DevStart Manifesto:

A Practical Guide for Experimental Research in Cognitive Science

Tommaso Ghilardi1, Giulia Serino1, Francesco Poli2

1 Centre for Brain and Cognitive Development, Birkbeck, University of London, London, UK

2 MRC Cognition and Brain Sciences Unit, University of Cambridge, Cambridge, UK

**Abstract**

Open-science practices have steadily advanced the transparency of research, yet significant gaps remain—particularly around the practical details of designing, running, and analysing experiments. DevStart addresses these needs by unifying historically fragmented resources into a single, hands-on guide for cognitive scientists. This paper outlines DevStart’s core framework and demonstrates how it streamlines the entire research pipeline: from installing free programming tools (Python and R) to building experimental paradigms and handling complex data (e.g., gaze, pupil size) with robust, open-source methods. By detailing step-by-step tutorials and openly sharing code and pipelines, DevStart lowers the barriers that hinder many early career researchers and students. Beyond tutorials, DevStart fosters community engagement through an online forum that supports collaboration, troubleshooting, and the exchange of ideas. In doing so, we hope to catalyse a vibrant research community that embraces robust, open, and collaborative science at every step of the experimental process.

# **Introduction**

Open science has progressed steadily over the last decade (Freese et al., 2022; Hamlin, 2017; Korbmacher et al., 2023; Nosek et al., 2022). However, change has been slower than anticipated (Tenney et al., 2021) and systemic biases in accessibility still persist (Bahlai et al., 2019). In addition, not all steps of experimental research have been made equally open. While it has now become common practice to share data in open repositories, the same thoroughness does not apply when it comes to sharing statistical analysis scripts, and even less to other aspects of research such as reproducible code for designing stimuli, running experimental paradigms or pre-processing data. Moreover, even when code is provided, many scripts rely on proprietary licenses, limiting true reproducibility for individuals who cannot afford those tools.

Even if many open tools exist, they often lack integration and connection, forcing researchers to navigate a fragmented ecosystem which can lead to frustration and discouragement. This can be especially daunting for students and early career researchers who are starting to orient in a patchwork of scripts, guidelines, and platforms. Even when scientific articles give helpful advice on how to carry out experimental (Roth et al., 2025) or computational (Wilson & Collins, 2019) research, they lack the practical aspects over which academics tend to get stuck.

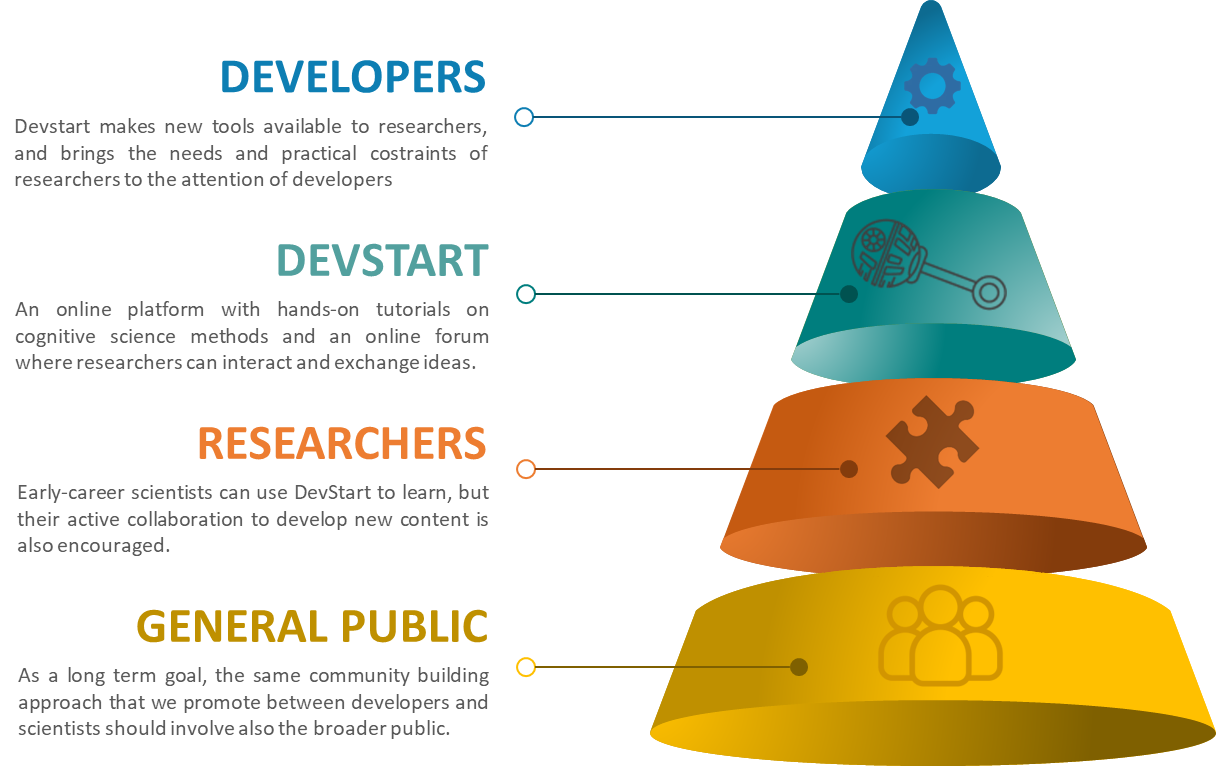
Overall, the resulting inefficiency and potential discouragement runs contrary to the aims of open science, which aspire to foster collaboration, transparency, and accessibility for everyone. Indeed, many PhD students report feeling overwhelmed by uncertainty during the early stages of their research, often struggling with where to start and how to proceed. This problem is exacerbated in under-represented groups, who are at risk of unequal access to resources, support, and opportunities (Nicholls et al., 2022). This uncertainty about the research process has been identified as a significant factor affecting both student wellbeing and degree completion (Cornwall et al., 2019).

