Documentation: Identifying Consecutive Numbers in a DataFrame

Overview

- This documentation provides a detailed explanation of how to identify numbers that appear at least three times consecutively in a given DataFrame. The DataFrame consists of two columns: id (which acts as a primary key) and num (which contains the numbers to be analyzed).

Problem Statement

- Given a DataFrame with columns id and num, the task is to find all numbers that appear consecutively at least three times. The result should be a DataFrame containing a single column ConsecutiveNums, listing the numbers that meet this condition. The output can be in any order.

Input

- **DataFrame Structure:**
  - ***id:*** Integer (primary key, auto-incremented).
  - ***num:*** String or Integer (values to be analyzed).

Example Input:

1	1
2	1
3	1
4	2
5	1
6	2
7	2

Output

- ***Expected Output Structure:***
  - ➢ The output is a DataFrame containing a single column ConsecutiveNums, which includes all unique numbers that appear consecutively at least three times.

Example Output:

ConsecutiveNums
1

Steps to Implement

1. **Initialize Variables:**

- Start by initializing an empty list result to store numbers that meet the criteria.
- Create a variable count to track the consecutive occurrences of each number, starting at 1.

2. Iterate Through the DataFrame:

- Begin iterating through the DataFrame from the second row to compare each number (num) with its previous value.

3. Check Consecutive Numbers:

- If the current number matches the previous one, increment the count by 1.
- *If the current number differs from the previous one, evaluate the count:*
  - ➢ If count is 3 or more, append the previous number to the result list.
  - ➢ Reset count to 1 for the new sequence.

4. Handle the Last Sequence:

- After the loop, check if the last number in the DataFrame has a count of 3 or more. If so, append it to the result.

5. Ensure Uniqueness:

- Use pd.unique() to remove any duplicate numbers from the result list, ensuring that each number appears only once in the output.

6. Format the Output:

- Convert the result list into a DataFrame with a single column named ConsecutiveNums.

Edge Cases

- **All Numbers are the Same:** If all the numbers in the DataFrame are identical, the code should correctly identify this as a valid sequence.

- **No Consecutive Sequences:** If there are no numbers that appear at least three times consecutively, the output should be an empty DataFrame.

- **Short DataFrames:** For DataFrames with fewer than three rows, there should be no valid output, as no number can appear three times consecutively.

Performance Considerations

- **Time Complexity:** The algorithm runs in linear time relative to the size of the input DataFrame, as it involves a single pass through the DataFrame.

- **Memory Usage:** Memory usage is minimal, as it only stores the result list and a few additional variables.

Conclusion

- This documentation outlines the steps and logic required to identify numbers that appear at least three times consecutively in a DataFrame. The approach is designed to be efficient and handle various edge cases gracefully. The result is a DataFrame containing the unique numbers that meet the criteria, ready for further analysis or reporting.