```
%For this data I imported the NB_RT matlab file from the F folder in
%data_mgl

%Here I am setting my requirements for the trials I want to include - the
%first part states that the trials must be correct trials (percorrSum = 1)
%and then the second part states that the reaction times in these trials
%must be at least 0 but not greater than 1.2
expressCutoff = 0.0
```

```
expressCutoff = 0
```

```
Ltrials = percorrSum == 1 & tRxnSum > expressCutoff(1) & tRxnSum < 1.2
```

```
Ltrials = 1×4050 logical array
   0   0   0   1   0   1   1   1   1   0   1   1   1   0   1   1   1   1   1   1···
```

```


%This is code for me to specify the reaction times of the trials that fit
%the conditions I defined above
data_ = { ...
    tRxnSum(Ltrials & numdirSum == -1 & labelSum == 1), ...
    tRxnSum(Ltrials & numdirSum == -1 & labelSum ~= 1), ...
    tRxnSum(Ltrials & numdirSum ==  1 & labelSum == 1), ...
    tRxnSum(Ltrials & numdirSum ==  1 & labelSum ~= 1)};
RTs = data{1}
```

```
RTs = 1×3563
    0.2175    0.2256    0.2855    0.3033    0.2721    0.2325    0.3382    0.2272···
```

```
% Step 2 of the exercise is to define the objective function which we will
% define as the negative log-likelihood
laterErrFcn = @(fits) -sum(log(normpdf(1./RTs, fits(1)./fits(2), 1./fits(2))));
```

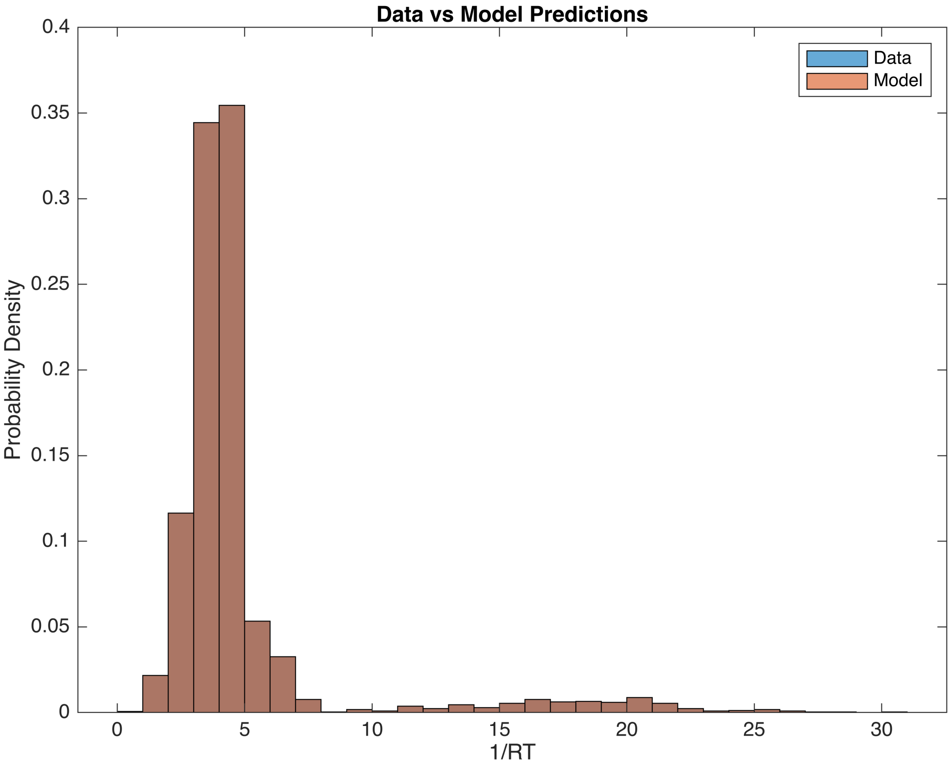
```
% Step 3 of the exercise is to define our initial conditions in which we
% will use mean and standard deviation of the reciprocal RTs
reciprocalRTs = 1./RTs;
initial_muR = mean(reciprocalRTs);
initial_deltaS = 1/std(reciprocalRTs);
initialValues = [initial_muR, initial_deltaS];
```

```
% I was having issues running my code because I had not defined the
% variable fits
fits = zeros(1,2);  % Assuming you're fitting two parameters, muR and deltaS
```

```
% Step 5 of the exercise is to plot our data in order to compare it to our model predi
histogram(1./RTs, 'Normalization', 'pdf'); % Observed data
hold on;
```

```
x_vals = linspace(min(1./RTs), max(1./RTs), 100);
y_vals = normpdf(x_vals, fits(1)./fits(2), 1./fits(2));

plot(x_vals, y_vals, 'r', 'LineWidth', 2); % Model predictions
xlabel('1/RT');
ylabel('Probability Density');
legend('Data', 'Model');
title('Data vs Model Predictions');
hold off;
```



```
%I guess in this case it looks like a complete overlap of the model and the
%actual data? Not entirely sure if that is correct to be honest but I was
%at least able to plot both the model and actual data on the same graph
```