DevStart is a new initiative that addresses these needs in cognitive science, providing hands-on tutorials that bring together historically disconnected aspects of research (Figure 1). Our platform offers step-by-step guidance through the entire experimental pipeline: from designing and implementing experimental paradigms, to collecting data, to pre-processing with robust techniques, to analysing results with appropriate statistical frameworks. In addition, DevStart provides carefully simulated datasets that directly correspond to each tutorial's experimental design—unlike resources that rely on generic sample data, our datasets allow users to work with realistic data that meaningfully connects to the presented research questions. Thus, rather than treating “openness” as a checklist of requirements, DevStart emphasizes the core ethos of accountability and transparency: It serves as a gateway to more robust, accessible, and integrated research practices, in the form of an online, hands-on manual that is freely accessible to all.

By offering guidance on everything from designing experiments to analysing data in openly available software, DevStart creates a unified learning path for those entering the field. In doing so, it lowers the barriers to *truly* open research, helping researchers adopt open practices while avoiding hours of frustration learning niche or scattered tools. This might be especially useful for early career researchers working in smaller labs or those who are beginning to explore experimental research independently — where access to experienced colleagues may be limited. DevStart aims to address these challenges through open collaboration and community building. We see this initiative as the start of a growing community for cognitive scientists. If formal guidelines do not yet exist, we can begin by sharing our own workflows, highlighting what has worked well, and helping to shape better practices for the field. In this way, we hope to ease the learning curve for future researchers, and to spare them some of the challenges we faced ourselves.

Since we are developmental cognitive scientists by training, many of our examples focus on developmental populations. At times, we also offer additional advice specifically for developmental scientists, for example on how to best calibrate an eye-tracker with infants. However, this should not discourage all cognitive scientists from using our resources. In fact, working with developmental populations requires the reliance on especially flexible pipelines which can deal with noise and missingness in the data (Frank et al., 2017; Hessels et al., 2017). However, these hurdles are familiar to many researchers, for example who works in the field of animal cognition (Howard & Barron, 2024) or with naturalistic settings (Sonkusare et al., 2019) that inevitably lead to noisier dataset. Hence, adopting robust pipelines can be beneficial to all cognitive scientists, beyond the ones focusing on development.

In this paper, we introduce DevStart’s framework giving an overview of the hands-on tutorials that are available. These tutorials create meaningful connections between methodology, analysis, and implementation, serving as stepping stones for researchers to develop an integrated understanding of essential research tools and practices.

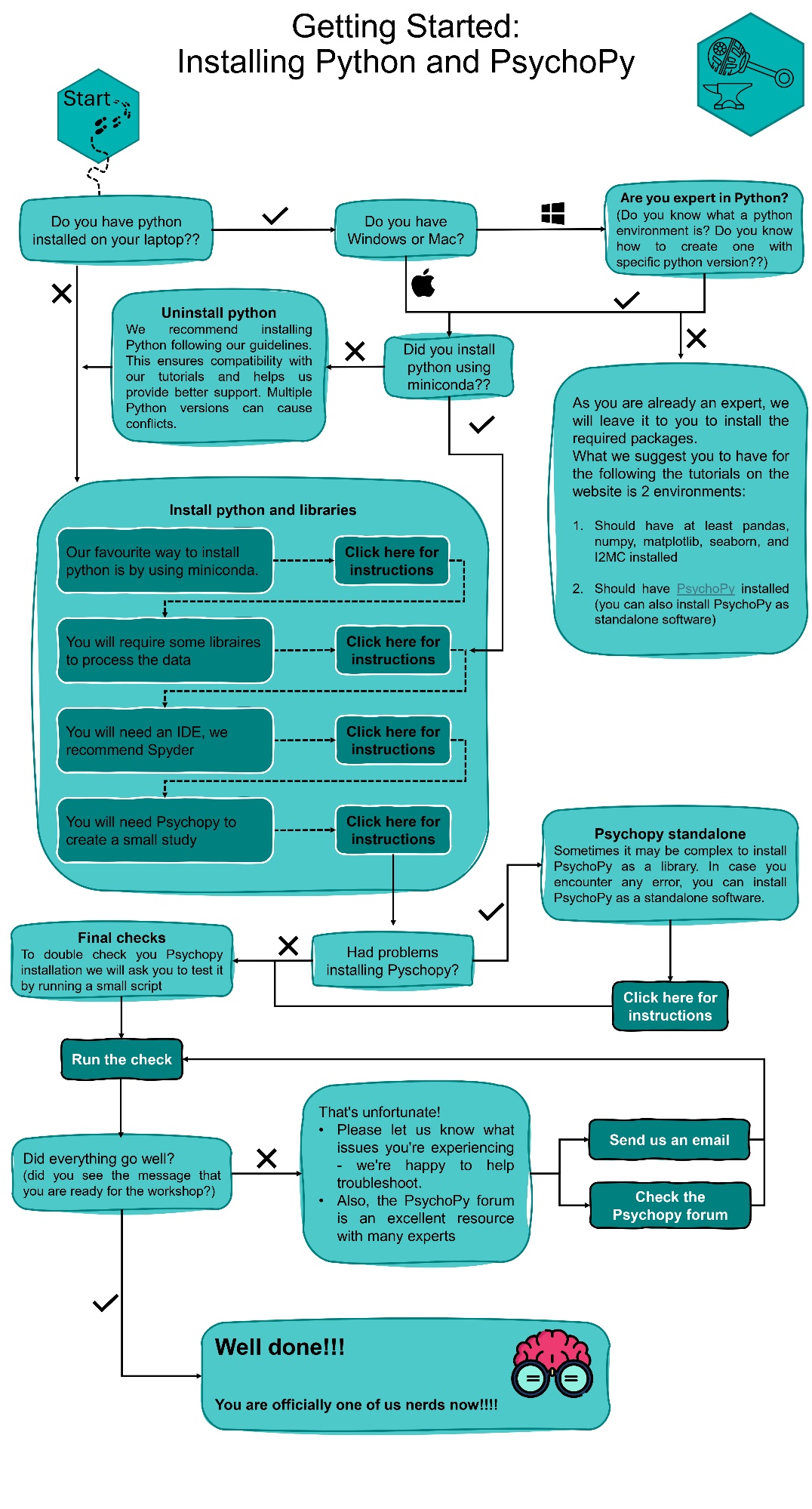
**Figure 1.** DevStart offers an online guidebook and a forum for cognitive scientists to learn, share, and connect. This community building effort involves researchers and developers, but will widen to the general public as a long-term plan.

