脳画像解析の観点での Matlabの基本

筑波大学医学医療系精神医学
根本清貴

Ver.20191110.01

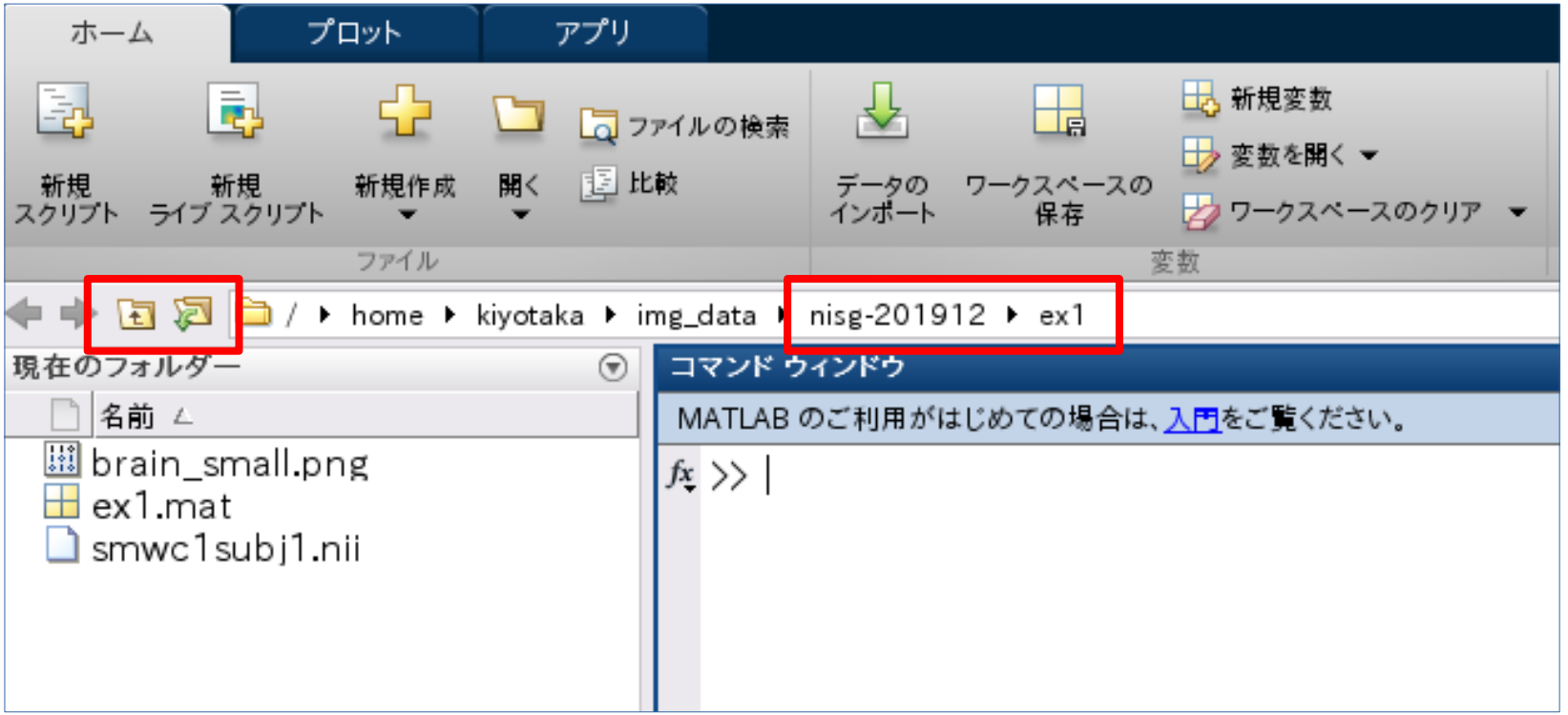
本日全体の作業ディレクトリ

- Windows
  - C:\img_data\nisg-201912 とします
  - データパーティションとしてDドライブがある方は、D:\img_data\nisg-201912 でもかまいません
  - ポイントはディレクトリのパスに日本語がないことです

- Mac
  - /Users/ユーザ名/nisg-201912 とします

この時間の作業ディレクトリ

- nisg-201912/ex1 がこの時間のディレクトリです
- Matlabでまず、ここに移動してください



チートシート

- ex1_talk.m にこのスライドにある全てのコマンドが記載されています
- タイプがつらくなった時などにどうぞ

```
>> edit ex1_talk.m
```

- 「実行して次に進む」が便利

Why Matlab?

