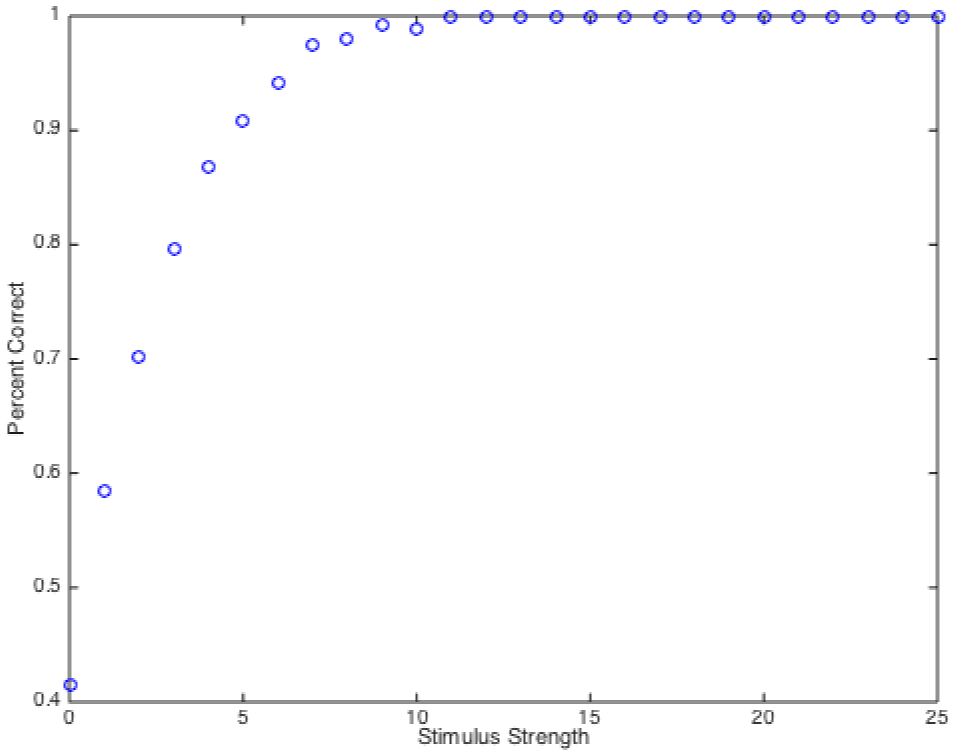
PERCEPTION SDT ASSIGNMENT

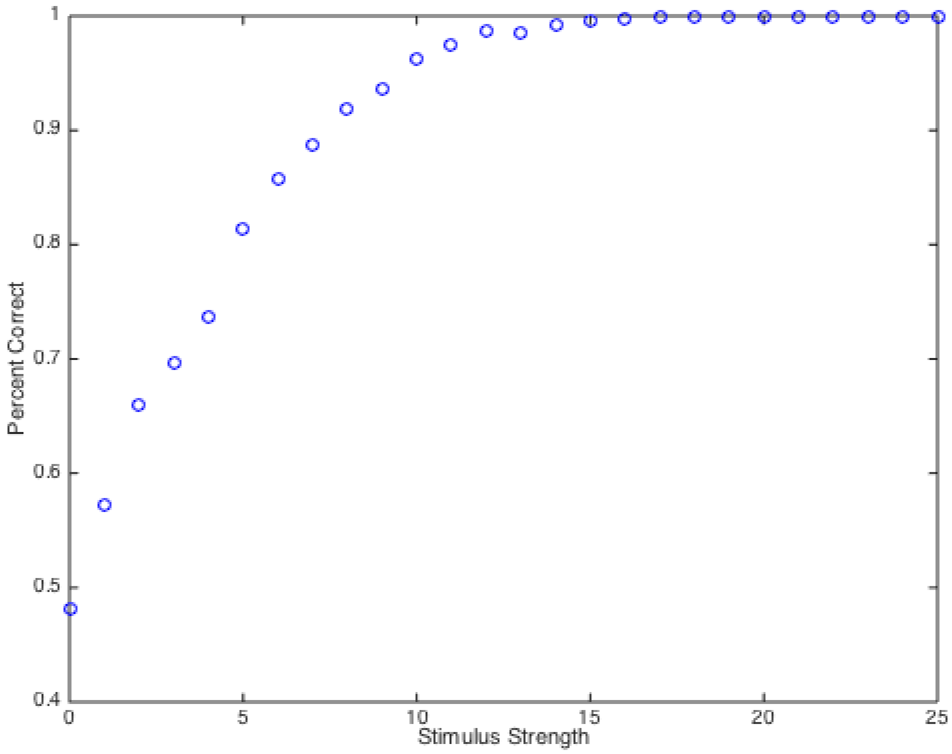
1) (a) Psychometric curve with Poisson noise For all stimulus strengths between 0 and 25, I generated 1000 noise responses and 1000 signal responses. The noise responses were drawn from a Poisson with a mean of 3 and the signal responses were Poisson with a mean of 3 + stimulus strength. The number of correct trials was the number of trials where the stimulus response was greater than the noise response. I then plotted it as a function of stimulus strength.

```
darkLight=3;
stimulusStrengths=[0:25];
nTrials=1000;
numCorrect=zeros(length(stimulusStrengths),1);
for s=stimulusStrengths
    j=s+1;
    % n test trials
    xrand=rand(nTrials,1);
    noiseAloneResponses=poissinv(xrand,darkLight);
    xrand=rand(nTrials,1);
    signalPlusNoiseResponses=poissinv(xrand,darkLight+s);
    % choose correctly when signalPlusNoiseResponse is bigger
    numCorrect(j) = sum(signalPlusNoiseResponses > noiseAloneResponses);
end
percentCorrect=numCorrect/nTrials;
plot(stimulusStrengths,percentCorrect,'ob')
xlabel('Stimulus Strength');
ylabel('Percent Correct');
```