# **Installing free programming tools**

Only a minority of social sciences researchers have a background in data or computer science (Ferrie et al., 2022). While an increasing number of universities integrate programming courses into psychology and cognitive science degrees, this is not always the case. Hence, the very first hurdle encountered by researchers is to even properly install programming languages. Instructions are often made for experts, and navigating among all the available options can be daunting. For this reason, our guidebook starts with guidelines on how to install free programming software. More specifically, we chose to focus on Python and R.

These two languages are completely free of costs, have reliable documentation, and regular updates. Thanks to Psychopy (Peirce et al., 2019) and other libraries, we find Python to be superior to any other language when it comes to build experimental paradigms, display visual or auditory stimuli, and interact with other equipment such as eye-tracking, EEG, or fNIRS devices. On the other hand, R remains indubitably the best language for performing statistical analyses thanks to the vast and very active community of statisticians, which developed highly useful collection of packages such as easystats (Lüdecke et al., 2019).

Given that even the very first steps on how to install these programming languages are often unclear, we made a flowchart that guides people though the installation process with as much clarity and simplicity as possible (Figure 2). These instructions might change over time depending on updates in computers (e.g., chips) or softwares (e.g., new python releases) but we commit to keeping them up to date on the website for the foreseeable future.

**Figure 2. A flowchart to get started with scientific programming.** By following the flowchart depending on specific needs and available resources, we guide researchers throughout the process of installing free open-source software for programming.

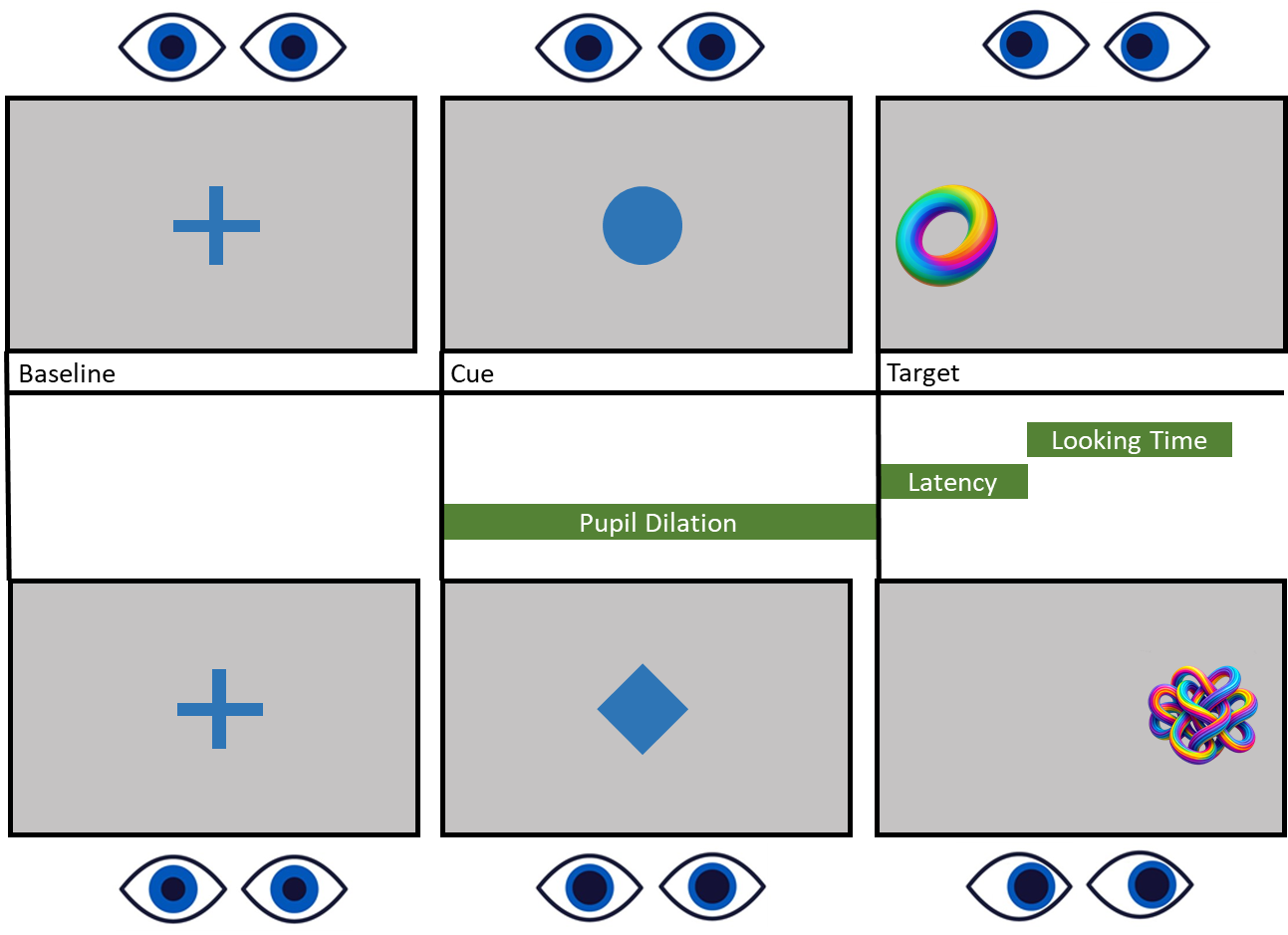
# **Building an experimental paradigm**

We believe that research should start from a strong theoretical foundation. Theories about how the mind or the brain work lead to generating specific, clear hypotheses which can be then tested empirically. In this perspective, the experimental side of research, where data is collected to verify or confute existing hypotheses, is of paramount importance. Within cognitive (neuro)science, many practical ways to test existing hypotheses exist, such as behavioural, eye-tracking, and neurophysiological tools (Purves et al., 2013). However, an aspect that underlies all of these tools is the existence of an experimental paradigm: A specific set of stimuli or experimental conditions that allows to tear apart or test different hypotheses.

