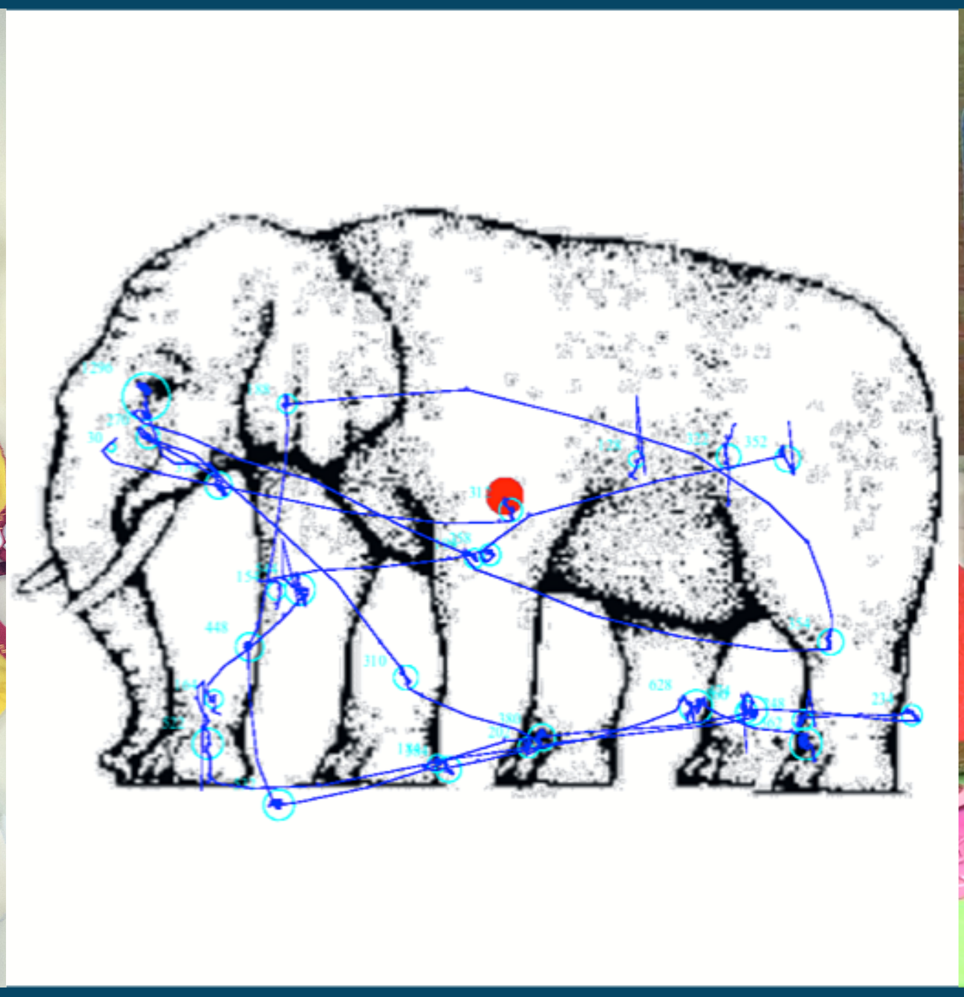
A Practical Introduction to Psycholinguistic Methods

A reader for conducting experimental research at Leiden University Centre for Linguistics


Eleanor Dutton & Bobby Ruijgrok

A Practical Introduction to Psycholinguistic Methods

A reader for conducting experimental research at
Leiden University Centre for Linguistics

Published by GBWusic
Kalkhaven 27
4201 BA Gorinchem
The Netherlands
www.gbwusic.com
info@gbwusic.com

ISBN: 978-90-816359-4-3
NUR: 616
DOI: 10.5281/zenodo.1339334

This publication and supplementary materials do not contain any DRM. They can be downloaded freely at https://zenodo.org/.

Copyright © 2018 Eleanor M. Dutton and Bobby J. Ruijgrok

Contents

How to use this reader

This reader is written for the BA courses "Experimental Methods I & II" and the MA course "Methods in Experimental Linguistics". These courses provide a comprehensive and practical introduction to the research methods used in the LUCL Labs and offer the chance to implement one of these techniques in an independent research project.

For the student...

Experimental Methods I: Design & Techniques consists of two learning units: "Experiment design & preparation" and "Software and hardware in the labs". In these two units, you gain practical knowledge of the first steps in individually designing, carrying out, analysing, and presenting a psycholinguistic experiment.

Experimental Methods II: Practicals & Projects consists of the learning units "Workflow of data collection & analysis" and "Practical project", after which you'll have the necessary competence to individually design, carry out, analyse, and present a psycholinguistic experiment.

This reader may seem like a lot of text, but in fact in these courses the emphasis is on learning by doing! Sometimes we'll be explaining the background or theory - then you'll see there are **Questions** to get you thinking about the material. When it gets to the technical stuff, you'll find there are often step-by-step instructions like `File > Open...` to help you get to grips with software and hardware. But the main practical learning will come through the **Exercises**.

We recommend that you complete all of the **Questions** and **Exercises** – not only is this the best way to get the most out of the course (and check you understand the content), but you may also be required to go back to your answers in a later exercise, and there won't be time to go back and catch up.

We strongly recommend that you use this reader in digital form as it will enable you to use embedded links that redirect you to different parts of the text as well as external resources such as websites. External links are written in this colour; you can jump to chapters, sections, figures and tables within the reader by clicking the numbers in the text or in the table of contents.

For the teacher...

For BA students, Experimental Methods I and Experimental Methods II should be taught in two semesters. Ideally, the two courses, of 5 ECTS each, should be completed during the third and fifth semesters of the BA programme respectively, with a one-semester statistics course in semester four. This learning trajectory is designed to provide students with the basis for doing an experiment-based Bachelor thesis project in the sixth (final) semester.

In the MA, the two halves of the course should be taught within one semester, creating a compact 10 ECTS course – Experimental Methods I (5 ECTS) in block 1 and Experimental Methods II (5 ECTS) in block 2. The MA version of the course is also ideal for PhD candidates and staff members who wish to gain a practical grounding in experimental design and techniques.

In both courses, the material in this reader must be supplemented with demonstrations and hands-on practice sessions in the labs. Supporting documents for teaching this course are available from the authors on request, by contacting b.j.ruijgrok@umail.leidenuniv.nl. These include sample class schedules used in the first two cycles of the courses (with deadlines and grade weightings) and useful documents to help students plan and structure their individual projects.

Acknowledgements

We would like to thank the following esteemed and valued colleagues for their unmissable contributions to this reader:

Bastien Boutonnet (PsychoPy tutorial materials)
Andreea Geambasu and Joyce Wijnmalen (our Baby Lab ladies)
Olga Kepinska (sample bibliography)
Saskia Lensink (Eye-Tracking)

and these cherished coworkers for their excellent reviewing skills:

Lisette Jager, Olga Kepinska, Saskia Lensink, Gulnaz Sibgatullina, Cesko Voeten, Man Wang...

Part I

Experiment design & preparation

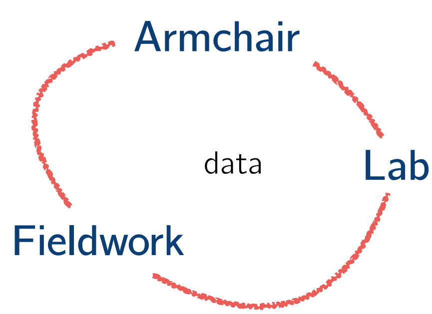
1 Meet the labs & experimental design introduction

LEARNING GOALS

- know which techniques are used at LUCL
- identify which variables to manipulate, measure and control depending on the research question

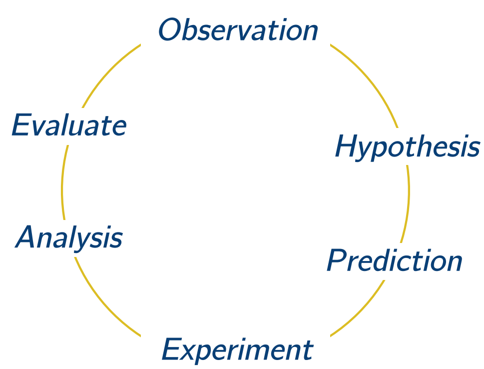
1.1 LUCL experimental linguistics labs

Leiden University Centre for Linguistics (LUCL) has as its mission to study structure and variation in the languages of the world. This is done from different perspectives in different workspaces while at the same time interdisciplinary research is encouraged. Within LUCL you can find theoreticians (sometimes referred to as armchair linguists), people who do (descriptive) fieldwork and people who collect and analyse data in a lab, either by consulting databases or by carrying out experiments with human subjects. This course focuses mainly on lab work. However, we cannot stress enough that to understand linguistic data it is important to integrate insights from different corners of linguistic analysis.



To illustrate this, let's use an analogy (from Ruijgrok, 2017): imagine Anne, sitting in her garden chair, noticing that the ants in her garden are walking faster as the temperature increases. While sitting there, she comes up with a function rule for this phenomenon. Her neighbour Eddy embarks on a jungle trek in South America and tries to apply the function to the Amazonian ants, without any success. However, he does notice that there is a high level of humidity. Already questioning the domain and range of Anne's function, their mutual friend Onno checks the limits of Anne's proposed function in his beloved botanical garden, manipulating both temperature and humidity as possible factors. He further relates his findings to the physical properties of several kinds of ants. Finally, the three friends arrive at an integrated theory of the ant's walking speed.[1] So, while we are mainly concerned with experimental lab work, you will see, during this course, that we try to integrate different methods, ultimately amounting to a better understanding of language.

The ant example also recalls the *scientific method* (sometimes referred to as the *empirical cycle*). From an observation a hypothesis is formulated and based on that a prediction is tested. Analysis and evaluation of the data leads to an adjustment of the starting hypothesis. At this point the cycle may start again:



This course will acquaint you with practical sides of the scientific method as applied to experimental linguistics. How do you conceptualise, design, carry out and analyse a linguistic experiment? Additionally, we will explain the techniques that can be used in LUCL's Labs. In the Lipsius building four labs can be found. In the order of year of opening, we have:

[1] This function appeared in an actual maths assignment in the 1980s in the method *Getal en Ruimte* (Noordhof Uitgevers.). To the amusement of the class an ant could end up walking backwards at some degree below zero.

- Phonetics Lab (opened 1983)
- Eye-tracking Lab (opened 2010; moved to a bigger room 2017)
- EEG Lab (opened 2013; between 2006-2013 in the Social Science building)
- Baby Lab (opened 2016; between 2008-2016 in the Social Science building)

Figure 1: LUCL's Labs consist of (1) the Phonetics Lab, (2) the Eye-tracking Lab, (3) the EEG Lab, and (4) the Baby Lab.

Since other courses teach you how to use the equipment in the Phonetics Lab, we will be focusing on teaching you about the other three labs. However, it's worth knowing that the labs are organised in the same way as much as possible: so for example, if you know how to record sound in the Phonetics Lab, you should be able to do so in all the other labs.

Before we can explain the *what* and *how* of the other lab techniques, we need to explain some basics about research ethics & scientific integrity in Chapter 2. After that, in Chapter 3, we will start at the very beginning of the experimental design process: collecting literature in order to formulate a research question. To set the scene for all that, in the rest of this chapter we will explain the basics of experimental research.

1.2 Experimental design basics

Let's start by thinking about the "philosophy" behind the methodology of experimental linguistics. Experimentation as a method of investigation is not something obscure that has no bearing on everyday life. It's a very intuitive approach to understanding how things work. If you come across something you don't understand, it's natural to try to figure it out by isolating different aspects of the situation, tweaking them, and seeing what happens. Let's take an example: imagine that every time you try to cook your evening meal, the electricity suddenly goes off. Is it the oven, the microwave or the rice cooker that's causing the problem? The obvious way to figure it out is to try each appliance one by one, in a methodical way, to narrow down the cause of the power surge.

To start exploring experimentation in more detail, it's handy to start with a definition. Of course this might not be the only possible definition, but it's good enough for our purposes:

> **"Experimentation is examining the relationship between two (or more) variables, while trying to keep all other variables constant."**

In this section we'll unpack this definition, and consider how we use this methodology to draw conclusions about the things we observe in our research.

1.2.1 Changing and measuring variables

The word "variable" means something that varies (obviously). A more useful description for us would be: "something that can be controlled, changed, or measured in an experiment." Let's say that we think that when X changes, Y changes. To test this out we would change X in some uniform way, and measure the effect on Y.

