A variable already borrowed as immutable was borrowed as mutable.

Erroneous code example:

```
fn bar(x: &mut i32) {}
fn foo(a: &mut i32) {
    let y = &a; // a is borrowed as immutable.
    bar(a); // error: cannot borrow `*a` as mutable because `a` is also borrowed
            //        as immutable
    println!("{}", y);
}
```

To fix this error, ensure that you don't have any other references to the variable before trying to access it mutably:

```
fn bar(x: &mut i32) {}
fn foo(a: &mut i32) {
    bar(a);
    let y = &a; // ok!
    println!("{}", y);
}
```

For more information on Rust's ownership system, take a look at the References & Borrowing section of the Book.