Here (and on the DevStart website, see [tinyurl.com/devstarthome](https://tinyurl.com/devstarthome); Ghilardi et al., 2024), we give a simple example of an experimental paradigm. However, we want to stress that the aim of the tutorials is to give researchers the tools which would allow them to generate any other paradigm, going beyond the specifics of our example. What we hope researchers to focus on are not the details of this paradigm, but the flexibility that Python and Psychopy afford when it comes to code virtually anything.

This experimental paradigm will be used across the rest of the DevStart tutorials. This point is crucial, because it allows us to show how to go from designing a specific study to every other aspect of the research process: collecting data, pre-processing it, analysing it, and interpreting it. Let’s imagine we would want to investigate the mechanisms underlying infants’ curiosity. Current theories of curiosity argue that more attention is allocated towards stimuli that offer greater learning opportunities (Gottlieb & Oudeyer, 2018; Poli et al., 2024). We might want to test whether this is indeed the case starting from infancy. More specifically, our hypothesis could be that infants should be more motivated to predict when and where a target stimulus will appear if they can learn more from it.

To do this, we devised a very simple paradigm in which two cue stimuli (either a circle or a square) reliably predict the location of the following target stimulus (either on the right or on the left, respectively). Crucially, we will manipulate how much they can learn from these two target stimuli. The stimulus associated with the circle will be visually complex, and it will thus offer many components which infants can slowly unpack and learn about. The stimulus associated with the square will be visually simple, and it will thus offer very little to learn. The exact paradigm is illustrated in Figure 3. Please not that we generated a very simple paradigm for educational purposes, and this is thus not fit for a robust, fully-fledged experimental inquiry.

******Figure 3. An example experimental paradigm.** After a fixation stimulus appears in the centre of the screen, a cue follows. The cue can either be a square or a diamond. The square always signals the appearance of a low-complexity stimulus on the left side of the screen, and the diamond always signals the appearance of a high-complexity stimulus on the right side of the screen. Hence, the cues are predictive both of the target stimuli’s location and their complexity. With this paradigm, we can measure pupil size while looking at the cues, the saccadic latency (or eye reaction time) when the target stimulus appears, and how long participants look at the target stimulus on each trial.

Crucially, we can collect multiple measures to test our hypothesis. In other words, we can make more specific predictions about how our hypothesis (i.e., that infants will be more motivated to predict the location of the more complex stimulus) will be tested. First, infants usually become faster over time at predicting the target location when the cue stimulus is presented (Poli et al., 2020; Romberg & Saffran, 2012). However, if our hypothesis is correct, they should display this pattern more so for the circle-complex stimulus association than for the square-simple stimulus one. Second, infants tend to look longer at stimuli they are more interested in (Stahl & Feigenson, 2017). Hence, we should observe longer looking time to the more complex stimulus. Finally, infants’ pupil size before a stimulus is presented might reflect the information content they expect to receive from that stimulus, with greater anticipatory pupil dilation for stimuli with greater information content (Zénon, 2019). Hence, we expect greater pupil dilation when the circle is presented compared to when the square is presented.

The focus of this section does not want to be specific theory and hypothesis we outlined. The focus should be how to go from a theory to hypotheses, and from hypotheses to specific predictions. These predictions concern specific variables of interest. In this case, the variables are saccadic latency (i.e., how quickly one directs their gaze to a stimulus when it appears on the screen), looking time (i.e., how long one looks at a given stimulus), and stimulus-evoked pupil dilation (i.e., how pupil dilates in response or anticipation to a given stimulus).

On the DevStart website, we first review in more detail the pros and cons of all these measures. Then, we show how to import and display experimental stimuli in Python using Psychopy, looping over multiple trials to dynamically present different cue and target stimuli. Next, we show how to use Python to interact with an external device. In our case, this is an eye-tracker, but similar principles would apply to other devices as well, such as EEG or fNIRS systems. By using the tobii\_research library, we connect to an eye-tracker. With additional functions to collect, store, and save data, we show how to interact with the eye-tracker via Python to extract the information we need (i.e., gaze and pupil signal, the validity of the signal, and its timing). A final, crucial point of a fully-fledged experimental paradigm is the ability to reliably record when and where the stimuli where presented on the screen. This can be done via “events” (also called “triggers”) which contain information about the timing of the stimuli and any additional information we might want to record about them, such as what stimuli where presented and where. We show how to make a function that can reliably collect information about these events.

# **Data collection**

Once the experimental paradigm is ready, we can proceed with data collection. When working with human participants, it is essential to obtain ethical approval from the ethics committee of the institution we work for, as well as informed consent from the participants — or from their caregivers in the case of developmental populations. This is standard practice across all types of studies involving human subjects (Manti & Licari, 2018).

Each technique has its own specific pipeline of steps that must be followed to ensure high-quality data collection. For eye-tracking, the first and most critical step is calibration. Calibration allows the system to account for individual differences in participants' eye features and enables the eye-tracking algorithms to accurately detect and track gaze. This is a fundamental step for ensuring reliable data.

There are different calibration techniques, but in general, participants are asked to look at a series of points displayed in different locations while keeping their heads still. This allows the eye tracker to encode the reflections generated by the pupil and cornea as the participant gazes in different directions. The specific number and placement of calibration points will vary depending on the research question. For screen-based paradigms, calibration points are typically presented on-screen to align with how the participant will explore visual stimuli. In contrast, for real-world research settings, calibration points may be 3D objects placed in physical space. In either case, the goal remains the same: to capture eye features accurately within the range of gaze behaviour expected during the task. On the DevStart website, we show how to launch a calibration from Python using Psychopy.

