1.  
--------------  
remember right now we hard-coded the data?  
  
i want you now to use the y's that you generated from a fixed (pi,Q,R) when you GENERATED the data.  
  
so first be god and generate x's and y's.  with a fixed (pi,Q,R)  
  
then I want you to take those same y's and run the nonlinear filter.  
  
i want you to watch the posteriors on x and compare the to the true x values that god generated.  
  
u should be able to manage that.

**Notes SR July 18, 2013:**

* Files that I am using to accomplish Aim 1 above:
  + Functions:
    - GenerateRandomBinaryOutcomeS: this function requires two inputs: pmf and z
      * This function compares z to the pmf vector (which is binary here).
      * If z is less than pmf(1), the output is state 1, else output is state 2
    - generateDataWithGivenPiQR\_july18\_v1: this file is used to generate a series of x’s and y’s. this is the “Playing God” file.
* Update:
  + Made good progress on this aim-i.e. I can now run the code nonlinearfilterElegant\_PlayingGodData and accomplish the following:
    - #1: Play God and generate a sequence of X’s and Y’s
    - #2: Calculate a posterior belief about every X from 1:T,
    - #3: Compare the posterior belief at each time t from 1:T with the X’s generated by “God”. When I do this, the posteriors consistently match with my God-generated X’s. Only found one instance with a mismatch so far
  + Things I’m still optimizing:
    - How do I generate graphs for the posteriors at each time point?

2.  
--------------  
I want you to keep X discrete but now I want y's to be continuous random variables.  I want you to no longer make R a matrix. But make it a conditional density.  
  
Rather than saying that P(Y=k | X=j) = R(j,k),  
I now want you to say that f\_{Y|X}(v | j) to be a density.  
what this means is that, if X=1 then Y is a continuous random variable with one density, f\_{Y|X}(v|1).  
and if X = 2 then Y is a continous random variable with a different density, f\_{Y|X}(v|2)  
  
in particular, I want you to deal with Gaussians.  Remember a Gaussian has a mean and a variance.  
so there will be a (mu1,var1) pertaining to the mean and variance of Y if X=1.  
likewise, there will be a (mu2,var2) pertaining to the mean and variance of Y if X=2.  
  
so now your "R" matrix is not probabilities but rather it is the parameters of the Gaussian PDF.  You could still let the "R" matrix be somethign like   
  
R(1,1) = mu1;   R(1,2) = var1  
R(2,1) = mu2;  R(2,2) = var2  
  
So I want you to have a new M file which is generating data for a fixed (pi,Q,"R") but now R represents as I said above.  
  
Note that you will generate the discrete X's just like you did in the code before.  
But now for any particular t, when you generate Y at time t, you first look at X at time t.  
if x\_t =1, then I want you to make Y\_t to be a Gaussian random variable with expectaion given by mu1 (*namely, R(1,1)* ) and variance given by var1 (*namely, R(1,2)* ).  
Similarly for if x\_t=2.  
  
do you remember in class how, if I give you Z which is a Gaussian random variable of mean 0 and variance 1, how you can generate Y = a Z + b so that Y has mean mu1 and variance var1?  
  
Matlab can give you a zero-mean, unit variance random variable by calling randn.  
  
I want you to next work on trying to be god in thsi scenrio.  
  
3. then after you nail this, you can do the filter for these continuous Y's.  
  
but first things first, let's work on 1 and 2.  I might have time at 9:30am to talk abou this, or around noon.  otherwise we can chat on friday.   first you take the lead on trying to hammer this out?

https://mail.google.com/mail/u/0/images/cleardot.gif