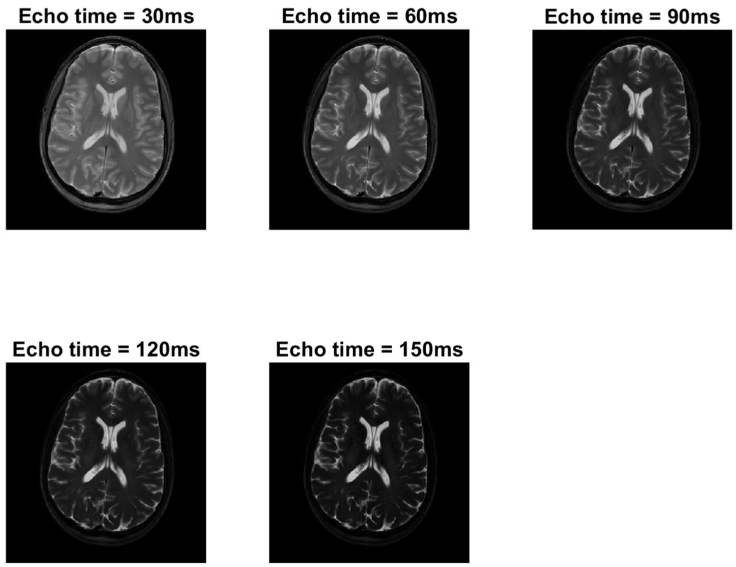
Project 1

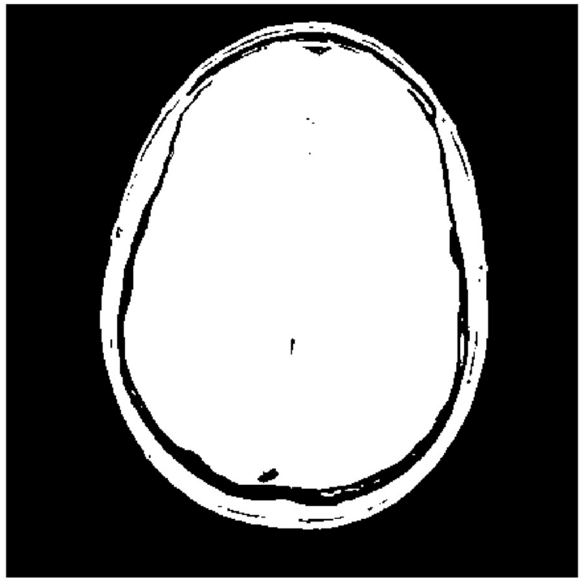
Part I: Mapping T2 Relaxation Times

```
clc
clear
load('proj1aData_QFI')

[nRows, nCols, nTe] = size(image_3d) ;

figure
subplot(2,3,1)
imagesc(squeeze(image_3d(:,:,1)))
intLimits_v = get(gca,"CLim");
axis image
axis off
colormap(gray)
title(['Echo time = ', num2str(te_v(1)*1000),'ms'])
for index = 2:nTe
    subplot(2,3,index)
    imagesc(squeeze(image_3d(:,:,index)), intLimits_v)
    axis image
    axis off
    title(['Echo time = ', num2str(te_v(index)*1000),'ms'])
end
```



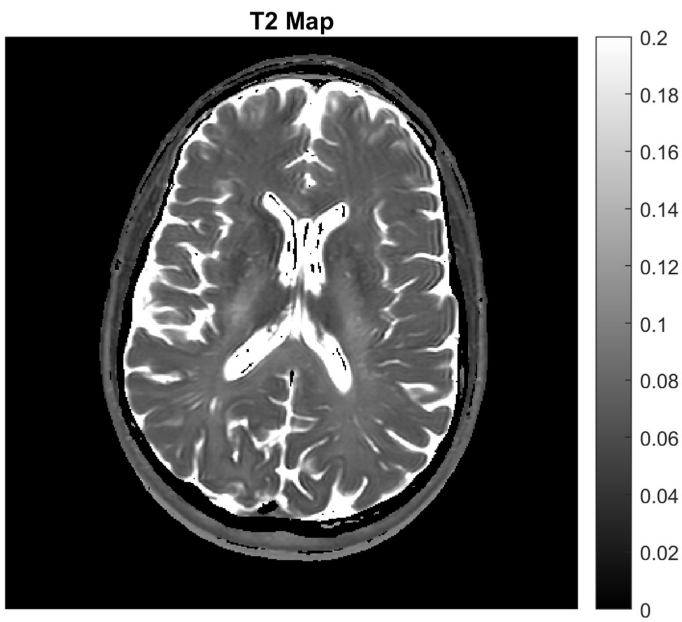

```
% 2. Use the magnitude of the first image to form a "binary mask" matrix

image_1 = squeeze(image_3d(:,:,1)) ;

mask_1 = (image_1 > 0.1 * max(image_1(:))) ;
figure
imagesc(mask_1)
colormap(gray)
axis image
axis off
title("Head mask")
```