- 脳画像解析は、脳のデータを「変数の線形結合」で表示するから
  - 線形結合：掛け算と足し算で表示
- 行列で扱ったほうが計算が楽だから
- 脳画像は「行列」で表示するとわかりやすいから

線形結合

- 以下を満たすβとCを求めなさい

$$y_1 = \beta x_1 + C$$
$$y_2 = \beta x_2 + C$$

- これは、中学数学で行う連立方程式にすぎない
- Matlabにはこれを一気に解く強力なコマンド `inv` と `pinv` がある

行列の基本

- 行列の掛け算 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$
- 逆行列をかけると、単位行列になる

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} a & b \\ c & d \end{bmatrix}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- 逆行列を使えば連立方程式が解ける

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

連立方程式を行列で解く

- 次の連立方程式をときなさい

$$2x+5y=12$$
$$3x+4y=11$$

$$\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}12\\11\end{bmatrix}$$

$$\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}^{-1}\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}^{-1}\begin{bmatrix}12\\11\end{bmatrix}$$

$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}^{-1}\begin{bmatrix}12\\11\end{bmatrix}$$

$$A=\begin{bmatrix}2 & 5\\3 & 4\end{bmatrix}$$
$$X=\begin{bmatrix}x\\y\end{bmatrix}$$
$$B=\begin{bmatrix}12\\11\end{bmatrix}$$ とすると

$$AX=B$$

$$A^{-1}AX=A-1B$$

$$X=A^{-1}B$$

Matlabで以下を実行

```
>> A = [2 5; 3 4]
>> B = [12; 11]
>> X = inv(A)*B
X =
 1.000
 2.000
>> A*X
ans =
   12.0000
   11.0000
```

$$AX = B$$
$$X = A^{-1}B$$

行列A、Bを規定できれば、
行列Xは簡単に求められる

A*Bは確かにXになっている

連立方程式を行列で解く

- 次の連立方程式をときなさい

$$
\begin{aligned}
&a+b+c+d+e=0\\
&2a+b-c+3d+4e=8\\
&5a+2b+3c-4d+6e=-3\\
&2a+3b+4c-6d+7e=6\\
&3a+4b+2c+5d+e=-5
\end{aligned}
$$

$$
A=\begin{bmatrix}1&1&1&1&1\\2&1&-1&3&4\\5&2&3&-4&6\\2&3&4&-6&7\\3&4&2&5&1\end{bmatrix}
$$

$$
X=\begin{bmatrix}a\\b\\c\\d\\e\end{bmatrix}\quad B=\begin{bmatrix}0\\8\\-3\\6\\-5\end{bmatrix}
$$

$$
AX=B
$$

Matlabで以下を実行

```
>> A = [1 1 1 1 1; 2 1 -1 3 4; 5 2 3 -4 6; 2 3 4 -6
7; 3 4 2 5 1]

>> B = [0; 8; -3; 6; -5]

>> X = inv(A)*B

X =

   -3.0000

   -1.0000

    0.0000

    1.0000

    3.0000

>> A*X %きちんとBになっているかどうかを確認
```

$$AX = B$$
$$X = A^{-1}B$$

変数の数がどんなに増えても、同じ X=inv(A)*B でXを簡単に求められる
これが行列の強み

行列の要素を修正

- もし、行列の要素の入力を間違ってしまった場合、ピンポイントで修正できる。
- 書式は以下
- **修正したい行列（行番号，列番号）＝修正したい要素**
- 行列Aを行列Hとして複製し、Hの5行4列目の値を（5でなく）3に変更してみる

```
>> H = A

>> H(5,4) = 3

H =

     1     1     1     1     1
     2     1    -1     3     4
     5     2     3    -4     6
     2     3     4    -6     7
     3     4     2     3     1
```

値の推定

- 本来ならば、N元一次方程式に一意の解を与えるためには、N個の式が必要
  - 2元一次方程式→式が2つ必要
  - 変数Nに対して方程式がNに満たない時→解が一意に定まらない
  - 変数Nに対して方程式がNより多い時→正解がない
- 逆行列を使うと、変数と方程式の数が一致しなくても解を推測できる
- 正方行列（行と列の要素数が同じ）でない時に使う逆行列のコマンドが **`pinv`** (pseudo inversion)

