IntermediateProblemAnswers

May 26, 2017

0.1 Metaprogramming Project

```
In [1]: macro ~(y,ex)
            new_ex = Meta.quot(ex)
            quote
              inner_ex = $(esc(new_ex))
              data_name = Symbol(string(inner_ex.args[end])[1])
              eval_ex = Expr(:(=),:data,data_name)
              eval(Main,eval_ex)
              new_X = Matrix{Float64}(size(data,1),length(inner_ex.args)-1)
              cur_spot = 0
              for i in 2:length(inner_ex.args)
                if inner_ex.args[i] == 1
                  new_X[:,i-1] = ones(size(data,1))
                else
                  col = parse(Int,string(string(inner_ex.args[i])[2]))
                  new_X[:,i-1] = data[:,col]
                end
              end
              $(esc(y)),new_X
            end
        end

        y = rand(10)
        X = rand(10,4)
        y~1+X1+X2+X4

        function solve_least_squares(y,X)
          X\y
        end
        solve_least_squares(tup::Tuple) = solve_least_squares(tup...)
        solve_least_squares(y~1+X1+X2+X4)

Out[1]: 4-element Array{Float64,1}:
          0.153788
          0.742984
         -0.268836
          0.320996
```

0.2 Distribution Dispatch Problem

This is from Josh Day's talk: https://www.youtube.com/watch?v=EwcTNzpQ6Sc
    Solution is from: https://github.com/joshday/Talks/blob/master/SLG2016_IntroToJulia/Slides.ipynb

```julia
In [ ]: function myquantile(d::UnivariateDistribution, q::Number)
            θ = mean(d)
            tol = Inf
            while tol > 1e-5
                θold = θ
                θ = θ - (cdf(d, θ) - q) / pdf(d, θ)
                tol = abs(θold - θ)
            end
            θ
        end

        for dist in [Gamma(5, 1), Normal(0, 1), Beta(2, 4)]
            @show myquantile(dist, .75)
            @show quantile(dist, .75)
            println()
        end
```