Head mask



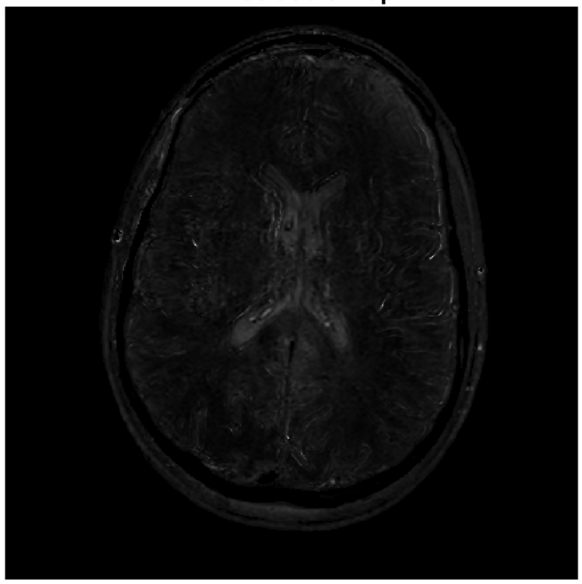
```
% 3. For each pixel in the head, calculate the time constant of the
% decaying signal.

t2_m = zeros(nRows, nCols) ;
s0_m = zeros(nRows, nCols) ;
norm_m = zeros(nRows, nCols) ;

for row = 1:nRows
    for col = 1:nCols
        if (mask_1(row,col)==1)
            signal_v = squeeze(image_3d(row,col,:)) ;
            coeff_v = polyfit(te_v, log(signal_v), 1) ;
            slope = coeff_v(1) ;
            logS0 = coeff_v(2) ;    % intercept
            S0 = exp(logS0) ;
            t2 = -1/slope ;
            % store results:
            t2_m(row,col) = min(t2,5) ;         % from class
            t2_m(row,col) = t2 ;
            s0_m(row,col) = S0 ;
            % calculate norm for each pixel:
            norm_m(row,col) = norm(signal_v-S0.*exp(-te_v./t2)) ;
        end
    end
end
```

```
% display the T2 map with values up to 0.2 seconds
imagesc(t2_m, [0,0.2])
title('T2 Map')
axis image
axis off
colorbar
```




```
% t2 residual map:

figure
imagesc(norm_m)
colormap(gray)
axis image
axis off
title("T2 Residuals Map")
```

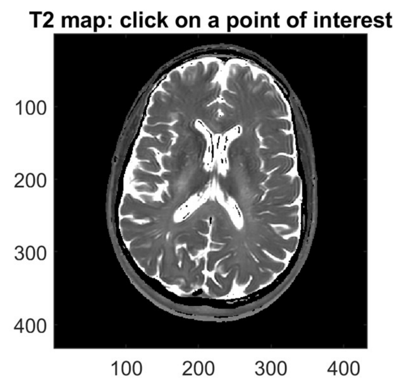
T2 Residuals Map



```
% 4. Make a new figure with two subplots. In the first show the T2 map:

% View fit:
figure
subplot(1, 2, 1)
t2Max = 0.2;                        % 0.2 s maximum value for display.
t2ceil_m = min(t2_m, t2Max*ones(nRows, nCols));
red0_m = t2ceil_m/t2Max;
green0_m = t2ceil_m/t2Max;
blue0_m = t2ceil_m/t2Max;
color_3d = cat(3, red0_m, green0_m, blue0_m);
image(color_3d)
axis image
title('T2 map: click on a point of interest')

% Construct a loop that allows the user of your program to select an
% arbitrary number of pixels to examine. For each selected pixel, plot the
% signal intensity versus TE along with the best-fit curve.

nPoints = 50;
contFlag = 1;
while(contFlag==1)

    % Get position of mouse-click on image:
    [x,y] = ginput(1);
    row = round(y);
```

```
    col = round(x);
    % Exit loop if the mouse click is outside the image:
    if (row<1 | row>nRows | col<1 | col>nCols)
        contFlag = 0;
        continue
    end
    red_m = red0_m;
    green_m = green0_m;
    blue_m=blue0_m;
    % Show the position of the pixel in red:
    red_m(row,col)=1;
    green_m(row,col)=0;
    blue_m(row,col)=0;
    color_3d=cat(3,red_m,green_m,blue_m);
    image(color_3d)
    axis image
    title('T2 map: click on a point of interest')

    % Show fit:
    t2 = t2_m(row,col);
    S0 = s0_m(row,col);
    s_v = squeeze(image_3d(row,col,:));
    % create an array of nPoints TE values from the minimum to maximum TE:
    tiFit_v = linspace(min(te_v),max(te_v),nPoints);
    % find corresponding signal at each TE using estimates of T2 and s0:
    sFit_v = S0.*exp(-tiFit_v./t2) ;
    S0
    subplot(1,2,2)
    plot(tiFit_v,sFit_v,':',te_v,s_v,'+')
    title(['At row =', num2str(row),', col = ',num2str(col),': T2 = ', ...
        num2str(t2*1000),'ms'])
end
```

