2023 ■ Vol. 19 ■ no. 2

Implementing multi-session learning studies out of the lab: Tips and tricks using OpenSesame

Stephan F. Dahm [a], Eduard Ort [b], Christian Büsel [a], Pierre Sachse [a] & Sebastiaan Mathot [c]

[a]Universität Innsbruck, Faculty of Psychology and Sports Science, Department of Psychology, Innsbruck, Austria
[b]Heinrich Heine University, Department of Biological Psychology of Decision Making, Düsseldorf, Germany
[c]University of Groningen, Department of Psychology, Groningen, Netherlands

Abstract ■ Here, we provide tips and tricks for running multisession experiments out of the lab using OpenSesame, a user-friendly experimental tool that is open source and runs on Windows, MacOS, and Linux. We focus on learning experiments that involve the measurement of reaction times. We show how such experiments can be run with traditional desktop-based experiment software on participants' own notebooks (i.e., out-of-the-lab, but not in a browser). Learning experiments pose specific challenges: accessing individual identifying numbers, accessing session numbers, and counterbalancing conditions across participants. This article includes helpful code and provides hands-on implementation tips that will be useful also beyond the presented use case. The aim of this article is to illustrate how to create multisession learning experiments even with little technical expertise. We conclude that, if done right, out-of-the-lab experiments are a valid alternative to traditional lab testing.

Keywords ■ repeated testing, experimental application tools, serial reaction time task. **Tools** ■ OpenSesame.

Stephan.Dahm@uibk.ac.at

10.20982/tqmp.19.2.p156

Acting Editor ■ Denis Cousineau (Université d'Ottawa)

Reviewers
■ One anonymous reviewer

Introduction

Consider an experiment in which a large number of participants needs to complete multiple sessions spread over multiple days. To complicate matters further, the order of the sessions as well as the order of the trials within each session must be carefully controlled. This scenario (Dahm, Weigelt, & Rieger, 2023; Ferrand et al., 2018; Koch et al., 2018; Verwey et al., 2015) is common in the field of learning research and other fields, and poses a number of unique challenges. To illustrate these challenges, we present an example use case in the context of motor sequence learning (Verwey et al., 2015) where humans' ability to learn a novel motor task was investigated. Here, the time course of the acquisition of motor-sequence representations during extensive practice is assessed (Dahm, Hyna, & Krause, 2023; Dahm, Weigelt, & Rieger, 2023; Dahm & Rieger, 2023). To this end, participants respond to a repeating sequence of stimuli during practice. After the practice phase, participants perform the practiced sequence faster than random control sequences, even if they were not aware of the sequence during practice (Dahm, Hyna, & Krause, 2023; Dahm & Rieger, 2023). The random control sequences are created within the experiment by creating a new sequence for each new block. In comparison with a prebuilt quasi-random control sequence, participants therefore cannot learn these random sequences when performing several random blocks. This is particularly important for control groups that practice random blocks of items (Dahm, Weigelt, & Rieger, 2023). Similar considerations may apply also to other experimental designs.

Traditionally, studies of this kind have been conducted by inviting participants to the lab on multiple days to complete a new experimental session each time. The main advantage of this lab-based approach is that it offers maximum control over the experimental setting, and that the experimenter has direct control over the machines on which the experiment runs. Software solutions for studies in the lab are for instance E-Prime (https://www.scienceplus.com), OpenSesame (Mathôt et al., 2012), Presentation (https://www.neurobs.com), PsychoPy (Peirce et al., 2019), or Psychophysics toolbox (Brainard, 1997). The

Table 1 ■ Pros and cons of lab experiments, online experiments, and the intermediate ‘out-of-the-lab’ approach suggested in the present manuscript.

