

Algorithms and Data Structures

Encoding

Number encoding

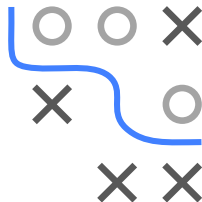
Input: 15

2	15		
2	7	—	1
2	3	—	1
	1	—	1

Learning goals

- Codes for numbers

Binary number: 1111



CODES FOR NUMBERS

- The basic arithmetic operations (and many other arithmetic operations) are performed directly by the CPU. The fewer bits per number, the faster.
- For technical reasons, a number should be encoded by a **fixed number of bytes**, thus using N bits only.
- We are looking for a function that maps sets of numbers like \mathbb{Z} or \mathbb{R} to the set of the 2^N available machine numbers.
- A fallacy: "Computer calculations are always correct."
- Basic knowledge of computer arithmetic is essential for anyone who mainly uses computers for calculations, i.e. especially for statisticians.



CODES FOR NUMBERS / 2

"Bug"-Report in R:

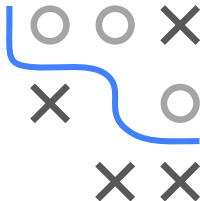
```
From: focus17@libero.it  
To: R-bugs@biostat.ku.dk  
Subject: error in trunc function  
Date: Fri, 6 Jul 2007 15:03:58 +0200 (CEST)
```

the command get a wrong result

```
> trunc(2.3 * 100)  
[1] 229
```

Answer Duncan Murdoch:

That is the correct answer. 2.3 is not representable exactly; the actual value used is slightly less.

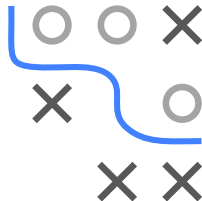


CODES FOR NUMBERS / 3

From: wchen@stat.tamu.edu
To: R-bugs@biostat.ku.dk
Subject: [Rd] match() (PR#13135)
Date: Tue, 7 Oct 00:05:06 2008

The match function does not return value properly.
See an example below.

```
> a = seq(0.6, 1, by = 0.01)
> match(0.88, a)
[1] 29
> match(0.89, a)
[1] NA
...
> match(0.94, a)
[1] 35
```

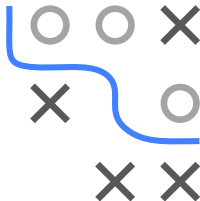


CODES FOR NUMBERS / 4

Answer Brian Ripley:

FAQ Q7.31 strikes again!

0.89 is not a member of `seq(0.6,1,by=0.01)`, since 0.01 cannot be represented exactly in a binary computer.

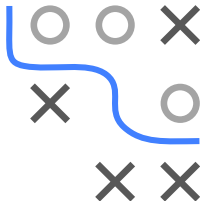


CODES FOR NUMBERS / 5

From: Friedrich Leisch <friedrich.leisch@stat.uni-muenchen.de>
To: Antonio Linan <antoniolvsa@hotmail.com>
Cc: <cran@r-project.org>
Subject: Re: Bug in R?
Date: Thu, 5 Nov 2009 13:57:03 +0100

>>>> On Thu, 5 Nov 2009 13:35:09 +0100,
>>>> Antonio Linan (AL) wrote:

```
> Hi, I'm not sure if it's really a bug:  
> When you execute:  
>> (2 / 3) * (0.6 / (1 - 0.6))  
> the result will be:  
> [1] 1  
> but if you execute:  
>> (2 / 3) * (0.6 / (1 - 0.6)) == 1  
> the result is:  
> [1] FALSE
```

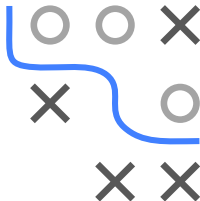


CODES FOR NUMBERS / 6

```
> Note: I'm using version 2.9.2, (and tried it in  
> 2.9.1 in 2.9.1 too) with Microsoft Windows XP  
> [Version 5.1.2600].  
> Thank you.
```

FAQ 7.31 strikes again:

```
R> 1 - (2 / 3) * (0.6 / (1 - 0.6))  
[1] 2.220446e-16  
R> .Machine$double.eps  
[1] 2.220446e-16
```



CODES FOR NUMBERS / 7

From: Marc Schwartz <marc_schwartz_at_me.com>

Date: Fri, 09 Jul 2010 09:00:10 -0500

On Jul 9, 2010, at 8:46 AM, Trafim Vanishek wrote:

```
> Dear all,  
>  
> might seem an easy question but I cannot figure it out.  
>  
> floor(100 * (.58))  
> [1] 57  
>  
> where is the trick here?  
> And how can I end up with the right answer?
```

See \texttt{R} FAQ 7.31

```
> sprintf("%.20f", 100 * .58)  
[1] "57.99999999999999289457"
```