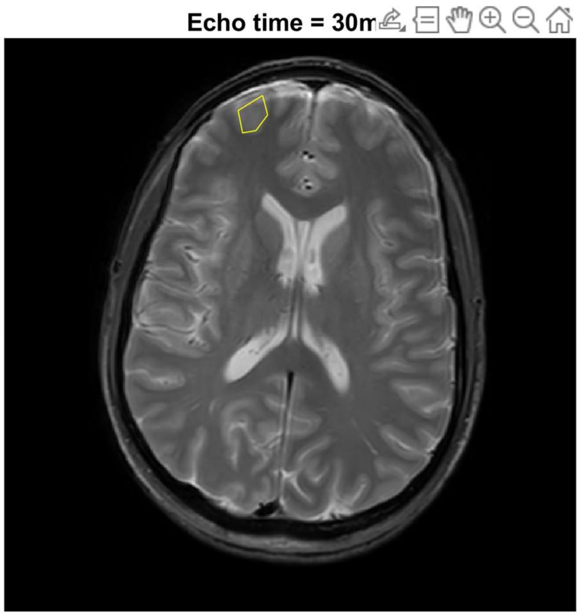
```
S0 = 1.2594e+04
```



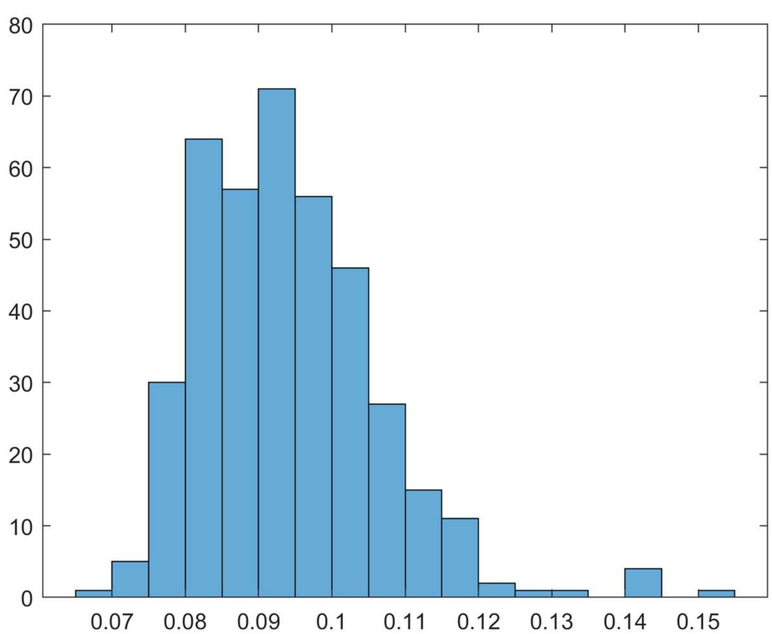

```
Error using ginput (line 84)
Interrupted by figure deletion
```

```
% 5. Display the T2 data as a gray scale map in a new figure.

image(color_3d)
axis image
axis off
colormap(gray)
datacursormode('on')
```

```
% 6. Identify the gray matter and white matter in the first image (shortest
% TE image). Look at examples on Google Images.

% 7. Ise the roipoly command to draw a region of interest (ROI) in each
% tissue (one ROI for gray matter, another for white matter). Make the
% regions as large as you can without including the other tissue type.

% plot first image again & draw polygon around gray matter

[nRows, nCols, nTe] = size(image_3d) ;

figure
image1 = imagesc(squeeze(image_3d(:,:,1))) ;
intLimits_v = get(gca,"CLim");
axis image
axis off
colormap(gray)
title(['Echo time = ', num2str(te_v(1)*1000),'ms'])

[mask_gray, x_v, y_v] = roipoly ;
line(x_v, y_v, 'Color', 'y') ;
```