	Lab approach	Online approach	Intermediate “out-of -the-lab” approach
Control of experimental environment	✓	✗	✗
Control of machines (e.g., hardware, firewalls)	✓	✗	✗
Time saving for experimenters	✗	✓	✓
Flexibility: no travelling for participants	✗	✓	✓
Laboratory resources required	✓	✗	✗
Versatility of software	✓	✗	✓
Automatic data gathering: single data file	✗	✓	✗
No software installation by participants	✓	✓	✗

main disadvantage of this lab-based approach is that it is time and resource intensive: The lab space needs to be available, researchers need to be available to conduct the testing, and participants need to travel to and from the lab. More recently, accelerated by the Covid-19 crisis, a shift has occurred towards conducting studies of this kind online, in which case participants receive URLs to complete the experimental sessions in a web browser on their own computer (Grootswagers, 2020; Sauter et al., 2020). The main advantage of online experiments is that they are not very time or resource intensive: a single researcher can quickly recruit a large number of participants, and no lab space is required. However, online experimenting is subject to several limitations: (1) The functionality that is offered by the browser, which may considerably reduce precision in measuring reaction times (Bridges et al., 2020), (2) functional restrictions of the library that runs the experiment in the browser, such as OSWeb (Mathôt & March, 2022), jsPsych (de Leeuw, 2015), Lab.js (Henninger et al., 2022), and PsychoJS (Bridges et al., 2020), (3) the server that hosts the experiments and collects the data, such as JATOS (Lange et al., 2015) or Pavlovia (Bridges et al., 2020), and (4) the lack of control over the experimental environment.

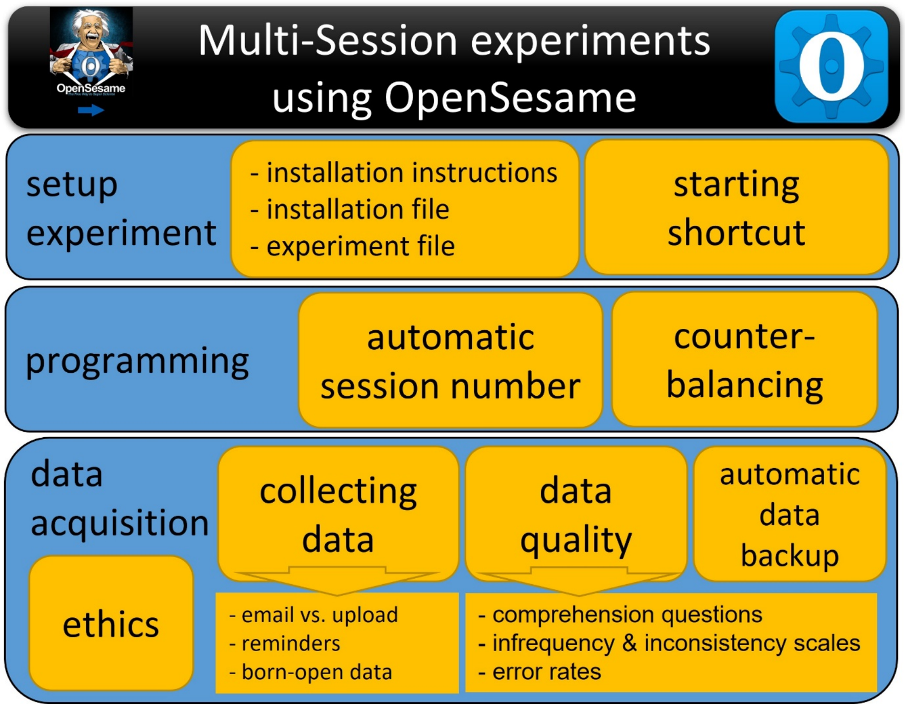
The pros and cons of traditional lab-based testing versus online experiments have been discussed in detail elsewhere (Grootswagers, 2020; Mathôt & March, 2022; Sauter et al., 2020). The goal of the present paper is to introduce a third approach (Table 1), which in a sense is an intermediate approach between lab-based and online testing: To have participants run experiments at home and by themselves using traditional lab-focused software. In contrast to online assessments (Mathôt & March, 2022), the proposed approach has the advantage that there is no internet connection required during data collection. However, a disadvantage is that the installation process and the data collection may be a hurdle for some volunteers willing to par-ticipate in such a study. As we will discuss below, this hurdle can be overcome. The presented out-of-the-lab approach is currently rarely used but we believe that, especially for specific use cases, it offers significant advantages, which we will outline in the present manuscript.

