Calculate age in Date class

Seems like there's enough complexity in determining the right values and it's something a ton of apps have to address. I've seen a bunch of StackOverflow and other blog posts attempt to address this, not all of them accurately.

Specifically age would calculate the number of years since a person's birthdate (or how old they are).

So if today is 2018, 6, 28:
Date.new(2016, 3, 6).age #=> 2
Date.new(2016, 6, 29).age #=> 1

I am not sure if .age() is a "proper name" or not, so I can't really decide on that. But I specifically wanted to add two things:

a) not all names necessarily fit towards the ruby object model as-is. Ultimately you have to ask matz about what fits and what does not fit, but several methods in the active* ecosystem (the gems) do not make a lot of sense from a "pure" OOP point of view, in my opinion. That is not to say that these methods never make sense, mind you (when I suggested ... I think it was .append() or something, I was not even aware of the active* ecosystem; I only noticed that from a twitter message how ruby core adds active* stuff ... but I don't use rails etc.. so I came from a completely different point of view, yet the suggestion seems to have made sense to matz back then.)

So TL;DR, I think one has to decide on a case-by-case basis whether it fits or not, API-wise that is.

b) as for your other comment, in regards to StackOverflow, but even more importantly, what people use with ruby, I actually agree. I think if people keep on having the same issue over and over again, it may be where the core of ruby could be improved.

Date/Time related stuff is something that I feel is coming up every now and then, so in this general area I agree - also because I think it is something lots of people use ruby with.

By the way, it may also help to add a description what the method .age() does, because with a missing description, others have to fill in so it adds to their burden of implementation. It's much easier for the core team to add methods when there is a clear documentation associated with it - but it is your suggestion, not mine, feel free to proceed in any way that you see fit.

I personally use the "poor man's approach", always calculating in seconds. :P

I actually opened a pull request in Rails to add this to ActiveSupport, but they told me to come here!
https://github.com/rails/rails/pull/33245
derosm2 (Mike DeRosa) wrote:

```ruby
Date.new(2016, 3, 6).age #=> 2
```

It might be nice to be able to specify a unit of time other than year.

```ruby
Date.new(2016, 3, 6).age #=> 2
Date.new(2016, 3, 6).age(:year) #=> 2
Date.new(2016, 3, 6).age(:century) #=> 0
Date.today.iso8601 #=> "2018-07-18"
Date.new(2016, 3, 6).age(:month) #=> 28
```


#6 - 08/09/2018 06:44 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I don't think the calculation of age from the birthday is not a responsibility of Date class.
There may be a chance for the method for more generic duration calculation of dates/times.

Matz.