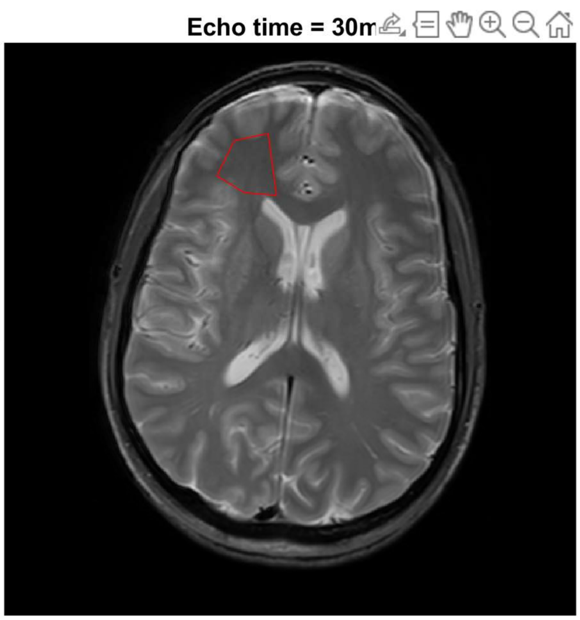



```
% Make a histogram of T2 values in each region and draw the ROI that the
% histogram represents. Find the mean and standard deviation of T2 for gray
% and white matter.
histogram(t2_m(mask_gray))
```

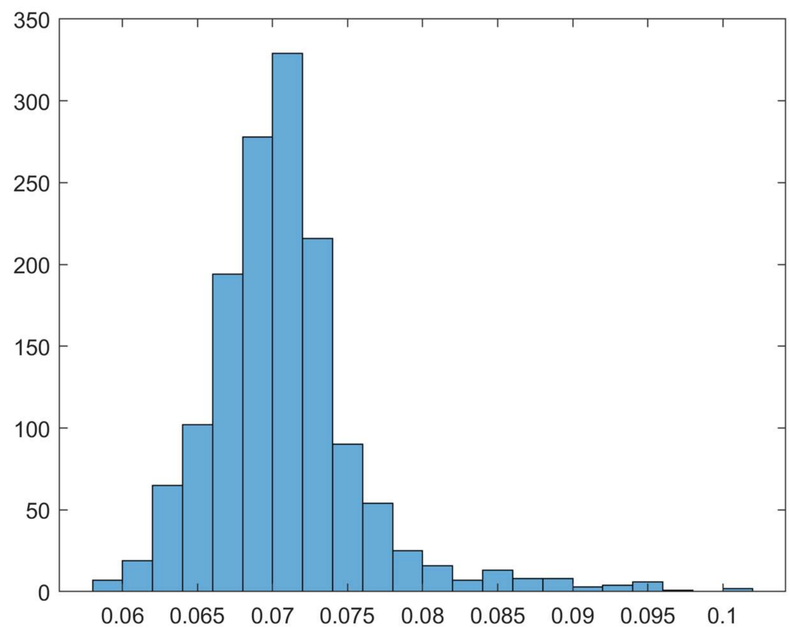



```
% plot first image again & draw polygon around white matter
```

```
[nRows, nCols, nTe] = size(image_3d) ;

figure
image1 = imagesc(squeeze(image_3d(:,:,1))) ;
intLimits_v = get(gca,"CLim");
axis image
axis off
colormap(gray)
title(['Echo time = ', num2str(te_v(1)*1000),'ms'])
datacursormode('on')

[mask_white, x_v, y_v] = roipoly ;
line(x_v, y_v, 'Color', 'r') ;
```




```
% Make a histogram of T2 values in each region and draw the ROI that the
% histogram represents.
histogram(t2_m(mask_white))
```




```
% Find the mean and standard deviation of T2 for gray
% and white matter.

t2_gray = t2_m(mask_gray) ;
mean_gray = mean(t2_gray(:))
```

```
mean_gray = 0.0942
```

```
std_gray = std(t2_gray(:))
```

```
std_gray = 0.0121
```

```
t2_white = t2_m(mask_white) ;
mean_white = mean(t2_white(:))
```

```
mean_white = 0.0707
```