While there are several tools that would allow experiments to be conducted at home by participants themselves, in this article we focus on OpenSesame (Mathôt et al., 2012), which has a number of features that make it especially suitable for this approach: It is free and open source (i.e., everyone can inspect and contribute to its source code). OpenSesame is cross-platform and works on the most common operating systems: Windows, MacOS and Linux (Mathôt et al., 2012). It is easy to use by providing a graphical user interface and drag and drop options for building experiments, but also allows for flexibility by providing a Python interface, such that complex experiments can be implemented. For example, complex pseudorandom sequences that control for trial transitions and cannot be implemented with built-in randomization and constraint functionality (e.g., as provided by ‘loop’ items in OpenSesame) can easily be realized with custom Python code embedded in the experimental file. Therefore, we provide example Python code for the present use-case for motor learning (see points 5, 6, and 7 below).

In this article, we offer solutions for the following issues (see Figure 1):

1. Instructions for participants on how to install OpenSesame on their own computers and how to run studies on it.
2. How to prepare the installation files.
3. How to prepare the experiment file.
4. How to create a shortcut that can be used in every session.
5. How to access data from previous sessions and to use it in the current session (e.g., for automatically setting the session number).
6. How to create a standardized data backup.
7. Techniques for counterbalancing the assignment to groups and the order of conditions.

Figure 1 ■ Depiction of considerations when conducting out-of-the-lab multi-session studies.



8. Practical and ethical issues related to data collection.
9. And issues related to data quality.

The present article addresses several ways in which these challenges can be solved using free and open-source software. While the tips presented here are focused on a specific use-case, most of them are also applicable to other scenarios and are therefore useful for a larger readership. They may not only be helpful when used together, but each solution may also be useful on its own.

In the present use case, we aimed to have three groups practicing a predefined sequence and a control group practicing random sequences. All groups perform short practice sessions spread over ten days.

Challenges and solutions

Instructions for participants to download the software and run the experiment

Many participants have little technical knowledge (e.g., how to install an app or software on their laptop). In this section, we will outline how participants can be instructed to download the software so that they can conduct the experiment themselves at home. We assume that participants first got in contact with the experimenter in some way (e.g., per email) to show their interest in par-ticipating in the study. Next, the experimenter may assign an individual identifying code number (ID or subject number) to the participant and send (links to) experimental files.

From the authors' past experiences only ≈60% of those participants that expressed an initial interest in participating in the study, did in fact participate. The reasons for not participating can be manifold and were not systematically assessed. Still, common reasons are: participants did not manage to install the required software; participants did not want to install any software on their personal notebooks; participants lost their interest after being informed about the large number of sessions.

Importantly, step-by-step installation instructions should be provided to the participants, because not all are familiar with installing apps on their computers. This needs to be done separately for each operating system (e.g., Windows, MacOS, or Linux) as the installation processes differ. Here, we will focus on Windows and MacOS as these are the most common operating systems (>99% of participants in the authors' studies). Installation instructions may start with a brief description of the content and the procedure of the study. Experimenters should be careful in finding the right level of detail for such instructions. While short instructions might not provide sufficient detail to participants to properly set up the study, excessive instructions might bore participants and demotivate them to read carefully, or even worse, cause them to cancel their participation.

