Note: this error code is no longer emitted by the compiler.

You gave too many lifetime arguments. Erroneous code example:

```
fn f() {}

fn main() {
    f::<'static>() // error: wrong number of lifetime arguments:
                   //        expected 0, found 1
}
```

Please check you give the right number of lifetime arguments. Example:

```
fn f() {}

fn main() {
    f() // ok!
}
```

It's also important to note that the Rust compiler can generally determine the lifetime by itself. Example:

```
struct Foo {
    value: String
}

impl Foo {
    // it can be written like this
    fn get_value<'a>(&'a self) -> &'a str { &self.value }
    // but the compiler works fine with this too:
    fn without_lifetime(&self) -> &str { &self.value }
}

fn main() {
    let f = Foo { value: "hello".to_owned() };

    println!("{}", f.get_value());
    println!("{}", f.without_lifetime());
}
```