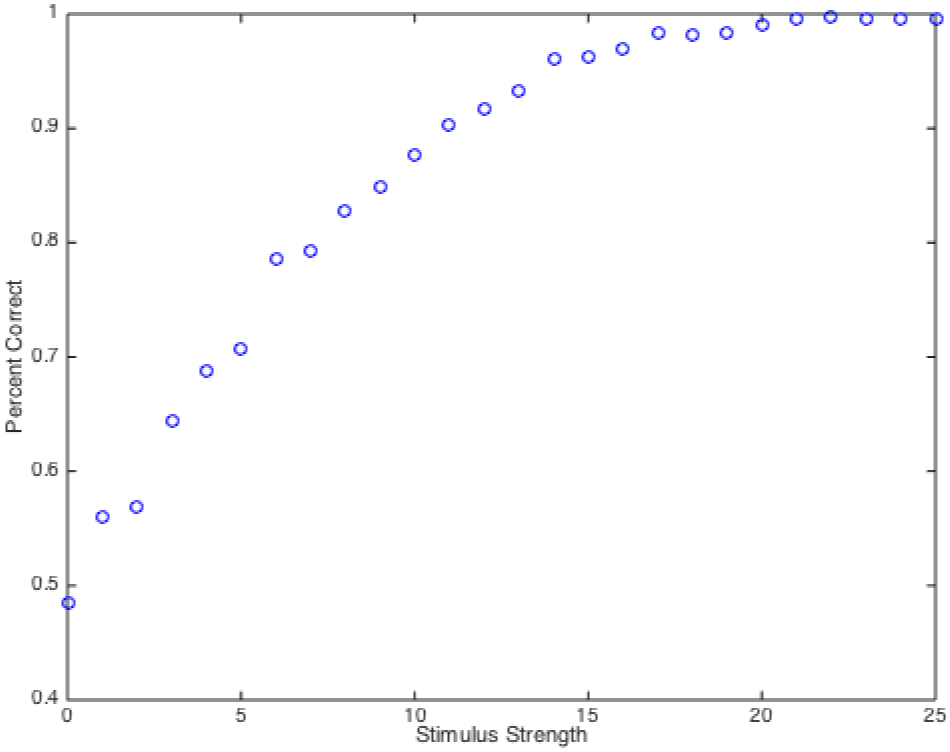
(b) Psychometric curve with Normal noise (variance = 4) Same as above but with responses drawn from a normal with a variance of 4. The means are the same as the Poissons above.

```
numCorrect=zeros(length(stimulusStrengths),1);
sigma = 4;
for s=stimulusStrengths
    j=s+1;
    % n test trials
    xrand=rand(nTrials,1);
    noiseAloneResponses=norminv(xrand, darkLight, sigma);
    xrand=rand(nTrials,1);
    signalPlusNoiseResponses=norminv(xrand, darkLight+s, sigma);
    % choose correctly when signalPlusNoiseResponse is bigger
    numCorrect(j) = sum(signalPlusNoiseResponses > noiseAloneResponses);
end
percentCorrect=numCorrect/nTrials;
plot(stimulusStrengths,percentCorrect,'ob')
xlabel('Stimulus Strength');
ylabel('Percent Correct');
```