We have found it useful in previous studies to specify “dos” and “don’ts” for participation within the installation instructions. For instance, sessions should be performed between 8 am and 8 pm, so that participants do not take part at night (e.g., after a party). No alcohol or other drugs should be consumed during or before participation. The conditions should be as similar as possible for all sessions (e.g., sitting at a table). For each session, 30 minutes without interruptions should be planned. For an example of instructions in German and English, including illustrations see: https://osf.io/xmn6z/.

Providing a download file to install OpenSesame

Although the installation file for OpenSesame can be downloaded free of charge on the website (https://osdoc.cogsci.nl/), we suggest that researchers upload this file together with the installation instructions so that participants can find everything they need in one place. Further, providing the installation file has the advantage that all participants will be using the same version, because on the website the most recent version may change during data collection. This needs to be done separately for each operating system (e.g., Windows, MacOS, or Linux) as the installation files differ.

Providing the experiment file

We found that the most convenient way to provide the experiment files is by putting them on a cloud directory (of the university), such as seafile (www.seafile.com), together with the installation instruction and the installation file. Here, OpenSesame has the advantage that the experiment file (`studyX.osexp`) is one single file that includes all stimuli (visual and audio) together with the experimental script. The same experiment file can be used on all operating systems (e.g., Windows, MacOS, or Linux).

On Windows: Using the shortcut and placement of the experimental file

Instead of opening the experiment file and starting the experiment from within OpenSesame, it is possible to set an icon (shortcut) on the desktop that launches the experiment with a double-click for every session using the opensesamerun utility. By right clicking on the shortcut, it is possible to access the settings. In the settings, participants may set the personal subject number (e.g., `--subject=999`) only once before starting the first session. Using the shortcut has the advantage that subject number and log file name are automatically set. Hence, participants do not need to insert the subject number and define a log file name to save the experiment in every single session. This prevents input errors of participants and guarantees that the subject number is always the same in every session. To place the experiment file, we chose C: because it is a standard path that is available at any computer running with Windows. The logfile is best placed in the same folder to be able to use relative paths in the experiment. In the shortcut settings, the path of the log file can be defined using `--logfile=C:\NameOfFolder`.

Automatically setting the session number by reading existing data files

When conducting multi-session experiments, a subjects’ respective session number needs to be tracked in each result file. Especially when experimental designs differ between sessions, the exact session number is crucial. While it is possible to ask participants to report the session number at the beginning of each test session (e.g., via a single choice or number input in OpenSesame), experience has shown that participants frequently make mistakes when asked to repeatedly provide the same information (e.g., subject number). With respect to session numbers, this could lead participants to skip or repeat sessions by entering the wrong session number. Hence, a reliable and automatic way to set the session number for each new session eliminates this source of error.

So far, a non-trivial way to accomplish this in OSWeb and JATOS does not exist (complex multi-session experiments are possible using OSWeb and the JATOS `batchSessionData`, see e.g., Zhou et al., 2022, however, this currently requires some technical know-how to implement). However, with a local installation of OpenSesame on the participants’ computers, this can be accomplished by asking OpenSesame to use a relative path to access the folder containing the log files from previous sessions. An example code provided in Listing 1 checks which sessions have already been completed by the participant and automatically sets the next session number. The code may be placed in an inline_script at the beginning of the experiment.

If you need to retrieve parameters from previous sessions, rather than merely knowing which sessions have been completed, you can use a variety of formats, including yaml, json, or Python pickles, to store and retrieve data across sessions (Ort et al., 2019).

Saving a backup of the data

Saving the data file (.csv) at the end of the experiment by creating a backup copy (in addition to the original data file) can be useful. First, doing so prevents overwriting data sets that already exist by using the actual date and time as part of the filename, which are almost guaranteed to be unique.

