.000000	89.000000	5.0
4	5	female	NaN	80.000000	77.000000	2.0
5	6	female	31.907745	94.000000	88.000000	1.0
6	7	male	21.000000	90.000000	90.000000	NaN
7	8	male	20.000000	32.378673	23.660945	2.0
8	9	female	19.000000	88.000000	NaN	1.0
9	10	female	20.000000	91.000000	94.000000	NaN
10	11	male	19.000000	93.000000	95.000000	3.0
11	12	male	NaN	NaN	90.000000	2.0
12	13	NaN	22.000000	85.000000	97.000000	1.0
13	14	male	NaN	93.000000	100.000000	NaN
14	15	male	21.000000	99.000000	76.000000	3.0

In [163…
```
plt.figure()
plt.hist(df_zscore["score"], bins=8)
plt.title("Histogram of Score (After Z-Score Outlier Treatment)")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[163], line 2
      1 plt.figure()
----> 2 plt.hist(df_zscore["score"], bins=8)
      3 plt.title("Histogram of Score (After Z-Score Outlier Treatment)")
      4 plt.xlabel("Score")

NameError: name 'df_zscore' is not defined
<Figure size 640x480 with 0 Axes>
```

In [153…

```
print(df.head())
print(df.info())
print(df.columns)
```

```
   student_ID  gender   age  score  attendance  study_hours
0           1    male  20.0   90.0        99.0          2.0
1           2  female  22.0   99.0        90.0          1.0
2           3  female  21.0    NaN        75.0          3.0
3           4    male  19.0   89.0        89.0          5.0
4           5  female   NaN   80.0        77.0          2.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   student_ID   15 non-null     int64
 1   gender       14 non-null     object
 2   age          12 non-null     float64
 3   score        13 non-null     float64
 4   attendance   14 non-null     float64
 5   study_hours  12 non-null     float64
dtypes: float64(4), int64(1), object(1)
memory usage: 852.0+ bytes
None
Index(['student_ID', 'gender', 'age', 'score', 'attendance', 'study_hour
s'], dtype='object')
```

In [147…

```
# Histogram - Exam Score
plt.figure()
plt.hist(df["score"], bins=8)
plt.title("Histogram of Exam Score (Before Outlier Treatment)")
plt.xlabel("Exam Score")
plt.ylabel("Frequency")
plt.show()

# Boxplot - Study Hours
plt.figure()
plt.boxplot(df["study_hours"])
plt.title("Boxplot of Study Hours (Before Outlier Treatment)")
plt.ylabel("Study Hours")
plt.show()

# -----------------------------
# Z-Score Outlier Detection
# -----------------------------
z_scores = np.abs(stats.zscore(df[numeric_cols]))
df_cleaned = df[(z_scores < 3).all(axis=1)]
```

```
# ----------------------------
# AFTER Outlier Treatment
# ----------------------------

# Histogram - Exam Score
plt.figure()
plt.hist(df_cleaned["score"], bins=8)
plt.title("Histogram of Exam Score (After Z-Score Outlier Treatment)")
plt.xlabel("Exam Score")
plt.ylabel("Frequency")
plt.show()

# Boxplot - Study Hours
plt.figure()
plt.boxplot(df_cleaned["study_hours"])
plt.title("Boxplot of Study Hours (After Z-Score Outlier Treatment)")
plt.ylabel("Study Hours")
plt.show()
```