**Adaptive auditory discrimination task – Manual**

*Introduction*

In the field of cognitive psychology, the auditory discrimination task is used to in order to examine a couple of different phenomena. This experiment focusses on the threshold where two sounds become so alike in frequency that people can not distinguish between them anymore. The task presents two sounds after one another and then asks the participant whether or not they’ve heard a difference. One of the main differences in this task however is the fact that this task adapts its difficulty to the estimated ability level of the participant. With the task comes a .csv file with a large list of items that vary on difficulty. Every item has an id, a difficulty score and a frequency for both sound 1 and 2 ranging from 475 Hz to 520 Hz. The task pools from this list and after every trial re-estimates the ability of the participant, after which it pools a new item that comes closest to the participant’s ability. In IRT-modelling, a difficulty that more or less corresponds with the ability of the participant would relate to a probability of about 50% of answering correctly. So with every new trial the outcome is uncertain. The updating system that is used is derived from ELO ranking system in chess (Klinkerberg, 2011). The Formulae that are used to update both are as follows:

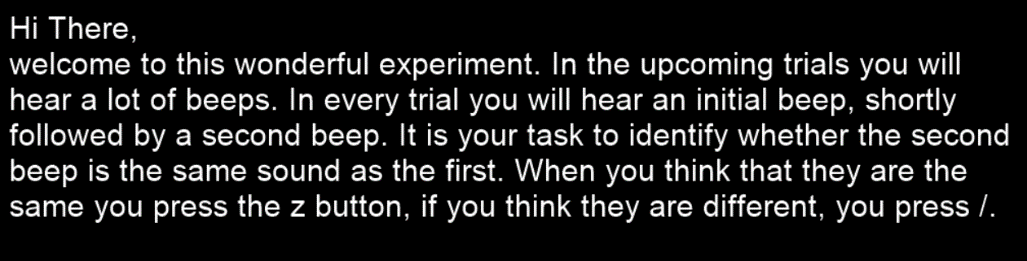
Where is the ability of participant *J*, the difficulty of item *I, S* the outcome of the trial (0 for wrong and 1 for right), and *E(S)* the expected outcome based on the following formula:

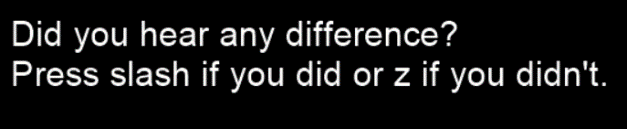
Lastly, *K* and *C* are constants that can be implemented based after own insight (in this case). A good example of *K* is to make it correct for frequency that an item has occurred thus taking into account that the uncertainty might decrease over time. In the experiment *K* has been set to one. *C* can be used to scale the scores. In chess, rankings can differ hundreds of points between players. Therefore *C* is set to 400 to scale back. In this experiment it is set to one.

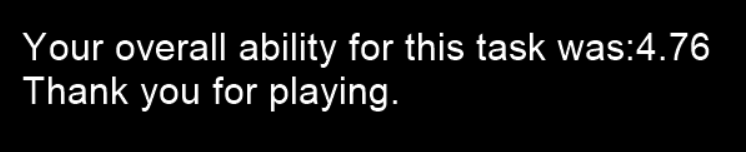
*Flow of the experiment*

The experiment starts with an opening text that explains the task (example 1). After pressing the task starts with presenting two beeps after one another. After these a small text asks the participant whether they’ve heard a difference and reminds them of the action options (press z for no difference and / for difference, example 2). This repeats for a certain amount of trials after which they are presented a thank you message along with their ability estimate (example 3).

*Examples:*







Behind the scenes everyone starts with an ability of 0.5. The item that is closest to 0.5 in difficulty is selected from the .csv file and the frequencies of the sounds are used to generate new sounds. When the participant selects an answer a new ability for the participant and a new difficulty for the item are calculated. The item is stored in a separate data frame and removed from the list so it can’t be presented twice in one experiment. This repeats for a preset amount of trials, after which the used and unused trials are combined and saved in the .csv file. The end text is presented and the experiment is over. The whole process is shown in the next figure.

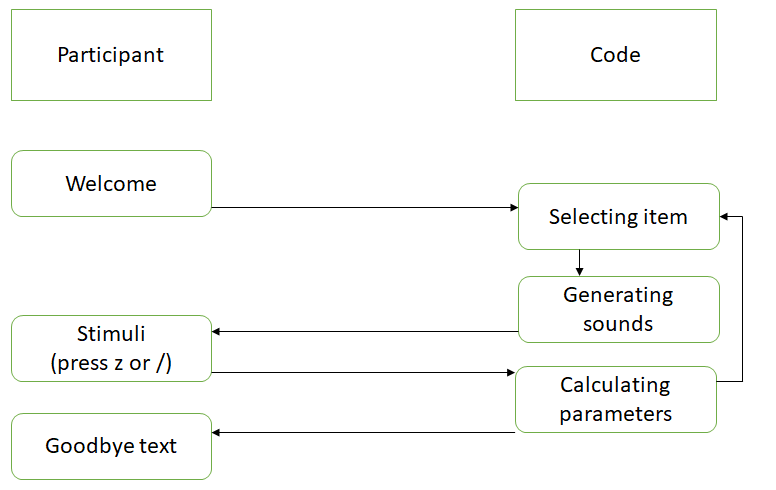


Figure 1. Flowchart of the experiment with user perspective (left) and mechanisms (right)

*Installation and use*

In order to run this experiment python 2.7 is required. It might be interesting to download the Anaconda navigator with the correct python version and use the Spyder builder for Psychopy. A couple of recommended package versions are also included in the code itself. In order to install these package versions you can go to the Anaconda prompt and use the command: pip install package-name==version-number. Moreover, it is necessary in order to use the list of difficulties that you have this file in the same folder as the experiment. You can select the folder by using the os.chdir(“path name”) command in the code. In order to tweak the experiment to your preference you can set different parameters like the amount of trials or the participant number in the code itself. The output will appear as a text file named participant number\_data.txt and consists of a data frame including item id, difficulty of each used item prior to use and ability after each trial. The output will be saved in the same folder as the experiment and the difficulty list are in.