Experimental Design and Analysis Review (EXDAR)

Designing and programming experimental paradigms is hard work that requires enormous attention to detail. There are many different things that can go wrong, at many different stages of an experiment. “Small” mistakes made early and not caught can cascade. In the best case scenario, these mistakes “only” cost tens or hundreds of hours of our time, and tens or hundreds of thousands of dollars. In the worst case scenario, they can lead to artifactual/non-reproducible results, retractions, and other headaches.

The purpose of a data review is two fold: 1) to ensure that all experimenters and PI are on the same page regarding task parameters, and 2) to provide consistent data checks before and during subject acquisition.

What are the stages of an EXDAR?

Phase 1: Experimental Design. At this stage, we are finalizing our experimental design - what we’re manipulating, what we’re measuring, and trial structure. The primary purpose of this stage is to ensure that we’ve operationalized our question in a way that will allow us to answer it, and to confirm that experimenter and PI are on the same page before moving forward.

Phase 2: After development, before pilot data acquisition. At this stage, experimenters and PI should be reasonably certain about what the task looks like. The purpose of phase 2 is to verify the experiment script file (i.e. that stimulus presentation and outputs produced by the experiment script file comports with expectation and project needs). In this phase, experimenter completes the scripting checklist, and compiles a task overview and trial sequence that is reviewed with PI.

Phase 3: Interim review during piloting. *After each subject is acquired*, experimenter compiles a task overview and trial sequence for PI. If fMRI, this is done for each run and as a concatenated “total.” The purpose of Phase 3 is to make sure that the data that we want is acquired consistently across pilot subjects, to enable the experimenter and PI to make changes to design as needed, and to confirm that that any agreed-upon pilot version changes are rapidly and appropriately incorporated.

See below for details of Phases 1-3

Phase 4: Post-piloting, before active data acquisition. The experimenter compiles a summary of the pilot data that will enable us to determine if desired manipulations affect dependent variables in predicted direction.

Phase 5: Interim review during active data acquisition. Experimenter compiles individual subject reviews after each of the first 5 subjects, and summary reviews every X subjects thereafter.

Phase 1: Developing an Experimental Design

Task Parameter Overview

Include a narrative summary of the task: why we're using it, what it measures, what the IVs and DVs are etc.

- Name of task:
- Type of design: (e.g. w/in subjects, between subjects, mixed)
- Task conditions:
    - -{Condition 1 name}: (description)
    - -{Condition 2 name}: (description)
- Levels of each condition:
    - -Condition 1, Level 1 (description)
    - -Condition 1, Level 2 (description)
    - -Condition 2, Level 1 (description)
    - -Condition 2, Level 2 (description)
- Are conditions are based on continuous variables that the experimenter manipulates between trials (e.g. reward delay, reward magnitude)? If so, which ones?
    - -How are these determined for each trial (fixed or randomly generated based on formula?)
    - -If applicable, what *should* the means of these variables be for each condition?

  Is trial order randomized? If so, how - on the fly? Predetermined run-orders counterbalanced across subjects?

- Total number of trials:
- If trials are blocked, how many blocks:
- If trials are blocked, how many trials/block:
- Trial number breakdown table:

	Condition 1	Condition 2

Level 1	25	25
Level 2	25	25

- Total length of trial:
- Trial sequence
    - (*include slide that shows what the subject sees, with timing, for a given trial)
    - -Part 1, XXms
    - -Part 2, XXms
    - -Part 3, XXms
    - mean ISI:
    - range of ISI:
    - How ISI determined:

For fMRI Tasks, add:

- How many runs:
- How many TRs/run:
- How long each run:
- Trial number, TR and timing breakdown table *total* and *by run*

Phase 2: After development, before pilot data acquisition.

Experiment Scripting Checklist

In general, you should get in the mind-set of checking your script against actual reality rather than merely double-checking for coding mistakes. For example, rather than simply looking at the code to ensure that the proper number of trials is being displayed, create a manual record of events and check it against the output file. Likewise, rather than look at the data output to see how long a script runs, use a stopwatch to time the actual fact of the matter. It doesn't matter whether your script says it's timing to the millisecond, if in reality it's not! Your script cannot be considered ready until at least the following criteria are met:

1. Scripting output checklist

○ Does the script record a data file that contains all the information needed to analyze data later? The data file should consist of *n* rows, where each row corresponds to one trial of the experiment. Each row should include at least the BIDS specification event information, as well as the following (note: there is currently overlap between BIDS specification column names and below; where there is overlap, use BIDS specification name).

1. Run number
2. Trial number
3. Trigger time (if fMRI study)
4. Condition(s) associated with each trial (e.g., congruent, neutral, incongruent); alternatively, each condition could be represented in a separate column
5. The stimulus presented to subjects (e.g., the code for the images or text displayed to subject)
6. Clock time at the start of the trial
7. Clock time for each meaningful intra-trial event (e.g. “delay” vs. “feedback”) and ISI onset
8. Expected length of the event (e.g., 4000 ms or the length of the NULL event)
9. Subject response
10. Subject reaction time, calculated as the difference between the clock time when the trial starts and when the subject makes a response.