Listing 1 ■ Code to automatically set the session number by checking with already existing data files

```
#Session number
import os

MAX_SESSION = 12

for var.session_nr in range(1, MAX_SESSION + 1):
    # Get the full path to the log file
    log_path = os.path.join(var.experiment_path, "Logfile",
    'ID' + str(var.subject_nr) + '_Session' + str(var.session_nr) + '.csv'
    )
    # If this name of the log file doesn't exist yet, break the loop and use this name
    if not os.path.exists(log_path):
        break
else: # Executed after the whole for loop, but only if no break occurred
    complete = Canvas() # create a canvas for stimulus presentation
    complete.text('Either all sessions have been finished or there was an error. Please
    contact the experimenter.') # fill in some text into the canvas
    complete.show() # present the canvas
    log.flush()
    clock.sleep(4000) # show the text message for 4 seconds
    exp.end() # end the experiment right here
# Re-open the logfile
log.open(log_path)
var.logfile = log_path
```

Second, rather than using error-prone file names generated by the participants, this procedure automatically generates file names which include the subject and session numbers in a standardized manner. This is important, as we experienced that participants produce typos when inserting a file name manually or even worse use the incorrect session number. Third, the experimenter may see which sessions have been fully completed without opening the files (participants may sometimes abort a session). Possible code for an inline_script is shown in Listing 2.

Assigning participants to groups based on the subject number

Most experimental designs require random assignment of participants to experimental groups (between-subject design) or a random order of experimental conditions (within-subject designs). Randomization has the advantage that, assuming an adequate sample size, the groups do not differ from each other except for the manipulated variables. However, using full randomization may result in uneven distributions of group sizes, such that by chance more participants end up being assigned to one group than to another group. Therefore, as an alternative to full random assignment, participants can be assigned to groups quasi-randomly, that is, not based on any relevant factor for the experiment. This procedure, referred to as counterbalancing, ensures equal group sizes in the final sample.

Listing 3 provides a counterbalancing example with four experimental groups and four experimental conditions. In the present use case (Dahm & Rieger, manuscript in preparation), the four experimental groups were a visual mental practice group (VMP), a kinesthetic mental practice group (KMP), a physical practice group (PP), and a control practice group (CP). During the tests, we used two sequences (A and B) and their mirrored counterparts (AM and BM). Each participant practiced only one of these four sequences which was counterbalanced across participants and groups while the CP group practiced random sequences. For this, we used the modulo (%) operation, which divides the subject number by the number of groups resulting in an integer (whole number) with a remainder. For example, 9 modulo 4 equals 1 because 9 / 4 = 2 remainder 1. The remainder is then used for group assignments. Mind that the first solution for a remainder is always zero, otherwise (using `1234` instead of `0123` in the example) counterbalanced group assignment will fail.

How to transfer data to the experimenter

In the suggested approach, it is indispensable to retrieve the data at the end of participation as this is stored locally on the participants' computers. Depending on applicable privacy regulations, regular email may be problematic as this violates the participants' anonymity if the experimenter has connected information about ID and Emails

Listing 2 ■ Code to automatically save a copy of the data at the end of the experiment

```
#create backup of data file
import shutil,os,datetime

backup_path = os.path.join(var.experiment_path, "Logfile")
if not os.path.isdir(backup_path):
    os.makedirs(backup_path)

now = datetime.datetime.now()
current_time = now.strftime("%Y-%m-%d__%H-%M")

filename = "ID" + str(var.subject_nr) + "_Session" + str(var.session_nr) + "_Date-" +
    current_time

shutil.copyfile(var.logfile, os.path.join(var.experiment_path, "Logfile", filename+".csv"))
```

(oftentimes including the names). Similarly, anonymity is violated if participants receive their ID per mail. As an alternative, an upload solution might guarantee anonymity; specifically, participants may pick an ID from a list and upload their data in a cloud which makes the exchange of data anonymous.

