Introduction to Matlab and Psychtoolbox (P657)

Tuesdays & Thursdays, 11:30A - 12:45, Synchronous
Zoom Location: https://iu.zoom.us/j/96254085856?pwd=UEhndER5RmV5NWV4eG1UT1Vsa1B1Zz09
Zoom Meeting ID: 962 5408 5856
Zoom Password: P657
Note: Login with your iu.edu SSO (single sign-on) account

Instructor:	Dr. Dan Kennedy
Contact:	dpk@indiana.edu
Office Hours:	Wednesdays, 2-3 https://iu.zoom.us/j/96254085856?pwd=UEhndER5RmV5NWV4eG1UT1Vsa1B1Zz09 **Meeting ID:** 962 5408 5856 **Meeting Password**: P657

Course Objectives:
The goal of this course is for you to become comfortable programming in MATLAB & using Psychtoolbox (PTB). This course is not intended to provide an exhaustive background covering all the things that MATLAB & PTB can do, but rather, it is intended to give you the foundation to get you started and on which to build upon in the future. By the end of the semester, you should be able to (and actually will) carry out your own behavioral experiments and preliminary analyses in MATLAB from beginning to end. This includes stimulus presentation, data collection, data organization, basic data analysis, and creating publication-quality figures -- all within MATLAB.

Course Format:
The best way to learn MATLAB is by actually using it to solve problems. Therefore, you will have lots of in-class problems to work on, some homework, and a final project.

Unless we're reviewing a homework, most days will generally begin with me providing a tutorial on some aspect of MATLAB - for example, how to use PsychToolBox (PTB) to display images on a screen. Following this tutorial, I will give you some example problems, and you will usually be given some time to work through those problems during class. You might be asked to work in small groups at times (especially for more complex problems).

There are often different and creative ways to solve problems. So, after you've had a bit of time to work on in-class problems, we will regroup and go over a few examples of solutions you came up with, discuss alternative strategies and tricks that the class might have used, and go over various problems that were encountered.

The class is intended to be ***highly interactive*** - questions, comments, and discussion are strongly encouraged -- otherwise, it's very hard for me to know what you understand and what you don't. You should expect to learn not only from me and on your own, but also from each other, especially when going over homework, and during in-class group work and the discussions that will follow.

And, you should be very open to *sharing incorrect answers.* These are especially helpful for learning, not only for you but for the entire class, because then we can altogether try to debug the code and/or figure out more efficient and effective solutions.

Note about Online Platform: I prefer if everyone leaves their cameras on so we can easily communicate back and forth and closely simulate the classroom environment. Also, we will be able to share screens easily via

Zoom, which will make it very easy to help each other debug problems and share alternative strategies. This is one of the best ways to learn how to program, and so I see this actually as a big advantage over the traditional classroom.

Do Not Share Course Materials (including videos, images, slides, etc.). I hold the exclusive right to distribute, modify, post, and reproduce course materials, including all written materials, study guides, lectures, assignments, exercises, and exams. Some of the course content may be downloadable; however, access is limited to students in the class (and perhaps my future classes). You should not distribute, post, or alter any of this material. While you are permitted to take notes on the online materials and lectures posted for this course for your personal use, **you are not permitted to re-post in another forum, distribute, or reproduce content from this course without my express written permission**. That includes Twitter, Youtube, Instagram, TikTok, Messenger, email, and so on.

Grading/Evaluation:
I assume you're attending the class because you want to learn MATLAB. This requires time, effort, attention, and practice. It is important that you attend each class, as each class builds on previous classes (and ultimately culminates in an independent final project which will require you to put everything you've learned together). You will know if you're not keeping up with the rest of the class - for instance, you won't be able to solve in-class and take-home assignments on your own. If you feel you're falling behind, it's your responsibility to let me know and we can figure out how to get you back on track.

Your grade will come from your participation/sharing in class (20%), take-home assignments throughout the semester (40%), and your final project (40%). Your final project will be to create and run a complete experiment from start to finish, fully conducted in MATLAB. It should include scripts and functions used for data collection, data organization and analysis, and the creation of publication-quality figures. You can generate fake preliminary data that you can analyze by running yourself through the experiment a few times, or perhaps we will ask each other in the class to serve as pilot participants. Because you are all busy graduate students, this project ideally should be something you want to program for your own research (just check with me for suitability), or I can suggest ideas. In the end, you will be asked to submit all the necessary files (.m and .mat files, plus any stimulus files, etc.) necessary for me to run the experiment and analysis on my own, along with the figures saved as publication-quality pdf files.

Keep in mind that there may not be a single solution to many of the problems I assign, and as you get more comfortable with various MATLAB functions, the approaches taken will diverge even more. That is OK and expected -- just because you programmed something differently from someone else or myself does not mean that you are wrong.