Does the script record a data file that contains all of the relevant participant info?

At a minimum, this should include::

1. Subject ID
2. Subject sex
3. Date on which the data were collected

· Does the data output match what actually takes place during the run? Scripts have been known to record a list of trials that differs from what subjects actually see. To ensure that your data output corresponds exactly to the trials shown to participants and outputs the actual responses they make, you need to manually record what happens during a run and compare it to the data output file. For example, you would record in a notebook that the trials shown by the script went [cong, incon, cong, incon, incon, go, nogo...] alongside the button presses you made to each trial. Details regarding stimulus presentation (e.g. does the stimulus go away when the subject responds?) should be noted as well. One may need to extend trial length and reduce the number of trials for testing in order to make this feasible. Then, you would compare what you know actually happened (from your notebook) to what the data output records as having happened. If these are not identical, something important is wrong. If possible, it may be useful to collect a screen

recording while you perform the task, so that you can go back and review details more easily.

For all tasks:

You should try to "break" your data output by testing various situations that can arise when the experimenter or the subject acts in a nonstandard fashion. For example:

- What happens if the experimenter attempts to re-use a subject code that has been used previously? The script should not overwrite existing files (for obvious reasons), but instead append a modifier to the filename or ask for a unique subject code.
- What happens if the subject does not respond in time? The script should nevertheless write the trial to the data output, and should record *something* in the response column. In general, it may be better to record something that is a different variable type as true responses; e.g., if your response column comprises integers from 1 to 5, it will be easiest if no-responses are recorded as N/A.
- What happens if a subject responds after the stimulus disappears? Is the response collected?
- What happens if the subject responds twice during a single trial? Is the initial response overwritten or is the second response ignored? If there are multiple responses and only one is recorded, it should be the first response.

For fMRI: What happens during NULL (fixation) events? Data should be recorded correctly during NULL events: at a minimum, the same number of columns should be written out during NULL events as during experimental trials. In addition, you should check what happens if a subject responds during a NULL event. Is the response recorded or ignored? Under either scenario, ensure that the event still times correctly.

To establish that your data file includes the minimum information needed to analyze your imaging data, you should run a "dummy" analysis in which you (i) run yourself as a subject or use simulated fMRI data; (ii) read the data file into SPM using the scripts that you plan to use for the actual data analysis; (iii) use the resulting onset vectors to model and estimate an arbitrary set of BOLD images collected for an earlier experiment. The results will obviously be garbage, but this will establish that your data file can be used to analyze your actual data.

2) Randomization and Trial Counts

- Does the script run the correct total number of trials? Count them, both in your data output and in your hand record.
- Does the script run the correct number of trials of each type? For example, if you want each run to include 20 Go and 20 NoGo, does your script actually do that? You will need to count the number of trials of each type in your output file or in your hand record of events.

· Is each stimulus item appearing the expected number of times? For example, if you have 100 items and each is supposed to appear once and only once, is it the case that (i) all 100 are used and (ii) no item is used more than once? This is likely best done by sorting the output file(s) in Excel and writing a macro to compare cells in adjacent rows of the same column.

· Does the script obey any additional randomization constraints? For example, if you have an experiment that includes two presentations of each stimulus item, does each item appear once before any item appears a second time? When the item appears a second time, does it do so in the proper condition (e.g., if an item appeared first in a 'Go' condition, you may want it to appear in the 'NoGo' condition the second time... does this happen for each item?).

· Does each run of each subject comprise a unique random order of trials types? Or, if trial order is fixed, is the fixed trial order consistent across subjects/runs? If you have multiple fixed orders that are randomized across subjects, is the script accurately randomizing assignment of trial order across subjects?

· Does each subject see a unique random order of stimulus items? Or, if counterbalanced fixed run orders, are the run orders appropriately distinct? For example, the same stimulus item should not always be shown first.

Note: in general it is preferable if possible to avoid randomization on the fly, and to pre-generate any random sequences that will be used.

3) Timing

· Does the script run for the correct amount of time? The only way to confirm this is with a stopwatch. If the script is longer or shorter in real time than it should be, something important needs to be fixed.

· Is each trial the correct length of time? Generally, this should be checked in two ways. Using a screen recording may be helpful.

  ○ Output file. The data file should record a "time stamp" that records the clock time when the trial starts: are the time stamps exactly *n* milliseconds apart, where *n* is the expected trial length (e.g., 4000). It is not necessary to time at better than the millisecond level.

  ○ Stopwatch. Time a few dozen trials with a stopwatch. These will necessarily be messy (because of the noise associated with your own reaction time), but do the trials appear to be timing as expected?