In the present use case, communication through email was used. In the authors' experience and when permitted, Emails have been proven a more convenient solution because it facilitates communication with the participants in case of missing data. For instance, about ≈10% of the participants fail to send the correct file. Rather than sending a file with the data, they send a copy of the empty file from the downloads path during the installation process. If this was detected early by the experimenter, this could always be resolved reminding them to focus on the path and send the correct file including the data. For more clearness in one's own outbox and inbox, it has proven useful to include an experiment name along with the ID of the participant in the subject line of each Email.

Born-open data: automatically and immediately making data public

Another approach to collect the data could be to automatically upload the data after participation in an online-repository. Similar solutions have been proposed by the born-open framework (Rouder, 2016), which goes even one step further, by uploading the data directly to an open-access repository. Such an approach has already been implemented in Eprime (Cousineau, 2020) and jsPsych (Cousineau, 2021) to upload the data on GitHub (https://github.com/). However, for OpenSesame there is currently no out-of-the-box solution for born-open data. To create a born-open solution, one may create a custom inline-script that stores the data on a public repository such as GitHub. While born-open data increases scientific transparency (for an overview see Rouder, 2016), it also raises potential privacy concerns. As an experimenter of a study, one needs all data, also those that can be considered private (e.g., age which could identify some participants), to provide information about the sample. Hence, when choosing to make data public automatically, special care must be taken that no personally identifiable information is inadvertently enclosed, and that this particular form of open data is in accordance with institutional privacy regulations. To solve this issue, a second data set including socio-demographic information could be stored on a private repository.

Sending reminders

In multi-session studies, reminding participants to participate on each day of testing is essential to prevent them from dropping out (due to forgetting the study). The authors' personal experience has shown that it is better if participants perform a session every day than every three days. Forgetting to participate was much higher when participants did not practice daily; that is, participants tend to either participate very regularly or not at all. Sending reminders can prevent some participants from dropping out of the study when practicing in a three-day rhythm but was not necessary in a one-day rhythm. We sent such reminders in the afternoon, as we assumed it is placed better after study or work hours. Furthermore, some participants had already participated at this time, thereby reducing the number of reminders to be sent. Reminders can be sent per phone or per email. In both cases, personal data needs to be collected. Therefore, reminders should be optional.

In the present use case, we explicitly asked participants to report in an online sheet when the study had been started if they wanted to receive reminders. This was also helpful to collect the data, as many participants forget to send the data after completion of the study. In the online sheet the experimenter had a full overview of the starting

Listing 3 ■ Example for counterbalancing with four groups and four conditions

```
#set between subject variables
#counterbalance practice groups
if var.subject_nr % 4 == 0:
    var.PG = 'VMP'
elif var.subject_nr % 4 == 1:
    var.PG = 'KMP'
elif var.subject_nr % 4 == 2:
    var.PG = 'PP'
elif var.subject_nr % 4 == 3:
    var.PG = 'CP'
#Counterbalance practice sequences taking into account the above groups
if var.subject_nr % 16 in [0,5,10,15]: #0=VMP, 5=KMP, 10=PP, 15=CP
    var.Pract_Seq = 'A'
elif var.subject_nr % 16 in [1,6,11,12]: #12=VMP, 1=KMP, 6=PP, 11=CP
    var.Pract_Seq = 'AM'
elif var.subject_nr % 16 in [2,7,8,13]: #8=VMP, 13=KMP, 2=PP, 7=CP
    var.Pract_Seq = 'B'
elif var.subject_nr % 16 in [3,4,9,14]: #4=VMP, 9=KMP, 14=PP, 3=CP
    var.Pract_Seq = 'BM'
```

time and completion time of each participant. Reminders after completion of the study were sent in a weekly rhythm. It should not be sent too long after a participant finished the experiment as some participants delete the folders of the experiment after some time.

Data monitoring and data quality

As in online studies, the suggested approach takes place outside the lab in an uncontrolled environment, which might result in more participants showing non-compliance with the study instructions (Huang et al., 2012). Therefore, we provide some suggestions on how to, before the experiment, prevent non-compliance (or indifference) in the study as well as how to, after the experiment, detect participants who showed non-compliance.

