

Homework 3 1D Array Analysis and Plotting

November 8, 2016

1 Import Packages

```
In [6]: import numpy as np
import matplotlib.pyplot as plt
import os
from os import path
```

2 Load Each Data File into NumPy Arrays

These files are found inside the “Homework 3” Directory, which can be downloaded from the github link. The data was taken from a publicly-available dataset from Boston College. In the study, students performed the Stroop Task (https://en.wikipedia.org/wiki/Stroop_effect), where they needed to say, as quickly as possible, the **color a word was printed in**, and ignore the **word itself**. There are many subjects in this dataset, divided up into files that each contain a single array for a given experimental variable, with elements for every single trial the subjects performed: - ‘subject.txt’: subject numbers - ‘trialnum.txt’, trial number for a given session - ‘stimword.txt’, what the stimulus word said (i.e. “Y” if the word was “YELLOW”) - ‘stimcolor.txt’: what color the stimulus was (this is what the subject was asked to report) - ‘respcolor.txt’;: what color the subject reported.

- ‘reaction_time.txt’: response time (msecs)

Please use the `numpy.loadtxt()` function to load these files. The length of each of these arrays should be equal to each other.

Note: for files containing strings, you’ll need to set the dtype to `str` in the `loadtxt` function.

```
In [7]: dir_name = path.join('Data', 'Stroop Data')
dir_name
```

```
Out[7]: 'Data/Stroop Data'
```

2.1 How Many Trials were there in total across all subjects?

```
In [ ]:
```

2.2 What was the Maximum number of trials any subject did? What was the Minimum?

```
In [ ]:
```

2.3 How Many subjects (i.e. Unique Subject IDs) were there in this study?

In []:

2.4 What is the mean response time, across all subjects and conditions?

In []:

2.5 Plot a histogram of the response times

(Don't forget to make sure that `%matplotlib inline` has been run already in this notebook, or the plot may not show up!)

In []:

2.6 Log Transform the Response Times

You should see that the responses are not normally distributed (if you can't see this, try using a higher number of **bins** in your histogram function). While this isn't really a problem, it's very convenient when you have normally-distributed data. Let's try transforming it to get something a bit nicer for statistical analysis...

- Make a new response time array that is the **log** of the response times.
- Plot a new histogram!

In []:

2.7 Reject Bad Data

Now, that's a little nicer! But wait... there's are some trials that seem to be **outliers** (data points that someone decides aren't representative of the data set): those with **log response times less than 5.7**. After much consideration and deliberation with my supervisor, I've decided to remove those trials from my whole analysis.

Let's do this in a few steps:

2.7.1 Select the the trials that we want to Keep (Not the outliers)

(Hint: You will be making a boolean array here)

In []:

2.7.2 Use Logical Indexing to get the response times from those trials into a new array, effectively removing the bad response times.

plot the histogram of the new log response time array to be sure those trials are gone.

In []:

2.7.3 Remove the bad trials from all of your other arrays, too.

Important: Confirm that all of your arrays have the same length after filtering before moving on, so you can be sure that each index corresponds to the same trial!

Important: It may be wise not to overwrite your old variable names. While not a problem in principle, it can lead to issues when you try to re-run the same cell of code.

```
In [ ]:
```

2.7.4 From here on, use the Rejected data!

2.8 Overall, what was the accuracy of the responses?

- How many trials in total were correct (stimcolor matched respcolor),
- how many were incorrect, and
- what percent accuracy was there, overall?

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

2.9 Matched vs Unmatched Colors:

- What was the mean log response time for Congruent trials (the stimulus word and color matched each other)?
- What was the mean log response time for Incongruent trials (when they did not match)?

```
In [ ]:
```

```
In [ ]:
```

2.10 Make a Bar Graph of Log Response Time

The bar graph should show Mean Log Response Times for the Congruent and Incongruent conditions, with Standard-Deviation Error Bars. Put text labels on the x axis showing which bars go with which condition.

For help and an example, take a look at http://matplotlib.org/examples/api/barchart_demo.html

```
In [ ]:
```

3 Make a Bar Graph of Response Time

The logged data doesn't make a very good impression of just how big an effect this is, sadly. Let's un-log it by taking the **exponent** (numpy.exp) of the logged response time data, and plot that bar graph to get the data in milliseconds!

```
In [ ]:
```

4 Change in Performance over Time

In experiments like this, we assume that each trial is independent of each other. Often, this isn't the case, and not taking that into account can sometimes lead to some wrong conclusions. Let's see what we can find...

4.0.1 How did reaction time change over the course of each subject's session?

Were earlier trials faster or slower than later trials in a session, overall? Make a scatter plot of the response times over the course of a session!

```
In [ ]:
```

4.0.2 What about between each subject? Do all subjects show this difference in response time between the Congruent and Incogruent conditions, or is this just coming from a few subjects?

Make two arrays, one for each condition, which contains the mean log reaction times for each subject for that condition.

```
In [ ]:
```

Plot a histogram of the difference in mean log reaction times between conditions for each subject!

```
In [ ]:
```

What do you think, should anyone be rejected?

4.1 Participant Response Speed vs. Size of Stroop Effect

Make a scatter plot showing the relationship between how quickly subjects responded for congruent and incongruent stimuli!

```
In [ ]:
```

That's an interesting effect, don't you think? Anyway, I think that's enough analysis. Let's publish that last figure and move on!

4.2 Save Figure to File

Save the figure using `fig.savefig()` to your data directory as a png file! Great job!

```
In [ ]:
```