連立方程式を行列で解く

- 次の連立方程式をときなさい

$$
\begin{aligned}
&a+b+c+d+e=0\\
&2a+b-c+3d+4e=8\\
&5a+2b+3c-4d+6e=-3\\
&2a+3b+4c-6d+7e=6
\end{aligned}
$$

$$
C=\begin{bmatrix}1&1&1&1&1\\2&1&-1&3&4\\5&2&3&-4&6\\2&3&4&-6&7\end{bmatrix}
$$

- 変数が5つあるのに、式は4つしかない
- このような場合、疑似逆行列が役立つ

$$
X=\begin{bmatrix}a\\b\\c\\d\\e\end{bmatrix}\quad D=\begin{bmatrix}0\\8\\-3\\6\end{bmatrix}
$$

$$
CX=D
$$

Matlabで以下を実行

```
>> C = A(1:4,:) %Aから1行から4行目までだけを取り出して、新たなCという行列に代入

>> D = B(1:4,:) %上と同様

>> X = pinv(C)*D

X =

   -2.7027

    0.7207

   -1.0721

    0.6036

    2.4505

>> C*X %別のXでもCX=Dを満たす可能性がある
```

$$CX = D$$

$$X = C^{-1} D$$

連立方程式を行列で解く

- 次の連立方程式をときなさい

$$
\begin{aligned}
&a+b+c+d=0\\
&2a+b-c+3d=8\\
&5a+2b+3c-4d=-3\\
&2a+3b+4c-6d=6\\
&3a+4b+2c+5d=-5
\end{aligned}
$$

- 今度は変数が4つに対し、方程式が5つある
- このような場合も、疑似逆行列が役立つ
- 脳画像解析はほぼこのパターン

$$
E=\begin{bmatrix}1&1&1&1\\2&1&-1&3\\5&2&3&-4\\2&3&4&-6\\3&4&2&5\end{bmatrix}
$$

$$
X=\begin{bmatrix}a\\b\\c\\d\end{bmatrix}\quad F=\begin{bmatrix}0\\8\\-3\\6\\-5\end{bmatrix}
$$

$$
EX=F
$$

Matlabで以下を実行

```
>> E = A(:,1:4) %Aから1～4列目までだけを取り出して、新たなEという行列に代入

>> F = B %Bと同じFという行列を準備

>> X = pinv(E)*F

X =

   -0.7184

    8.0151

   -8.9355

   -3.0615

>> E*X %Fにやや近いがFと一致しない
```

$$EX = F$$

$$X = E^{-1}F$$

Matlabのお作法：セミコロンとコロン

- セミコロン（;）の使い方2つ
  - 行列をあらわす時に、改行する

    ```
    [2 4; 3 5]
    ```

  - コマンドの結果を出力しない

    ```
    X=pinv(A)*B;
    ```

- コロン（:）の使い方2つ
  - 数字a:数字b:数字c　で、aからcまでb刻みで連続する値を意味

    ```
    1:2:10 %1～10まで2刻み→「１３５７９」と同義
    ```

  - 単独で使うことで、「全行」「全列」を意味

    ```
    A(:,2) %行列Aの全行第2列
    ```

Matlabのお作法:パーセント

- パーセント(%)はコメント
  - % が出てきた時は、これ以降行末まではMatlabは無視する
  - %% と2つ続けた場合は、セクションの開始となる
- スクリプトを書く際はコメントを書く習慣をつける
- シェルスクリプトやPythonでのコメントは # であることに注意

脳画像解析の本質は「係数の推定」

- 脳画像解析は、今行ってきたことをやっているにすぎない
- 画像のボクセル値を、年齢、性別、頭蓋内容積、心理検査の結果などの線型結合で表示し、それぞれの項の係数と定数を推定する
- 例：年齢と相関する領域を求めたい
  - 式をたてる
    - ボクセル値＝β×年齢＋C（定数）
  - 係数を求める
    - βを計算
  - 係数が0と異なるかを評価する

Y = β × age + C

- Y = age × β + 1 × C
  - Y（ボクセル値）とage（年齢）から、βとCを求める
- $Subject_1$のボクセル値をY_1、年齢をage_1とすると、行列は次のように表せる

