**Errors and Drawbacks**

1. **Global Variable Usage**:
   * **Error/Drawback**: The variable number is declared globally and could lead to unintended side effects if accessed or modified elsewhere in the code.
2. **Name Scope Issue**:
   * **Error/Drawback**: The variable name is defined in the intro() function but is used in the pick() function without being passed explicitly, which can lead to a NameError if name is not defined in the expected scope.
3. **Magic Numbers**:
   * **Error/Drawback**: The number of allowed guesses (6) and the range (1-200) are hard-coded, making the code less flexible and harder to maintain or modify.
4. **General Exception Handling**:
   * **Error/Drawback**: The except block catches all exceptions, which is not recommended because it can suppress unexpected errors and make debugging difficult. Specific exceptions should be caught instead.
5. **Repeated Code for Play Again**:
   * **Error/Drawback**: The expression playagain=="yes" or playagain=="y" or playagain=="Yes" is repeated, making the code unnecessarily verbose and prone to errors.
6. **Unnecessary time.sleep Calls**:
   * **Error/Drawback**: Excessive use of time.sleep can make the program feel slow and unresponsive, especially in a game where quick feedback is desired.
7. **Potential Infinite Loop**:
   * **Error/Drawback**: The main loop could potentially run indefinitely if the play again input is not handled correctly or if an invalid input is given.

**Recommendations for Fixes**

* **Encapsulate Variables**: Encapsulate the number variable within a function or class to avoid global state issues.
* **Pass Variables**: Ensure that variables such as name are passed explicitly to functions that need them.
* **Use Constants**: Define configurable parameters like the number of guesses and the range as constants.
* **Specific Exceptions**: Use specific exceptions like ValueError to handle known error conditions.
* **Simplify Logic**: Simplify conditional checks and avoid repetitive code.
* **Reduce time.sleep**: Minimize the use of time.sleep to improve user experience.

These are the key errors and drawbacks in the current code that need to be addressed to improve its efficiency and maintainability.