Psychometric curve with Normal noise (variance = 6) Same as above but with a variance of 6

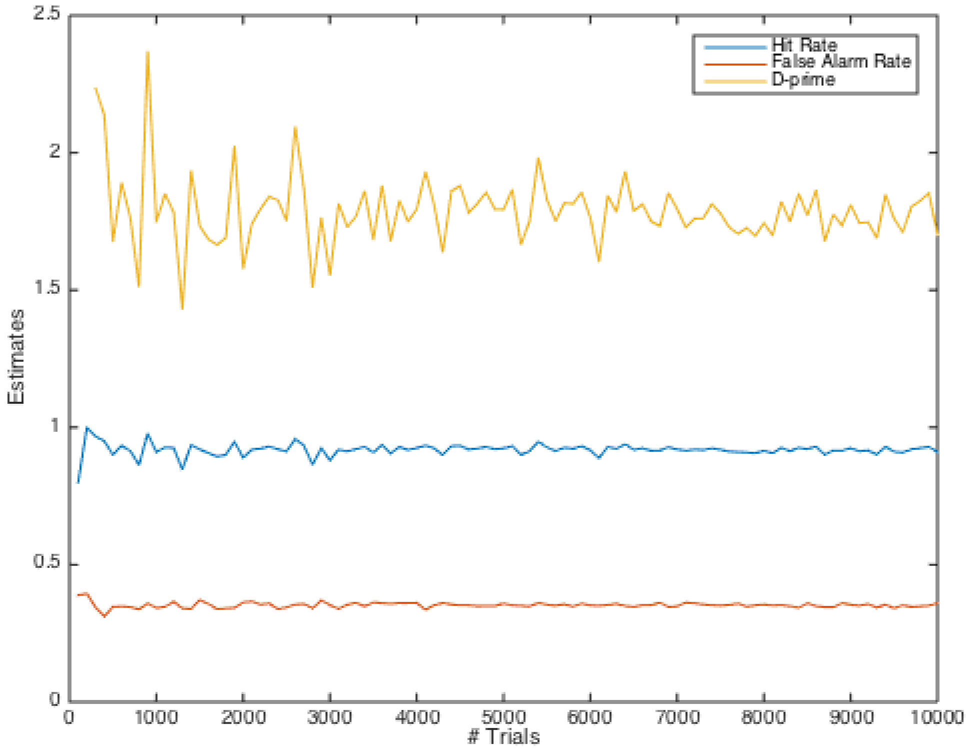
```
numCorrect=zeros(length(stimulusStrengths),1);
sigma = 6;
for s=stimulusStrengths
    j=s+1;
    % n test trials
    xrand=rand(nTrials,1);
    noiseAloneResponses=norminv(xrand, darkLight, sigma);
    xrand=rand(nTrials,1);
    signalPlusNoiseResponses=norminv(xrand, darkLight+s, sigma);
    % choose correctly when signalPlusNoiseResponse is bigger
    numCorrect(j) = sum(signalPlusNoiseResponses > noiseAloneResponses);
end
percentCorrect=numCorrect/nTrials;
plot(stimulusStrengths,percentCorrect,'ob')
xlabel('Stimulus Strength');
ylabel('Percent Correct');
```



2. (a) Hit Rate, False Alarm Rate and D' estimates as a function of experiment length Here we investigate estimated parameters as a function of the number of trials. For experiment lengths of 100-10000 trials, I generated signal and noise responses as above and computed the hit rate and false alarm rate. I also computed d-prime (Φ(HR) - Φ(FAR))