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} age_1 & 1 \\ age_2 & 1 \\ \vdots & \vdots \\ age_n & 1 \end{bmatrix} \begin{bmatrix} \beta \\ C \end{bmatrix}$$

$$Y = XB$$

- 行列Xを計画行列（Design matrix）という

  ※これまで扱ってきたBやXとは違うことに注意

Matlabのお作法:構造体

- Matlabは変数を入れる「箱」を準備できる
- この箱を「構造体 struct」という
- 構造体の中に複数の変数を入れることができる
- 今、構造体Sにこれまでの変数を入れてみる
- Matlabから以下を実行

```
>> S.A = A; %構造体Sの中に変数Aを設定

>> S.B = B;

>> S.X = X;
```

構造体を確認

- Matlabで構造体を確認

```
>> S

S =

  フィールドをもつ struct:

      A: [5×5 double]

      B: [5×1 double]

      X: [4×1 double]
```

- Aは5行5列、Bは5行1列、Xは4行1列の行列であることがわかる

構造体の中身にアクセス

- 構造体の中を見るには、　構造体.変数　とする
- 以下を確認

```
>> S.A

>> S.B

>> S.X
```

- SPMは構造体を頻用している
- 構造体であることを意識すれば様々な変数を自在に取り出すことができる

構造体の保存

- 構造体はmatファイルとして保存できる
- 書式は、以下

  `save('ファイル名.mat','保存する構造体名')`

- 今の構造体を`S.mat`として保存してみる

```
>> save('S.mat','S')
```

- ワーキングディレクトリに `S.mat` ができていることを確認

matファイルの読み込み

- matファイルは **load** コマンドで読み込める
- 書式は以下の2つ

  >> load matファイル %matファイルをそのまま読み込む

  >> 変数=load('matファイル') %構造体の中に構造体が入り込む

matファイルの読み込み

```
>> clear S  %変数(構造体)Sを消去

>> S  %関数または変数'S'が未定義です

>> load S.mat  %S.matの読み込み

>> S  %先程作成した構造体が確認できる

>> T=load('S.mat')  %構造体Tの中に構造体Sを読み込み

>> T.S  %Tの中にSが入っていることがわかる
```

Y=β × age + C（再掲）

- Y= age × β + 1 × C
  - Y（ボクセル値）とage（年齢）から、βとCを求める

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} age_1 & 1 \\ age_2 & 1 \\ \vdots & \vdots \\ age_n & 1 \end{bmatrix} \begin{bmatrix} \beta \\ C \end{bmatrix}$$

$$Y = XB$$

- ex1.mat にY, age, Xを格納してあり、Matlabで読み込み

```
>> load ex1.mat

>> ex1 %中身を確認するクセをつける
```

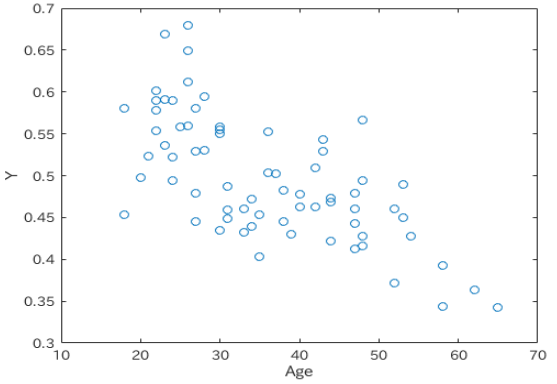
YとageのScatter図を描画

- 散布図を描くには、以下を用いる

```
>> plot(変数1, 変数2, 'o')
```

- o: 各点の形。o: ○, x: ×, s: □, ^: △
- Matlabから以下をタイプ

```
>> plot(ex1.age, ex1.Y, 'o')
>> xlabel('Age')
>> ylabel('Y')
```



βとCの推定

```
>> B = pinv(ex1.X)*ex1.Y
B =
    -0.0043 %傾き
     0.6493 %切片
```

- ここから推定できる直線を先程の散布図に描いてみる

```
>> y = ex1.X*B  % y = -0.043 × Age + 0.6493
>> hold on %グラフを上書きでなく、追記する
>> plot(ex1.age,y)
```