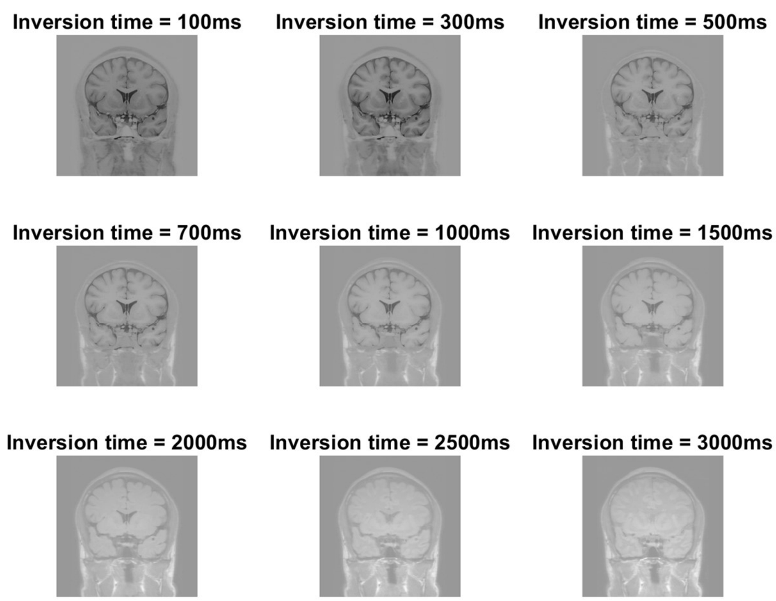
```
std_white = std(t2_white(:))
```

```
std_white = 0.0052
```

Part II: Mapping T1 Relaxation Times

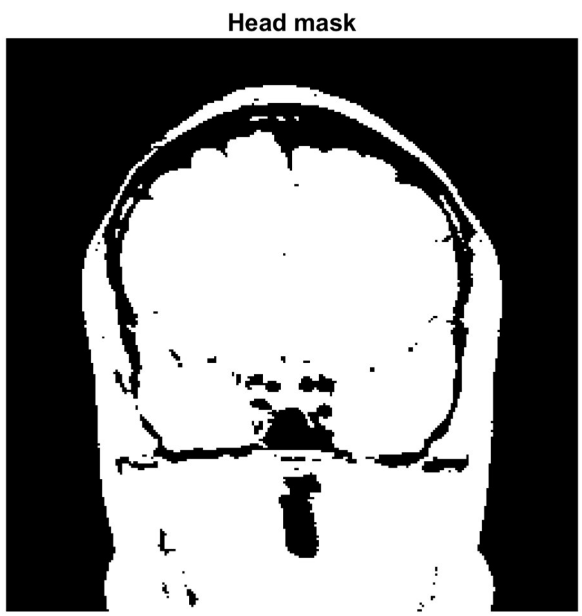
```
load('proj1bData_QFI')

[nRows, nCols, nTi] = size(irImage_3d) ;

% Display all inversion recovery images in one figure
```

```
figure
subplot(3,3,1)
intLimits_v = [min(irImage_3d(:)), max(irImage_3d(:))];
imagesc(squeeze(irImage_3d(:,:,1)), intLimits_v)
axis image
axis off
colormap(gray)
title(['Inversion time = ', num2str(ti_v(1)*1000),'ms'])
for index = 2:nTi
    subplot(3,3,index)
    imagesc(squeeze(irImage_3d(:,:,index)), intLimits_v)
    axis image
    axis off
    title(['Inversion time = ', num2str(ti_v(index)*1000),'ms'])
end
```



