Visualization of NIfTI Images
Again we read in the data

t1 = neurobase::readnii("training01_01_t1.nii.gz")
Density of an Image

Let's do a marginal density of the values:

```r
plot(density(t1))  # large spike at 0
```

![Density plot](image)
Density with a mask

You can also pass in a mask to most standard functions:

```r
plot(density(t1, mask = t1 > 0))
```
Similarly: a Histogram

Note the high intensities:

\texttt{hist(t1)}
Orthographic view with additions

The `neurobase::ortho2` function displays nifti objects in 3 different planes.

```
neurobase::ortho2(t1)
```
Brightening up the image

We see a dark image; a this single large value affects how values are mapped. The function `robust_window` calculates Winsorizes an image, by default the 0 (min) and 99.9th quantile, and sets values outside of this range to that quantile (97.5th below).

```r
ortho2(robust_window(t1, probs = c(0, 0.975)))
```
Robust Density

Note the x-axis:

Density of T1

Density of Windowed T1

N = 31752192  Bandwidth = 5.03

N = 31752192  Bandwidth = 4.786
Overlaying images in `ortho2`

For the rest of the slides we will use the robust $t_1$ for plotting

Here we plot the T1 and a mask of values over 300:

`ortho2(t1, y = t1 > 300)`
Double orthographic view

Sometimes you would like to represent 2 images side by side, of the same dimensions and orientation of course (useful for checking registration), use `double_ortho`

```python
double_ortho(t1, y = t1 > 300, col.y = "white")
```
Lightbox: view all slices

The `oro.nifti::image` function shows a lightbox view, all slices of an image:

```r
image(t1, useRaster = TRUE) # look at average brightness over each slice
```

![Slices of an image](image.png)
Viewing specific slices

The `slice` function can plot individual slices:

```cpp
oro.nifti::slice(t1, z = c(60, 80))
```
Different Planes

We can specify $z$ the same way but change the plane to be different to get a different slice of the brain (could also do coronal):

```c
oro.nifti::slice(t1, z = 125, plane = "sagittal")
```
Overlaying slices

We can also overlay one slice of an image upon another using the `oro.nifti::slice_overlay` function.

```
slice_overlay(t1, y = t1 > 300, z = 80)
```
Smoothing an Image (not extensively covered)

If you want to do 3D Gaussian smoothing, the `extrantsr::smooth_image` is helpful:

```r
library(extrantsr)
sm_img = smooth_image(t1, sigma = 2)
double_ortho(t1, sm_img)
```
Conclusions

- ortho2 - show orthographic images (and with overlays)
- image - shows multiple slices of an image
- slice - shows only specified slices
- slice_overlay - similar to image but with an overlay
- double_ortho - similar to ortho2 but side-by-side
- robust_window - good for setting high values to not so high
Website

http://johnmuschelli.com/imaging_in_r