相関解析

- ageとYの相関係数を見れば、この2つに有意な相関があると言える
- Matlabでは、相関解析を行うことができる
- [相関係数　確率　95%下限　95%上限] = corrcoef(変数1,変数2)

```
>> [r p rl ru] = corrcoef(ex1.age,ex1.Y)

r =

    1.0000   -0.6812
   -0.6812    1.0000

p =

    1.0000    0.0000
    0.0000    1.0000

rl =

    1.0000   -0.7891
   -0.7891    1.0000

ru =

    1.0000   -0.5325
   -0.5325    1.0000
```

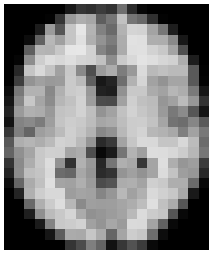
r	Age	Y
Age	1.0000	-0.6812
Y	-0.6812	1.0000

相関解析のまとめ

- 相関係数：-0.6812
- p: $<6.28\times10^{-11}$　%p(1,2)で詳しい値がわかる
- 95%信頼区間：-0.7891 ～ -0.5325
- Matlabは関数によっては、出力結果が複数出てくる
  - 今の場合、**[r p rl ru]** = corrcoef(ex1.age,ex1.Y) がそれ
- むしろ、そのように指定しないと、結果が出てこない

画像のMatlabへの取り込み

- Matlabでは、JPEG画像、PNG画像など、そのまま行列に取り込むことができる
- 取り込むコマンド: **imread**
- 'brain_small.png'を行列 brain に取り込んでみる



```
>> brain = imread('brain_small.png')
```

- brainの属性を確認

```
>> whos brain
  Name      Size            Bytes  Class     Attributes

  brain     24x20x3          1440  uint8
```

uint8

- **u**nsigned **int**eger **8**bit の略
- 符号なし整数型8ビット
- 符号なし＝マイナスはなし。プラスのみ
- 8bit = 2^8 = 256
- 信号値を0～255で表示

double と single

- double: 倍精度浮動小数点 64bit
- single: 単精度浮動小数点 32bit
- Matlabのデフォルト: double型
- uint8型をdoubleに変更する方法は簡単

```
>> brain = double(brain)
```

24x20x3

```
>> whos brain
  Name      Size            Bytes  Class    Attributes
  brain     24x20x3         11520  double
```

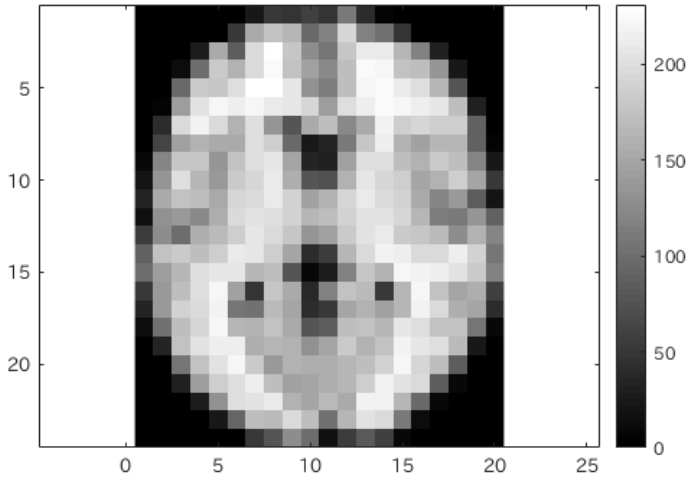
- 画像はRGB要素に分けて行列に入っている
- 今の脳画像はモノクロのため、24 x 20の同じ行列が3つ入っている
- 扱いやすくするため、1つの行列だけ取り出す

```
>> brain = brain(:,:,1)
>> whos brain
  Name      Size            Bytes  Class    Attributes
  brain     24x20           3840   double
```

行列の可視化

- 行列は **imagesc** で可視化できる
- ひとつの要素をひとつの長方形であらわす
- 要素を正方形にするには、**axis equal** とする
- カラーマップは **colormap** で変更できる
  - **colormap**までうって、スペースを入れた後、Tabキーを押せば、カラーマップの候補が表示される