An example is helpful here. You may be familiar with the notion of word frequency, that is, the frequency with which a given word occurs in the language (typically measured by means of counting the number of times it appears in a corpus). Word frequency has been shown to affect linguistic processing in a number of ways, including the speed with which we can respond to word stimuli (by reading them aloud, or perhaps deciding whether the word is an actual word in the language or a made-up pseudoword).

How would you test this hypothesis that word frequency affects word recognition speed? Well, clearly you would need to measure Y (word recognition speed), and you would do this at different levels or values of X (word frequency). This is fairly easy to implement: we get our corpus, we find out the frequencies of some words, we get people to read the words, we measure how long it takes them, and we then test to see if there's a (statistical) relationship between the variables: whether the scores on the two scales are **correlated**.

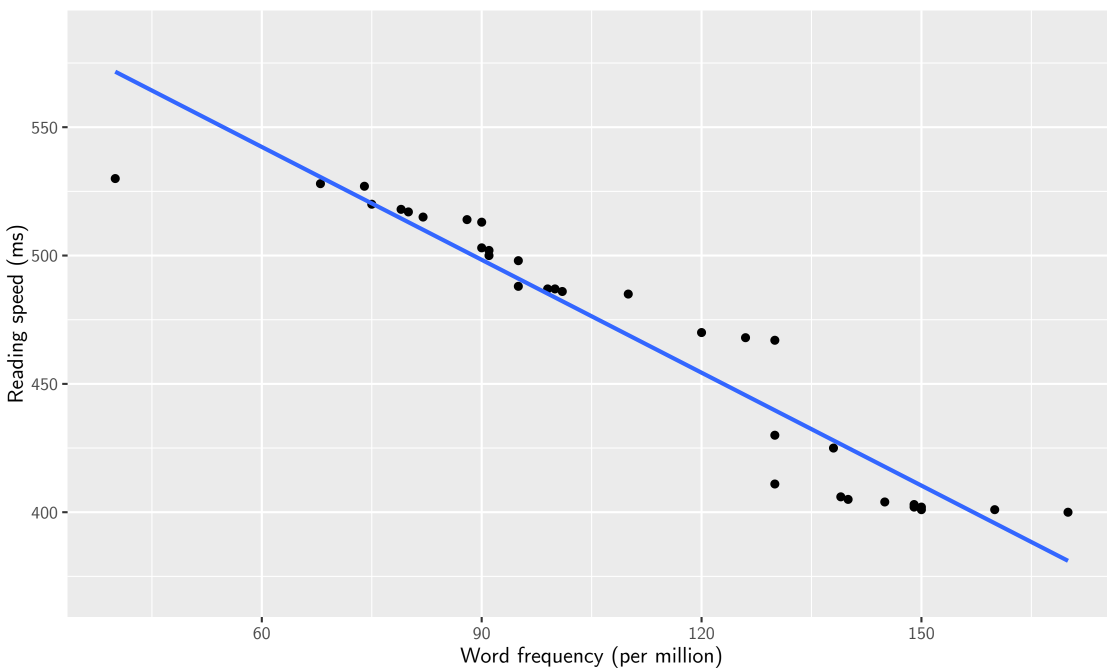


Figure 2: Pretend correlation scatter plot of word frequency and word reading speed. The blue line represents the direction and strength of the correlation. In Chapter 15 you'll learn how to create a fancy graph like this.

So far so good. But we are not always working with variables that are on **scales** (such as speed, frequency and age) where predictions can take the form of "the more X, the more Y". Sometimes we just want to know, are values of Y significantly different when measured for different **categories** of X? An example of a categorical variable could be mother tongue, or put differently, whether or not you are a "native speaker" of a language (by some criteria). We might then ask, do native speakers read words more quickly than non-native speakers? So then we measure how quickly people read words and compare the speed between native speakers and non-native speakers.

The variable we change is often called the explanatory variable or independent variable, and the one we measure is often called the outcome variable or the dependent variable.

1.2.2 Controlling variables

When you do find a relationship between the explanatory and outcome variables, you want to be sure that the relationship you're looking at isn't actually caused by something else. This is a fundamental aspect of how you design the task and the stimuli, and how you recruit your participants. In fact, most time in experimental design and preparation is spent on making sure that all the variables that aren't being measured or changed are well **controlled**. You can control variables by keeping them constant (e.g. the same number of trials per participant), but you can also just make sure that if they do vary, the variation is random. The point is that no extra variable should vary *systematically* with the variables you are investigating. A variable that does vary systematically with one of the variables of interest is called a **confound**.

To make this clearer, let's look again at our example of the relationship between word frequency and speed of reading the words aloud. It's also the case that more frequent words tend to be shorter words. So if people are quicker at reading more frequent words this could be down to the fact that those words are shorter (and so quicker to process), *unless*, when you design your experiment, you make sure that among the words you're using, this pattern doesn't hold. So you can use words of various lengths, but you will make sure that length of word and frequency of word are not correlated in your set of stimuli.

Question 1 - Explain how the following situations may lead to confounds. Propose solutions for these shortcomings.

- the experimental stimuli are shown always in the same fixed order.
- while one participant is doing the experiment, other participants are sitting around watching while they wait their turn.
- after testing a participant, you tell them exactly what you are researching. They chat to their friends about your experiment and then their friends come along to take part.

- the only remaining speakers of the language you want to test are over 50 and hard to find. You'll contrast it with Dutch, because for Dutch you can just ask your friends at university.

1.2.3 Drawing conclusions

So let's say we run a well-controlled experiment and we find an effect, whether that's a correlation of word frequency and word reading speed, or a systematic difference between native and non-native speakers. What we're really interested in are effects and relationships that exist "in the wild". In our experiment, we won't test every speaker of the language - we'll test just a few people. And we won't test every possible word in the language, either. We'll just test a sample. To what extent does this actually tell us something about the entire language or speaker community?

Hypothesis testing

Well, we can't directly test what happens in every possible instance, so we can never be 100% sure that our hypothesis is true. There might always be another case out there in the world that disproves our hypothesis! What we do is use statistical testing, because what statistics does is to turn the hypothesis around to give us something we **can** test. The hypothesis that we can test is called the null hypothesis. With the null hypothesis, we hypothesise that there is *no* difference between the groups, or *no* relationship between our scales (that's why it's called the *null* hypothesis).

Then we can ask, **what is the chance of getting our result if the null hypothesis is actually true?** In the real world, it could be the case that whether you're a native speaker or not doesn't actually affect how quickly you can read out words, but it just so happened that in our experiment, there was a difference between the two groups we tested. The statistical test will tell you the probability of finding this difference between the native and non-native speakers, if in the real world, being a native speaker makes no difference. We consider that a probability below 5% (0.05) is so extremely unlikely, that it's most probably the case that the null hypothesis (the hypothesis that there's no real difference) isn't actually true. This is a complex topic; the important things to note here are that statistics doesn't give the entire answer (there's still that 5% chance!) and that all we can do is reject – or not reject – our null hypothesis. We don't get to *prove* anything (unfortunately).

It should be clear by now that statistics isn't a recipe which allows you to hit a few buttons on your screen and go home early, or a slot machine which you enter data into and get little pieces of truth out. It's a tool for assessing your data and coming up with robust and valid conclusions. Use it wisely, and be alert to others who don't.

Validity

As mentioned above, the way you choose your participants is important for controlling variables. It's also important for generalising our findings to the wider population using statistical tests. We need to make sure that the set of people we test do actually form a good representation of the actual population. Otherwise even if we do good statistics, we are drawing conclusions about the wrong people. In other words, our conclusions need to be **valid** for the population we are interested in.

Question 2 - How can you ensure that the test sample is representative of the population? What is the danger if you only get data from a few people?

Validity comes down to how well your experiment represents the (real-world) situation you are trying to investigate, and so it applies to all aspects of experimental design. To make sure your conclusions are valid, you also need to be cautious about how you generalise your results.

Causality

Beware: just because you find a relationship between two variables, it does not mean that it is a **causal** relationship. For example, just because word frequency and word reading speed correlate, it doesn't mean that higher word frequency is necessarily the *cause* of speedier word reading. As noted above, in this case the pattern in the data could be due to another, unseen variable (see section 1.2.2), but relationships observed between variables can also sometimes just be due to chance. If you need to be convinced that correlation doesn't imply causality, take a look at Tyler Vigen's Spurious Correlations site.

Question 3 - There used to be an advertisement on television for a breakfast cereal called Shredded Wheat. The advert stated that "people who eat Shredded Wheat tend to have healthier hearts." What is misleading about this advertising claim?

We will discuss more about the process of experimental design in Chapter 4. In the meantime we recommend that you read the first chapter of *How to design and report experiments* by Andy Field and Graham Hole (available from the University Library) to get a better understanding of the topics covered in this chapter.

Question 4 - In 2015, this article appeared on the Science Daily website with the headline 'Climate affects development of human speech'. Read the article and answer the following questions:

- **What were the actual findings of the study mentioned in the article (rather than the interpretation made in the headline)?**
- **What is the physiological explanation (i.e. relating to the human body) that the article suggests for the relationship between climate and tone?**
- **What are two possible alternative explanations for the association that the study found between climate and tone?**
- **One of the researchers states that this study can be taken to "imply ... that climate can, over the long haul, be one of the factors that helps shape languages". Do you agree? (If not, what evidence would you need to see to convince you that climate can affect the evolution of language?)**

2 Research ethics & scientific integrity

LEARNING GOALS

- recognise types of misconduct relating to integrity and ethics
- be able to discuss and propose solutions for problematic ethical cases

Research is subject to certain rules of conduct. In this chapter you will learn why and how we implement such rules in experimental research. We will focus on how researchers should behave in relation to their community and how they should respect individuals who are taking part in experiments and fieldwork.

2.1 Scientific integrity

Research misconduct relates to dishonesty. As a researcher you can be dishonest about the way you compose your writing but also about the way you collect, handle, present and interpret your data. Hopefully, you already know that for the papers and theses that you write as a student, you cannot just copy and paste from books, journal articles and internet without proper referencing. In Chapter 3 you will learn skills to turn the usually tedious task of referencing into a fast and smooth operation.

Exercise 1 - Find an example of plagiarism outside academic research. Describe the negative consequences.

There are numerous websites offering (scientific) text services, but be aware of the fact that it is not OK to pay a ghostwriter or website to write your stuff. Everything that you produce should be your own and original. When you have finished writing your piece, it is good practice to have it proofread by some others. They can make small corrections or suggest to rewrite bits but they should not rewrite whole paragraphs.

If you have been following the Dutch news in the last few years, you may have come across the name of Diederik Stapel, a former professor of social psychology. He was fired for intentionally fabricating and manipulating his data. Making up your data is clearly wrong. Manipulating data is also unacceptable, but it's harder to spot. One form of manipulation is the concept of *HARKing* (Hypothesising After the Results are Known) as described by Kerr (1998). This means that you change your hypothesis to fit your results, which is a misrepresentation of the research process. The bottom line is that you should be transparent and honest about your hypotheses, predictions and results – even when they are totally different from your expectations.

Question 5 - What are the consequences of invented data for the research community? And for the general public?

2.2 Ethical treatment of research participants

If you do experimental research where you test participants – as all of you probably will as part of your studies in Leiden – you are not just responsible for your conduct regarding your own writing and data handling, but you are also responsible for treating the participants in your research appropriately.

It is clear enough that you should approach participants in a polite and respectful manner, but there is more to research ethics than just this. Firstly, each researcher at LUCL is required to read and sign the LUCL Ethics Code. This is a document setting out the code of conduct you are expected to stick to.[2] Secondly, the standard procedure for experiments at LUCL requires that you ask each participant for "informed consent". What this means is that you provide each participant with an explanation of what they will be doing (without revealing your experimental design!), and ask them to sign a form to declare that they agree to undergo the procedures described. The information sheet also states that participants are permitted to withdraw their consent at any time and without giving a reason. In such a situation, you should also delete any data you have collected from them.

Often we give participants some money to compensate them for the time they have spent taking part. When this is the case, we are required to have participants give minimally their name, date of birth and signature on receiving the payment (preferably also their address).

Informed consent procedures are fairly straightforward for most experiments we do. However, issues can arise when working with different types of participants, or in non-laboratory settings. Firstly, if your participants are under 18, they cannot give informed consent. The parent or legal guardian must give the consent, alongside the participant's own agreement (which is termed 'assent').

Question 6 - At LUCL we have a Baby Lab. The parents give consent for their babies to participate in the research. However, it's also important to be assured of the infant's 'assent'. How do you think you could judge this?

[2] Currently you need to access Sharepoint to download this; we will introduce Sharepoint in Chapter 11.

