Homework 5

As always, show your work, use docstrings and good style, and don't forget to test your code! Write your solutions in a file named `FirstnameLastname_homework5.py` and submit to m.spacek@lmu.de before class 11 (July 26).

images

1. Use the `skimage` library to load the file `nissl.tif` as a colour image array. NOTE: you might have to run `pip install imagecodecs` in order for skimage to be able to read .tif files. What shape is the array? What is its datatype (`dtype`)?

2. Display the image in matplotlib using `imshow()` . Check each channel (red, green and blue) programmatically to see if it has signal, i.e. non-zero pixel values. Which colour channel has signal?

3. Convert the 3D RGB array to grayscale by slicing out the one channel with signal. Check the shape. You should now have a 2D array.

4. Invert the contrast. Consider the `dtype` of the array. Watch out for integer overflow!

5. Subsample the image by a factor of 2, i.e. select every other pixel in both x and y. Check the shape of the resulting array.

6. Display the image again in a new figure window. Use the `gray` color map.

7. Save the image to a .png name `nissl_gray.png` .

pandas

The "Stroop" psychophysical task involves presenting the name of a colour (e.g. "red", "green", "blue") in various colours, either congruently (the word "red" written in red) or incongruently (the word "red" written in blue). In this case, the task of the subject was to report the written color name, regardless of the color it was displayed in. Response accuracy is generally lower and response time is generally slower when the stimulus is incongruent. See https://en.wikipedia.org/wiki/Stroop_effect

`stroop_data.csv` contains results from experiments on multiple subjects. Each row in the file represents one trial from one subject, and each subject has a unique subject ID. Reaction times are in ms.

1. Load the data in `stroop_data.csv` into a pandas DataFrame.

2. How many trials were there in total across all subjects?

3. Group the data by subject and call `.describe()` on it to get a full summary by subject. How many subjects were there?

4. Find the average reaction time across subjects.

5. Plot (with labels) the distribution of reaction times of all trials in the data. Is it normally distributed?

6. Plot the distribution of the **log** of the reaction times. Hint: specify an appropriate set of logarithmic bin edges using `np.logspace()` , and then use `ax.set_xscale('log')` to change the display to also be logarithmic. Does the log distribution look normal?