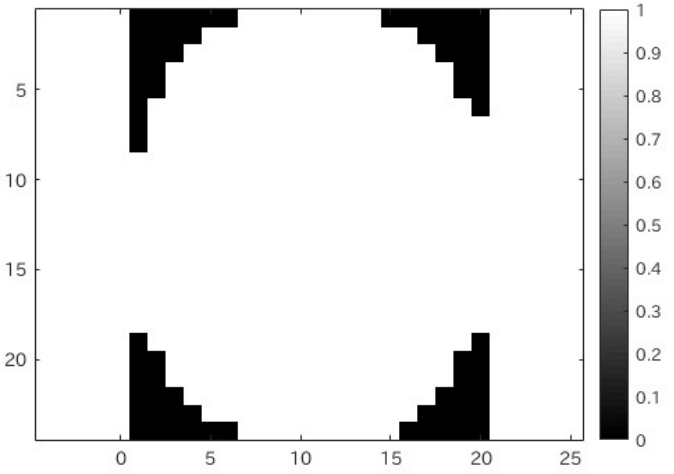
```
>> figure
>> imagesc(brain)
>> axis equal
>> colormap gray
>> colorbar
```



マスク画像の作成

- Matlabには「論理値 logical」がある
- "1"と"0"の2値だが、これは数字ではなく、"True"と"False"の意味
- 行列brainで0より多いものだけTrue,それ以外をfalseとした行列maskを作成し、可視化してみる

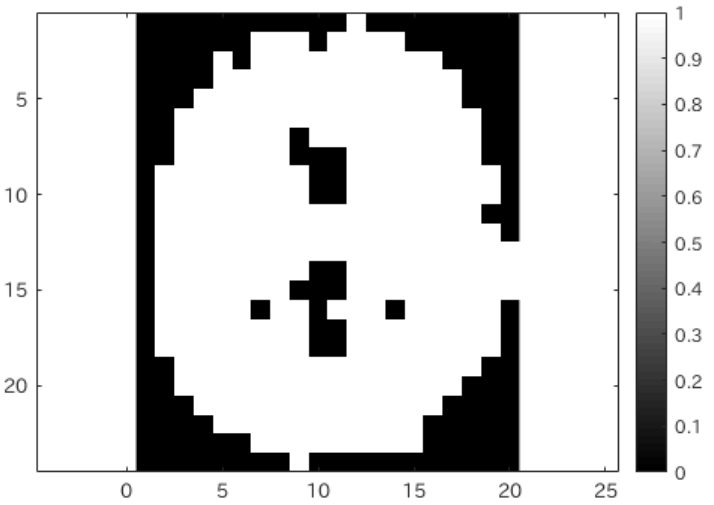
```
>> mask = (brain>0)
>> whos mask %ClassがlogicalになっているFigure
>> figure
>> imagesc(mask)
>> axis equal
>> colormap gray
>> colorbar
```



マスク画像の修正

- 今作ったマスクは大きすぎる
- 閾値を100とした行列mask2を作成し、可視化してみる

```
>> mask2 = (brain>100);
>> figure
>> imagesc(mask2)
>> axis equal
>> colormap gray
>> colorbar
```



変数のクリアとFigureの消去

- Matlabでの変数のクリアは以下の通り

```
>> clear 特定の変数
>> clear all        %すべてをクリア
```

- Figureを閉じる方法は以下

```
>> close figure  Figure番号
>> close all        %すべてのFigureを消去
```

NifTiファイルの読み込みと書き出し

- SPMの関数を使うと、NifTiファイルをMatlabに読み込むことができ、同時に書き出すことができる

```
spm_select        %NifTiファイルの選択
spm_vol           %NifTiヘッダの読み込み
spm_read_vols     %NifTiの行列への読み込み
spm_write_vol     %画像データへの書き出し
```

Matlabでのmask画像の作成

- smwc1subj1.nii を行列に読み込み、信号値が0.2より高い値を使ったマスク画像を作成してみる
- まずは、データを読み込む

```
>> P = spm_select(1,'image')  %GUIでsmwc1subj1.niiを選択
>> V = spm_vol(P)             %選択した画像のヘッダをVに読み込み
>> Y = spm_read_vols(V);      %実データを行列Yに格納
>> whos Y
 Name Size            Bytes     Class
 Y    121x145x121     16983560  double
```

