10/6/2022

Project 1

BME 7450

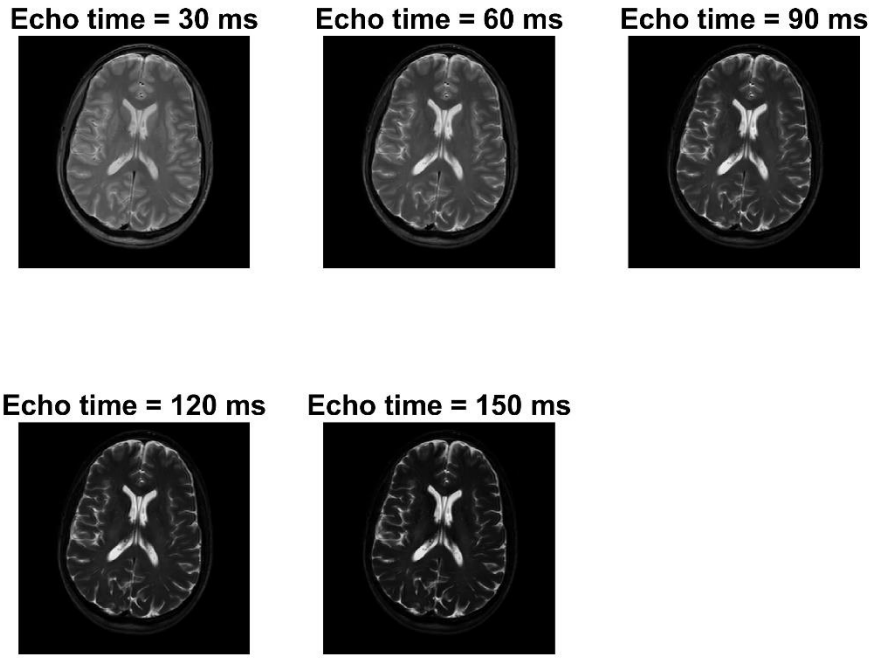
Submitted by,

Rana Mozumder

DEPARTMENT OF BME, VANDERBILT UNIVERSITY

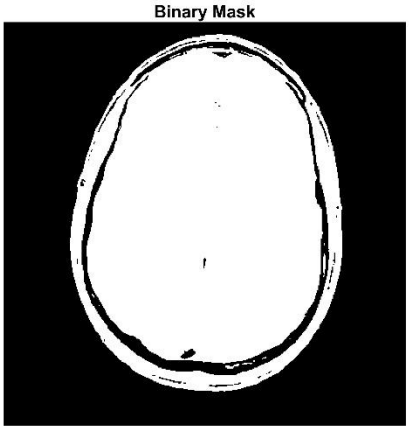
VANDERBILT UNIVERSITY

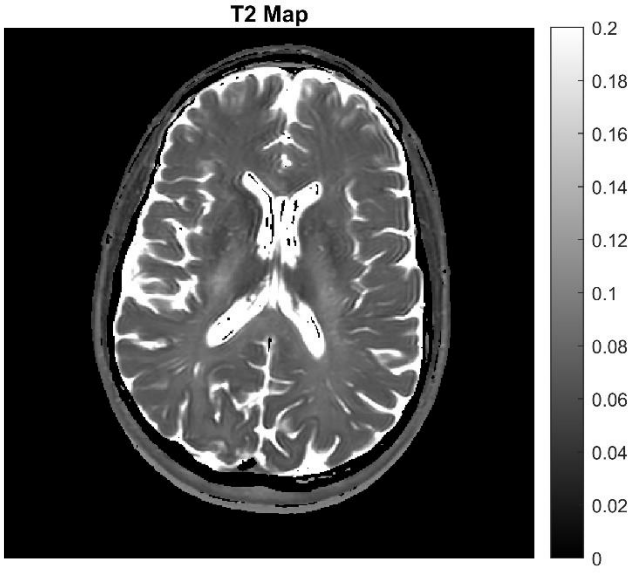
1.



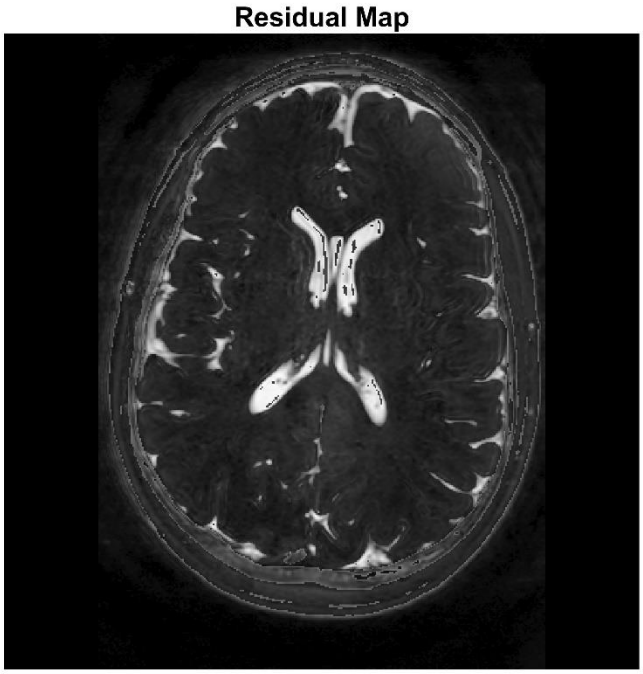
All images are scaled to the lowest and highest intensity of the first image (TE=30 ms).

2.



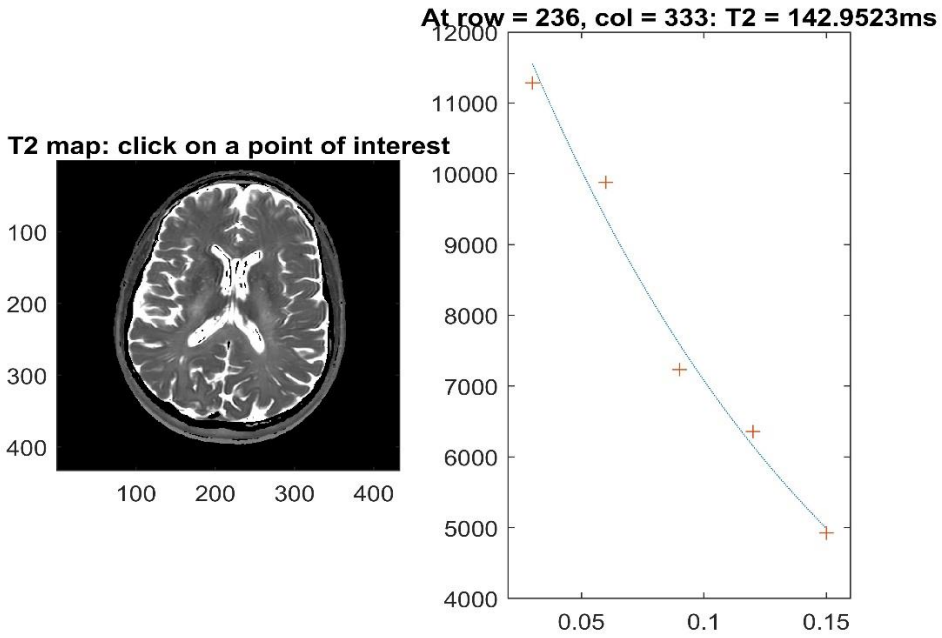


3.

