fMRI Python Discounting Task Manual

Mikhail Koffarnus, PhD
Associate Professor
University of Kentucky School of Medicine
koffarnus@uky.edu

Links to program script and associated files, including this manual:

- Dr. Koffarnus' onedrive repository: https://goo.gl/l8ZbhB
- Githum repository: https://github.com/micned/IDT

What awaits you here

Within these folders are two discounting programs I've written in the python programming language that I've made available for use. Both tasks depend on a number of python add-on modules as well. All of these are free and I've uploaded the files here for your use. Before you use either discounting task, install python and the associated modules.

Python installation instructions

System requirements:

- A Windows computer. I'm not sure what the minimum system requirements are, but they're minimal. You won't need a very fast computer.
- A display with a horizontal resolution of at least 1024 pixels (this includes essentially all monitors and displays).
- A user account with administrator privileges.

Python 2.7 and all of the necessary dependencies are in the "Install first", "Install second", and "Install third" folders, each of which contains one or more executable files. Each of these executable files should be separately installed in the order indicated by the folder names. Installation programs within a folder can be installed in any order. Files can be downloaded one at a time and installed, or entire folders can be downloaded at once. If you download a whole folder, OneDrive will automatically compress it as a .zip file and you will need to extract the files within before installing them. To install the psycopg2 package in the install second folder, copy the whole psycopg2 folder to your C:\Python27\Lib\ directory (this path may be different if you specified a different installation location when installing Python in the first folder).

Note: On some computers, errors are generated during installation of some of the python modules. If you get this, just click 'ok' or 'continue' or whatever and don't worry about it. I'm not sure what causes these errors, but they don't seem to interfere with the tasks.

The discounting tasks

The tasks can be run by double clicking on them. The out-of-scanner program ("Out-of-scanner discounting.py") needs to be run before the in-scanner program ("In-scanner discounting.py") will work. This can be done immediately before the in-scanner session or on a different day, although it is best if

these are closer together so the out-of-scanner discounting rate estimate is accurate. If the programs are in the same folder on the same computer or run from a mutual flash drive or network share, nothing needs to be done for this cross-talk to happen. If the programs will be installed locally on separate computers, then some file transferring will need to be done for each participant. To do this, the .p file that is generated by the out-of-scanner program located in data\subjID\ will need to be copied to a data\subjID\ folder on the in-scanner computer. The algorithm for the out-of-scanner task was developed by others and is described in this paper: Du, W., Green, L., & Myerson, J. (2002). Cross-cultural comparisons of discounting of delayed and probabilistic rewards. *Psychological Record, 52*(4), 479-492.

I wrote each task to be flexible and accommodate a variety of experimental questions, so each task will initially present options to configure these settings. The same subject ID and session name should be used for the in-scanner and out-of-scanner sessions (this is how the in-scanner program finds the out-of-scanner data). After ID and session name, all of these replicate common forms of discounting tasks found in the literature, and each response should be one of the numbers in the brackets:

- Subject ID: Put whatever you want here to identify the subject, but limit yourself to the first ASCII 128 characters and don't use characters that cannot be used in windows filenames (i.e., ~, *, #, %, &, *, {, }, \, :, <, >, ?, /, +, |, or ").
- Session: Put whatever you want here too with the same character limitations as above.
- Delay [0] or probabilistic [1] discounting: Put a 0 if you want delayed discounting, put a 1 if you want probability discounting.
- Gains [0] or losses [1]: This will determine if the task presents the participant with gains of the commodity [0], or losses of that commodity [0].
- Future [0] or past [1] events: This only applies to delay discounting. Discounting of future, delayed events is more common [0], but discounting of events that occurred in the past can also be selected [1].
- Implicit [0] or explicit [1] zero: Implicit zero [0] is the default here and will present choices between two amounts of money, one immediate and one delayed. Explicit zero [1] reframes the questions so the same amounts and delays are used, but informs the participant that none of the commodity will be received at the delay not chosen.
- Cross-commodity discounting [y/n]: The answer to this should be a 'y' or an 'n'. If you select 'y', then the task will be a cross-commodity one where the immediately available and delayed commodities can be different. If you select this, you will need to specify separate immediate and delayed commodities and amounts. The amounts you specify should be the amount that you want to be discounted for the delayed commodity and the equivalent amount for the immediate commodity. You'll need to determine this equivalent amount separately.
- Commodity(ies) with unit: This should be the thing that is being discounted. If you want to use money, just put a dollar sign ($). The same character limitations as for subject ID apply.
- Delayed or probabilistic amount(s): Put the larger-later amount here. This should be a number (decimals are ok).
- Stop at the end on the Task Complete screen [y/n]: If you put a 'y' here, the task will display a "Task Complete" screen at the end until the user presses 'escape' to exit. If you put an 'n' here, the task will just end after the final choice trial.