Secondly, it is possible that you might work with groups of participants who are not comfortable with giving their personal details, or with signing forms (especially when they have low levels of literacy); you may even work with a language which is not written.

Question 7 - How could you fulfil the requirement of informed consent for such populations?

Your obligation to your participants does not end when they walk out of the door - you have a duty to keep all of their personal details and data private. The issue of data privacy becomes complex when you consider that you may store this data for many years.

Question 8 - You gather informed consent from your participants, and the form participants sign states that the data is held and processed only by the researcher. If in 30 years the language you worked with is no longer spoken, should you allow other linguists to gain access to this increasingly valuable data?

Lastly, you should be aware that there are some forms of research that require approval by an ethics committee because they are invasive or have the potential to cause harm to the participants. It's unlikely that you will have to go through this process, because the kinds of procedures that are typical in experimental linguistics do not involve this kind of risk. However, any time you come into physical contact with participants you must maintain hygiene and safety standards to ensure that you do no harm.

Research ethics is an ongoing discussion, meaning that although today you may find you are able to get away without paying much attention to ethical procedures, you could in the future discover that your data is not considered acceptable for publication. Keep up with developments at LUCL and in the wider field, so that you are not caught out.

Exercise 2 - Find an example of scientific misconduct reported in the media. Write a short paragraph about the case, explaining the type of misconduct involved and the possible impact of this misconduct.

3 Bibliography handling: finding, managing and referencing literature

LEARNING GOALS

- set up a bibliography: search, save and organise
- present references appropriately in written texts

We expect you to already be aware from your ongoing studies of the essentials of referencing and discussing different sources. In this chapter, we will introduce you to techniques and tools for conducting a literature search and review in experimental linguistic work, including how you should present your references according to different style requirements.

3.1 Literature search

When conducting any independent research, you will need to first spend time becoming familiar with the relevant literature in that area. This means knowing the key publications and latest developments on your topic; it also means being able to draw together results of studies similar to yours and be able to summarise and compare findings that may not always converge on a single theoretical interpretation. To be able to do this effectively, you need to have a straightforward and transparent procedure for searching for relevant papers and books.

3.1.1 Where should I look for literature?

One obvious place to start is the University Library Catalogue. The University Library also employs subject librarians, who may be able to guide you in gathering references for your topic of interest. Check out the University Library's linguistics subject guide page for more information and contact details.

You can, of course, simply use your favourite search engine to start looking for literature related to your topic. Google also has a dedicated academic literature search engine, Google Scholar, which gives extra functionality for finding academic publications. In this situation – and in the case that you look for publications directly through journal websites – remember that to access most papers you need to be logged in to the University Library's proxy to be granted permission to download the content. It's easiest to do this by locating the paper through the University Library

Catalogue, which will prompt you to log in.[3] PubMed Central is another useful tool for doing a literature review. It is an archive of biomedical and life sciences literature, including experimental psycho/neurolinguistics.

Find out if there are any existing literature reviews or meta-analyses[4] in your area of interest. These not only bring together relevant references, but also provide you with a summary discussion of the theory.

An important skill is to adopt the habit of looking through the reference lists of papers you read – especially recent ones – as this will lead you to additional relevant publications (some online journals will even propose related papers when you download a copy of a publication).

The main online academic networks, Academia.edu and ResearchGate, can be useful for accessing research that is not (yet) officially published (including conference posters) and for getting in touch with researchers, asking questions to the community and so on. Many well-established researchers also host copies of their publications on their own websites, especially when these are not easily accessible elsewhere.

3.1.2 How do I know that I have all the right papers?

There is no real way to "know" or "judge" that a bibliography is correct; however it should minimally contain the important, well-known publications related to the topic and be up to date with the latest publications. In a long-term research project, it can happen that in the meantime a highly relevant or even overlapping study is published – it's better to find that out in good time, rather than after submitting your work to a journal!

The best way to tackle this issue is to allow enough time for a thorough literature search, perhaps making summary notes for each paper you read in an organised manner so that you can quickly refresh your knowledge. As with many aspects of research, it's advisable to approach knowledgeable colleagues or supervisors for their feedback – they may suggest important publications that you have missed.

Of course, you will want to use a program to help you organise and keep track of all the literature you are collecting (and you need to get this set up before you start). Luckily there are freely available tools for this purpose, as you'll read in the

[3]If you use Google Scholar, you can get it to link directly to the University Library catalogue. Go to settings then library links, add Leiden University, and save. Now you'll see a *GetIt@Leiden* link next to your Scholar search results.

[4]A meta-analysis is a kind of review where the results of a range of studies are pulled together into a single statistical analysis, rather than simply discussed.

next section.

3.2 Managing your bibliography

Whether you are searching for specific references or just browsing, it is good practice to stay organised. With the help of a reference manager you can do so quickly and efficiently:

- you can collect bibliography data in just one click (only rare titles need to be added by hand).
- you can easily search your database and make sub-libraries for different projects.
- you can instantly use your database during writing (we will show you how in section 3.4).

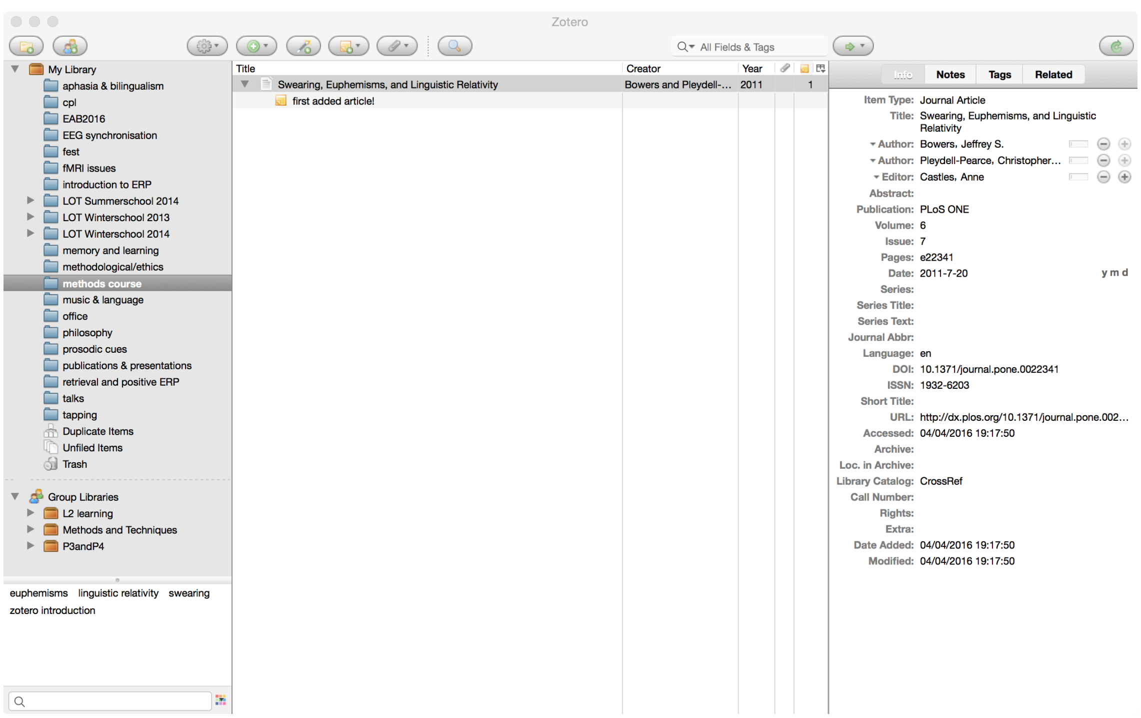


Figure 3: Zotero's standalone desktop workspace

Some reference managers will also allow you to invite others to work in a group.

Zotero offers all of these functions. There are about 30 other such managers around – and you are free to choose one of your liking – but we think that Zotero is the best free open source option to go for. You can run a standalone version on a Mac, Windows or Linux, or use it in the internet browser Firefox. Beware though: your online account only allows you to store up to 300 MB of files (unless you pay a small fee). If your library gets bigger than this (and you don't want to pay to upgrade) you'll need to use the standalone version. If you want to, you could continue to sync the library with your online account, but save your large files (e.g. PDFs) separately and make links to them within Zotero[5]. Alternatively, you can just abandon syncing altogether and store everything in Zotero Standalone. If you'd like to store everything in Zotero, calculate 2 MB (give or take) per entry.

In any case, it's important to keep your library backed up – read more about this on the Zotero FAQ pages.

3.2.1 Getting started

After you have signed up here you can start working. On university computers it's probably best to use Zotero in Firefox – which is the original way. Alternatively, you could use Google Chrome. If you normally work on your own computer or laptop you might want to try the standalone version. Open Firefox[6] or Google Chrome and install the Zotero add-on here. You may need to restart your browser as part of the installation process. Once it's complete, you'll see the Zotero button in the Toolbar of your browser. By clicking the Z-icon you can view your library while browsing.

Exercise 3 - Make a sub-folder in your Zotero library (i.e. add a folder), give it a name, search for the paper Bowers & Pleydell-Pearce (2011), which is an article on euphemisms. As soon as you arrive at a page with information about a paper Zotero will notice this and you'll be able to save it to a folder in you library using the Zotero button in the toolbar (it's next to the Z-icon). How do you get it directly saved to the appropriate folder? Where does your data go otherwise?

[5] Linking to files will only work if both Zotero and files are located on the same system folder!

[6] Maybe you need to install it: open an internet browser and search for Firefox and follow the instructions. It'll be done in a minute.

3.2.2 Collecting and organising

There are many ways to import data into your Zotero library. Check the buttons left from the search button in the toolbar.

Exercise 4 - Find the easiest way to import data of the following Digital Object Identifier (DOI): 10.1016/S0165-0173(99)00060-0. Assign it an extra tag "swearing".

If you already have PDFs you can drag them into the library/folder pane and let Zotero extract the meta-data: right mouse click on the item(s) and choose "Retrieve Metadata for PDF". Really cool! Now, take a look at Figure 3 and do the following exercise.

Exercise 5 - Add the same tags that you can find in the left corner to the two files that are in your folder. Add a note to both entries. Search your library on the basis of the tag "swearing".

As said, you can save data directly to the folder you want. All data can always be found in "Library". You can check the "Duplicate Items" folder to find out if you've downloaded any references more than once. This can be very useful if you decide to merge different project folders later on - it makes the process of removing duplicate copies much easier.

Lastly, note that while Zotero works well most of the time, occasionally wrong or incomplete metadata may be retrieved. Keep an eye on the process and check your bibliography over before you submit any work.

3.3 Presenting references in your written texts

This section concerns the way in which you present the details of referenced literature when you write up your work. This includes both references in the running text ("citations"), and the reference list at the end of the text. Although there are some basic requirements for this, such as author surnames, titles, years of publication and so forth, you will find that there are numerous different styles for the exact formatting of the information. What we mean is that different journals and publishers have specific demands about such miniscule things as whether to use brackets around the publication year, whether to state the authors' full first names or simply initials, how many authors there need to be to use the phrase "et al." in the running text, and so on.

When submitting a paper to a journal, there will normally be a detailed style guide provided, informing you exactly how to format not only your references but also your graphs, tables and diagrams, abstract, title and so on. For the references, you may alternatively be simply asked to use a certain established style. In our field, we typically use what is known as "APA" style. This is based on a publication manual produced by the American Psychological Association. You can find guides to this style online, but in doubt you should consult the manual itself which you can find in the library.

Here's an example of a citation and a reference in APA 6th Edition style (produced by Zotero):

citation:

(Bowers & Pleydell-Pearce, 2011)

reference:

Bowers, J. S., & Pleydell-Pearce, C. W. (2011). Swearing, Euphemisms, and Linguistic Relativity. PLoS ONE, 6(7). http://doi.org/10.1371/journal.pone.0022341

To compare, here is the format required by the Chicago Manual of Style 16th Edition:

(Bowers and Pleydell-Pearce 2011)

Bowers, Jeffrey S., and Christopher W. Pleydell-Pearce. 2011. 'Swearing, Euphemisms, and Linguistic Relativity.' PLoS ONE 6 (7). doi:10.1371/journal.pone.0022341.

While these may seem like annoying little details, they can take up a lot of time if you are not aware of them in advance and have to go back through your references correcting the formatting. Luckily, there are a number of inbuilt styles in Zotero which means that you can simply choose which format you require (e.g. APA) and your reference list comes out with the dots and brackets in all the right places!

Exercise 6 - Create references like the ones above (i.e. the complete reference in different styles). You'll find this option in Zotero when you right-click an item and choose `Create Bibliography from Item...`

Paste them into a text file.