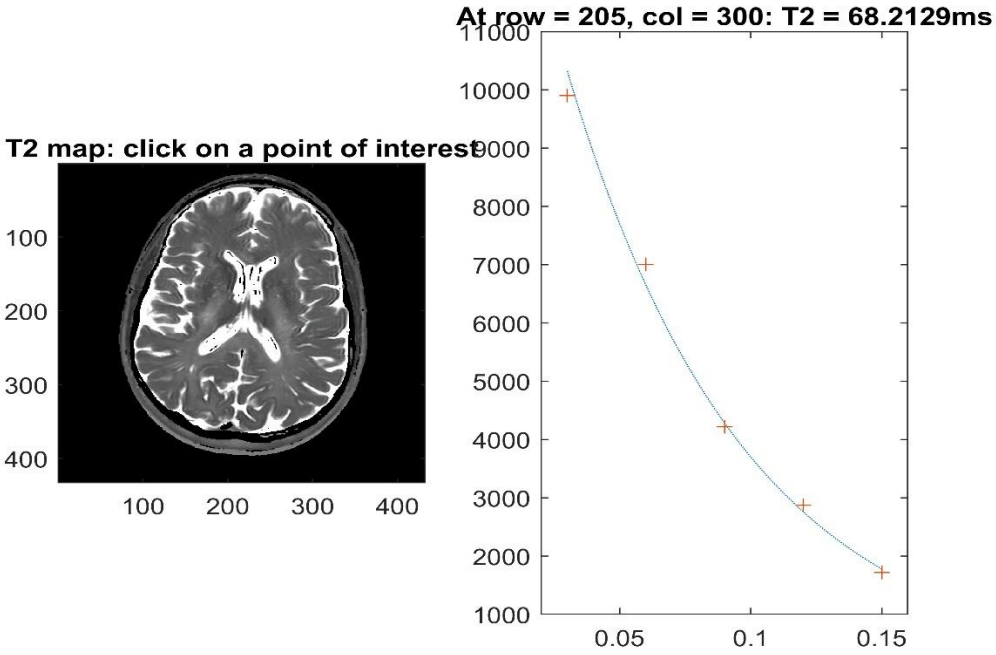


4.

T2 fit for a gray matter pixel



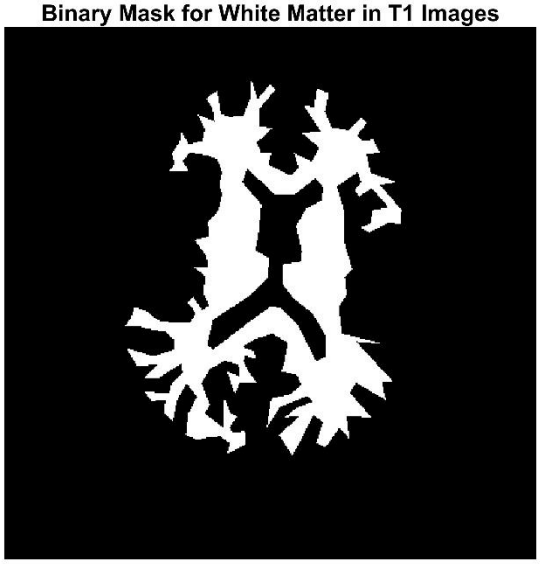
T2 fit for a white matter pixel



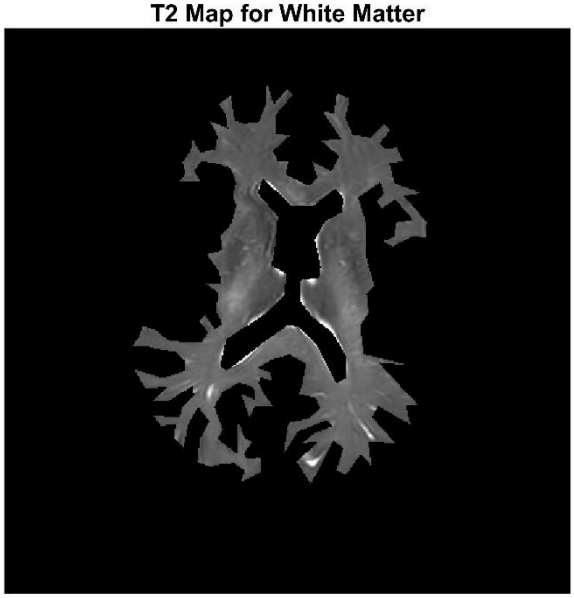
5.

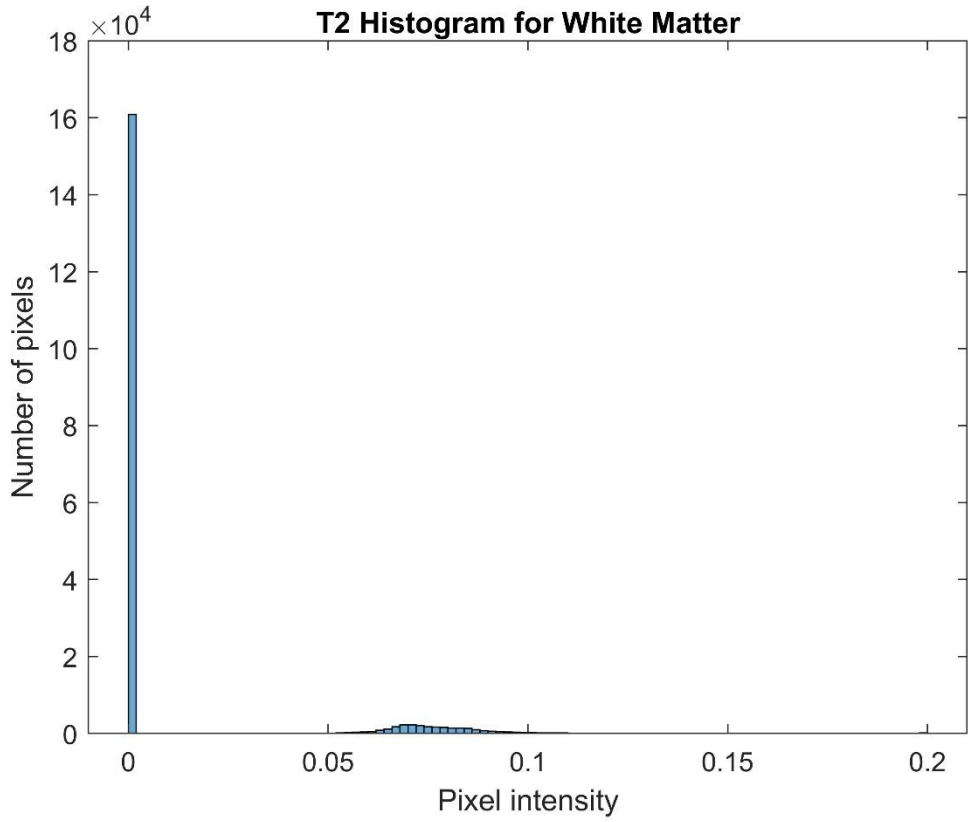
For White Matter:

Binary Mask for White Matter in T1 Images

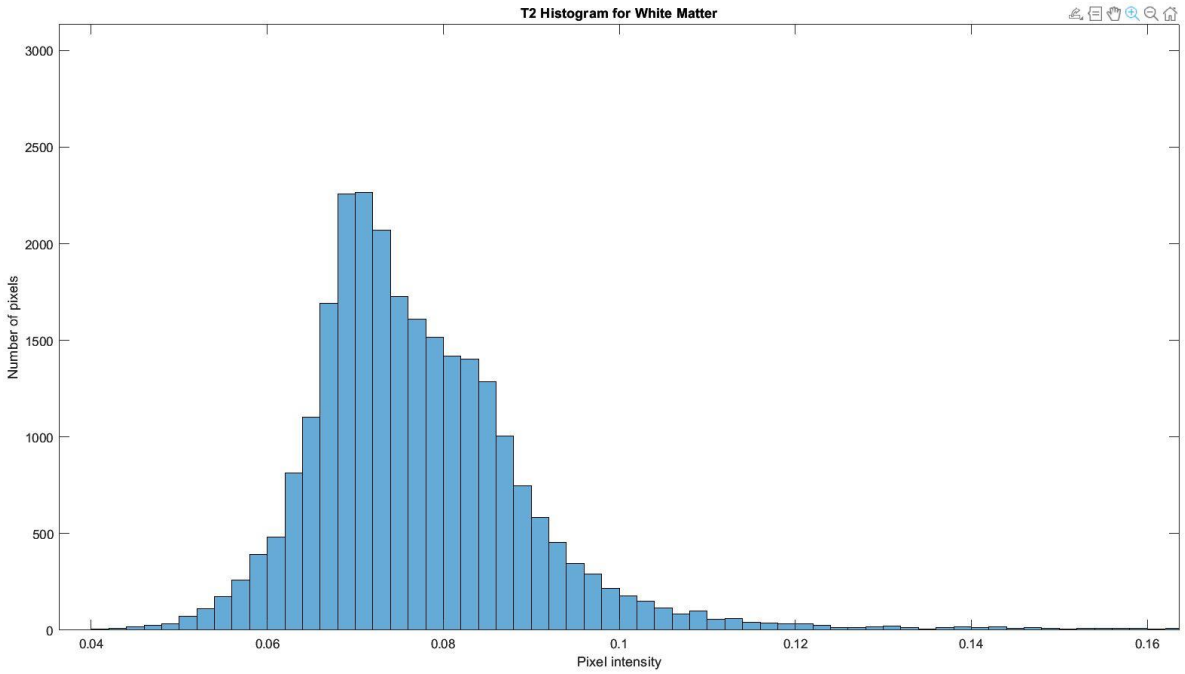


T2 Map for White Matter





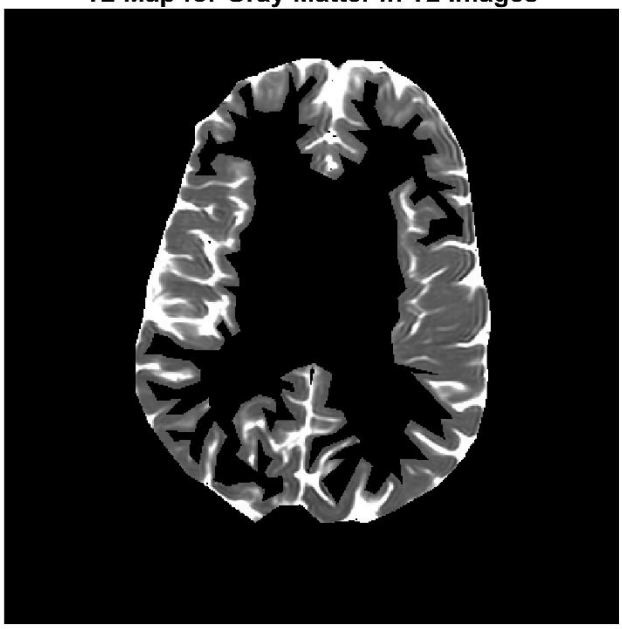
Zoomed version:

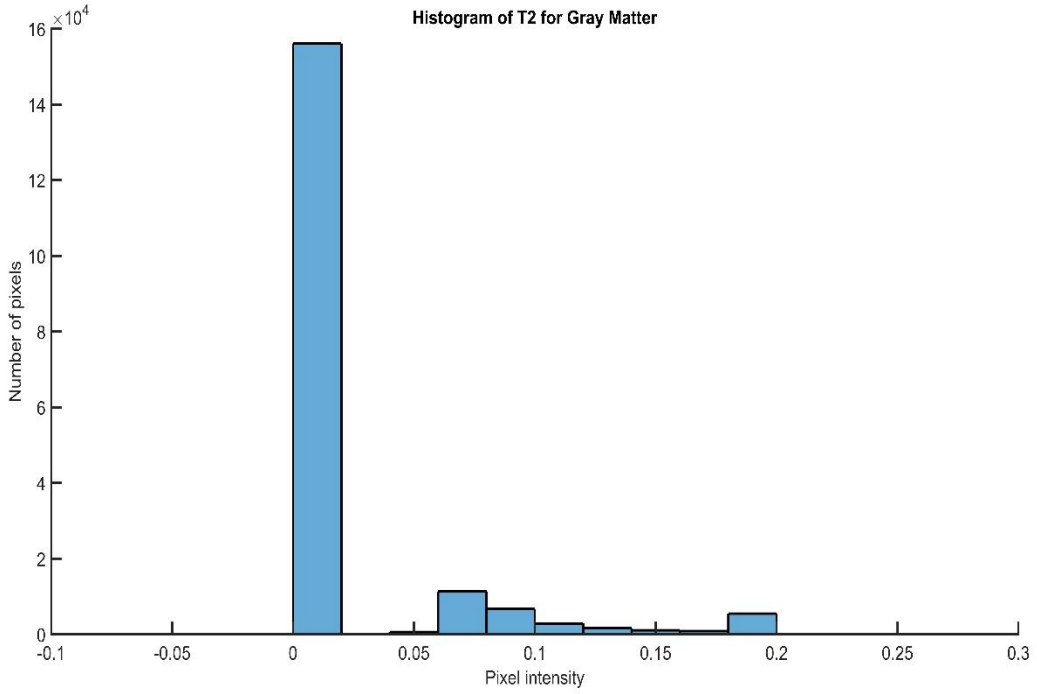


For Gray Matter:

Binary Mask for Gray Matter in T2 Images

T2 Map for Gray Matter in T2 Images





Corresponding Matlab Code:

```
clc; close all;

load('proj1aData_QFI');
[nRows, nCols, nTe] = size(image_3d);

% plotting all the TI images in the same intensity scle
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

% creating a binary mask
```

```
image_m = squeeze(image_3d(:, :, 1));
mask_m = (image_m>0.1*max(image_m(:)));

figure
imagesc(mask_m)
colormap(gray)
axis image
axis off
title('Binary Mask')

% creating T2 map
t2_m = zeros(nRows, nCols);
s0_m = zeros(nRows, nCols);
for row=1:nRows
    for col=1:nCols
        if (mask_m(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.
             t2 = -1 / slope;
            t2_m(row, col) = t2;
            s0_m(row, col) = exp(logS0);
        end
    end
end

for row=1:nRows
    for col = 1:nCols
        if t2_m(row, col)>0.2
            t2_m(row,col)=0.2;
        end
    end
end

figure
imagesc(t2_m, [0, 0.2])
colormap(gray)
colorbar

% creating residual map
res=zeros(nRows,nCols);
for row=1:nRows
    for col = 1:nCols
        %%%may need to change
        signal_res_v = squeeze(image_3d(row, col, :));
        res(row, col)=sqrt(norm(signal_res_v ...
            - s0_m(row,col)*exp(-te_v./t2_m(row,col))));
    end
end

figure
imagesc(res)
intLimits_v = get(gca, 'CLim');
axis image
```

