Matrices in $n$ dimensions

\[
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix}
\]

\[
y(:,:,1) = \begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\]

\[
y(:,:,2) = \begin{bmatrix}
5 & 6 \\
7 & 8
\end{bmatrix}
\]

\[
y(:,:,3) = \begin{bmatrix}
9 & 10 \\
11 & 12
\end{bmatrix}
\]

size(y) $\rightarrow [2 \times 2 \times 3]$

- Typical brain data: location of neuron $(x, y, z)$ + time + response

Heat-maps

\textbf{imagesc(Data)} – view 2D matrix of scaled data as image
- Red is highest value, blue is lowest value

Visualize a 2D slice of brain data (size(brainData) $\rightarrow 128 \times 128 \times 88$)

\[
A = \text{squeeze(brainData(:,:,20))} \rightarrow \text{slice 20 of brain}
\]

\textbf{imagesc(A)}

Multiple plots

\textbf{figure} $\rightarrow$ opens new plotting window

\textbf{subplot(c,0,j)} $\rightarrow$ creates grid of plots with
- $r$ rows
- $c$ columns
- Fill in position $i$

\[
A = \text{squeeze(brainData(:,:,10))};
\]

\[
\text{subplot(1,3,1); imagesc(A)};
\]

\[
B = \text{squeeze(brainData(:,:,20))};
\]

\[
\text{subplot(1,3,2); imagesc(B)};
\]

\[
C = \text{squeeze(brainData(:,:,30))};
\]

\[
\text{subplot(1,3,3); imagesc(C)};
\]

Scaling vs. not-scaling

\textbf{imagesc(Data)} – view 2D matrix of scaled data as image
- Red is highest value, blue is lowest value

\textbf{image(Data)} – view 2D matrix of data as image
- Red is 64 or higher, blue is 0 or lower

\[
A = \text{squeeze(brainData(:,:,10))};
\]

\[
\text{figure; imagesc(A)};
\]

\textbf{``}

\[
\text{figure; image(A)}
\]

Repetition

Show four copies of brain slice 10:

\[
A = \text{squeeze(brainData(:,:,10))};
\]

\[
\text{figure; imagesc(A)};
\]

\[
\text{figure; imagesc(A)};
\]

\[
\text{figure; imagesc(A)};
\]

\[
\text{figure; imagesc(A)};
\]

\textbf{OR}

\[
A = \text{squeeze(brainData(:,:,10))};
\]

\[
\text{for } k=1:4,
\]

\[
\text{figure; imagesc(A)};
\]

\[
\text{end};
\]
Execution of \texttt{for}:

- Assign \( k \) each value counting up from \texttt{start} value to \texttt{finish} value, repeating listed actions for each new value of \( k \).

```plaintext
for k=start:finish,
  action1 to repeat
  action2 to repeat
  action3 to repeat
end;
```