3.4 Cite as you write

Rather than just exporting citations from Zotero afterwards, you can do it as you write in Google Docs, LaTeX and Word. As the functionality is rather poor in Google Docs and sometimes a bit complicated[7] in LaTeX, we will look at the integration with Word. For this to work you will need to download a special plugin here. Install it and restart the program if asked. Once it's complete, open a Microsoft Word document. You should see a new `Zotero` tab along the top menu of your Word window.

If you work with a standalone version, you can install the Word add-in from the preferences pane as in shown in figure 4.

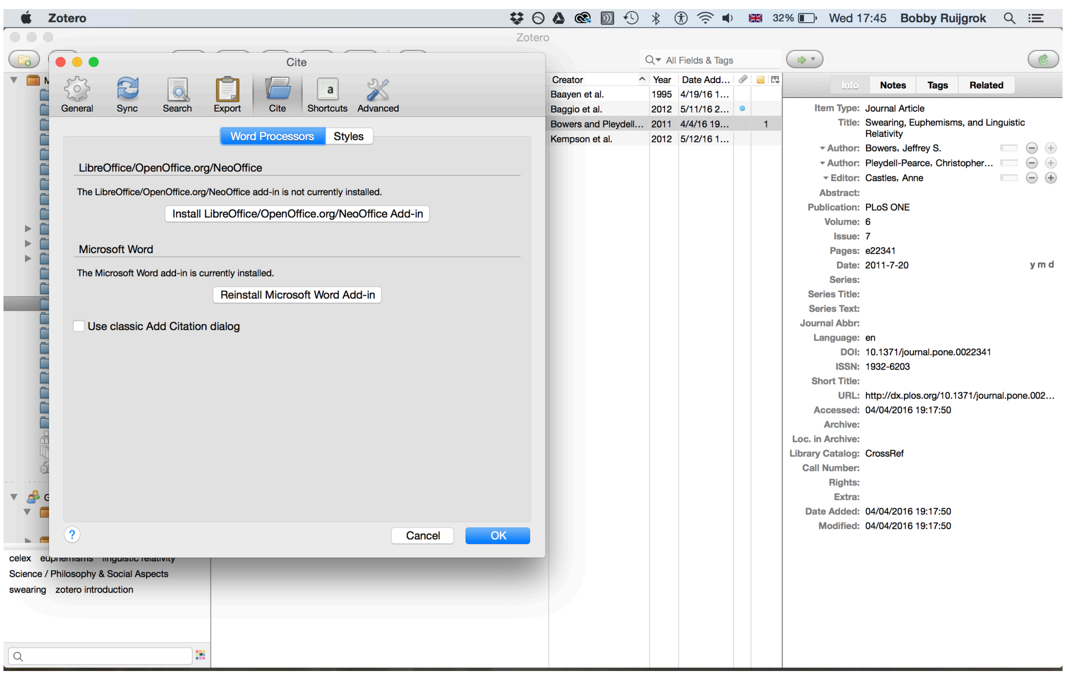


Figure 4: Preference pane to install add-ins.

[7] If you want to try LaTeX, start by using an online editor such as ShareLaTeX. They have many templates available which make it much easier to get started. By the way, the file you're reading right now was created in LaTeX.

Exercise 7 - Have a look at this instruction video. After that, open an empty Word document. Create some text and add two different citations such as: (Bowers & Pleydell-Pearce, 2011). Make a separate header "References". Move your cursor underneath this header and choose "Add Bibliography".

3.5 Putting it all together

The final assignment of this chapter (to be completed as homework) is to use what you have learned in the previous sections of this chapter to research and propose a bibliography based on a specific research question.

You'll be working on a real research question that was investigated by Olga Kepinska, a former PhD candidate at LUCL, as part of her PhD thesis. Once you have made your bibliography, you'll be able to compare it with an example bibliography proposed by Olga.

Exercise 8 - For the following research question: *Does language aptitude, as measured by language aptitude tests, correspond to particular localization patterns of brain activity during L2 learning?*

(a) use online resources to gather a list of ten important references for a paper addressing this research question (see section 3.1);

(b) collect and organise your references in Zotero (see section 3.2);

(c) export and present your bibliography in APA style in a Word or PDF document (see section 3.3).

4 Conceptualising an experiment

LEARNING GOALS

- determine explanatory (independent), outcome (dependent) and control variables appropriate for addressing a given RQ
- propose feasible operationalisations for IV and DV (given practical limitations)

Conceptualising an experiment means going from your research question, which might be quite broad, to a design for an experiment that you could actually carry out. In this chapter we build on the material from Chapter 1 from a practical point of view.

4.1 Independent, dependent & control variables

In Chapter 1 we defined experimentation as investigating the relationship between two (or more) variables, while trying to keep all other variables constant. We introduced the terms **independent**, **dependent** and **control** variables, and we learned that when we do an experiment we are interested in whether our dependent (or "outcome") variable changes systematically with changes in the independent (or "explanatory") variable. Typically we do this by *manipulating* the independent variable in some way, and *measuring* the outcome on the dependent variable, and testing to see if there's a significant relationship. We additionally want to be sure that any apparent relationship between the independent and dependent variables can't be explained by some **confounding** variable, so for this reason we make sure to control all other variables that could have an influence. This is a useful summary, however, when we get down to the practicalities of running real experiments, we find that it's a bit more complex than this.

- The independent variable is not always a variable that the experimenter actively *manipulates*. For example, imagine that we want to see if there's an effect of age on reaction time - does it take longer to name pictures the older you are? Age is the independent variable here, but of course it's not the case that we can increase or decrease a participant's age to see the effects! We must instead select a number of participants of different ages, record their ages, and use this as the explanatory variable in our analysis.

- A variable is simply something that can change – it's the experimental design which tells you whether a variable is independent, dependent or control. For example, imagine that we've done our experiment to find out if age affects reaction time, and we've found out that older people tend to be slower at responding. Imagine that now we want to investigate whether time of day (morning or afternoon) affects reaction time. We better be careful to check our participants' age before we test them! Age is the independent variable in the first experiment, but it's a control variable in the second experiment.

- It should start to become clear by now that controlling for something doesn't always mean *keeping it constant*. Often it does – for example, we give every participant the same instructions so that everyone should have the same understanding of the task. But often it's simply not possible to keep things the same. For example, when age is a control variable, we could decide that all participants should be exactly the same age, but in practical terms that may not be realistic. Remember from Chapter 1: the key is to make sure that control variables do not vary *systematically* with the independent and dependent variables. For example, when we test whether time of day affects reaction time, it's a problem if all the people we test in the afternoon are significantly older than the people we test in the morning. In section 4.3 we discuss some straightforward ways to approach this issue.

- Not everything that is *measured* by the experimenter is a dependent variable. Sometimes you need to measure a variable in order to control for it. This is either because it's just not possible for you to actively control it, or it's just so that you can check later that this variable really didn't have an effect. For example, imagine we are investigating the effect of mother tongue on learning a second language - we might also measure participants' memory abilities, because we want to be able to say with confidence that their language learning scores weren't simply due to how good their memory is. So, although we measure memory performance as part of our experiment, it is a control variable, not a dependent variable.

Question 9 - Reaction time is typically a dependent variable. Can you think of an experiment where reaction time is the independent or control variable?

4.2 Between and within participants designs

When we test a number of different values or levels of the independent variable – for example, the time of day in our morning vs. afternoon reaction time experiment – we call these different levels "conditions". We are interested in comparing the outcome (e.g. reaction time) in one condition with the outcome in the other condition. There are two different ways of setting this up:

- We could test each participant on one of the conditions. So, each participant is assigned to either the morning or the afternoon group, and we compare the outcomes of the two groups. This is called a **between participants design**, because we compare outcomes between participants.
- We could alternatively test each participant in the morning and then again in the afternoon. Then we compare each participant's morning reaction time against their afternoon reaction time. this is called a **within participants design**, because we compare outcomes within each participant.[8]

How do you decide which design to use? Well, sometimes it's clear from your variables which to choose. For example, if you have an independent variable such as age or mother tongue, which is a characteristic of an individual, you have to compare between participants because you can't actually manipulate these variables. However, when it's not decided by the variables, it's usually better to choose a within-participants design. The main reason for this has to do with control. Remember that when we control variables, we're trying to make sure that these variables don't vary systematically with the independent variable. When you test your experimental conditions on different groups of people, there's a risk that some characteristic variable like age, gender, working memory etc. will end up being significantly different between the groups, simply because people vary in many ways. When you are comparing *within* participants, things like participant age and gender will obviously be the same between testing conditions.

Question 10 - Decide whether each of the following hypotheses should be tested within or between participants.

- Left-handed people are faster at lexical decision tasks.

[8] These are also known as between subjects and within subjects designs – "subject" is just another word for participant.

- The older we get the harder it is to learn a new language.
- Tongue twisters get easier the more you say them.
- MA students use longer words than BA students.
- Studying linguistics makes you better at learning foreign languages.

4.3 Control techniques

Controlling variables and making sure that you haven't got any confounds is a difficult task. In our experience, this is the part of experimental linguistics which takes the most time and effort! This is because control affects every part of the experimental setup process – how you recruit your participants, how you design your experiment, how you select your stimuli, how you instruct your participants, how you present your stimuli, how you plan your analysis... the list goes on. You'll be pleased to know however that there are a few standard "tricks" used by every researcher to make this difficult task more manageable.

4.3.1 Randomisation

Probably the most simple and effective way to reduce the risk of confounding variables is to randomise. When something is random, it should not have any systematic pattern, and therefore it is the easiest way to reduce the chance of the kind of systematic variation that leads to confounds. The technique of randomisation can be applied in many ways. For example, if you have to test between participants, you should put the participants *randomly* into one of the groups. Normally you should show your stimuli in a *random order* - and even better, the order should be *randomised* again for each person, so that everyone sees a different order!

Question 11 - What confounds may occur if you don't randomise the order of stimuli?

Of course, you may have already realised that randomisation doesn't totally guarantee that differences are not systematic by accident. It's always good to double check that no systematic differences have crept in, especially with variables that are known to be influential (remember our example above of age as a control variable in a reaction time experiment!)

4.3.2 Counterbalancing

Another important technique for controlling variables is called counterbalancing. As the name suggests, this has to do with balancing groups and conditions - essentially that instead of randomly grouping or ordering things, you construct several groups or orders that are balanced in a way that cancels out systematic variation. For example, imagine that you want to test whether priming a word makes it easier to recognise. To do this, you decide to prime each word with either a related word, or a made-up word (pseudoword). Then you want to compare the reaction times for the related-prime condition with the pseudoword condition. You don't want to use a between-subjects design, because individual people have quite different reaction speeds. However if you show each participant both conditions (i.e. using a within-subjects design), they will see each word twice, which means that many of the reaction times you measure will be reaction times to a word that was already seen earlier!

This is where counterbalancing comes in handy. You split your list of target words into two: part one and part two. Then half the participants see part one with related-word primes and part two with pseudoword primes, and the other half of the participants see the reverse: part one with pseudoword primes and part two with related-word primes. Figure 5 makes this a bit clearer!

GROUP A

PRIME TYPE	TARGET
related word	fish
related word	skirt
related word	spade
related word	cherry
related word	floor
pseudoword	badge
pseudoword	dog
pseudoword	tree
pseudoword	frog
pseudoword	bell

GROUP B

PRIME TYPE	TARGET
pseudoword	fish
pseudoword	skirt
pseudoword	spade
pseudoword	cherry
pseudoword	floor
related word	badge
related word	dog
related word	tree
related word	frog
related word	bell

Figure 5: Example of counterbalancing two parts.

When do you randomise, and when do you counterbalance? The answer to this depends on the experiment. For example, in a within-participants design, we normally present trials from all conditions mixed up in a randomised order. However for some experiments this isn't possible (imagine if your two conditions are morning vs. afternoon!). In this case, you could counterbalance the order in which you test the conditions. Typically, in a within-subjects experiment you will combine randomisation and counterbalancing techniques.

Question 12 - How could you use counterbalancing to avoid confounds when testing reaction times in the morning vs. the afternoon?

4.3.3 Norming

Another very important control technique - and one that takes a bit more time - is norming. As the name suggests, this technique is about finding normal scores on certain variables so that you can avoid systematic differences in these variables when setting up your experiment. One variable which can influence reaction time when naming pictures is the *name agreement* of the image. Name agreement refers to how likely it is that everyone names a picture with the same word. The idea is that if a picture has several equally appropriate names, it's less easy to choose a name, and so it may take longer for participants to respond to that picture.