To prevent non-compliance, warnings have been shown to be effective (Huang et al., 2012). If participants are told at the beginning of the study that participation is only valid if the data quality is good, participants are more committed to the study instructions. Furthermore, comprehension questions after the main instructions may ensure that participants have read and understood the instructions.

To detect non-compliance, several approaches exist. One may implement infrequent catch trials among regular trials to check whether participants were paying attention throughout the experiment. Another approach is including a questionnaire that checks for infrequency (Meade & Craig, 2012) and inconsistency (Maniaci & Rogge, 2014). In addition to participants who respond without variance (e.g., always choosing the same response option), these scales also detect participants who are not really reading the items and respond randomly. The infrequency scale contains items that are usually responded by all participants in the same manner, such as: "I have been to every country in the world". The inconsistency scale contains paired items that are placed separately from each other but have the same meaning, for instance "I am an active person" and "I have an active lifestyle". Such measures (Maniaci & Rogge, 2014) resulted in approximately 5% of non-compliant participants in the authors' student samples. Additionally, the most common forms of non-compliance in the authors' studies were a) participants clicked through the study as fast as possible without paying attention to instructions or stimuli (≈6% of student samples), or b) participants were distracted, for instance by listening to the radio during participation (≈3% of student samples). This may be visible in the data with error rates around chance level combined with particular patterns in reaction times (e.g., low response-time variance or a series of very short reaction times in the raw data).

Conclusions

In the present article, we addressed challenges that come up with multi-session experiments that take place out-of-the-lab (but using regular experimental software), using learning studies as a prototypical example, and presented possible solutions using OpenSesame (Mathôt et al., 2012). The presented approach may not only be helpful for the research questions addressed in the present use case, but also each one on its own in other experimental settings.

Authors' note

S.D. is a recipient of an APART-GSK Fellowship (APART GSK / 11963) of the Austrian Academy of Sciences at the Department of Psychology at the University of Innsbruck.

References

Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision*, *10*(4), 433–436. doi: 10.1163/156856897X00357.

Bridges, D., Pitiot, A., MacAskill, M. R., & Peirce, J. W. (2020). The timing mega-study: Comparing a range of experiment generators, both lab-based and online. *PeerJ*, *8*(e9414), 1–9. doi: 10.7717/peerj.9414.

Cousineau, D. (2020). Born-Open Data for E-Prime. *PsyArXiv*. doi: 10.31234/osf.io/kyuvs.

Cousineau, D. (2021). Born-Open Data for jsPsych. *PsyArXiv*. doi: 10.31234/osf.io/rkhng.

Dahm, S. F., Hyna, H., & Krause, D. (2023). Imagine to automatize: Automatization of stimulus–response coupling after action imagery practice in implicit sequence learning. *Psychological Research*, *87*, 1–9. doi: 10.1007/s00426-023-01797-w.

Dahm, S. F., & Rieger, M. (2023). Time course of learning sequence representations in action imagery practice. *Human Movement Science*, *87*, 10305–9. doi: 10.1016/j.humov.2022.103050.

Dahm, S. F., Weigelt, M., & Rieger, M. (2023). Sequence representations after action-imagery practice of one-finger movements are effector-independent. *Psychological Research*, *87*(1), 210–225. doi: 10.1007/s00426-022-01645-3.

de Leeuw, J. R. (2015). Jspsych: A javascript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*(1), 1–12. doi: 10.3758/s13428-014-0458-y.

Ferrand, L., Méot, A., Spinelli, E., New, B., Pallier, C., Bonin, P., Dufau, S., Mathôt, S., & Grainger, J. (2018). Megalex: A megastudy of visual and auditory word recognition. *Behavior Research Methods*, *50*(3), 1285–1307. doi: 10.3758/s13428-017-0943-1.