The number, type, and duration of calibration points also depend on the population being tested. For example, when working with infants, calibration points are usually colourful, animated stimuli accompanied by attention-grabbing sounds. These stimuli remain on screen until the infant fixates on them, then quickly shift to a new location to prompt further fixation. With adult participants, calibration is often more precise, using 9 or more static points, and repeating them if needed to improve accuracy. With infants, it is necessary to find a balance between data quality and participant fussiness. A typical benchmark is that 3 out of 5 successful fixations is sufficient to begin the experiment.

Another key difference between adult and infant studies is the positioning of the participant and eye tracker. For adults, the eye tracker is usually placed directly in front of the participant. In contrast, infants may be lying in their caregiver’s lap or in a crib, so it is better to position the screen centrally and parallel to their face to facilitate quick eye detection and reduce calibration time. With infant participants, calibration efficiency is essential—though it may seem counterintuitive to prioritize speed over precision. A protracted calibration procedure risks inducing boredom or fussiness in infants, potentially compromising their attention not only for the calibration itself but for the subsequent experimental task. This delicate balance between calibration thoroughness and maintaining infant engagement exemplifies the methodological insights typically acquired through hands-on laboratory experience rather than formal methodological literature.When working with adults, it is also important to remind participants not to wear eyeliner or mascara, as these can interfere with data quality. If possible, participants should also avoid wearing glasses. Finally, ask participants to take a comfortable and stable position, as head or posture changes during the task can alter eye-tracker accuracy relative to the original calibration, thereby affecting data quality. Once calibration has been successfully completed, the experiment can start and data can be collected and saved.

# **Data Pre-processing**

## ***Gaze data***

After data collection, you’re left with a large spreadsheet filled with numbers — which likely do not resemble the clear measure of interest (e.g., saccadic latency) you were planning to analyse. A critical step in eye-tracking research is transforming these raw data points into interpretable eye-tracking metrics. But how is this done?

What an eye-tracker records are the x–y coordinates over time indicating where the participant’s gaze was on the screen. To extract meaningful metrics from these coordinates, various pre-processing algorithms are available. On the DevStart website, we chose to present “Identification by two-means clustering (I2MC) algorithm (Hessels et al., 2017), an algorithm which is particularly robust to noise. This makes it widely applicable not only to adults, but also developmental populations (meaning that as it is robust to both missing data and imprecise signal commonly observed in children).

In simple terms, I2MC first determines whether the gaze points recorded by the eye tracker are stable or unstable. If a series of x–y coordinates are close to one another in both time and space, it likely means the participant was fixating on a particular area of the screen. In that case, I2MC classifies the sequence as a fixation. Once fixation periods have been identified, the second step of I2MC involves interpolating missing data. Missing samples are usually due to blinks, head movements, or momentary loss of tracking. I2MC performs a robust interpolation procedure that estimates the most likely gaze position during these gaps, but only when the surrounding data is sufficiently stable. This ensures that the algorithm fills in brief signal losses without introducing false fixations. After interpolation, I2MC applies additional filters to remove noise and outliers, producing a clean sequence of fixations characterized by their onset and offset time, duration, and mean x–y position.

In addition to its robustness to noisy data, the I2MC algorithm is also straightforward to implement in Python. Before running the algorithm, it is necessary to adjust a few key parameters, such as the sampling frequency used during data collection, to ensure the algorithm is properly optimised for your dataset. Hessels and colleagues (2017) provide clear and practical guidelines for each of these steps, making the setup process accessible even for researchers with limited programming experience.

The final – and possibly the most challenging – aspect of eye-tracking data pre-processing is moving from fixations to variables of interest. Given the lack of universally accepted guidelines, early-career researchers are left not only with the task of implementing (often from scratch) the code to extract relevant metrics from the data, but also to make critical decisions about the pre-processing: How do I define Areas of Interest (AOIs)? What is the optimal time window in which the variable of interest should be measured? On DevStart, we offer a set of practical guidelines drawn from our experience and the established practices used in our research groups. Specifically, we provide the code we typically use to define AOIs and to extract key gaze metrics such as saccadic latency and looking time.

In our example paradigm, we define Areas of Interest (AOIs) specifically around the target stimuli to cleanly capture both saccadic latency and looking time. These AOIs are drawn so they do not overlap with each other or with the preceding cue’s region, ensuring that any gaze within an AOI can be unambiguously attributed to the target. In addition, each AOI has the same size, preventing artificial differences in measured gaze behaviour that could arise from larger or smaller regions. Regarding the time of interest, it varies between the two measures of interest. For saccadic latency, we include the inter-stimulus interval between the cue and the target so we can detect whether infants make anticipatory saccades (i.e., they shift their gaze to the target’s AOI before it actually appears). Given that it takes a minimum of 200 ms to plan and execute a saccade, any saccade that lands on the target before 100 ms after target appearance can be considered as anticipatory, because its planning must have started before the target was presented. For looking time, the time of interest is the total duration within the target’s AOI only from the moment the target is displayed, capturing how long infants visually engage with the stimulus once it is on screen.

What we hope to illustrate with these examples is broader than the specific decisions that concern our task: We want to show how to think critically about the crucial pre-processing steps involving the decision of the optimal areas and times of interest. This way of thinking can then be applied to any experimental paradigm or measure of interest, no matter its specifics.

## ***Pupillometry data***

Pupillometry—the measurement of changes in pupil diameter—has become a powerful tool for studying both attentional and affective processes in cognitive science (Mathôt, 2018). DevStart showcases how to pre-process pupil signals using the open-source R package PupillometryR (Forbes, 2020). PupillometryR is particularly useful when the amount of missing data is high, because it operates on a *trial-by-trial* basis: This allows single trials with poor data quality to be discarded without excluding the entire participant.

The pre-processing pipeline we present on DevStart begins with reading raw pupil data (including separate traces for the left and right eyes) and organizing the data such that each trial has a clear start and end time. We then handle noisy segments by regressing one eye’s data onto the other, removing or interpolating blink artifacts, and filtering out high-frequency fluctuations that do not reflect genuine physiological changes. Next, PupillometryR’s functions identify trials with excessive missingness (e.g., due to fussiness or poor calibration) and remove them according to user-defined thresholds. The clean segments are subsequently downsampled to a lower frequency (e.g., 20Hz) to further reduce noise, and a baseline correction is applied so that each trial’s pupil signal is aligned to a stable reference point prior to stimulus onset.