To get information about this variable, you need to conduct a norming survey, where you show a number of people the pictures and ask them to name each one. You then collect all the responses and calculate the name agreement score, for example by taking the most common name and calculating how big a proportion of responses it accounts for:

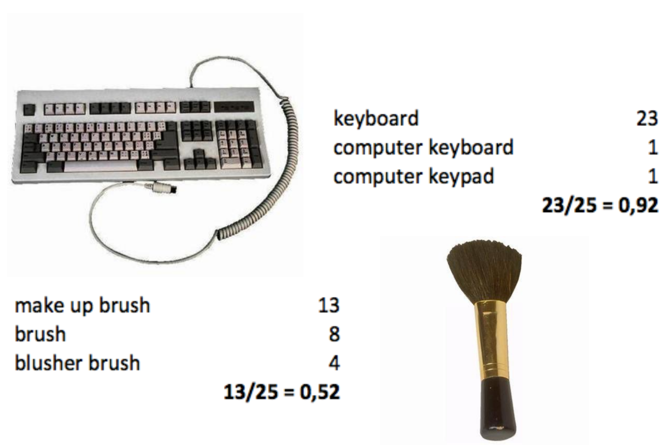


Figure 6: Example name agreement responses and scores for two images.

You can then use these scores to check that the name agreement of your pictures is not (significantly) related to any variables you're testing.

Norming surveys must be carried out with people who are not taking part in the actual experiment (otherwise you're just going to get even more confounds!) and it's a good idea to follow standard experimental procedures, such as screening participants, randomising the order of stimuli and giving each person the same instructions. Survey software such as Qualtrics is handy for this, even allowing you to distribute your survey online to save you the hassle of interviewing each person yourself! You'll learn more about this later in the course (see Chapter 7).

Exercise 9 - Come up with at least two variables that you might need to use a norming survey to gain information about. You could look at published papers to find examples. How would you go about collecting the norming data (e.g. what kind of scale or task to use, how to calculate the final scores)?

4.3.4 Fillers and cover stories

Not all confounding arises from design or stimuli. Sometimes it comes from the participants themselves. Humans are on the one hand curious, and on the other hand always looking for the easiest way to get things done. Consequently, participants typically try to figure out what you're researching, and how to complete the task with minimal effort. You can use filler trials to disguise the hypothesis – trials that are just there to fill up the experiment and aren't related to your hypothesis. You should also think about how you will present the task to the participants: give them an explanation that satisfies their curiosity but doesn't reveal too much. However you should avoid actually lying to your participants! You may wish to offer a 'debrief' once the experiment is finished.

4.4 Operationalisation

Looking back over this chapter, you will see that you can often express your variables in two ways – a general way that makes clear the idea you are testing, and the practical way that enables you to actually test it. For example, when our research question is "Does time of day affect reaction time?", our independent variable can generally be stated as the "time of day", but in order to actually test this, we have a practical definition of time of day which is "morning vs. afternoon". This process of coming up with a practical, testable version of a variable is called operationalisation.

Sometimes the variable and its operationalisation are not really different: when age is your independent variable you will simply test people of different ages. Typically however you'll need to come up with an operationalisation for each variable, and there may be different possibilities. For example, we might test the variable "tiredness" by measuring reaction time, but we could also ask people to report how tired they feel on a scale of 1 to 10. Some variables are almost impossible to operationalise – for example, imagine the research question "Do bilinguals dream in both their languages?" Of course you can ask people what they think about this question, but testing it scientifically is beyond our capabilities.

Clearly how you operationalise your variables has consequences for your experimental design. For example, an experiment where we operationalise "alertness" as a score on a memory test will be a very different experiment from one where we manipulate "alertness" by giving people different amounts of coffee. In any case, when you operationalise your variables you need to make a trade-off: on the one hand you need to stay as close as possible to the idea you are testing to ensure validity (see Chapter 1), but on the other hand you need to make it practical so it's possible to actually do the experiment!

Exercise 10 - Come up with two different operationalisations for each of the following variables:

- **Foreign accent proficiency**
- **Level of politeness in language use**
- **Language degeneration due to cognitive decline**

5 Stimuli preparation

LEARNING GOALS

- extract information from CELEX for stimuli preparation
- use Excel to prepare & organise stimuli sets

5.1 Navigating the lexical database CELEX

We have talked about word frequency as control variable earlier in section 1.2.2 and you may have implemented it in the last assignment. In this section we are going to take a look at CELEX (CEnter for LEXical information, Baayen et al., 1995), a lexical database that you can extract data from. It contains information about orthography, phonology, morphology, syntax and, importantly, word frequency. Especially if you are running experiments in Dutch, English or German, this tool is the one to know about.

CELEX was developed at Radboud University Nijmegen, the *Instituut voor Nederlandse Lexicografie*, the *Instituut voor Perceptie Onderzoek in Eindhoven* and the Max Planck Institute for Psycholinguistics in Nijmegen. It provides lexical information for English, Dutch and German, and is based on a number of sub-corpora (for example COBUILD for English, INL for Dutch). There exists a web interface (celex.mpi.nl) but it may be rather slow at times. Also, it may take a while to get to grips with all the abbreviations that are used. So, we'll explore the 'disk version' and learn how to extract the data that you need first. Once you're familiar with the basic concepts you could try the online version.[9]

Once you've opened the CELEX folder you'll find the data for the three different languages arranged in three different folders. Within these folders you'll find subfolders starting with the letter of the appropriate language (**d**utch, **e**nglish, **g**erman), then a letter telling you whether it is about **o**rthography, **p**honology, **m**orphology, **s**yntax or **f**requency. The last letter refers to '**l**emma' or '**w**ord forms'. Now, let's have a look at frequency data of lemmas in the English directory. A lemma, as you probably know, refers to a word in all its different inflected forms and is listed under its 'base form' – basically like a 'dictionary' or 'citation' form (for more about lemmas, take a look at the introduction of the CELEX User Guide).

[9]As a student following this course, you will be given temporary access to CELEX files, for learning purposes. Please do not distribute this data among others!

Exercise 11 - Open the CELEX folder and go into the English folder. Open the folder called 'efl'. In this folder you'll find a .cd file which you can open with a text editor (e.g. Notepad++ or Wordpad on Windows; or TextEdit on a Mac). Also open the README using the same program.

You'll notice that backslashes separate columns in the CELEX entries. Every folder in CELEX contains a README giving you information about what the different columns contain. As you can see in the README of the efl file, the first number in a row is the IdNum, which is short for identity number: each lemma has its own ID number.[10] Here you can see the two different lemmas with the base form 'radial':

```
37037\radial\5\0\0\0\5\0\0\0\0\0
37038\radial\19\0\1\0\19\1\0\0\0\0
```

In the English folders, 'COB' indicates frequency data (because its data is drawn from the COBUILD corpus). Rather than the absolute frequency counts, the *frequency per million* (COBmln) is the figure that's generally used, as it is a standardised measure that allows comparison with other frequency corpora.

Sometimes it's hard to understand (or remember!) what the README abbreviations refer to. For this you'll need to get used to consulting the user guides, which have a wealth of information about the database and how it is constructed. Each language has its own user guide which can be found in the main folder for that language (e.g. **eug** is the **e**nglish **u**ser **g**uide).

Exercise 12 - Find out what the Dutch corpus abbreviations Inl and InlMln mean.

Sometimes you may find the number zero as the absolute frequency. This simply means that information is unavailable as the entry has been added to the database at a later stage.

When there are multiple lemmas for the same word, you will need to try and work out which is which. As all the information is coded in numbers, you may need to use Table 1 (below) as reference. These are the numbers used in the entries of esl.cd to indicate which word class each lemma belongs to (check the esl README to find out which column!)

[10] Note, that the lemma ID number is represented as IdNum in Lemma files, but as IdNumLemma in Wordform files!

Word class	Code
Noun	1
Adjective	2
Numeral	3
Verb	4
Article	5
Pronoun	6
Adverb	7
Preposition Conjunction Interjection	8
Single contraction Complex contraction	9
Affix	10

Table 1: Codes for word classes in esl.cd.

Exercise 13 - In efl.cd, search and save the information for all lemmas for *record* to a .txt file. Then use efl.cd and esl.cd together to find out which lemma for *record* is which. What is the frequency for 'record' as a noun?

Homophonous forms can be tricky to differentiate if they are in the same word class! However, sometimes there are ways to figure it out, by comparing the word forms listed for particular lemma numbers, for example.

Exercise 14 The Dutch wordform *hoeven* can be the plural of *hoef* or the plural of *hoeve*. Look up the entries for *hoeven* in dfw.cd. Which one is which?

An experienced awk-programmer could possibly consult different CELEX folders at once. Although this is way beyond the objectives of this course, there is one handy awk-script (written by Rinus Verdonschot) that you could use to get instant information for a whole list of words. This script ACR.exe can be found in the folder ACR, if you would like to take a look. However, it is sufficient for now to work with the CELEX files by opening them in a text editor.

Exercise 15 - Now let's gather some data which we can continue to work with in the next section. Make a list of 20 Dutch nouns, which have a range of different frequencies (per million). Find each word in dfl.cd and copy the entries for all your words into one .txt file. Aim to

get a mix of concrete and abstract nouns. Tip: it's handy to use the Notepad++ program to open the files and also for creating your own list.

5.2 Working with stimuli in Excel

You may associate Excel with spreadsheets for calculating finances and things like that, but in fact it's a really useful tool for setting up experiments and organising datasets. In fact, we use it so much that we think it's important you learn some basic tricks for using Excel. Even if you are familiar with Excel, the exercises in this section will be helpful in showing you how to apply these skills when designing experiments.

Excel is handy for stimuli preparation in a number of ways. The most obvious one is that it helps you to organise and keep track of your stimuli set – for example, if you are using counterbalancing, it'll be handy to create all the different lists and keep them in an Excel file. You can also use Excel to extract information about your set of stimuli, using formulas. For example, you might want to use Excel to calculate something for you, such as the length of each of the words in your stimuli set. In this section we'll take a look at these examples in more detail.

The exercises and examples in this reader were written for Excel 2010/2013 (and Excel 2011 for Mac). Key functions and settings should still remain in later versions of Excel, but be aware there may be differences in how menus and tools are organised in newer versions the program.

5.2.1 Regional settings (to comma or not to comma...)

In an international context such as Leiden University, it's important to be aware that different programs may use different regional settings, and that this can cause problems when you move data between programs.

Regional settings cover things like how numbers, dates and currency are displayed. Many European countries including the Netherlands use the comma (,) as decimal marker, and the point (.) is used as thousands separator. So the number *one thousand two hundred and five point three* is written **1.205,3**. However, other countries, particularly the UK and US, use the decimal point (.) and use the comma as the thousands separator - so the same number is written **1,203.5**. When one program thinks a comma is a decimal and another program thinks it a comma is a thousand separator, the numbers get completely chewed up.

The bottom line is: in the Dutch context, commas are used to indicate decimal points, so you can't use them for indicating anything else; however, this restriction on commas doesn't hold for the UK/US context. Particularly in Excel this leads to some clashes when working in an international context. We will point out the issues raised by this as we go through this chapter, so that hopefully you will not get caught out when you work with your own data.

There is also the language setting of Excel itself. This is actually separate from the regional settings. For example, you can have regional settings for Dutch (commas as decimal separators, etc) even while the program interface is displayed in English. **In this chapter, we use the Dutch regional settings with an English language interface.** Is your Excel interface displayed in Dutch? Before you start the tutorial, please change the display language by going to `File > Options > Language` (or in Dutch `Bestand > Opties > Taal`, of course...!).[11]

5.2.2 Getting a text or csv file into Excel

Before you can do anything, you need to get some data into Excel to work with. If you have a list or table in a text (txt) file or a csv (comma separated values) file, you can open it in Excel. To do this you need to open up a blank Excel file first, and then go to the `Data` tab, then click the button `From Text` (in the area of the ribbon labelled `Get External Data`).[12]

First you need to browse to the file with the data you want to import, and click `Import`. The text import wizard asks you for some info so that it knows how to format your file, for example if it's *delimited* or *fixed width*. You will probably mostly work with *delimited* files, which means the individual cells of a table are separated by ('delimited' by) something like a comma or a semi-colon or another character. For example, we often use the *comma separated values* format. *Csv* is a way to store tables of data in a simple text form that can be understood by many programs. When you see data in Excel, it's neatly organised in columns, but in a *csv* file, these separations between columns are indicated by commas, hence the name *comma separated values*. Although it's called *comma separated values*, you might also find *csv* files that use semi-colons or other punctuation marks as delimiters. Use the little preview window to figure out which delimiter you need and check how your data will look.