Grootswagers, T. (2020). A primer on running human behavioural experiments online. *Behavior Research Methods*, *52*(6), 2283–2286. doi: 10.3758/s13428-020-01395-3.

Henninger, F., Shevchenko, Y., Mertens, U. K., Kieslich, P. J., & Hilbig, B. E. (2022). Lab.js: A free, open, online study builder. *Behavior Research Methods*, *54*(2), 556–573. doi: 10.3758/s13428-019-01283-5.

Huang, J. L., Curran, P. G., Keeney, J., Poposki, E. M., & DeShon, R. P. (2012). Detecting and deterring insufficient effort responding to surveys. *Journal of Business and Psychology*, *27*(1), 99–114. doi: 10.1007/s10869-011-9231-8.

Koch, I., Poljac, E., M"uller, H., & Kiesel, A. (2018). Cognitive structure, flexibility, and plasticity in human multitasking—an integrative review of dual-task and task-switching research. *Psychological Bulletin*, *144*(6), 557–583. doi: 10.1037/bul0000144.

Lange, K., K"uhn, S., & Filevich, E. (2015). "just another tool for online studies" (jatos): An easy solution for setup and management of web servers supporting online studies. *PLoS ONE*, *10*(6), e0130834–e0130834. doi: 10.1371/journal.pone.0130834.

Maniaci, M. R., & Rogge, R. D. (2014). Caring about carelessness: Participant inattention and its effects on research. *Journal of Research in Personality*, *48*, 61–83. doi: 10.1016/j.jrp.2013.09.008.

Mathôt, S., & March, J. (2022). Conducting linguistic experiments online with opensesame and osweb. *Language Learning*, *11*, 1–32. doi: 10.1111/lang.12509.

Mathôt, S., Schreij, D., & Theeuwes, J. (2012). Opensesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, *44*(2), 314–324. doi: 10.3758/s13428-011-0168-7.

Meade, A. W., & Craig, S. B. (2012). Identifying careless responses in survey data. *Psychological Methods*, *17*(3), 437–455. doi: 10.1037/a0028085.

Ort, E., Fahrenfort, J. J., Reeder, R., Pollmann, S., & Olivers, C. N. (2019). Frontal cortex differentiates between free and imposed target selection in multiple-target search. *NeuroImage*, *202*, 116133–1. doi: 10.1016/j.neuroimage.2019.116133.

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., H"ochenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). Psychopy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. doi: 10.3758/s13428-018-01193-y.

Rouder, J. N. (2016). The what, why, and how of born-open data. *Behavior Research Methods*, *48*(3), 1062–1069. doi: 10.3758/s13428-015-0630-z.

Sauter, M., Draschkow, D., & Mack, W. (2020). Building, hosting and recruiting: A brief introduction to running behavioral experiments online. *Brain Sciences*, *10*(4), 1–9. doi: 10.3390/brainsci10040251.

Verwey, W. B., Shea, C. H., & Wright, D. L. (2015). A cognitive framework for explaining serial processing and sequence execution strategies. *Psychonomic Bulletin & Review*, *22*(1), 54–77. doi: 10.3758/s13423-014-0773-4.

Zhou, C., Lorist, M. M., & Mathôt, S. (2022). Categorical bias as a crucial parameter in visual working memory: The effect of memory load and retention interval. *Cortex*, *154*, 311–321. doi: 10.1016/j.cortex.2022.05.007.

Open practices

The *Open Material* badge was earned because supplementary material(s) are available on osf.io/xmn6z/

Citation

Dahm, S. F., Ort, E., Büsel, C., Sachse, P., & Mathot, S. (2023). Implementing multi-session learning studies out of the lab: Tips and tricks using OpenSesame. *The Quantitative Methods for Psychology*, *19*(2), 156–164. doi: 10.20982/tqmp.19.2.p156.

Received: 24/01/2023 ~ Accepted: 19/05/2023