

Can integers be caught in R, even if they have a value of 0?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *Can integers be caught in R, even if they have a value of 0?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=156625>

The R programming language has the ability to capture and manipulate various types of data, including integers. Even if an integer has a value of 0, it can still be caught and used in R. This means that R is able to handle and process all types of integers, regardless of their specific value. This feature makes R a versatile and powerful tool for data analysis and manipulation.

Catch integer(0) in R (With Examples)

Sometimes when you use the function in R, you may end up with `integer(0)` as a result, which indicates that none of the elements in a vector evaluated to `TRUE`.

For example, suppose we use the following code to check which elements in a vector are equal to the value 10:

```
#define vector of values
```

```
data <- c(1, 2, 4, 4, 5, 7, 8, 9)
```

```
#find elements in vector equal to 10
```

```
x <- which(data == 10)
```

```
#view results
```

```
x
```

```
integer(0)
```

Since none of the elements in the vector are equal to 10,

the result is an integer of length 0, written as `integer(0)` in R.

It's important to note that an `integer(0)` is not an error, but sometimes you may just want to be aware of when it occurs.

The following examples show how to catch an `integer(0)` in R.

Example 1: Catch `integer(0)` in R Using `identical()` Function

The easiest way to catch an `integer(0)` in R is to use the `identical()` function in the following manner:

```
#define vector of values
data <- c(1, 2, 4, 4, 5, 7, 8, 9)

#find elements in vector equal to 10
x <- which(data == 10)

#test if x is identical to integer(0)
identical(x, integer(0))
```

TRUE

Since our result is equal to `integer(0)`, R returns **TRUE**.

This lets us know that the result of the which() function is an integer of length 0.

Example 2: Catch integer(0) in R Using if else Function

Another way to catch an integer(0) is to define an if else function that returns something specific if an integer(0) occurs.

For example, we could define the following function to return the phrase "It is an integer(0)" if an integer(0) occurs:

```
#define function to catch integer(0)  
integer0_test <- function(data) {  
  
  if(identical(data, integer(0))) {  
    return('It is an integer(0)')  
  }  
  
  else {  
    return(data)  
  }  
  
}
```

We can then use this function:

```
#define vector of values
```

```
data <- c(1, 2, 4, 4, 5, 7, 8, 9)
```

```
#find elements in vector equal to 10
```

```
x <- which(data == 10)
```

```
#use function to test if x is integer(0)
```

```
integer0_test(x)
```

```
"It is an integer(0)"
```

And if x is not an integer(0), our function will simply return the result of the which() function:

```
#define vector of values
```

```
data <- c(1, 2, 4, 4, 5, 7, 8, 9)
```

```
#find elements in vector equal to 4
```

```
x <- which(data == 4)
```

```
#use function to test if x is integer(0)
```

```
integer0_test(x)
```

3 4

Our function returns 3 and 4 because these are the positions of the elements in the vector that are equal to the value 4.

Additional Resources

The following tutorials explain how to perform other common tasks in R:

ARABPSYCHOLOGY.COM