After importing, you should always check that your data looks OK – especially

[11] On Excel for Mac it's usually not possible to change the display language, unfortunately.

[12] On Excel for Mac, you can also find this by going to `File > Import...`.

the numbers! If you do find that your numbers get chewed up because of the decimal issue, there are a couple of workarounds. In Excel for Windows, the easiest solution is to change the separators in Excel's main settings. In Windows, you can do this by going to `File > Options > Advanced` and changing the separators (untick the "Use system separators" box and enter the correct separators in the fields below).[13] Another workaround is to open the file in Notepad before you import it and search & replace the decimal points/commas. In general however it's advisable to keep consistent regional settings across programs where possible.

Exercise 16 - Import your CELEX data from Exercise 15 into Excel following the steps above. You should end up with separate columns that represent the different pieces of information for each entry (e.g. Idnum, frequency, etc.). Add column headers for your dataset. In a new column at the end of the dataset, enter for each noun if it is a concrete or abstract noun.

In particular, *csv* files saved in Dutch versions of Excel typically use semi-colons instead of commas as separators (see section 5.2.1).

We'll first look at how we can use Excel to get more information out of the set of stimuli. Then, we can use this to help us decide how to organise the stimuli set into counterbalancing lists.

In this chapter, we assume that you know how to do the most basic things in Excel like inserting new columns and rows, and sorting columns of data. If you are already experienced with Excel, the exercises in this chapter will illustrate how you can apply your skills to setting up experiments, so you still need to work through it. However, you might want to skip straight to **Exercise 18** if you're already used to working with functions in Excel formulas.

5.2.3 Introducing formulas and functions

As mentioned above, we use formulas to ask Excel to tell us certain information about our stimuli set. A formula is a general term, meaning an operation that takes in some values and gives some result. You'll be familiar with formulas from maths class at school – a mathematical sum such as *5+7* is a formula, which returns the value *11* (you might also remember more complex formulas from algebra lessons, such as formulas that define straight lines or curves). Click on a cell somewhere outside of

[13] On a Mac, just change the computer's regional settings in the Localization preferences pane.

your data set, and type =5+7. Now press Enter, and you'll see that the cell displays 12. Also take a look at the bar above the spreadsheet, with *fx* on the left of it: this is the **formula bar**. When there is a formula in a cell, the cell only shows the result of that formula, but the formula bar will show you the formula.

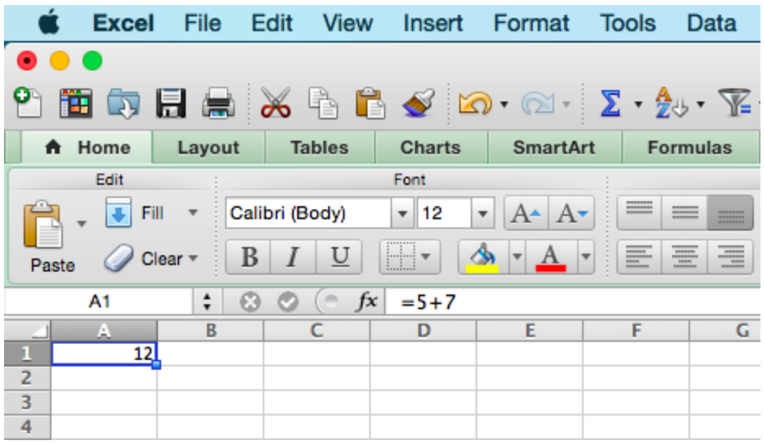


Figure 7: The Excel formula bar.

You might wonder why the equals sign goes at the beginning when you write a formula. It's just the way it works in Excel! You can think of it like this: the value of this cell *equals* the result of this formula. So, for the formula you just typed, the value of the cell *equals five plus seven*.

The values you give to the formula don't have to be numbers that you type in. They can be other cells in the data set. For example, instead of =5+7, you might want to add together two cells that each contain a number. It's very simple: each time instead of typing a number, you use the mouse to click on a cell.

Exercise 17 - In a blank cell, find out the sum of two other cells in your dataset, following the steps above.

Of course, we're not just going to use Excel to calculate mathematical sums. Most of the time, a formula contains a **function**. A function is basically a command – you command Excel to tell you some information about some cell(s). Functions make life simpler. Imagine if you wanted to add more than just two cells – maybe you want to find out the sum of a whole column of cells. You don't want to sit there all day clicking each cell and typing +! For this you would use the SUM function in Excel (which, as the name suggests, calculates the sum).

Let's take a closer look at the `SUM` function to see how this works. Here's how the `SUM` function looks if we want to add the value in cell A1 and the value in cell B5 together:

```
=SUM(A1;B5)
```

So you see the equals sign at the start, and then the function `SUM` and in brackets, the cells that you want to add, separated by a semi-colon `;`. All functions have this format: the function name, and then in brackets, the values or cells that you want the function to work on.

When the function requires a few separate values or cells, these are separated by semi-colons, as in this example above (`A1;B5`). If you're working with UK/US regional settings, it will be a comma, not a semi-colon (`A1,B5`). If you're not sure which you need to use, you can find out easily enough: start typing a formula and look at the helpful little box that pops up under the cell. This little box is always there to guide you in what form your function should take.

Some functions, like `SUM`, can work on a group of cells – also known as a *range* – so that you don't have to select each cell one by one when you're typing the formula. A range is defined by giving the starting cell, then a colon (which you could read in your head as 'up to' or in Dutch 'tot en met'), then the last cell. For example:

```
=SUM(A1:A20)
```

This means that you select all the cells from A1 to A20, and calculate the sum of all those values. Again, you don't actually have to type the range using the keyboard – you can highlight the area in the spreadsheet using the mouse (e.g. click on A1 and drag the mouse down to A20).

Question 13 - What is the difference between `=SUM(C1;C10)` and `=SUM(C1:C10)`?

Now you know about functions, you need to know where to find out what functions are available and what they do. There are two ways to do this. Often the best way is to just search online – Excel is so widely used that there is tons of information online, including the Excel section of Microsoft's own support website support.office.com and many other handy guides such as excel-easy.com/functions.html. The other way is to use the **formula builder**. This is Excel's built-in guide to using formulas and functions. You launch the formula builder by clicking on the *fx* by the formula bar (see Figure 7). It contains a list of functions and a window where you can enter the different parts necessary.

Exercise 18 - Use functions to get some basic information about the frequency of your words. Find out:

- **the maximum and minimum frequencies**
- **the range of frequencies**
- **the mean frequency**

Arrange the information in a separate table underneath your word set, and make it clear which value is which. Have you got any extreme outliers? Now is the time to replace any unacceptable stimuli by selecting something more appropriate from CELEX.

Is everything going smoothly? If you happen to find that the formulas are not calculating anything, make sure Excel understands that those cells are supposed to contain numbers and not text. You can do this by highlighting the cells, then select `Number` from the dropdown list in the centre of the ribbon.

There are many functions for all kinds of things in Excel, from the simple to the complex. If you are into programming, you can also look into the VBA (Visual Basic for Applications) programming language which you can use within Excel to write scripts for more advanced operations on your datasets (like merging sheets of data, extracting data into separate files, and even more exciting things than that).

We are working with an English interface, as explained in section 5.2.1. If you are working with Excel displayed in Dutch, you'll find that not only are the menus all displayed in Dutch, but all the formula names are in Dutch too – so, `=SUM(C1:C10)` becomes `=SOM(C1:C10)`. Having a different display language is frustrating (not all the translations are as obvious as "som"!); luckily, switching Excel's display language should not lead to issues with the data (because it's independent of the regional settings, see section 5.2.1).

5.2.4 Extracting information using formulas

In **Exercise 18** you used formulas to calculate information about frequencies. Frequency is a numerical variable, or **scale** variable, which means that we can do calculations on it – like finding the mean, as you did. However, sometimes we want to extract information that doesn't have to do with numbers.

When we have stimuli that are words, for example, we need to know properties of those words in order to make a properly controlled experiment. Take word length for example. Imagine you are going to use your set of words to test whether word frequency affects how quickly the words can be read aloud. It's not much good if, in your word set, word frequency correlates with word length – maybe the less frequent words take longer to read because they contain more letters!

String functions

Excel has various functions that can be used on cells that contain text. These are called 'string' functions, because 'string' is the term used by programmers to refer to pieces of text.

One of these is the `LEN` function, for finding the length of a string. This is how you would find out the length of the word in cell A1:

```
=LEN(A1)
```

Exercise 19 - Insert a new column next to the column of words. In this column, show the length of each word. Tip: check that there are no spaces on the end of any words – Excel counts spaces and punctuation as part of the string!

Did you type the formula again in each cell when you did **Exercise 19**? There's no need: you can copy and paste formulas. You can even paste a formula into many cells by highlighting the whole area before pasting.

Notice that when you paste the formula, the cell it refers to is different from the original formula. This is because cell references are *relative*. So each cell in the new column shows the length of the word next to it. If you select one of the cells you just pasted, and look in the formula bar, you'll see the cell reference inside the brackets is different for each one.

Using the online resources mentioned in section 5.2.3 you can explore what can be done with strings in Excel.

Logical functions

Another type of information you might need for controlling your stimuli is categorical information. Back in Chapter 4 we mentioned that some variables have to do with categories, such as whether someone is a native speaker or not. As for a word list, you might be interested in whether the word starts with a vowel or a consonant, for example. Then you would need to create another column in your dataset that gives that information for each word. To save yourself having to go through each word by eye, you can use **logical** functions. With logical functions you ask Excel to evaluate cells using some criteria that you decide.

For now we are going to concentrate on the `COUNTIF` function, which is going to be most useful to you for creating a controlled stimuli set. `COUNTIF` is a function that you can use to *count up* how many cells fulfil the criteria you provide. You do this by telling Excel the range of cells that you're interested in, and the criteria that each cell must fulfil in order to be counted.

```
=COUNTIF(range of cells; criteria)
```

The criteria can be of two types. Firstly, you could just ask Excel to check whether the cell matches some value – that value could be a number or a string, or another cell. In this case, you just have to type that number, string or cell reference in. When you enter a string in a formula, you must always put it between quotation marks so that Excel recognises that it's a string. For example, to count all cells in column B that contain the string "native speaker":

```
=COUNTIF(B1:B100;"native speaker")
```

If you already have the string or number that you want to match written in a cell somewhere, then you can just refer to that cell instead of writing the value into the formula. So, for example, if you have "native speaker" written in cell C24, the formula below will give you the same answer as the one above:

```
=COUNTIF(B1:B100;C24)
```

Secondly, the criteria for `COUNTIF` can be a logical condition – you ask it to calculate how many cells meet some logical criteria. For example, you could ask Excel to count all the cells containing a number bigger than 10, using the logical expression >*10*.

There is a special way of writing a logical expression into the formula (unfortunately it's not as simple as just writing `>10`!). First, you need to put the logical operator inside quotation marks: `">"`. Then you need to ask Excel to stick (or 'concatenate') this operator to what follows it, using an ampersand, i.e. the `&` symbol. Finally, you write the element to evaluate against – in this case, we want to evaluate whether cells are larger than 10; so we write `10` after the ampersand.

Here's how it will look:

```
=COUNTIF(B1:B100;">"&10)
```

Just like with the string example above, instead of a number, you can refer to the value contained in another cell, by using a cell reference. For example, the formula below counts all the cells in column B that are larger than the value in cell A24:

```
=COUNTIF(B1:B100;">"&A24)
```

This formula is now much more flexible and dynamic: if the value in cell A24 changes, the result of the `COUNTIF` formula will simply update by itself – no need to edit the formula!

Exercise 20 - Use COUNTIF to calculate:

- **how many of your nouns are concrete and how many are abstract**
- **how many of your words have a frequency bigger than the mean frequency**

To see all the different ways you can use `COUNTIF`, have a look at the Microsoft support page for this function. Or you could of course just google *COUNTIF*.

5.2.5 Organising experimental lists in Excel

So you have your list of noun stimuli, including information about frequency, word length, and concreteness. You also know how to find out information about both scale and categorical variables that might be relevant for controlling your stimuli set.

If you were using this stimuli set to test if word frequency affects word reading speed, you would probably just present all the words to all the participants in a random order, and calculate whether reaction time correlates with word frequency. In that case, word frequency is the independent variable. However, you could also use this stimuli set to test word reading speed for the same set of words under two different experimental conditions. A good example of this is a priming experiment: testing whether people read the words faster when they are primed. However, if you use a within-subject design, you need to counterbalance the conditions. So we will now show how to use the current stimuli set to create two counterbalanced lists for a priming experiment.