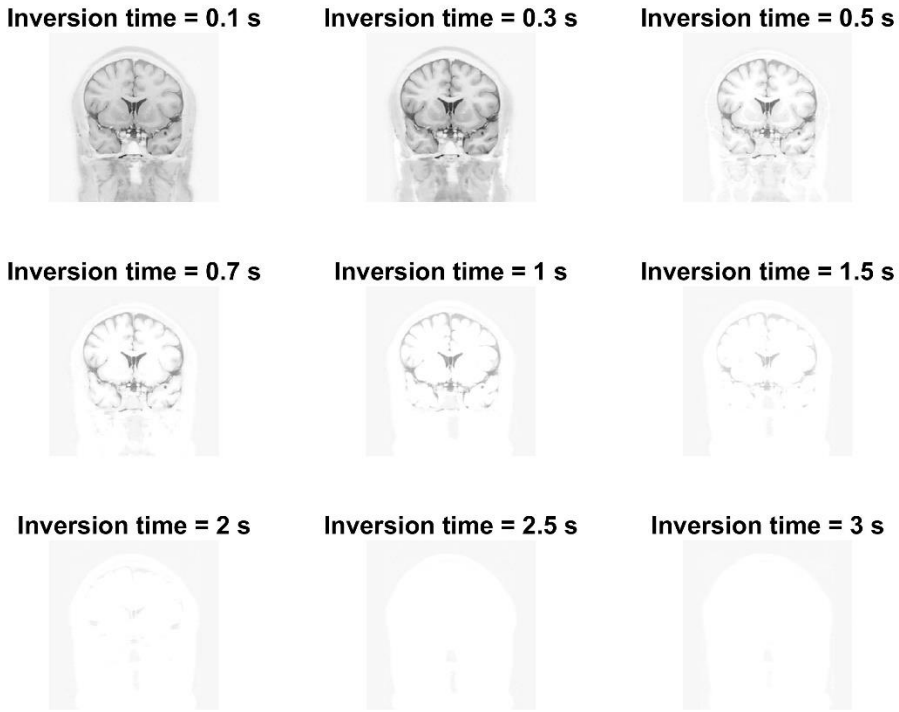
```
axis off
colormap(gray)
title('Residual Map')

% showing fit for different regions
figure
subplot(1, 2, 1)
t2Max = max(t2_m(:));
red0_m = t2_m/t2Max;
green0_m = t2_m/t2Max;
blue0_m = t2_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')

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, :));

    % Insert your code here to create an array of nPoints TE values from
    % the minimum to maximum te:
    teFit_v = linspace(te_v(1), te_v(5), nPoints);
    % Insert your code here to find the corresponding signal at each TE,
    % using your estimates of T2 and S0:
    sFit_v = s0*exp(-teFit_v/t2);
    subplot(1, 2, 2)
    plot(teFit_v, sFit_v, ':', te_v, s_v, '+')
    title(['At row = ', num2str(row), ', col = ', num2str(col), ': T2 = ', ...
        num2str(t2*1000), 'ms'])
end
```

```
% creating mask for white matter
figure
image1_m = squeeze(image_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask1_m, x1_v, y1_v] = roipoly;
line(x1_v, y1_v,'color', 'y')

figure
image1_m = squeeze(image_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask2_m, x2_v, y2_v] = roipoly;
line(x2_v, y2_v,'color', 'y')

mask_w_m = mask1_m - mask2_m;
figure
imagesc(mask_w_m)
axis image
axis off
colormap(gray)

% showing T1 map for white matter
t2_w_m = zeros(nRows, nCols);
for row=1:nRows
    for col=1:nCols
        if (mask_w_m(row,col)==1)
            t2_w_m(row, col)=t2_m(row,col);
        end
    end
end

figure
imagesc(t2_w_m)
intLimits_v = get(gca, 'CLim');
axis image
axis off
colormap(gray)

% histogram for white matter
figure
histogram(t2_w_m)

% calculating mean and standard deviation for white matter
w=1;
for row=1:nRows
    for col=1:nCols
        if (mask_w_m(row,col)==1)
            t2_w(w)=t2_m(row,col);
```

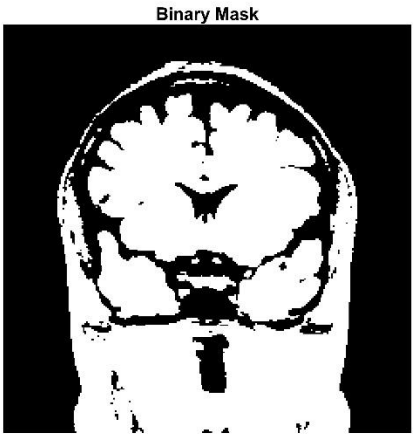
```
            w=w+1;
        end
    end
end

mean_white = mean(t2_w);
StdDev_white = std(t2_w);

% creating mask for gray matter
figure
image1_m = squeeze(image_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask4_m, x4_v, y4_v] = roipoly;
line(x4_v, y4_v,'color', 'y')

mask_g_m = mask4_m - mask1_m;
figure
imagesc(mask_g_m)
axis image
axis off
colormap(gray)

% showing T1 map for white matter
t2_g_m = zeros(nRows, nCols);
for row=1:nRows
    for col=1:nCols
        if (mask_g_m(row,col)==1)
            t2_g_m(row, col)=t2_m(row,col);
        end
    end
end

figure
imagesc(t2_g_m, [0 0.2])
intLimits_v = get(gca, 'CLim');
axis image
axis off
colormap(gray)

% histogram for gray matter
figure
histogram(t2_g_m)

% calculating mean and standard deviation for white matter
w=1;
for row=1:nRows
    for col=1:nCols
        if (mask_g_m(row,col)==1)
            t2_w(w)=t2_m(row,col);
            w=w+1;
        end
    end
```

```
end

mean_gray = mean(t2_w);
StdDev_gray = std(t2_w);
```

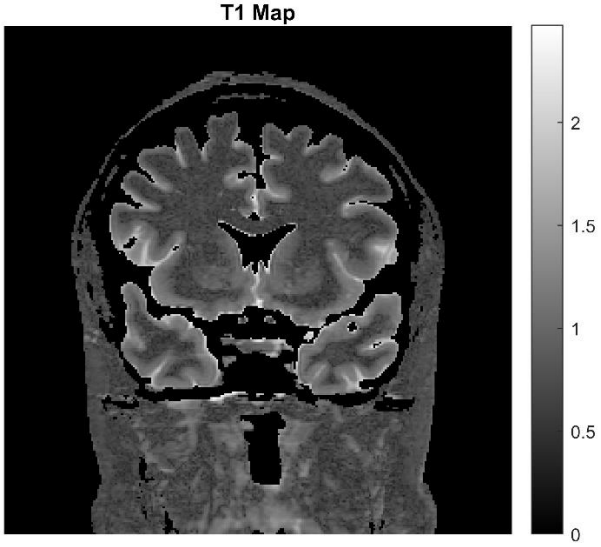
6.



All images are scaled to the lowest and highest intensity of the first image (TI=100 ms) and as we can see, with the increase in TI, the contrast between different regions of the brain get diminished.

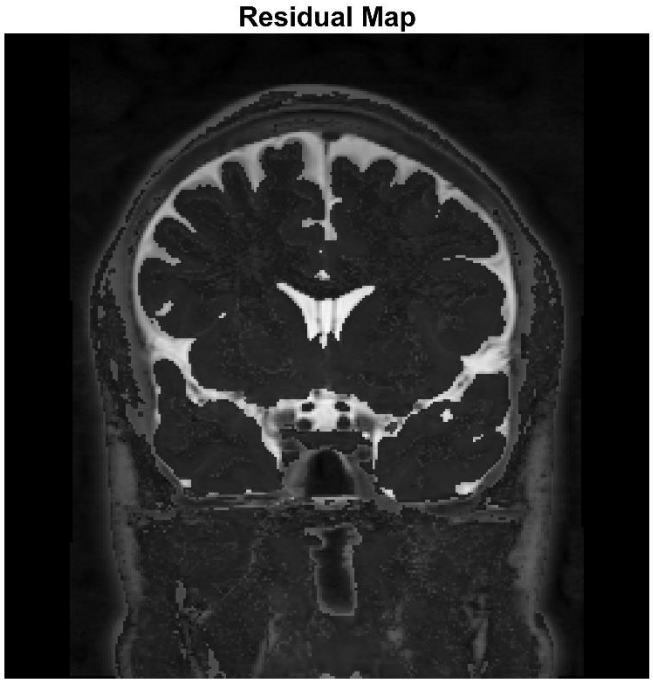


7.