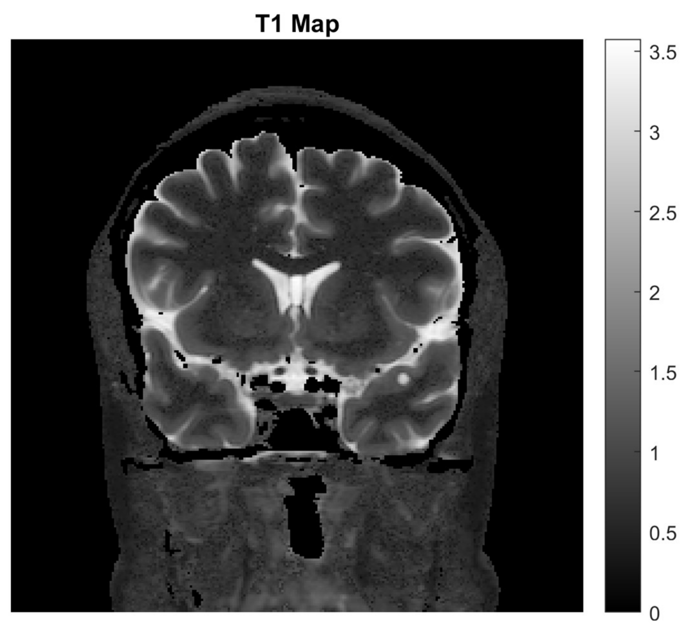

```
% 3. Estimate the T1 value for each pixel in the head and display the T1
% map for the entire slice. Also, calculate the residual of the fit (norm).
% Display your T1 and residual maps.

% first, make mask:

image_1 = squeeze(irImage_3d(:,:,end)) ;
```

```
mask_1 = (image_1 > 0.2 * max(image_1(:))) ;
figure
imagesc(mask_1)
colormap(gray)
axis image
axis off
title("Head mask")
```



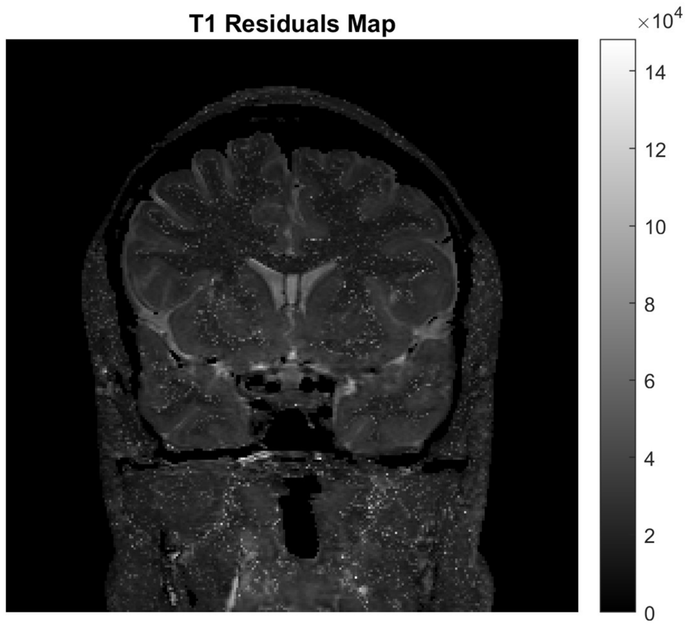
```
% calculate T1 for each pixel:

t1_m = zeros(nRows, nCols) ;
norm_m = zeros(nRows, nCols) ;

for row = 1:nRows
    for col = 1:nCols
        if (mask_1(row,col)==1)
            mz_v = squeeze(irImage_3d(row,col,:)) ;
            m0 = m0_m(row,col);
            inds =  find(mz_v < m0) ;
            coeff_v = polyfit(ti_v(inds), -log(0.5.*(1-(mz_v(inds)./m0))), 1) ;
            slope = coeff_v(1) ;
            t1 = 1/slope ;
```

```
            % store results:
            t1_m(row,col) = t1 ;

            % calculate norm for each pixel:
            norm_m(row,col) = norm(mz_v-m0.*(1-2.*exp(-ti_v./t1))) ;
        end
    end
end

% display the T1 for the entire slice
figure
imagesc(t1_m)
title('T1 Map')
axis image
axis off
colorbar
colormap(gray)
```




```
% display the residuals map
figure
imagesc(norm_m)
title('T1 Residuals Map')
axis image
```

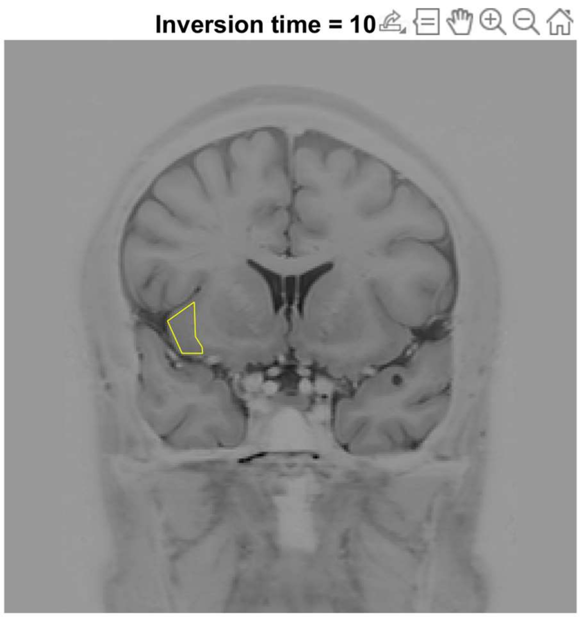
```
axis off
colorbar
colormap(gray)
```