- Custom delays or probabilities [y/n]: Put 'n' here to use the standard delays. If you want to use different delays, put 'y' here and enter the delays you want in the next few prompts exactly in format described in the prompts. If this task is being used in conjunction with the in-scanner task, custom delays should **NOT** be used.

The Starter Scripts

I wrote this program to be flexible with many settings, which you will be prompted to enter when you start it by double clicking on the 'Out-of-scanner discounting.py' and 'In-scanner discounting.py' scripts. Included in the folder are two additional optional scripts, 'Out-of-scanner starter.py' and 'In-scanner starter.py', which can be edited and used to automate the running of the main scripts with a given set of parameters that meet your needs. In the form that I've included them, each will run it's respective task as a $1000 delay discounting task with future gains and implicit zero. There are some brief instructions within this script and it can be edited with any plain text editor like Notepad (I recommend Notepad++ though, which is a free program). In python scripts, everything after a # sign in a given line is a comment that I've included to describe how to edit the script.

Frequently Asked Questions

Q: What instructions do you read to participants?

A: This varies somewhat depending on the specific experiment and task parameters, but here is an example of instructions for monetary discounting tasks with future gains:

> *"You will now complete a decision-making task. You will be asked to make choices between different amounts of money given to you now or after a delay. These are hypothetical choices, but please choose your answer as if the items were to be delivered as described. Each task will start with some brief instructions on the screen. Read these instructions, and press the 5 key on the keyboard when you are ready to begin. There are no right or wrong answers in the tasks, just choose which option you prefer in each case. Please take your time and answer thoughtfully. To select the option on the left side of the screen, press the left arrow, and to select the option on the right side of the screen, press the right arrow."*

Q: How do I make responses?

A: Press the 1 key or the left arrow on your keyboard to choose the left option and the 6 key or the right arrow for the right option. The 5 key starts each task. The participant or research assistant can press this for the out-of-scanner task. For the in-scanner task, this '5' should be initiated by the MRI to sync the task with the fMRI data.

Q: How can I stop the task?

A: To stop either task before it is complete, press the Escape key on your keyboard.

Q: How is the session data stored?

A: When you run either task, it will create a 'data' subfolder in the location that the task is run, and a subfolder within that matching the current subject ID (for this reason, you should only use characters in your subject IDs that can be used in Windows file and folder names). Data files for each Subject ID will be stored in their own folder, which will be automatically created by the programs if they're not already there. Both tasks store data files within these folders for each task run as .csv files. This file type is a text file of comma-separated values and can be opened in Excel. Each data file contains some session

information at the top of the file, trial-by-trial choice information for each choice trial, and session summary information at the bottom. For the out-of-scanner task, this summary information includes indifference points at each delay (expressed as a proportional value relative to the larger later amount), *k* value determined by nonlinear regression fit of Mazur's (1987) discounting function through the indifference points, and rule violations of the rules proposed by Johnson and Bickel (2008, Experimental and Clinical Psychopharmacology, 16, 264-274). The number that appears after "JB1" refers to the number of violations of the "bounce" criterion proposed in that paper and the number by "JB2" indicates whether the proposed "trend" criterion was violated. Whether either of these criteria were violated also appears on the screen at the end of the task.

Q: I'm using a Mac and this doesn't work.
A: Although Python works on Windows, Mac, and Linux, these programs are Windows specific.