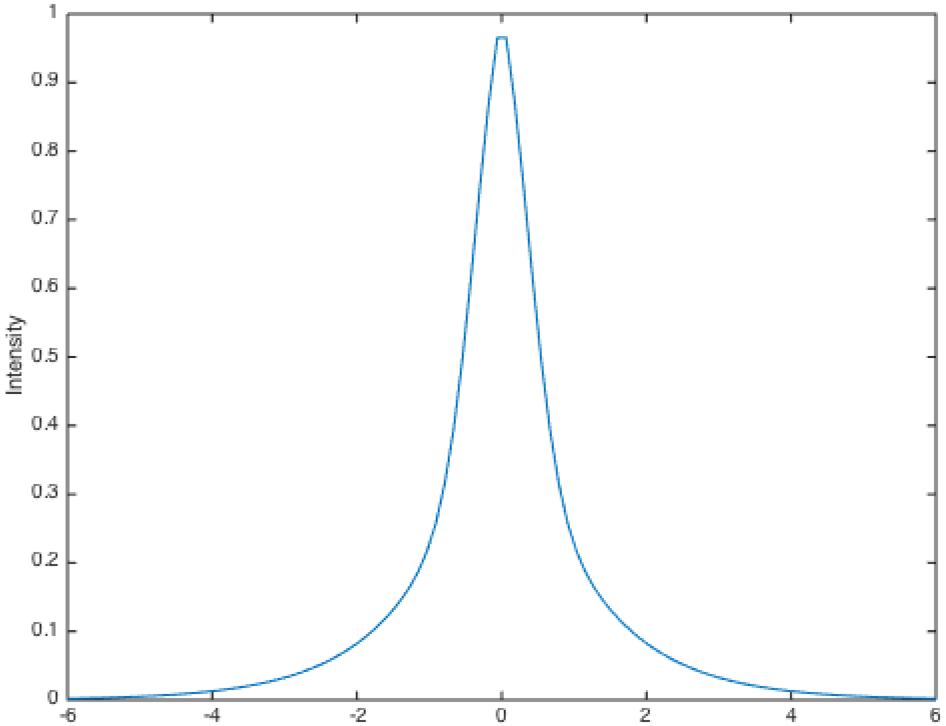
```
trials = linspace(100, 10000, 100);
hitRate=zeros(length(trials),1);
dPrime=zeros(length(trials),1);
falseAlarmRate=zeros(length(trials),1);
j = 0;
s = 4;
for nTrials = trials
    j = j + 1;
    noiseTrials = int16(nTrials * .9);
    xrand=rand(noiseTrials,1);
    noiseAloneResponses=poissinv(xrand,darkLight);
    signalTrials = int16(nTrials * .1);
    xrand=rand(signalTrials,1);
    signalPlusNoiseResponses=poissinv(xrand,darkLight+s);
    criterion=4;
    hitRate(j)=sum(signalPlusNoiseResponses>=criterion)/double(signalTrials);
    falseAlarmRate(j)=sum(noiseAloneResponses>=criterion)/double(noiseTrials);
    dPrime(j) = norminv(hitRate(j), 0, 1) - norminv(falseAlarmRate(j), 0, 1);
end
plot(trials, [hitRate falseAlarmRate dPrime])
legend('Hit Rate', 'False Alarm Rate', 'D-prime');
```

```
xlabel('# Trials');
ylabel('Estimates');
```