However, solutions can differ in their elegance -- how efficient they are, how straightforward/clear/easy-to-follow the programming is, how creative (when creativity is necessary) -- and how much they rely on the tools and principles we just learned. Grades on your homework and final project won't just reflect getting the job done, but will also reflect the approach. If you arrive at the right answer, but had to take a very obscure and inefficient route to get there (e.g., very repetitive, brute force code that is inefficient to program or run; hard-coding lots of variables, etc.), that's not as good as a straightforward solution.

This may seem somewhat abstract at this point, but will become clearer as the class progresses. To provide one quick example, say I ask you to create a list of numbers from 1-100, skipping every even number. A good solution would be to use a single line of MATLAB code (actually, you can do this with as few as 7 characters![1]) to create this list. A very bad way would be to type each individual number (1, 3, 5, 7, ... and so on). An inefficient and less elegant and more programmatically difficult way would be to create the whole list from 1-100 and then search for and remove each even number. It's inefficient in two senses -- both in the time it would take to program it, and the computational time it takes to run compared to the good solution mentioned above (the latter will be imperceptible to you, but these inefficiencies can add up and start to really matter within more complicated scripts).

Computer and Software: Please install MATLAB before the first class — it is free for students/faculty/staff at IU (https://iuware.iu.edu). You will also need to install some other free software (PTB3, GStreamer), but we will go over this in class later in the semester.

Resources (a book is *NOT* required):
You don't need to buy a book to learn MATLAB. This class will be enough to get you going. But, if you really want one, then I'd recommend the Rosenbaum book (2nd edition) listed below. I learned, in part, from the first edition of this book -- they've since added several nice chapters in the second edition, including psychtoolbox. I don't have as much familiarity with the Stormy Attaway book, but some people like it so I'm listing that one here as well. There are also a few others that might be OK but I have even less familiarity with those (e.g., for neuroscience, etc.). Again, you don't need to buy a book. Finally, perhaps the best reference is the Mathworks website. They have tutorials, messageboards, webinars, blogs, etc. You may find some of this helpful at some point. Google also knows a lot.

MATLAB for Behavioral Scientists, Second Edition. by David A. Rosenbaum
http://www.amazon.com/MATLAB-Behavioral-Scientists-Second-Edition/dp/0415535948

Matlab: A Practical Introduction to Programming and Problem Solving. by Stormy Attaway.
http://www.amazon.com/Matlab-Practical-Introduction-Programming-ebook/dp/B005DI9M44/ref=pd_sim_kstore_1

http://www.mathworks.com/matlabcentral/

Statement for Students with Disabilities. Every attempt will be made to accommodate qualified students with disabilities (e.g. mental health, learning, chronic health, physical, hearing, vision neurological, etc.) You must have established your eligibility for support services through the appropriate office that services students with disabilities. Note that services are confidential, may take time to put into place and are not retroactive; Captions and alternate media for print materials may take three or more weeks to get produced. Please contact Disability Services for Students (https://studentaffairs.indiana.edu/student-support/disability-services) at iubdss@indiana.edu or 812-855-7578 as soon as possible if accommodations are needed. The office is located on the third floor, west tower, of the Wells Library, Room W302. Walk-ins are welcome 8 AM to 5 PM, Monday through Friday. You can also locate a variety of campus resources for students and visitors that need assistance at: http://www.iu.edu/~ada/index.shtml

Tentative Class Schedule (very likely to change):

[1] 1:2:100

Week	Dates	topic
Week 1	Aug 24 & 26	Introduction, directories, variables, syntax, basic unix commands
Week 2	Sept 1 & 3	Vectors and Matrices; Indexing; Randomization
Week 3	Sept 8 & 10	Looping (for, if/then, while, case), writing scripts
Week 4	Sept 15 & 17	Functions
Week 5	Sept 22 & 24	Nesting Loops; Error handling
Week 6	Sept 29 & Oct 1	Data types (cells, structures)
Week 7	Oct 6 & 8	Debugging / String Operations
Week 8	Oct 13 & 15	Reading/writing text files
Week 9	Oct 20 & 22	Plotting
Week 10	Oct 27 & 29	Psychtoolbox - Displaying Text and Images
Week 11	Nov 3 & 5	Psychtoolbox - Playing Movies / Recording responses
Week 12	Nov 10 & 12	Building a complete experiment and Additional topics TBD.
Week 13	Nov 17 & 19	Putting it all together -- creating a complete experiment; [And, start on your own project]
Week 14	Nov 24 & 26; no class (Happy Thanksgiving!)	
Week 15	Dec 1 & Dec 3	Putting it all together -- finish your experiment & collect data
Week 16	Dec 8 & 10	Putting it all together -- analyze data / create figures
Finals Week	no class - Final projects due on Thursday (Dec 15) by 5pm.	