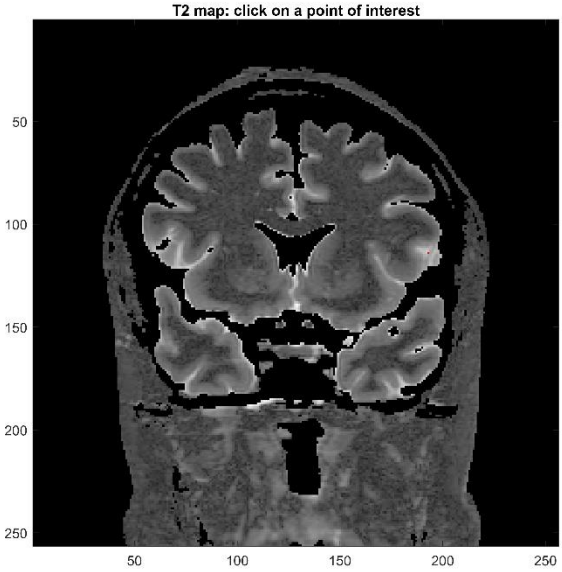
8.

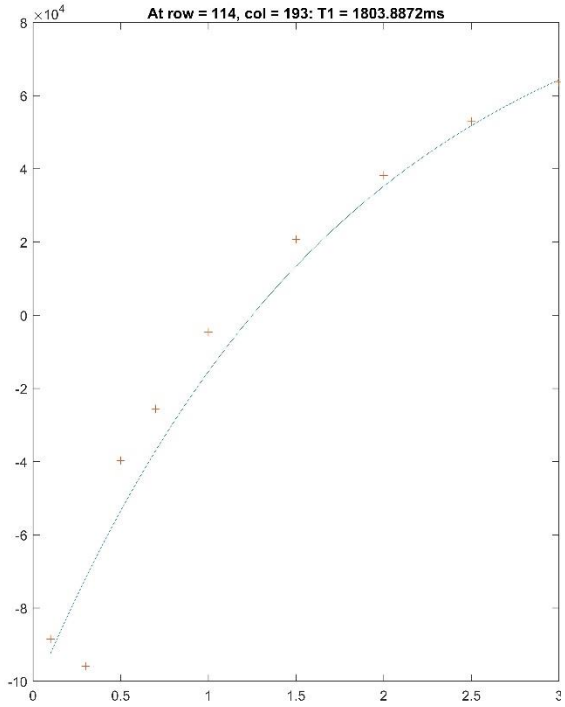
<u>Residual Map:</u>



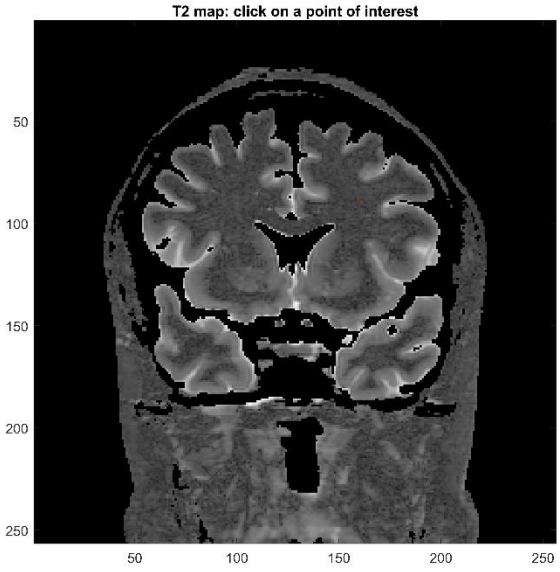
9.

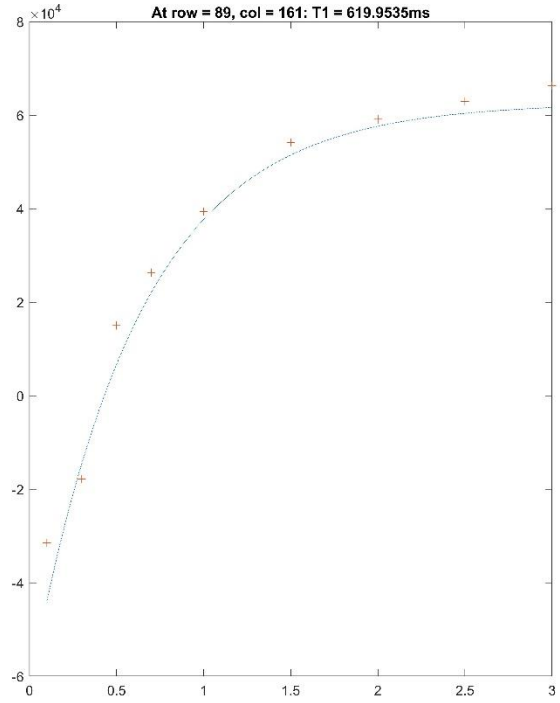
T1 fit for a gray matter pixel





T1 fit for a white matter pixel

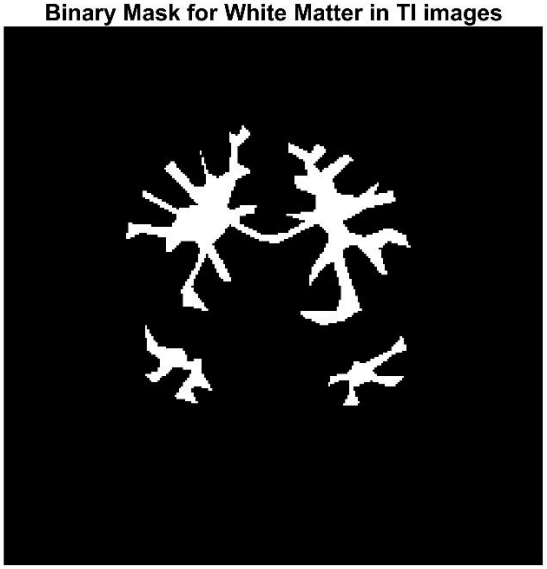




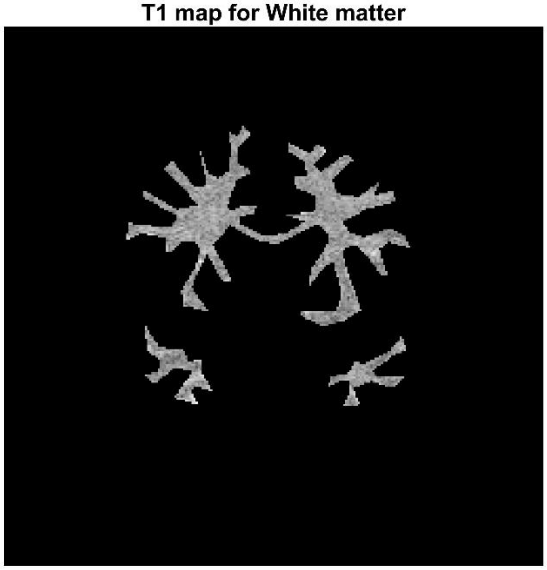
10.