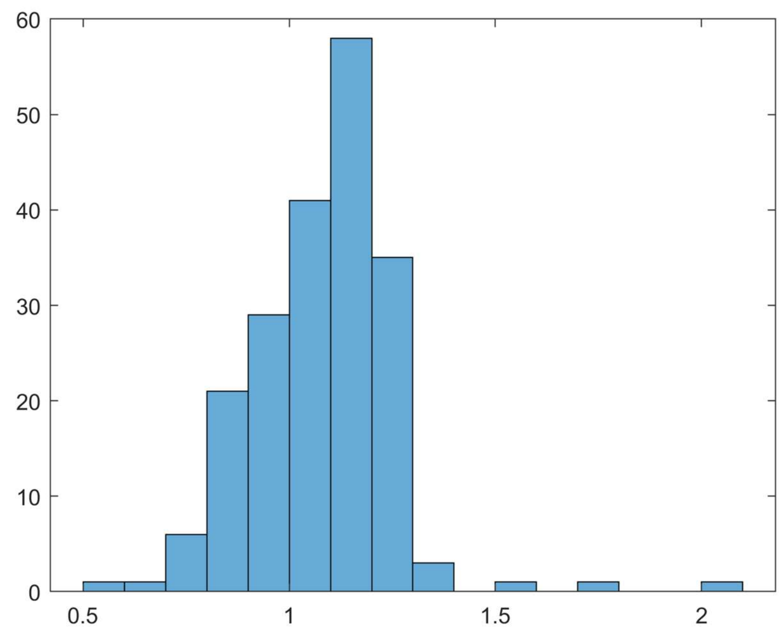

```
% 4. Add a loop to allow the user to show the T1 fit for an arbitrary
% number of pixels. For each selected pixel, plot the measured Mz versus T1
% along with the modeled curve using estaimate of T1.

% View fit:
figure
subplot(1, 2, 1)
t1Max = max(max(t1_m));
t1ceil_m = min(t1_m, t1Max*ones(nRows, nCols));
red0_m = t1ceil_m/t1Max;
green0_m = t1ceil_m/t1Max;
blue0_m = t1ceil_m/t1Max;
color_3d = cat(3, red0_m, green0_m, blue0_m);
image(color_3d)
axis image
title('T1 map: click on a point of interest')

% Construct a loop that allows the user of your program to select an
% arbitrary number of pixels to examine. For each selected pixel, plot the
% signal intensity versus TE along with the best-fit curve.
```

```
nPoints = 50;
contFlag = 1;
while(contFlag==1)

    % Get position of mouse-click on image:
    [x,y] = ginput(1);
    row = round(y);
    col = round(x);
    % Exit loop if the mouse click is outside the image:
    if (row<1 | row>nRows | col<1 | col>nCols)
        contFlag = 0;
        continue
    end
    red_m = red0_m;
    green_m = green0_m;
    blue_m=blue0_m;
    % Show the position of the pixel in red:
    red_m(row,col)=1;
    green_m(row,col)=0;
    blue_m(row,col)=0;
    color_3d=cat(3,red_m,green_m,blue_m);
    image(color_3d)
    axis image
    title('T1 map: click on a point of interest')

    % Show fit:
    t1 = t1_m(row,col);
    m0 = m0_m(row,col);
    mz_v = squeeze(irImage_3d(row,col,:));
    % create an array of nPoints Ti values from the minimum to maximum Ti:
    tiFit_v = linspace(min(ti_v),max(ti_v),nPoints);
    % find corresponding signal at each Ti using estimate of T1:
    mzFit_v = m0 .* (1-2.*exp(-tiFit_v./t1)) ;
    subplot(1,2,2)
    plot(tiFit_v,mzFit_v,':',ti_v,mz_v,'+')
    title(['At row =', num2str(row),', col = ',num2str(col),': T1 = ', ...
        num2str(t1*1000),'ms'])
end
% 5. Display the T1 map in a new figure. Use roipoly to define regions of
% interest in the gray matter and white matter. Plot histograms of the T1
% distributions in gray and white matter and show the ROIs used to extract
% the histogram data. Find the mean and standard deviation of T1 for gray
% and white matter.

figure
```