Finally, to illustrate how we link these pre-processed signals to the hypotheses in our example paradigm, we extract baseline-corrected pupil size as our main dependent measure during the 2-second observation window for each cue. Concretely, we subtract each participant’s pre-stimulus baseline from the pupil time course, ensuring that any changes in pupil diameter can be attributed to differences in the cognitive processes occurring during the presentation of the circle and square cue stimuli. This final, cleaned, and baseline-corrected metric is then ready for statistical analyses aimed at testing our original prediction (i.e., that greater pupil size is expected in anticipation of a more interesting stimulus). Van Rij and colleagues (2019) offer a very useful illustration on how to best perform statistical analysis on pupillometric data.

A close-up of a graph

AI-generated content may be incorrect.

**Figure 4. Overview of the pupil‐processing pipeline.** Top panel shows raw pupil traces from left (purple) and right (gray) eyes for six simulated subjects, with shaded bands marking event periods. Bottom-left panel walks through each preprocessing step—averaging the two eyes, filtering and downsampling, epoching, interpolation, and baseline correction—illustrating how the signal is cleaned at each stage. Bottom-right panel displays the group-average post-cue pupil response (solid line) alongside the model fit (dashed line), with shaded areas representing ± SE.

# **Data analysis**

The approach we advocate in DevStart goes beyond classic *t*-tests and ANOVAs, because those methods can be limiting when dealing with multiple predictors or repeated measurements; They can only compare a limited number of categorical conditions at a time, and struggle with complex designs or unbalanced data. In contrast, linear models—particularly mixed-effects and generalized variants—are not only free from these limitations, but also hold several other advantages. For example, they offer a flexible framework for capturing individual differences (through random effects) and for dealing with non-normal or bounded data (e.g., reaction times, which often benefit from gamma or lognormal distributions).

Moreover, extracting **predictions** and **contrasts** from your linear or generalized linear models is crucial to drawing correct conclusions. Rather than merely relying on p-values or omnibus tests (e.g., ANOVA output) to determine which effects “matter,” these model-based estimates allow you to directly quantify **how** levels of a factor differ (contrasts) or what the expected outcome is under specific conditions (predictions). This approach not only promotes more **transparent** reporting but also bolsters the validity of your conclusions, as each statement you make aligns directly with a formal statistical comparison produced by your model. While these analyses might seem complex, we keep our examples as simple as possible on the DevStart website, breaking down each step with careful explanations and illustrations.

A further extension is to use additive models, which allow non-linear relationships between predictors and outcomes (e.g., a smooth effect of time or trial number). Additive models enrich the linear framework by fitting smooth “curves” (instead of straight lines) and are particularly relevant for timeseries data such as pupil size. Such flexibility can greatly improve model fit and interpretability, revealing patterns you might otherwise miss with purely linear assumptions. Crucially, these additive components can still be paired with mixed or generalized modeling, giving you the best of both worlds: random effects for participant-level variability and non-linear functional forms for predictors.

To give an example, we analyse the simulated, pre-processed gaze and pupil data for our simple experimental paradigm. We show how to fit generalised mixed-effect models to looking time and saccadic latency to test our original hypothesis that infants are more motivated to predict the appearance of a more complex (and hence more interesting) stimulus. We show how to check whether our models respect the assumptions of independence, homoscedasticity, normality of residuals, and linearity. Then, we show how to perform the relevant statistical tests and plot the results.

The example results are depicted in Figure NNN for gaze data, and Figure NNN for pupillometry. We show that in our simulated data, over time, “infants” look faster at the more complex target stimulus, which would indicate a greater motivation to learn to predict the complex stimulus over the simple one. Regarding looking time, we show that they look at the more complex stimulus for longer, which would indicate greater interest in the complex stimulus compared to the simple one. Finally, we show that their baseline-corrected pupil size during the cue presentation is greater for the cue that is predictive of the complex stimulus. This indicates they expect to gather more information by looking at the complex target stimulus. Beyond a focus on these group-level results, we also show how to use random intercepts and slopes to look at individual variability in these effects.

A close-up of several graphs

AI-generated content may be incorrect.

**Figure 5. Model diagnostics and predictions for the looking‐time analysis.** Top row: Six panels of model checks (from the modelbased package)—(1) posterior predictive density overlaid on observed data, (2) residuals vs. fitted values to assess homogeneity of variance, (3) Cook’s distance vs. leverage for influential points, (4) variance‐inflation factors for collinearity, (5) Q–Q plot of standardized residuals to evaluate uniformity, and (6) Q–Q plots of random‐effect estimates for normality.

Bottom panel: Model‐predicted looking times (solid lines) across 20 trials for “NoReward” (salmon) and “Reward” (teal), with shaded ribbons showing ± 95 % confidence intervals.

# **Future Plans**

Looking ahead, we plan to expand DevStart with additional tutorials. First, some tutorials will address behavioural measures beyond eye-tracking—such as button presses, reaction times, or other motor responses that are often used in cognitive science. These expansions will cover how to set up experimental paradigms, data collection, and pre-processing of behavioural data.

It is important to acknowledge that while our current eye-tracking tutorials cover theoretical principles applicable to any eye-tracking system, the specific data collection protocols are optimized for Tobii eye-trackers. This limitation reflects our current hardware access rather than an intentional focus. We plan to expand these tutorials to include other popular eye-tracking systems as we gain access to additional hardware. This expansion will allow researchers using diverse equipment to benefit from our step-by-step guidance while maintaining the same principled approach to experimental design and data analysis.

