Exercise Data Preparation

Gibt es in eurem Datensatz Duplikate? Wenn ja, welche?

Ja, es gibt insgesamt 109 Duplikate aus 4080 Datensätzen.

```
import pandas as pd

# Load the uploaded file to examine its structure
file_path = '/mnt/data/fish_data.csv'
fish_data = pd.read_csv(file_path)

# Display basic information about the dataset to understand its structure
fish_data.info(), fish_data.head()

# Check for duplicate rows in the dataset
duplicates = fish_data[fish_data.duplicated()]

# Number of duplicates and preview of duplicate rows if any
num_duplicates = len(duplicates)
duplicates if num_duplicates > 0 else "No duplicates found", num_duplicates
```

Gibt es Daten, die nicht relevant sind?

Nein.

Überlegt euch eine Strategie wie ihr mit NAN-Werten umgeht. Probiert verschiedene Methoden aus. Wie wirkt sich die Strategie auf euren Datensatz aus?

Es befinden sich keine NAN-Werte im Datensatz.

Überlegt euch eine Strategie wie ihr mit Ausreißern umgeht. Wie könntet ihr Ausreißer in eurem Datensatz behandeln? Wie viele % sind betroffen?

- Length: Es wurden keine Ausreißer identifiziert.
- Weight: 455 Einträge (etwa 11,15 %) wurden als Ausreißer erkannt. Diese Werte liegen oberhalb von 6,01 kg.
- W/L-Ratio: 17 Einträge (etwa 0,42 %) wurden als Ausreißer identifiziert. Diese Werte liegen oberhalb von 0,6.

Strategie zum Umgang mit Ausreißern**:

Filter oder Entfernen: Ausreißer oberhalb der Schwellenwerte könnten entfernt werden, wenn sie als fehlerhaft oder untypisch für die Analyse gelten.

Ersetzen: Alternativ können die Ausreißer durch den Median oder Mittelwert der jeweiligen Spalte ersetzt werden, um den Einfluss extremer Werte zu mindern.

Transformation: Falls die Werte stark abweichen, könnten auch logarithmische oder robuste Transformationen helfen, die Verteilung anzupassen.

```python
def detect_outliers_iqr(df, column_name):
    Q1 = df[column_name].quantile(0.25)
    Q3 = df[column_name].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    # Identifizieren der Ausreißer
    outliers = df[(df[column_name] < lower_bound) | (df[column_name] > upper_bound)]
    return outliers, len(outliers)

outliers_info = {}
for col in ['length', 'weight', 'w_l_ratio']:
    outliers, num_outliers = detect_outliers_iqr(fish_data, col)
    outliers_info[col] = {
        'num_outliers': num_outliers,
        'percentage': (num_outliers / len(fish_data)) * 100,
        'lower_bound': outliers[col].min() if num_outliers > 0 else None,
        'upper_bound': outliers[col].max() if num_outliers > 0 else None
    }

outliers_info
```