For White Matter:

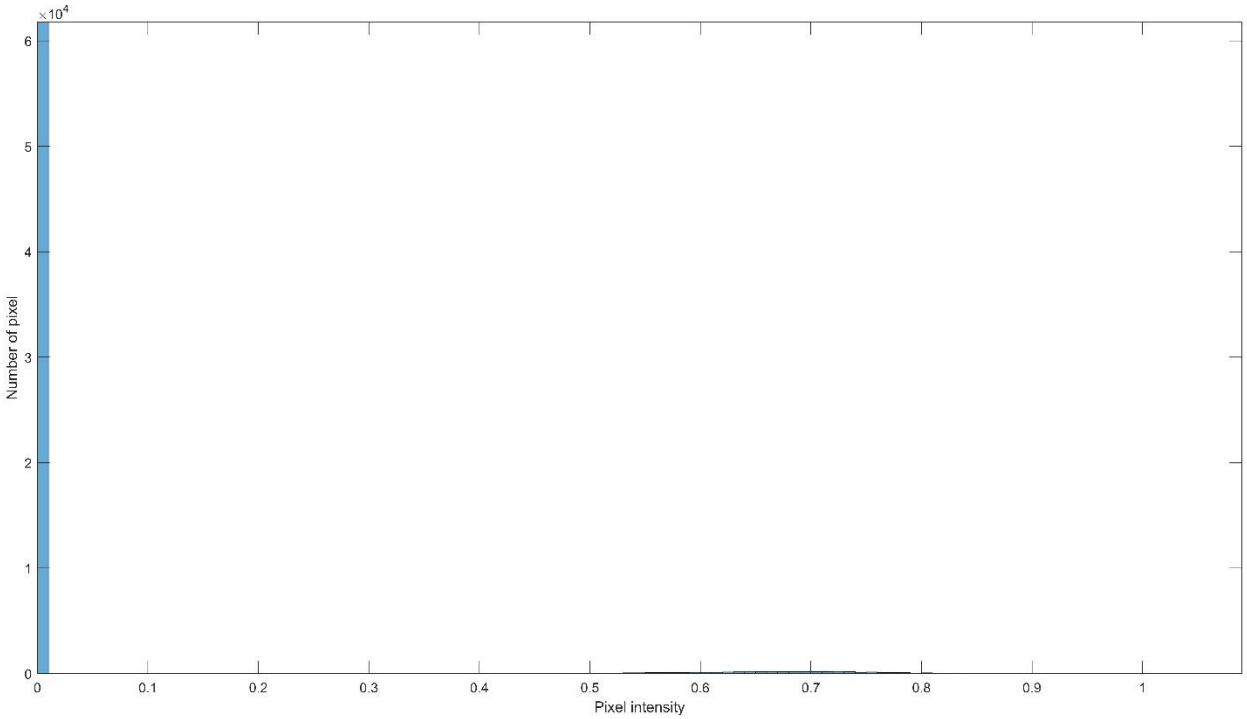
Binary Mask for White Matter in TI images



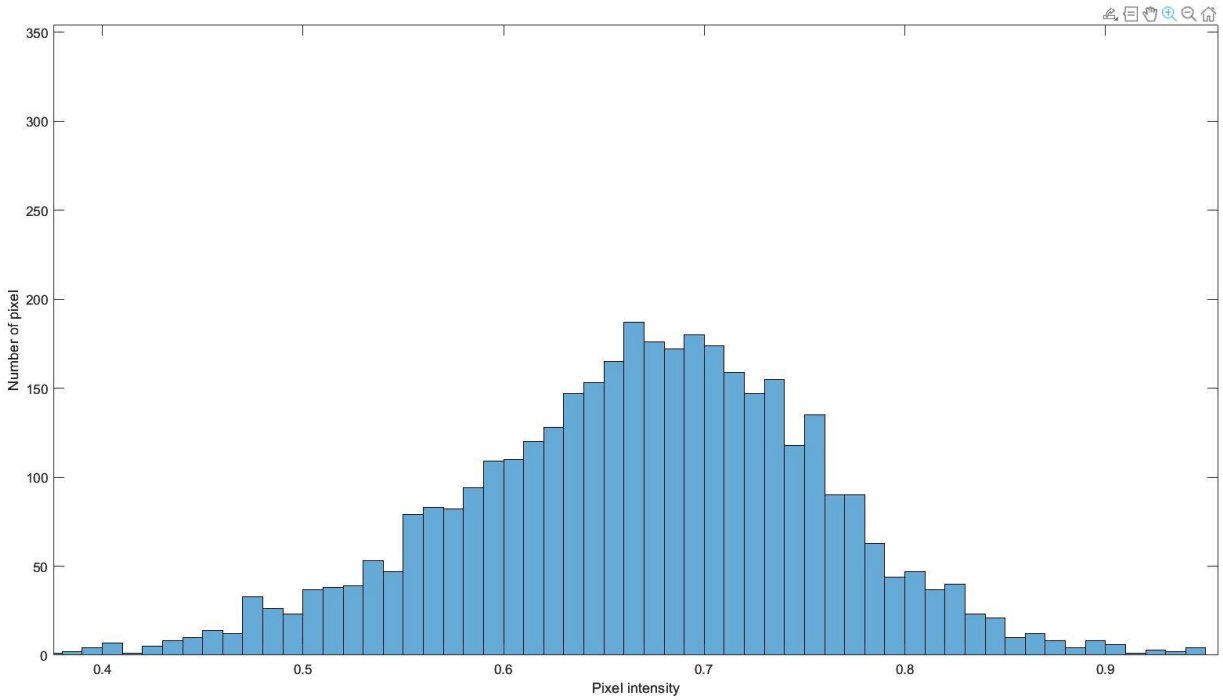
T1 map for White matter



T1 Histogram for white matter pixels

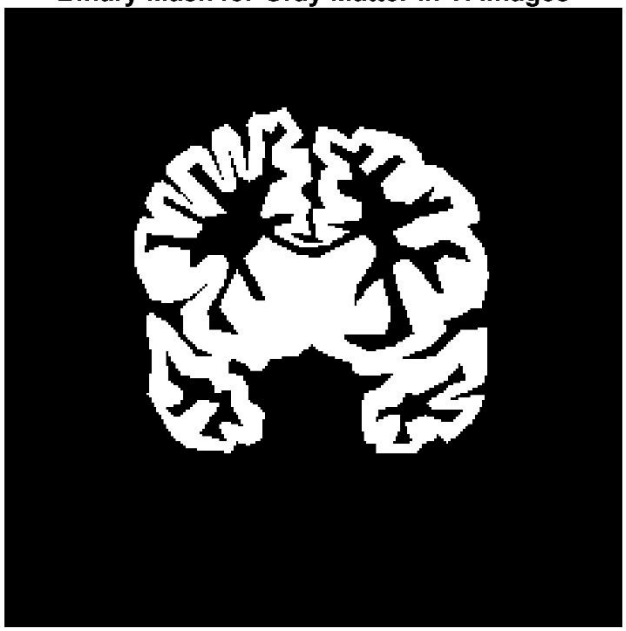


Since applying binary mask creates a lot of pixels with zero intensity, a zoomed version of the main pixels is shown below.

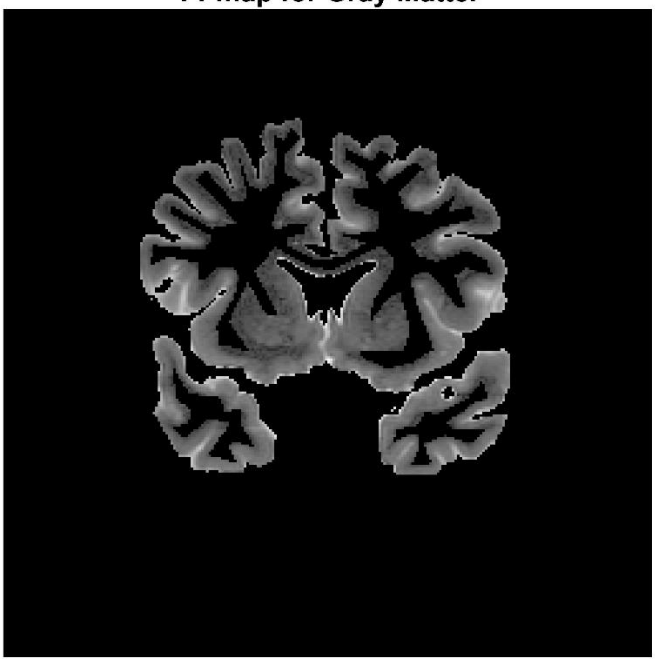


For Gray Matter:

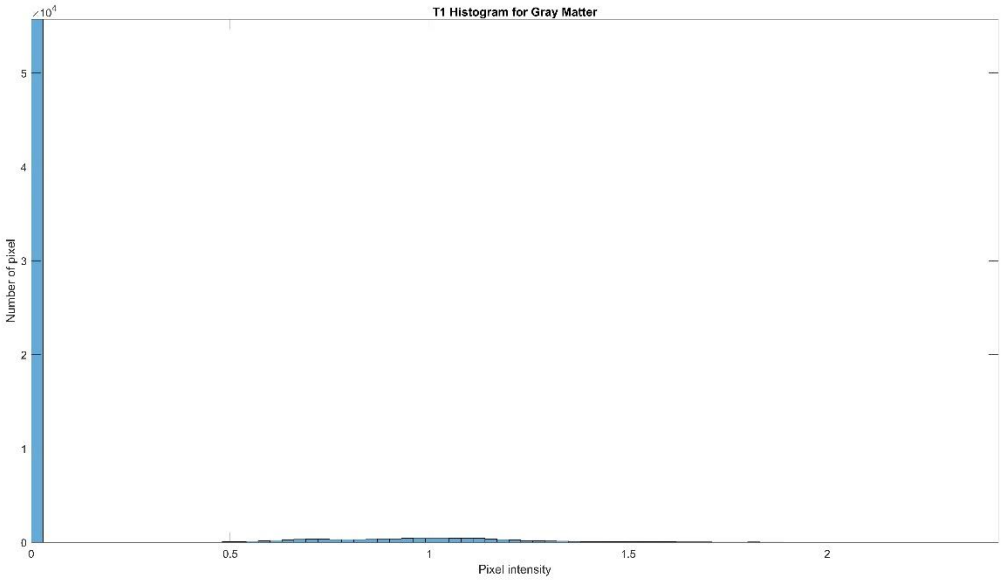
Binary Mask for Gray Matter in TI Images



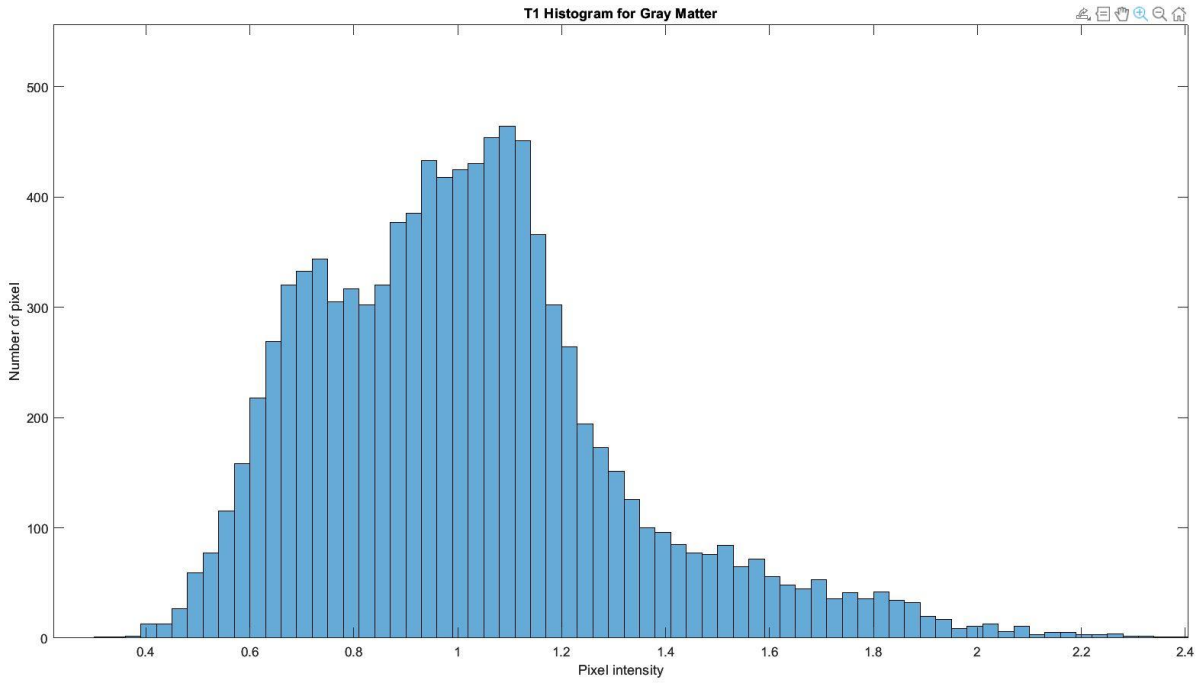
T1 map for Gray Matter



T1 Histogram for gray matter pixels



Zoomed version of the main pixels of gray matter's T1 histogram is shown below.



Corresponding Matlab Code:

```
clc; close all;

load('proj1bData_QFI');
[nRows, nCols, nTe] = size(irImage_3d);

% plotting all the TI images in the same intensity scle
figure
subplot(3, 3, 1)
imagesc(squeeze(irImage_3d(:, :, 1)))
intLimits_v = get(gca, 'CLim');
axis image
axis off
colormap(gray)
title(['Inversion time = ', num2str(ti_v(1)*1000), ' ms'])

for index = 2:nTe
    subplot(3, 3, index)
    imagesc(squeeze(irImage_3d(:, :, index)), intLimits_v)
    axis image
    axis off
    colormap(gray)
    title(['Inversion time = ', num2str(ti_v(index)*1000), ' ms'])
end

% creating a binary mask
image_m = squeeze(irImage_3d(:, :, 7));
mask_m = (image_m>0.2*max(image_m(:)));

figure
imagesc(mask_m)
colormap(gray)
axis image
axis off
title('Binary Mask')

% creating T1 map
t1_m = zeros(nRows, nCols);
for row=1:nRows
    for col=1:nCols
        if (mask_m(row,col)==1)
            Mz = squeeze(irImage_3d(row, col, :));

            numerator = 1- (Mz/m0_m(row,col));
            ln = -real(log(numerator*0.5));
            coeff_v = polyfit(ti_v, ln, 1);

            slope = coeff_v(1);
             t1 = 1 / slope;
            t1_m(row, col) = t1;
        end
    end
end
```

```
figure
imagesc(t1_m)
colormap(gray)
colorbar

% creating residual map
res=zeros(nRows,nCols);
for row=1:nRows
    for col = 1:nCols
        signal_res_v = squeeze(irImage_3d(row, col, :));
        res(row, col)=sqrt(norm(signal_res_v ...
            - m0_m(row, col)*(1-2*exp(-ti_v./t1_m(row,col)))));
    end
end

figure
imagesc(res)
axis image
axis off
colormap(gray)
title('Residual Map')

% showing fit for different regions
figure
subplot(1, 2, 1)
t1Max = max(t1_m(:));
red0_m = t1_m/t1Max;
green0_m = t1_m/t1Max;
blue0_m = t1_m/t1Max;
color_3d = cat(3, red0_m, green0_m, blue0_m);
image(color_3d)
axis image
title('T2 map: click on a point of interest')

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:
    t1 = t1_m(row, col);
    m0 = m0_m(row, col);
    s_v = squeeze(irImage_3d(row, col, :));

    % Insert your code here to create an array of nPoints TE values from
    % the minimum to maximum te:
    tiFit_v = linspace(ti_v(1), ti_v(9), nPoints);
    % Insert your code here to find the corresponding signal at each TE,
    % using your estimates of T2 and S0:
    sFit_v = m0*(1-2*exp(-tiFit_v./t1));
    subplot(1, 2, 2)
    plot(tiFit_v, sFit_v, ':', ti_v, s_v, '+')
    title(['At row = ', num2str(row), ', col = ', num2str(col), ': T1 = ', ...
        num2str(t1*1000), 'ms'])
end

% creating mask for white matter
figure
image1_m = squeeze(irImage_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask1_m, x1_v, y1_v] = roipoly;
line(x1_v, y1_v,'color', 'y')

figure
image1_m = squeeze(irImage_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask2_m, x2_v, y2_v] = roipoly;
line(x2_v, y2_v,'color', 'y')

figure
image1_m = squeeze(irImage_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask3_m, x3_v, y3_v] = roipoly;
line(x3_v, y3_v,'color', 'y')

% combining all the masks
mask_w_m=mask1_m + mask2_m +mask3_m;
figure
imagesc(mask_w_m)
axis image
axis off
colormap(gray)
```

```
% showing T1 map for white matter
t1_w_m = zeros(nRows, nCols);
for row=1:nRows
    for col=1:nCols
        if (mask_w_m(row,col)==1)
            t1_w_m(row, col)=t1_m(row,col);
        end
    end
end

figure
imagesc(t1_w_m)
intLimits_v = get(gca, 'CLim');
axis image
axis off
colormap(gray)

% histogram for white matter
figure
histogram(t1_w_m)

% calculating mean and standard deviation for white matter
w=1;
for row=1:nRows
    for col=1:nCols
        if (mask_w_m(row,col)==1)
            t1_w(w)=t1_m(row,col);
            w=w+1;
        end
    end
end

mean_white = mean(t1_w);
StdDev_white = std(t1_w);


% showing T1 map for gray matter
figure
image1_m = squeeze(irImage_3d(:,:,1));
imagesc(image1_m);
colormap(gray)
axis off
axis image
[mask4_m, x4_v, y4_v] = roipoly;
line(x4_v, y4_v,'color', 'y')

mask_g_m=mask4_m - mask1_m - mask2_m - mask3_m;
figure
imagesc(mask_g_m)
axis image
axis off
colormap(gray)

t1_g_m = zeros(nRows, nCols);
for row=1:nRows
```

```
    for col=1:nCols
        if (mask_g_m(row,col)==1)
            t1_g_m(row, col)=t1_m(row,col);
        end
    end
end

figure
imagesc(t1_g_m)
intLimits_v = get(gca, 'CLim');
axis image
axis off
colormap(gray)
histogram for gray matter
figure
histogram(t1_g_m)

% calculating mean and standard deviation for gray matter
g=1;
for row=1:nRows
    for col=1:nCols
        if (mask_g_m(row,col)==1)
            t1_g(g)=t1_m(row,col);
            g=g+1;
        end
    end
end

mean_gray = mean(t1_g);
StdDev_gray = std(t1_g);
```