Question 14 - Before continuing, make sure you can answer the following questions:

- **What will be the independent variable in the priming experiment?**
- **Why is it necessary to counterbalance?**
- **What about word frequency – is it still relevant for the priming experiment?**

If you're not sure about any of these questions, review Chapter 4.

The primes in your experiment could be anything you like. Priming is an interesting phenomenon that can be used to test lots of different associations, or parts of the production process. For example, you could test whether reading a word aloud is quicker when primed by a word starting with the same letter, by a word of the same semantic field, by translations of the word, and so on. Or you could do cross-modality priming, using spoken word stimuli, or pictures or sounds. Remember: priming can only be calculated by drawing a comparison of the priming condition with a control condition. Exactly what the control condition should be depends on the experiment and which variables need to be controlled.

Exercise 21 - Insert two columns to the left of your words. In one column, enter a prime for each of the stimulus words. In the other column, enter the control 'prime' for each word. The type of priming is up to you!

Now let's create two counterbalancing lists. As you know by now, this means splitting the stimuli in half (if this isn't making any sense, you need to look back over section 4.3.2). However, if you're not careful, this can create new confounds! If, for example, one half of the the list has mostly short words and the other half has mostly long words, your conditions are not well balanced within each participant.

Luckily we can fall back on the easiest control technique of all – randomisation. It would be great if you could just 'sort randomly' in Excel, but alas, life is not so simple. However, Excel does have a function for *generating* random numbers, `RAND`. If you type `=RAND()` into a blank cell and press Enter, you'll get a random decimal number between 0 and 1. All you have to do is insert a new column at the edge of your dataset, enter random numbers all down the column using `=RAND()`, and sort the rest of the data according to that column. We assume that you understand how to sort a dataset in Excel – just remember to make sure that you select *all and only* the stimuli set (not any extra rows, or headers!) before sorting, and use the `Custom Sort` menu to choose which column to sort by.[14]

Exercise 22 - Randomise the order of your stimuli set by inserting a column with random numbers and sorting the dataset based on that column.

Warning: you may notice that the random numbers keep changing all the time! This is because every time you do something in the worksheet, Excel regenerates the random numbers. This isn't anything to worry about, because you only need these numbers for the purpose of creating random orders.

Exercise 23 - Create two counterbalanced lists:

- **First, divide your stimuli into two halves, called list one and list two. Use formulas to check that the two halves have similar word lengths and word frequencies. Use `COUNTIF` to check that your halves have roughly equal numbers of abstract and concrete nouns. If the two halves don't seem balanced, try randomising again.**
- **Once your lists are balanced, create a new worksheet (within the same Excel file). In this sheet, present two counterbalanced lists for your priming experiment (check out Figure 5 for a guide). Each participant will be randomly assigned one of these two lists. Correct counterbalancing will mean that:**
  - **Every participant should see each stimulus word once only**
  - **Every stimulus should be presented in both priming and control conditions during the experiment**

Check that these statements are true for your pair of lists!

[14] If you need to brush up on the basics of sorting, take a look at the Office support Quick start guide.

5.3 Getting the most out of Excel

Excel is an invaluable tool, especially when coupled with the possibility to write VBA code. In this chapter we've only scratched the surface. However, to get the most out of it, when working in Excel, always keep in mind how you can best organise your data in order to be able to make use of the full functionality of Excel.

For example, it's natural that you want to keep your dataset looking nice and visually easy to understand. If you're used to working in Word or other similar programs, you may be tempted to format your information in Excel in nice-looking tables, using colours to show different types of data and leaving white spaces within columns or between sections, etc. However, if you want to be able to re-sort your dataset according to the different columns, or you want to be able to apply a complicated formula to all of the items in your dataset, having gaps in the information will make this impossible! A rule of thumb is that every row represents a "case" or an "item of data". This means that you should be able to select a single row of your data set without losing any information that relates to that case.

Part II
Software and hardware in the labs

6 Introduction to the LUCL labs

LEARNING GOALS

- know which techniques are used in which lab
- understand the basic hardware setup in each lab

6.1 General

Together with technician Jos Pacilly we try to keep the lab setups as uniform as possible. Despite some minor differences it basically means that if you know how to make a recording in the Phonetics Lab you'll also be able to make a recording in e.g. the Baby Lab. If you'd like to make high quality recordings outside the Phonetics Lab you could use one of the high fidelity microphones (there are two in the EEG Lab and one in the Baby Lab). All booths can be used for behavioural experiments.

6.2 Phonetics Lab

The Phonetics Lab is the oldest lab and was built together with the Lipsius building. But even before that, phonetics was a distinct department at Leiden University. You've probably worked in the Phonetics Lab during Jos's course. At present it consists of a teaching/meeting room and two sound-proofed experiment booths, one of which has an EEG setup installed in it. Various programs (e.g. DMDX, E-Prime, Praat, SPSS, R and Adobe Audition) are available for data collection and analysis. These programs are installed on booth computers as well as on the four workstations near the windows. If you're planning to do fieldwork, you can apply for portable recording equipment which includes various sound recorders and a range of condenser and dynamic microphones.

Figure 8: Layout of the recording booth setup.

(1) Computer

(2) Screens

(3) Participant's screen

(4) Amplifier

(5) Mixing console

(6) SRBox

(7) Sound card

In figure 8 you see the layout of the recording booth. There is a computer (1) to present stimuli on the screens (2) and (3). The amplifier (4) can be used but it's best to use headphones to not disturb others. The mixing console (5), SRBOX (6) and Sound card (7) will be explained soon.

By default, the details of the setup are as explained in figure 9. **This is how you should find the settings, and how they should be when you leave.** These settings will allow you to test the voice input of the participant who is in the booth. You can talk to someone in the booth using the microphone that is on the desk (or just next to the desk in the Phonetics lab). In every lab this works as follows:

1. Make sure everything is turned on, including computer and speakers in the booth.

2. Instead of speakers, the participant can use headphones. Make sure the right connector is plugged in!

3. Put on the experimenter's headphones.

4. Adjust the incoming signal on the mixing console (5).

The connections going to and from the mixing console can be a bit confusing, so let's look at it now in more detail:

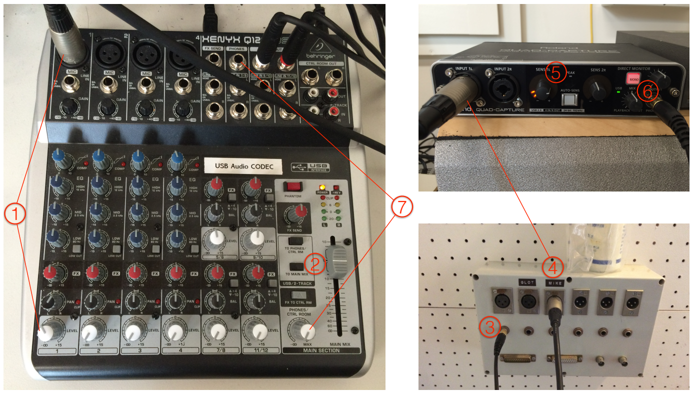


Figure 9: Layout of the recording booth setup.

(1) Experimenter's microphone

(2) Main mix

(3) Participant's speakers

(4) Participant's microphone

(5) Level for P's microphone

(6) Experimenter's headphones

(7) Additional output

The communication microphone is connected to the first input of the mixing console. Just below the connector the 'gain' should be almost fully open (nearing +60). Use knob (1) to adjust the volume. Don't forget to put it to zero as soon as the experiment starts, otherwise you'll interfere with the experiment! The main mix level (2) should be set as indicated and the little switch 'to main mix' should be pushed down.

In the booth the connector (3) to speakers or headphones should be plugged in as well as the microphone (4) which is connected to the sound card. With knob (5) you can adjust the level of the incoming signal. If 'Playback' in the audio control centre of the computer[15] is set to the external sound card, sound can be monitored by the experimenter by having the headphones (6) plugged in the connector of the sound card. In the other labs the headphones can be plugged in one of the speakers on the desk, so, switching between speakers and headphones is a matter of unplugging the headphones. But again, it's best to use headphones if there are other people working in the lab too. You may notice that in the EEG setups there is a cable connected to channel (7). It is connected to the speakers to transfer the output signal coming from the computer: in these setups the output is using 'USB Audio CODEC' as default sound out. Having the volume slightly open on the right speaker, the white knob under (7) can be used to adjust the sound output to speakers or headphones.

In principle you are **not** allowed to make changes to the setups. If you do wish to use a different setting, notify the Lab Manager(s) by sending an email with your

[15] Go to `Control panel > Sound > Playback` to verify this. You should also see in the control panel that (by default!) the sound card is 'checked' as recording device.

wishes to lucl-labs@hum.leidenuniv.nl. That way you can get assistance to create the setup you want. You should also use the same address to ask for help in case of any malfunction with the setup.

6.3 EEG Lab

As you might already know, electroencephalography (EEG) is a way to record electrical activity of the brain. Neurons communicate with each other by pumping ions from one to another. Ions are electrically charged. During the recording of EEG we can measure differences in ion exchange over time. This difference is measured as an electric potential that can be either positive or negative (this doesn't mean that positivity is 'good' and negativity is 'bad').

Typically, we measure event related potentials. This means that, during an experiment, we look at the EEG signal relative to specific time points in the experimental presentation. For example, when you mark the time point at which a stimulus appears on-screen, you can then analyse how the brain activity responds to that particular stimulus. Then, you can compare the overall difference in the brain response to stimuli of two different conditions.

We usually present participants about 20 or 30 trials per condition. Then we average the signal per condition per participant, and *then* we average the individual averages to get a "Grand Average" per condition. The difference between the Grand Averages per condition is what is called event-related potential (ERP).

Robust and replicable event-related potentials are often referred to as "components". Components have names such as N400, which is a negative peak between 350 and 500 ms after the onset of the stimulus.

ERPs are sensitive to many linguistically relevant dimensions, for example, changes in prosody, word frequency, and unexpected syntactic or semantic features of sentences. ERPs are the method of choice for investigating detailed timing of neural processes – but they are not so good for assessing the location of processes in the brain (which is better done with fMRI).

In the EEG Lab we have two setups to record EEG data. There is an additional booth in the Phonetics Lab. In figure 10 you see the layout of the setup 'EEG 1'.

① Stimulus presentation

② Data acquisition

③ Participant's screen

④ Screen switch

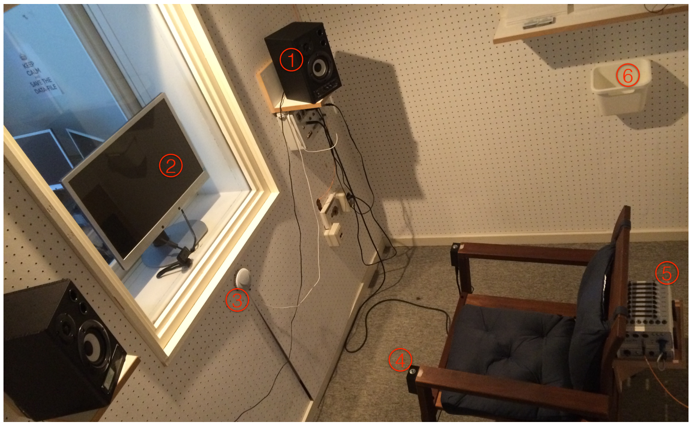
⑤ Digitiser

⑥ SR box

Figure 10: Lay out of an EEG setup.

As you can see, a setup has two computers and four screens. One computer ① is used to present your experiment script, the other ② is used to record the EEG data. On the participant's screen ③ you can change from data visualisation to stimulus presentation by switching from **B** to **A** on switch ④. The voltage signals coming from the booth are converted to a digital signal by the digitiser ⑤. The SR box ⑥ connects the response buttons that are installed on the participant's chair in the booth to the stimulus presentation computer ①.

Figure 11 shows the inside of the EEG booth. Inside the booth a participant can hear you talking (if necessary), or listen to auditory stimuli through the speakers ①. There's a screen ② and a microphone ③ to collect voice responses. Two response buttons ④ are installed on the chair – one on each arm. The electrodes need to be plugged in the Analog to Digital box ⑤ which sits on top of a battery. There is also a shelf so you can keep small items such as alcohol swabs and syringes to hand, and a bin to collect disposals ⑥.



1. Speaker
2. Screen
3. Microphone
4. Response button
5. AD box and battery
6. Bin

Figure 11: Lay out of an EEG booth.