マスクの作成

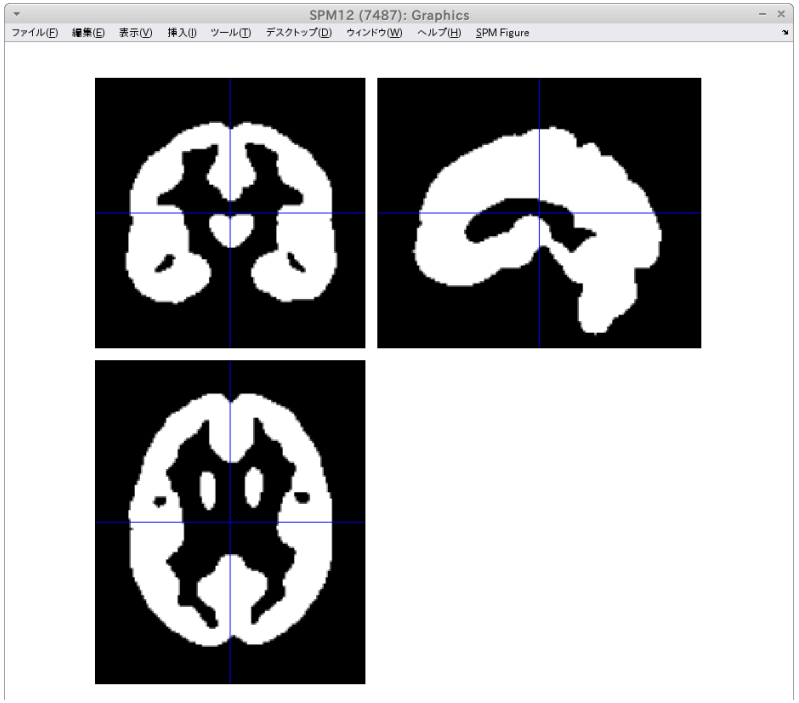
- 行列Yで0.2より高いところをマスクとする
- ヘッダVを利用して書き換える
- spm_write_volを使って書き出す

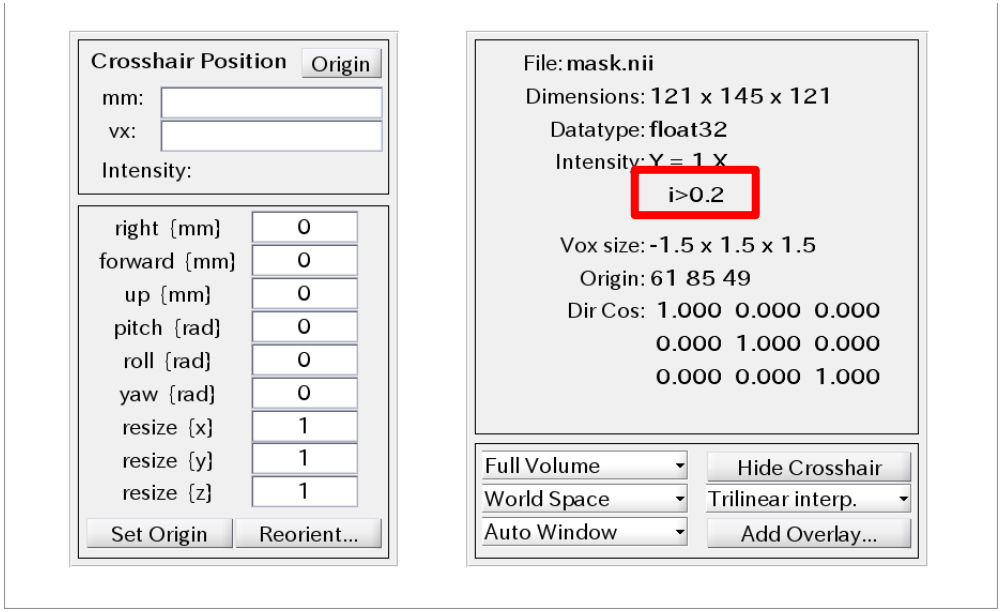
```
>> Ymask = (Y>0.2);
>> Vmask = V % ヘッダをコピー
>> Vmask.fname = 'mask.nii' % fnameを編集
>> Vmask.descrip = 'i>0.2'  % descipを編集
>> spm_write_vol(Vmask,Ymask)
```

作成したmask.niiの表示

- **`spm_image('Display',`**ファイル名**`)`**でSPMのGraphicsを呼び出すことができ、画像を表示できる

```
>> spm_image('Display','mask.nii')
```





どこで情報を手に入れているか

- SPMの機能に関しては、SPMのMLで情報が流れるたび、Matlabでその関数のヘルプを確認
- spm_imageであれば、

```
>> help spm_image
```

- ヘルプを読むとスキルがあがる

Questions?