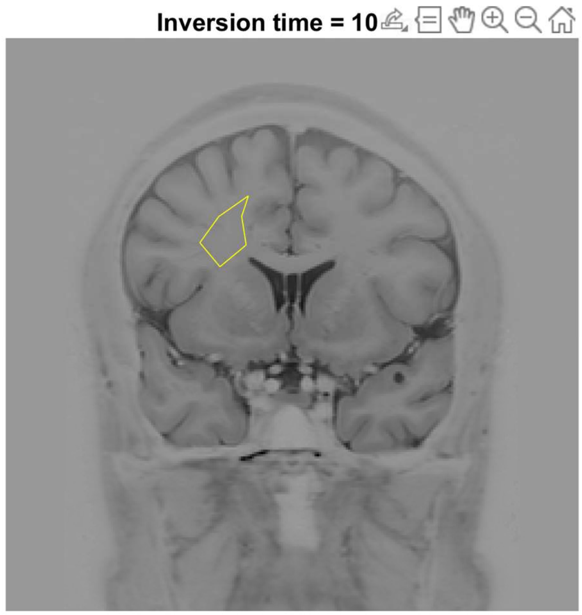
```
intLimits_v = [min(irImage_3d(:)), max(irImage_3d(:))];
imagesc(squeeze(irImage_3d(:,:,1)), intLimits_v)
axis image
axis off
colormap(gray)
title(['Inversion time = ', num2str(ti_v(1)*1000),'ms'])

[mask_gray, x_v, y_v] = roipoly ;
line(x_v, y_v, 'Color', 'y') ;
```



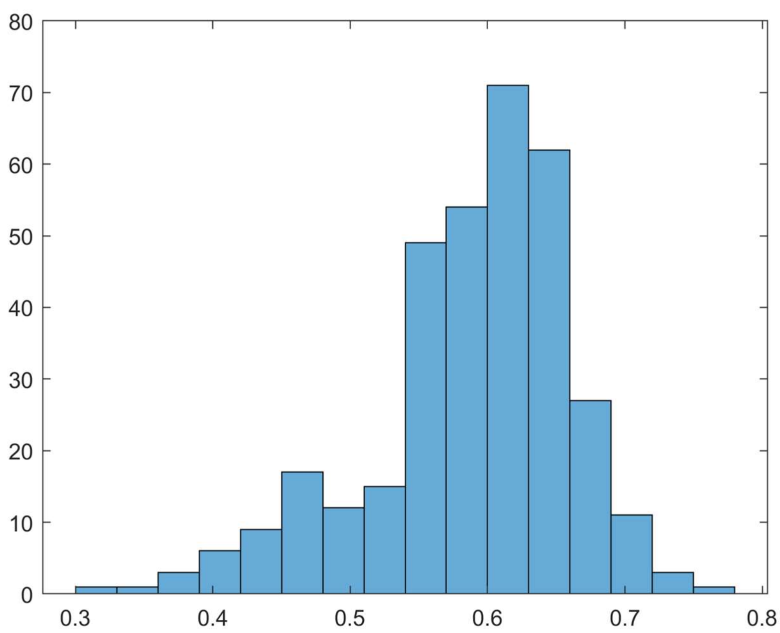

```
histogram(t1_m(mask_gray))
```




```
% define region of interest in white matter

figure
intLimits_v = [min(irImage_3d(:)), max(irImage_3d(:))];
imagesc(squeeze(irImage_3d(:,:,1)), intLimits_v)
axis image
axis off
colormap(gray)
title(['Inversion time = ', num2str(ti_v(1)*1000),'ms'])

[mask_white, x_v, y_v] = roipoly ;
line(x_v, y_v, 'Color', 'y') ;
```




```
histogram(t1_m(mask_white))
```




```
% Find the mean and standard deviation of T1 for gray
% and white matter.

t1_gray = t1_m(mask_gray) ;
```

```
mean_gray = mean(t1_gray(:))
```

```
mean_gray = 1.0776
```

```
std_gray = std(t1_gray(:))
```

```
std_gray = 0.1707
```

```


t1_white = t1_m(mask_white) ;
mean_white = mean(t1_white(:))
```

```
mean_white = 0.5885
```

```
std_white = std(t1_white(:))
```

```
std_white = 0.0734
```