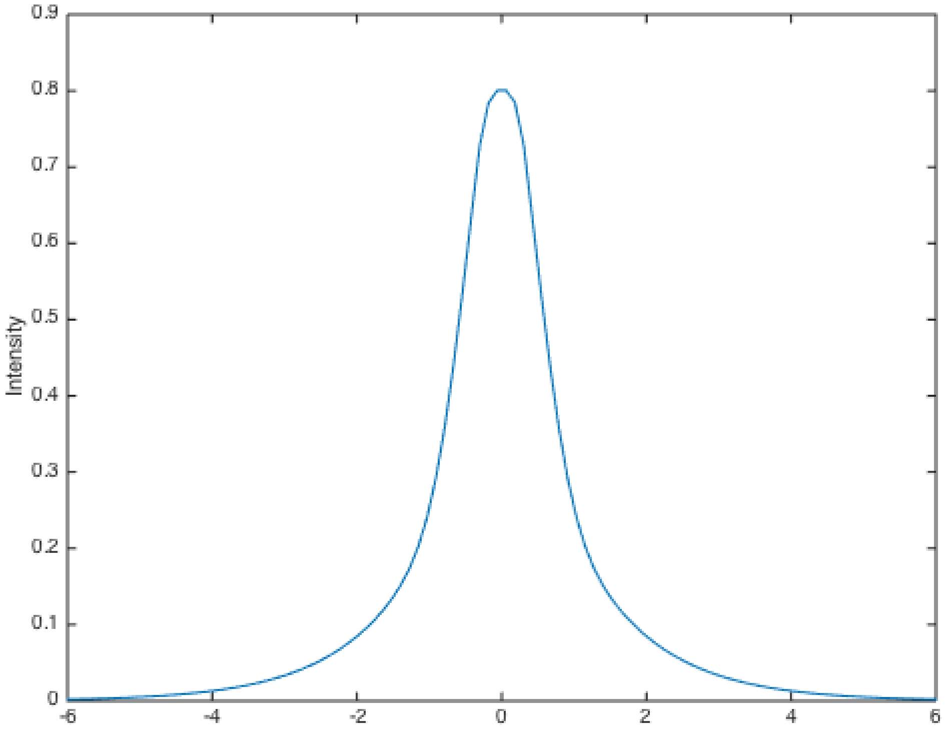
(b) Response to one line As given by the equation from chapter 2

```
i = linspace(-6, 6);
line = 0.47 * exp(-3.3 * power(i, 2)) + 0.53 * exp(-0.93 * abs(i));
plot(i, line)
ylabel('Intensity');
```



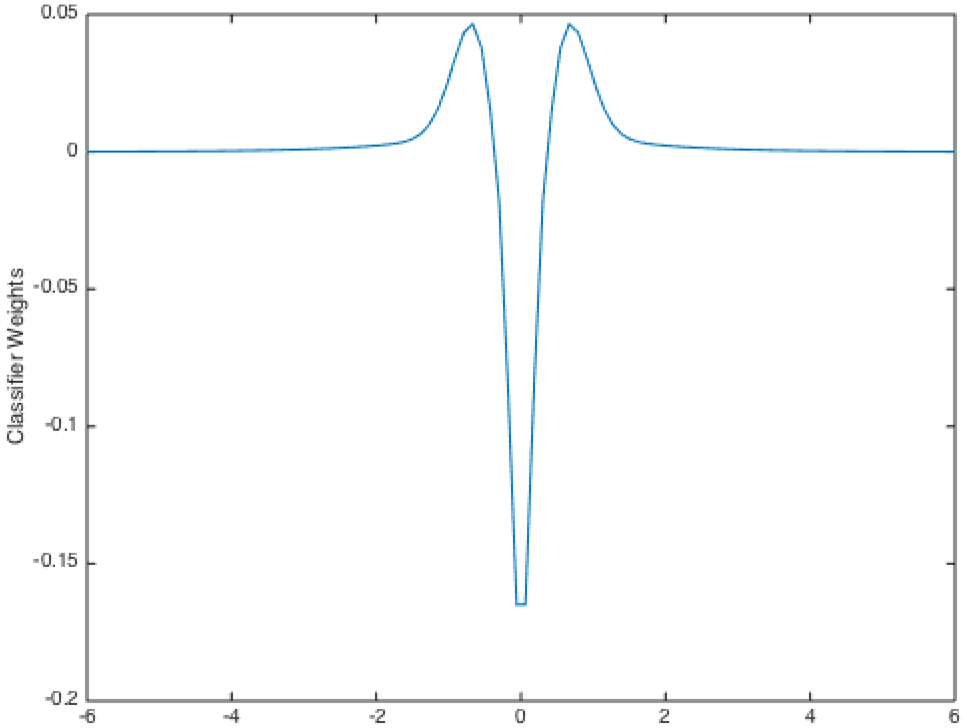
Response to two lines Same as above but shifted by 30 seconds for each line and with a half the gain for each line

```
dist = .5;
li1 = .5 * (0.47 * exp(-3.3 * power(i+(dist/2), 2)) + 0.53 * exp(-0.93 * abs(i+(di
li2 = .5 * (0.47 * exp(-3.3 * power(i-(dist/2), 2)) + 0.53 * exp(-0.93 * abs(i-(di
linePair = li1 + li2;
plot(i, linePair)
ylabel('Intensity');
```



Weights for Bayesian classifier Optimal decision variable is the response to the pair of lines - response to a single line

```
plot(i, linePair-line)
ylabel('Classifier Weights');
```



Published with MATLAB® R2014b