Questions:

1. *The large, bright "X" shaped structure in the center of the brain in part 1 is part of the ventricular system. The ventricles are cavities in the brain filled with cerebral spinal fluid (CSF). Based on the signal decay in the five T2–weighted images, does the CSF have longer or shorter T2 relaxation time than brain tissue? (Explain using qualitative observations, not your fitting results).*

   Answer:

   We know that CSF mainly consists of water molecules and because of the small sizes of the molecules, their transverse relaxation/decay takes more time than other brain areas. Since they take a longer time to decay, the signal acquired from them is stronger and hence, it appears white in T2 images.

2. *What is a typical T2 value for brain tissue? How much variation do you see over the brain?*

   Answer:

   A typical T2 value for brain tissue would be between 70 ms to 120 ms. There are some variations among different regions of the brain. Like, CSF has more than 200 ms of T2 value, white matter's T2 value is around 100 ms, and for gray matter it is around 180-200 ms.

3. *Are the residuals of the fit uniform in the head? If not, where are they larger and where are they smaller and what might cause the variation?*

   Answer:

   No, the residuals of fit are not uniform throughout the head. They are larger for CSF and gray matter, and smaller for white matter. So, we can say that, regions with higher T2 values are less well fitted to our exponential model. The reason could be since we are assuming that T2<=0.2, the areas with high T2 values are getting modulated/lowered. So, it affects the approximation made by our model and fitting is not good there.

4. *How well can you distinguish between gray matter and white matter on the basis of T2 values in this map? What is the contrast in T2 values between gray and white matter (i.e., mean T2 for the gray matter ROI minus the mean T2 for the white matter ROI)? If the 'noise' in this measurement is the standard deviation of the difference between T2 values, $\sigma T2(GM)-T2(WM)$, what is the contrast-to-noise ratio, CNR, between gray and white matter in the T2 map? Use the propagation of errors to find an expression for $\sigma T2(GM)-T2(WM)$ in terms of $\sigma T2(GM)$ and $\sigma T2(WM)$.*

   Answer:

   I think both gray and white matter regions can be distinguished well by just seeing. Because the gray matter has a higher T2 value than white matter. However, by using roipoly function, it becomes very tedious and difficult to distinguish the regions.

   The contrast between the two regions,
   Contrast = mean(gray matter) – mean(white matter) = 0.0978-0.0782=0.0196
   CNR = Contrast/std(gray matter – white matter)

=Contrast/ sqrt(var(gray matter) + var(white matter)) [using propagation of error formulas]

$=0.0196/\text{sqrt}(0.6401^2 + 0.0166^2)$

=0.0306

5. *Find the lateral ventricles in the inversion recovery images. Based on the Mz recovery as a function of TI, does CSF have longer or shorter T1 relaxation time than brain tissue? (Again, explain using your qualitative observations, not the fitting results).*

   Answer:

   In the images, the ventricles appear black which means they take longer time to recover magnetization in the Z-direction. Thus, we can say they have a larger T1 compared to other regions.

6. *What is a typical T1 value for brain tissue? How much variation do you observe over the brain?*

   Answer:

   Typical T1 value for brain tissue would be around 1-2 s. But it is not uniform and there are lots of variations. For gray matter, T1 is around 1.8-2.5 s; for white matter it is around 1-1.5s.

7. *Do the estimated (modeled) recovery curves match the measured data well? Are the residuals similar for different tissue types?*

   Answer:

   For most of the regions, the residuals fit well. But just like in the T2 images, the fit is not good around the ventricles and gray matter regions. It might because of the assumption that Mz <=m0.

8. *How well can you distinguish between gray matter and white matter on the basis of T1? What is the contrast in T1 values between gray and white matter? If the 'noise' in this measurement is the standard deviation of T1 values around the means, what is the contrast-to-noise ratio, CNR, between gray and white matter in the T1 map?*

   Answer:

   Compared to T2 images, it is a bit easier to distinguish gray and white regions by eye. As stated before, by using roipoly function, it becomes very tedious and difficult to distinguish the regions.

   The contrast between the two regions,

   Contrast = mean(gray matter) – mean(white matter) = 0.9009-0.6691=0.2318

   CNR = Contrast/std(gray matter – white matter)

   =Contrast/ sqrt(var(gray matter) + var(white matter)) [using propagation of error formulas]

   $=0.2318/\text{sqrt}(0.0.4341^2 + 0.0917^2)$

   =0.5224

9. *Which relaxation time, T1 or T2, provides the highest CNR between gray and white matter? If you had to classify each pixel in the brain as either gray or white matter (ignoring CSF), which relaxation time map would you use?*

   Answer:

   From the calculations, T1 provides a higher CNR value. So, its easier to distinguish gray and white matter from T1 images. Hence, I'd use T1 mapping to distinguish between them.

10. *Ideally, the residual maps reflect only random noise. Do you see evidence for non-random errors in the T1 or T2 fits?*

    Answer:

    I think there are some non-random errors in both T1 and T2 fits. For higher value of T1 and T2, the residual is bigger (it can be because of our approximations and assumptions as discussed earlier). This correlation suggests that these errors are not because of random noises.