We are also developing a Python package that will serve as a wrapper between tobii\_research and PsychoPy libraries. This package will streamline the interaction process with Tobii eye-trackers, making it more accessible for researchers with limited programming experience. Importantly, this development will not replace the current approach in DevStart tutorials—we believe researchers should first understand how to interact with eye-trackers at a fundamental level before using higher-level wrappers. Once researchers grasp the underlying principles, this package will offer a simpler interface to common functionalities. Both the low-level tutorials and the simplified wrapper will be available through DevStart.

Additionally, we plan to incorporate Bayesian statistical approaches using brms (Bayesian Regression Models using ‘Stan’). This will offer a unified approach that covers all linear, mixed, generalized, and even additive model within a single R package (Bürkner, 2017). Bayesian statistics bring clear additional benefits: they provide full posterior distributions for each parameter (rather than point estimates alone), allow for intuitive uncertainty quantification, and can incorporate prior knowledge (if available) about plausible effect sizes. By relying on these built-in tools, researchers can avoid the pitfalls of merely chasing *p*-values or simplistic omnibus tests, and instead arrive at robust, data-driven conclusions that reflect the complex nature of developmental phenomena. In addition, we hope to introduce Bayesian optimal stopping strategies that help researchers decide when enough data has been collected, thereby improving efficiency while maintaining robust statistical inference.

We also hope to integrate EEG and fNIRS tutorials using open-source tools—ensuring that researchers who do not have access to proprietary software can still analyse brain imaging data in a reproducible and cost-effective manner. We plan to walk through the pipelines for data acquisition, artifact rejection, and interpretation of developmental EEG or fNIRS results, all while emphasizing transparent, replicable workflows. However, this goal will be challenging for the time being, because data acquisition for these methods still requires proprietary software, and even analyses are often performed using MATLAB packages and scripts.

Finally, we aim to strengthen our community engagement by hosting interactive workshops and launching an online forum. Workshops will be periodically offered—both in-person and virtual—to provide hands-on guidance and foster collaborative learning. The forum will serve as a hub for troubleshooting, sharing code, and brainstorming novel ideas. Through these collective efforts, we hope to create a thriving, open-science community that continuously evolves and benefits researchers at every stage of their career.

# **Discussion**

In this paper, we presentedDevStart, an online, hands-on manual designed to lower the entry barriers for open, reproducible research in cognitive science. By consolidating fragmented resources and providing accessible tutorials, DevStart empowers researchers—particularly early career scientists or those in smaller labs—to navigate each step of the research pipeline more confidently. Through practical guidelines, publicly available code, clear demonstrations, and carefully simulated datasets, DevStart reduces the overhead of searching for disparate scripts or tutorials and creates a cohesive environment where theoretical, methodological, and analytical aspects of research can be learned together.

Currently, DevStart offers step-by-step guides on installing free programming tools (with a focus on Python and R), building experimental paradigms (with Psychopy for stimuli and external-device integration), processing gaze and pupil data, and performing advanced statistical analyses in R (including mixed-effects and generalized linear models). As outlined in the Future Plans section, we have a roadmap for expanding these resources while maintaining our commitment to accessibility and open science principles. While DevStart's main focus is on teaching research methods and skills, we recognize that clear guidance may also help address the uncertainty many PhD students face when starting research projects—an issue linked to poorer wellbeing in academic settings. By providing clear steps through the research process, we hope to reduce the feeling of being lost that often comes with early-stage research. Community workshops and an online forum will further support users, offering avenues for troubleshooting, collaboration, and sharing best practices.

It is worth noting that DevStart does not attempt to prescribe definitive analytical approaches or tools, as these naturally vary based on specific research questions and experimental designs. Instead, we offer foundational methodologies and practical skills that serve as stepping stones for researchers beginning their scientific journey. Through accessible examples and transparent workflows, we aim to build confidence in early career scientists, enabling them to embark on increasingly sophisticated research projects. Our tutorials function not as rigid templates but as educational foundations that researchers can adapt and extend as they develop their own unique methodological approaches.

In line with a broader movement within psychological sciences (Ghai et al., 2025), our overarching mission is to democratize the learning process for anyone in the field or entering it for the first time. Rather than imposing rigid protocols or adding bureaucratic layers to “check off” open-science requirements, DevStart emphasizes robust principles of transparency, accountability, and accessibility. By fostering an environment where researchers are encouraged to share workflows, propose improvements, and collectively refine methodologies, DevStart aspires to cultivate a more connected research community—one that continually evolves, embraces diversity of thought, and strives for reproducible excellence across all steps of the experimental pipeline.

# **References**

Bahlai, C. A., Bartlett, L. J., Burgio, K. R., Fournier, A. M., Keiser, C. N., Poisot, T., & Whitney, K. S. (2019). Ethics: Open Science Isn’t Always Open to All Scientists. *American Scientist*, *107*(2), 78–82.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*, 1–28.

Cornwall, J., Mayland, E. C., Van Der Meer, J., Spronken-Smith, R. A., Tustin, C., & Blyth, P. (2019). Stressors in early-stage doctoral students. *Studies in Continuing Education*, *41*(3), 363–380. https://doi.org/10.1080/0158037X.2018.1534821

Ferrie, J., Wain, M., Gallacher, S., Brown, E., Allinson, R., Kolarz, P., MacInnes, J., Sutinen, L., & Cimatti, R. (2022). *Scoping the skills needs in the social sciences to support data driven research*. https://eprints.gla.ac.uk/289150/

Forbes, S. H. (2020). PupillometryR: An R package for preparing and analysing pupillometry data. *Journal of Open Source Software*, *5*(50), 2285. https://doi.org/10.21105/joss.02285

Frank, M. C., Bergelson, E., Bergmann, C., Cristia, A., Floccia, C., Gervain, J., Hamlin, J. K., Hannon, E. E., Kline, M., Levelt, C., Lew‐Williams, C., Nazzi, T., Panneton, R., Rabagliati, H., Soderstrom, M., Sullivan, J., Waxman, S., & Yurovsky, D. (2017). A Collaborative Approach to Infant Research: Promoting Reproducibility, Best Practices, and Theory‐Building. *Infancy*, *22*(4), 421–435. https://doi.org/10.1111/infa.12182

Freese, J., Rauf, T., & Voelkel, J. G. (2022). Advances in transparency and reproducibility in the social sciences. *Social Science Research*, *107*, 102770. https://doi.org/10.1016/j.ssresearch.2022.102770

Ghai, S., Thériault, R., Forscher, P., Shoda, Y., Syed, M., Puthillam, A., Peng, H. C., Basnight-Brown, D., Majid, A., & Azevedo, F. (2025). A manifesto for a globally diverse, equitable, and inclusive open science. *Communications Psychology*, *3*(1), 16.

