image analysis exercises

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, data, filters, color
from scipy import ndimage
```

1. Load the `ohki2005.png` image into an array and display it in a matplotlib figure. Hint: use `skimage.io.imread`. What shape does the data have? Is it colour?

2. Rotate the image data 90 degrees *clockwise* and display it in a figure. Add a colorbar. Try plotting with different kinds of colormaps:

   i. sequential: e.g. 'gray', 'viridis'
   ii. Diverging: e.g. 'Spectral', 'coolwarm'
   iii. Categorical: e.g. 'Dark2', 'Pastel1'

See https://matplotlib.org/users/colormaps.html

Hint: you can use `im.set_cmap()` to change the existing colormap without having to create a new plot, where `im` is the image object returned by `imshow()`.

3. Plot a histogram of the pixel values. Make sure to set bins appropriately (one bin per possible pixel value). Use the `.ravel()` method on your array to flatten it to 1D before doing the histogram. what happens if you try and do a histogram on a 2D array?

4. Let's say we want to focus on the middle section of the image. Slice out the middle third of the image (vertically and horizontally), so you end up with an array with about 1/9th as many pixels as the original. Display it in a figure. Now scale up that subset by a factor of 2 using `scipy.ndimage.zoom` and display that as well.

5. Plot a histogram of the subset. Is it fairly different from the original histogram of the full image?

6. Apply the "Sobel" edge enhancement filter (`skimage.filters.sobel`) to the full image and display the result. Compare to the original image. Inspect both images to see why the algorithm worked better for some cells in the image than for others.

7. Load the immunohistochemistry example data from skimage (`immun = data.immunohistochemistry()`). Check its shape and display the image in a figure. What happens if you try and specify a colour map? Does it make sense to do so with this image?

8. Convert it to grayscale using `skimage.color.rgb2gray`. Check its shape and display it. Now does using a color map make sense?

9. Remove the blue channel from the image and display it. Hint: set the blue channel values to 0.

10. Bonus: try doing image segmentation by following the tutorial at http://scikit-image.org/docs/dev/user_guide/tutorial_segmentation.html

Edge-based segmentation:

```
from skimage.feature import canny

coins = data.coins()
f, ax = plt.subplots()
ax.hist(coins.ravel(), bins=np.arange(256))

edges = canny(coins)

f, ax = plt.subplots()
ax.imshow(edges, cmap='gray')
```

```
fill_coins = ndimage.binary_fill_holes(edges)

f, ax = plt.subplots()
ax.imshow(fill_coins, cmap='gray')
```

Region-based (watershed) segmentation:

```
from skimage.filters import sobel
elevation = sobel(coins) # edge detection

markers = np.zeros_like(coins)
# these markers designate foreground and background, see above histogram of coins
markers[coins < 30] = 1
markers[coins > 150] = 2

from skimage.morphology import watershed
segmentation = watershed(elevation, markers)
segmentation = ndimage.binary_fill_holes(segmentation - 1) # remove some holes
labels, _ = ndimage.label(segmentation) # label the coins
f, ax = plt.subplots()
ax.imshow(labels, cmap='Dark2')
```