A method was called on an ambiguous numeric type.

Erroneous code example:

```
2.0.neg(); // error!
```

This error indicates that the numeric value for the method being passed exists but the type of the numeric value or binding could not be identified.

The error happens on numeric literals and on numeric bindings without an identified concrete type:

```
let x = 2.0;
x.neg();  // same error as above
```

Because of this, you must give the numeric literal or binding a type:

```
use std::ops::Neg;

let _ = 2.0_f32.neg(); // ok!
let x: f32 = 2.0;
let _ = x.neg(); // ok!
let _ = (2.0 as f32).neg(); // ok!
```