

An invalid number of arguments was given when calling a closure.

Erroneous code example:

```
let f = |x| x * 3;
let a = f();           // invalid, too few parameters
let b = f(4);          // this works!
let c = f(2, 3);       // invalid, too many parameters
```

When invoking closures or other implementations of the function traits `Fn`, `FnMut` or `FnOnce` using call notation, the number of parameters passed to the function must match its definition.

A generic function must be treated similarly:

```
fn foo<F: Fn()>(f: F) {
    f(); // this is valid, but f(3) would not work
}
```