Ghilardi, T., Poli, F., & Serino, G. (2024). *TommasoGhilardi/DevStart: TIE* (Version v1.0.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.13361296

Gottlieb, J., & Oudeyer, P.-Y. (2018). Towards a neuroscience of active sampling and curiosity. *Nature Reviews. Neuroscience*, *19*(12), 758–770. https://doi.org/10.1038/s41583-018-0078-0

Hamlin, J. K. (2017). Is Psychology Moving in the Right Direction? An Analysis of the Evidentiary Value Movement. *Perspectives on Psychological Science*, *12*(4), 690–693. https://doi.org/10.1177/1745691616689062

Hessels, R. S., Niehorster, D. C., Kemner, C., & Hooge, I. T. C. (2017). Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC). *Behavior Research Methods*, *49*(5), 1802–1823. https://doi.org/10.3758/s13428-016-0822-1

Howard, S. R., & Barron, A. B. (2024). Understanding the limits to animal cognition. *Current Biology*, *34*(7), R294–R300.

Korbmacher, M., Azevedo, F., Pennington, C. R., Hartmann, H., Pownall, M., Schmidt, K., Elsherif, M., Breznau, N., Robertson, O., & Kalandadze, T. (2023). The replication crisis has led to positive structural, procedural, and community changes. *Communications Psychology*, *1*(1), 3.

Lüdecke, D., Waggoner, P., & Makowski, D. (2019). insight: A Unified Interface to Access Information from Model Objects in R. *Journal of Open Source Software*, *4*(38), 1412. https://doi.org/10.21105/joss.01412

Manti, S., & Licari, A. (2018). How to obtain informed consent for research. *Breathe*, *14*(2), 145–152.

Mathôt, S. (2018). Pupillometry: Psychology, Physiology, and Function. *Journal of Cognition*, *1*(1), 1–23. https://doi.org/10.5334/joc.18

Nicholls, H., Nicholls, M., Tekin, S., Lamb, D., & Billings, J. (2022). The impact of working in academia on researchers’ mental health and well-being: A systematic review and qualitative meta-synthesis. *PLOS ONE*, *17*(5), e0268890. https://doi.org/10.1371/journal.pone.0268890

Nosek, B. A., Hardwicke, T. E., Moshontz, H., Allard, A., Corker, K. S., Dreber, A., Fidler, F., Hilgard, J., Kline Struhl, M., Nuijten, M. B., Rohrer, J. M., Romero, F., Scheel, A. M., Scherer, L. D., Schönbrodt, F. D., & Vazire, S. (2022). Replicability, Robustness, and Reproducibility in Psychological Science. *Annual Review of Psychology*, *73*(1), 719–748. https://doi.org/10.1146/annurev-psych-020821-114157

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Poli, F., O’Reilly, J. X., Mars, R. B., & Hunnius, S. (2024). Curiosity and the dynamics of optimal exploration. *Trends in Cognitive Sciences*, *28*(5), 441–453.

Poli, F., Serino, G., Mars, R. B., & Hunnius, S. (2020). Infants tailor their attention to maximize learning. *Science Advances*, *6*(39), eabb5053.

Purves, D., Cabeza, R., Huettel, S. A., Platt, M. L., LaBar, K. S., & Woldorff, M. G. (2013). *Principles of cognitive neuroscience* (Vol. 83). Sinauer Associates Sunderland, MA. https://www.researchgate.net/profile/Audrey-Turchick/publication/274734343\_Pediatric\_Imaging\_Essentials\_Radiography\_Ultrasound\_CT\_and\_MRI\_in\_Neonates\_and\_Children/links/5810b00008aef2ef97b2ca3e/Pediatric-Imaging-Essentials-Radiography-Ultrasound-CT-and-MRI-in-Neonates-and-Children.pdf

Romberg, A. R., & Saffran, J. R. (2012). Expectancy learning from probabilistic input by infants. *Frontiers in Psychology*, *3*, 610. https://doi.org/10.3389/fpsyg.2012.00610

Roth, J., Duan, Y., Mahner, F. P., Kaniuth, P., Wallis, T. S., & Hebart, M. N. (2025). Ten principles for reliable, efficient, and adaptable coding in psychology and cognitive neuroscience. *Communications Psychology*, *3*(1), 62.

Sonkusare, S., Breakspear, M., & Guo, C. (2019). Naturalistic stimuli in neuroscience: Critically acclaimed. *Trends in Cognitive Sciences*, *23*(8), 699–714.

Stahl, A. E., & Feigenson, L. (2017). Expectancy violations promote learning in young children. *Cognition*, *163*, 1–14. https://doi.org/10.1016/j.cognition.2017.02.008

Tenney, E. R., Costa, E., Allard, A., & Vazire, S. (2021). Open science and reform practices in organizational behavior research over time (2011 to 2019). *Organizational Behavior and Human Decision Processes*, *162*, 218–223. https://doi.org/10.1016/j.obhdp.2020.10.015

van Rij, J., Hendriks, P., van Rijn, H., Baayen, R. H., & Wood, S. N. (2019). Analyzing the time course of pupillometric data. *Trends in Hearing*, *23*, 2331216519832483. https://doi.org/10.1177/2331216519832483

Wilson, R. C., & Collins, A. G. (2019). Ten simple rules for the computational modeling of behavioral data. *Elife*, *8*, e49547.

Zénon, A. (2019). Eye pupil signals information gain. *Proceedings of the Royal Society B: Biological Sciences*, *286*(1911), 20191593. https://doi.org/10.1098/rspb.2019.1593