Ruby master - Feature #10445

[PATCH 3/3] Extend Matrix#[]

10/29/2014 08:30 AM - gogotanaka (Kazuki Tanaka)

Status:	Feedback
Priority:	Normal
Assignee:	marcandre (Marc-Andre Lafortune)
Target version:	

Description

I've made patches which Matrix#[] returns new vector if either arguments is range,
and returns new matrix, if both arguments are range.

Like below.

```
# matrix[row, column]           -> obj        or nil
# matrix[row, col_range]        -> new_vector or nil
# matrix[row_range, column]     -> new_vector or nil
# matrix[row_range, col_range] -> new_matrix or nil

Matrix.diagonal(9, 5, -3)[1, 1]
  => 5

Matrix.diagonal(9, 5, -3)[1, 0..1]
  => Vector[0, 5]

Matrix.diagonal(9, 5, -3)[0..1, 0]
  => Vector[9, 0]

Matrix.diagonal(9, 5, -3)[0..1, 0..1]
  => Matrix[[9, 0], [0, 5]]
```

I'm not sure matrix[row, col_range] should return vector or matrix
But from my view, it's fine.

I'm not in a hurry. Take your time.

History

#1 - 11/19/2014 05:54 PM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Feedback

I understand the idea, but I'm not convinced.

Currently, [] is a simple access to the elements of a matrix. This proposal makes it more complex and changes completely the type of return depending on the arguments. The main question is: when would someone want to extract a minor from a matrix and much prefer calling [] instead of the clearer and explicit minor?

I feel like we should favor explicitness in this case.

#2 - 11/20/2014 06:47 PM - gogotanaka (Kazuki Tanaka)

@Marc-Andre Lafortune

Thank you for reply.

OK, the answer is when we expect Matrix to behave something like Array, I mean..

```
[1,2,3,4][1..2]
#=> [2,3]

Matrix[[1,2,3],[4,5,6],[7,8,9]][0..1, 0..1]
#=> Matrix[[1,2], [4,5]]
```

Actually I am also one of people who expect such a behavior.

Thanks.

Files

add_test.patch	1.05 KB	10/29/2014	gogotanaka (Kazuki Tanaka)
implement_matrix.rb.patch	1.96 KB	10/29/2014	gogotanaka (Kazuki Tanaka)
update_NEWS.patch	825 Bytes	10/29/2014	gogotanaka (Kazuki Tanaka)