· Is each part of a trial appearing for the expected amount of time? The only way to confirm this is with a stopwatch.

Note: also, for paradigms where timing is very important, or when using a new device with unknown timing characteristics, it may be necessary to characterize the timing response of the device using an external timer (e.g. an oscilloscope with a light-sensitive diode)

4) *Trial Sequence Check*

Provide trial sequence. If multiple versions (e.g. two fixed sequences counterbalanced across subjects) provide sequence for all versions.

Each row is a trial, with the following columns. Each row should include at least the BIDS specification event information, as well as the following (note: there is currently overlap between BIDS specification column names and below; where there is overlap, use BIDS specification name):

- Trial #
- Condition
- Values of any continuous variables that define conditions (e.g. reward delay, reward magnitude; one per column)
    - If trials are multi-alternative forced choice, continuous variables associated with each option (one column per variable per response option)
    - A column for the difference between the continuous variables for each option; one for each variable
        - e.g. a trial in which someone chooses between $1.50 today and 5$ in two weeks
        - include anything that the experimenter manipulates that varies from trial to trial
- Column for subject response
- Column indicating subject RT
- Column indicating if choice is "correct"
- Column indicating feedback to subject if appropriate
- Column indicating total length of trial from onset to start of ISI
- Column indicating post-trial ISI

For fMRI Tasks, add:

- Trial sequence for each run and concatenated (all runs, each row is a trial)

Phase 3: Interim Review During Piloting.

For each subject, provide the following; give a readout for total and one for each run (if fMRI)

Part 1) *Subject Specific Information and Task Parameter Review*

- Subject#:
- Date run:
- Name of task:
- Version of task revision (if appropriate)
- Version of task sequence (if counterbalancing)

- Task conditions:
  - -{Condition 1 name}: (description)
  - -{Condition 2 name}: (description)

- Levels of each condition:
  - -Condition 1, Level 1 (description)
  - -Condition 1, Level 2 (description)
  - -Condition 2, Level 1 (description)
  - -Condition 2, Level 2 (description)

- Total number of trials:
  - If trials are blocked, how many blocks:
  - If trials are blocked, how many trials/block:

- Trial number breakdown table, e.g.:

	Condition 1	Condition 2
Level 1	25	25
Level 2	25	25

- Are conditions are based on continuous variables that vary between trials (e.g. reward delay, reward magnitude)? If so, which ones?
- -How are these determined for each trial (fixed or randomly generated based on formula?)

Part 2) Summary measures:

- total # of trials
- # trials in each level of each condition
    - -Condition 1, Level 1:
    - -Condition 1, Level 2:
    - -Condition 2, Level 1:
    - -Condition 2, Level 2:
- mean ISI:
- range of ISI (if variable):
- means of continuous IVs for each level of each condition (if appropriate)
- means of RT for each level of each condition
- means of accuracy for each level of each condition (if appropriate)
- means of feedback for each level of each condition (e.g. money won; if appropriate)
- derived/calculated DVs by condition as appropriate (e.g. learning rate, *k*, subjective value)

Part 3) *Trial Sequence*

Provide trial sequence. If multiple versions (e.g. two fixed sequences counterbalanced across subjects) provide sequence for all versions.
Each row is a trial, with the following columns:

- Column for Trial #
- Column for Condition
- Column for Values of any continuous variables that define conditions (e.g. reward delay, reward magnitude; one per column)
    - If trials are multi-alternative forced choice, continuous variables associated with each option (one column per variable per response option)
    - If appropriate, a column for some operation (e.g. difference) between the continuous variables for each option; one for each variable
        - e.g. a trial in which someone chooses between $1.50 today and 5$ in two weeks

Condition	Delay-S	Magnitude-S	Delay-L	Magnitude-L	Difference-Delay	Difference-Mag
S/L	0	1.50	14	5	14	3.5

- Column for subject choice
- Column indicating subject RT
- Column indicating if choice is "correct" (if appropriate)
- Column indicating feedback to subject if appropriate
- Column indicating total length of trial from onset to start of ISI
- Column indicating post-trial ISI

- (Attach Histogram of ISIs; if variable)

Part 4) *Scripting output checklist*.

For each subject, and for each run, check that your script outputs at least the following:

a. Subject ID
b. Subject sex
c. Date on which the data were collected
d. Run number
e. Trial number
f. Trigger time
g. Condition associated with each trial (e.g., sooner/later, later/later)
h. The stimulus presented to subjects (e.g., the code for the images or text displayed to subject)
i. Clock time at the start of the trial
j. Clock time for each meaningful intra-trial event (e.g. “delay” vs. “feedback”) and ISI onsent
k. Expected length of the event (e.g., 4000 ms or the length of the NULL event)
l. Subject response
m. Subject reaction time, calculated as the difference between the clock time when the trial starts and when the subject makes a response.

Confirm that these match expectation!!!