All (spare) materials are kept in the main EEG lab (Lipsius 1.04). Also, there is a sink to wash materials after use. To make the connection between electrodes and scalp we use conductive gel, which gets stuck in the participant's hair (more about this in Chapter 12) – so we offer them the use of the sink and a towel (and hairdryer if they like) to wash or rub it off. So if you wonder why there's a washing machine in the lab, it's for the towels!

On the long desk by the windows you'll find two analysis computers (called PC1 and PC2) that have programs such as E-prime, Brain Vision Analyzer, Experiment Builder and Data Viewer (see 6.4) installed on them.

You can, as we've mentioned before, use the EEG booths for behavioural experiments. However, if you want to use the response buttons on the chairs as part of a behavioural task, both computers on the EEG experimenter's desk must be switched on (i.e. also the secondary computer that is normally for EEG recordings). This is because the button signal is routed through this computer.

6.4 Eye-tracking Lab

The point of eye-tracking is pretty clear when you think about reading research – by measuring how people move their eyes around the text, we can understand which parts demand more processing and which circumstances lead people to look back at previous words. However, it may be less clear why we would use it to investigate spoken language production or comprehension. The key thing to know here is that when we listen to or produce descriptions of objects in the visual environment, we have the tendency to move our eyes to the relevant objects while processing the linguistic information. When we are less sure, our eyes move around possible competitors before settling on the chosen object. So, if you wanted to test whether Dutch people can hear final voicing distinctions in English such as *cap* versus *cab*, you could show them a visual display including a picture of a baseball cap and a taxi cab and measure where their eyes move as they hear the auditory stimulus. What's more, when we're producing complex descriptions, such as describing scenes, the order in which we look at ('fixate') things within the scene can tell us about the order in which we go about encoding the visual information into a linguistic form.

Eye-tracking works with an illuminator and a camera. The illuminator shines infra-red light at the participant, and the camera picks up the reflection of this light (which is invisible). The different amounts of reflection of different parts of the eye make it possible to calculate the position of the pupil, and therefore the direction of the gaze (more about this in Chapter 13).

The eye-tracking lab has a sound-attenuated booth for the participant and a desk for the experimenter. You can see the layout in Figure 12.

① Host PC

② Experiment PC

③ Participant's keyboard, mouse and headrest

④ Eye-tracking camera

⑤ Participant's screen

⑥ Booth power switch

Figure 12: Layout of the Eye-tracking lab.

The experimenter's desk has two computers: the host PC ① and the experiment PC ②. The host PC is the one that runs the Eyelink eye-tracking system, while the experiment PC is the one that runs the experimental software, Experiment Builder, and the analysis software Data Viewer. Inside the booth, there is a table for the participant to sit at, with a keyboard and a mouse for the participant and a headrest, which you can use to help the participant keep their head still without getting too uncomfortable ③. Then on the table in front of the participant you'll see the eye-tracking camera itself ④. The camera consists of a lens, and an infra-red illuminator panel. On the table there are marks indicating an optimal position for the camera (in fact, you should try to avoid moving the camera around if possible). The eye-tracking camera is connected to the host PC. There's also a computer screen in the booth (5). This is where the experiment is displayed to the participant - basically this screen shows the same as the experiment PC screen. Finally, on the floor is an extension cable with a switch (6) - important to point out, because this is the switch you use to turn all the equipment in the booth on and off.

The software we use to design and run eye-tracking experiments is called Experiment Builder, and the software for viewing and processing eye-tracking data is called Data Viewer. Apart from the computer in the eye-tracking lab, Experiment Builder and Data Viewer are also installed on PC1 and PC2 (by the windows) in the EEG lab, so that it's also possible to work on your design or your data without occupying the eye-tracking lab. However, to use the software, you need the dongle (see 11 for more about dongles). Since there is only one eye-tracking dongle, it's important to bear in mind that the people working in the eye-tracking lab have priority. Therefore, **never** remove this dongle from the labs, and always return it to the eye-tracking lab after use.

It's important to check that all hardware devices have the correct settings, so that everything works properly with the Eyelink system. To check the settings of the Eyelink camera, you click the `Devices` tab in Experiment Builder and then select `EYELINK` at the top of the tree structure. The properties window will now show the properties for the Eyelink system. For our setup, the following settings are required:

Tracker Version:	Eyelink 1000
Camera Mount:	Desktop
Desktop Version:	Illuminator on Right
Mount Usage:	Monocular Remote

6.5 Baby Lab

The Baby Lab can be used to test infants, toddlers, and children in a variety of behavioural paradigms. It is primarily designed for testing infants in the Headturn Preference Paradigm and the Visual Fixation paradigm, both of which measure infants' level of interest in various stimuli. Both of these paradigms test infants' learning and discrimination abilities – for example, whether they can tell the difference between different speech sounds.

A typical paradigm begins by exposing the infant to a short ‘familiarisation’ phase to the audio/visual stimuli, after which they are tested on their ability to tell the difference between stimuli that are similar to what they were familiarised with and stimuli that are different in some way. With the Headturn Preference paradigm, we measure the way they turn their head to attend stimuli on different sides of the booth; with the Visual Fixation paradigm we test simply when and how long they look at the screen. In the end, we can calculate the total looking times for the different types of stimuli, and test whether these looking times are different between conditions. If the infants have learned something and become bored with it (i.e. if they have become ‘habituated’), they will prefer to attend longer to stimuli that are novel. Typically we only study one contrast per infant.

You should be aware that there is a great deal of individual difference between babies, and within one experimental group, some of the babies might prefer novel stimuli while others may prefer familiar stimuli. The direction of preference infants show has been shown to be affected by age (younger infants prefer familiar stimuli) and by complexity of the stimuli (complex stimuli yields preference for familiar stimuli); for a disussion on habituation of looking time see Oakes (2010).

As you’ll learn in Chapter 14, the experimenter plays an active facilitating role during Headturn and Visual Fixation experiments, by monitoring the baby’s behaviour from outside the booth and using this to guide the presentation and duration of the stimuli. The scripts for presenting stimuli in the Baby Lab may be written in E-prime, or more recently, a program called LOOK (which you will learn more about in Chapter 14). The baby’s response behaviour is recorded using a video camera, and the video is then manually coded in ELAN (Brugman and Russel, 2004). ELAN is a video annotation software developed by the Max Planck Institute for Psycholinguistics in Nijmegen. Manual coding means that after the experiment, the experimenter reviews the recording and labels the video to categorise what the infant is doing at each moment (for example, looking left, looking right, looking away, etc). The categories for labelling depend on what is relevant for each experiment.

(1) Welcome area

(2) Experimenter's area

(3) Recording booth

(4) Large screen & video camera

(5) Side panels (screens, lights, speakers)

(6) Participants' seat

Figure 13: Layout of the Baby Lab

The Baby Lab is a bit brighter than the other labs – it's intended to be a friendly place for babies (and their parents). The lab has a welcome area with seating for participants (and plenty of toys!) (1). Around the corner are the experimenter's desks and computers (2). Finally there is a large sound-attenuated booth (3) where the experiments are recorded. Inside the booth there is a large TV screen with a video camera underneath (4) and two side panels (5). The side panels each comprise a speaker, a small screen and an LED light. Finally there's a chair in the middle (6), where the parent sits holding the baby on his/her lap facing the screen. To avoid influencing the baby's behaviour, the parent should use the iPod and headphones provided to listen to music while the experiment is running.

7 Qualtrics survey software

LEARNING GOALS

- use Qualtrics to design and launch a survey
- create and evaluate basic norming questionnaires

Qualtrics is a web-based survey software, where you can make questionnaires, distribute them using a link, and export the results in various formats. Unlike free survey tools you may be familiar with, Qualtrics offers lots of options for survey design and content. The university has a licence for Qualtrics so that students and researchers can all make use of it.

In experimental linguistics, we often use Qualtrics for conducting **norming surveys** (look back to section 4.3.3). In this chapter you'll learn how to use Qualtrics for this purpose, by working through several exercises to set up a norming survey for pictures. You could, however, use the same format to test words and sentences, as audio files or as written text, or even build on this template to develop a basic online experiment.

Another thing Qualtrics is useful for is creating **background questionnaires**. A background questionnaire is a list of questions which you ask participants to fill in before an experiment, in order to get information about important control variables like age, sex and linguistic background. **Remember: norming surveys and background questionnaires are two different things.** Norming surveys collect information about the stimuli you want to use in your experiment, background questionnaires collect information about the participants.

In the rest of this chapter we will walk through the steps necessary for creating a basic norming survey. Your assignment for this chapter is to create your own survey.

7.1 Making a norming survey in Qualtrics

Before you begin the exercises, decide what pictures you are going to use, and which variable you are going to norm. For the pictures, we advise you to download an existing set of psycholinguistic image stimuli called POPORO (Kovalenko et al., 2012). For ideas about which norming variable you want to gather data on, you can look back at your response to **Exercise 9**.

To log in, go to the Leiden University Qualtrics login page leidenuniv.eu.qualtrics.com. You should be able to use your university username and password to log in. Once you're in, click on the button `Create Project` on the top right. This gives you a blank template to work with. You'll need to come up with a name for your survey and you can also specify a folder in which the survey is to be saved.

7.1.1 Creating questions and blocks

The basic units of the survey are questions, that can be grouped in blocks. First of all you need to create a block that includes a welcome text (the first thing your survey participants will see when they begin the survey) and some questions about the participant's background - for example, questions to check whether they are native speakers and any other necessary information. There are a number of question types to choose from.

To start building your first question block, click on `Create a New Question` and Qualtrics will automatically generate a question in multiple choice format for you. Clicking on the downward arrow next to `Create a New Question` will let you choose a different question type. You can also change the type of an existing question using the panel that appears on the right.

Exercise 24 - Create the first survey block.

When you've finished creating the first block of questions, you can add a new block by clicking on the `Add Block` button below your first block. This will generate a new separate block where you are going to add your norming questions. Before you do that, you'll need to upload the pictures you want to norm.

7.1.2 Uploading and using pictures in your survey

To upload pictures, click on `Library` at the top right of the screen and choose the `Graphics Library` tab (top left). This is where all your graphics are stored so that you can access them when creating questions in any of your surveys. Create a new folder for your norming pictures using the folders panel on the left. When you are in the folder where you want to upload the pictures, click the `Upload Graphic` button.

Exercise 25 - Upload (minimally) 10 stimuli pictures from the POPORO set.

To return to your survey, click `Projects` at the top right of the screen.

It's important to start the norming section of your survey in the new block you've created. The reason for this is that it is then possible to randomise questions within this block, without it randomising your first block as well. Tip: first play around with question types and appearances and decide how you want your basic norming question to look, then copy this basic question to generate the rest of the norming questions in your survey (you'll just have to update the picture in each one).

To insert your picture into a question, you'll need to use the `Rich Content Editor` function within one of the question text fields.

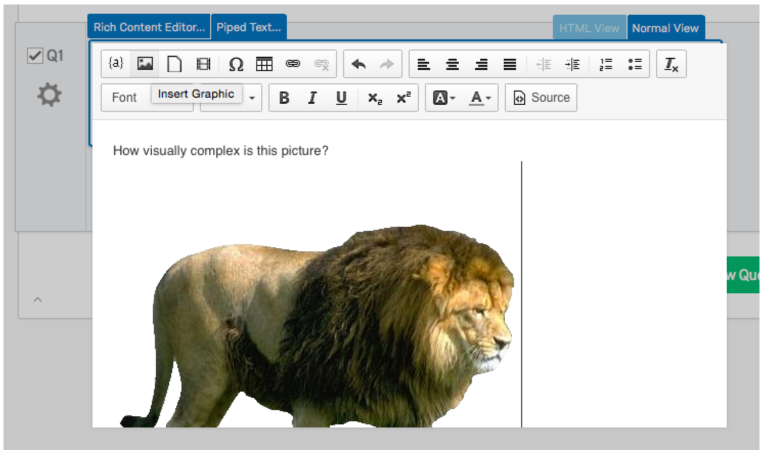


Figure 14: Adding content into Qualtrics question fields.

You should make sure to rename the questions in your survey (click where it says Q8 or whatever the default question number is, on the top left of the question) to reflect the picture shown in each question, or at least code it in some meaningful way. Otherwise, when you download the data, you won't know which response refers to which picture.

Exercise 26 - Create your norming questions.

Don't forget to randomise the questions in the norming block using the `Block Options` menu at the top of the block.

Previewing questions is possible by clicking the `Preview Question` button on the bottom right, and you can preview the whole survey using the `Preview Survey` button on the bar along the top. You'll see that the survey has a default theme. In the next section you'll find out how to alter this.

7.1.3 Finalising and distributing your survey

To finish off your survey, you should create a final block which includes a closing text thanking the participant, and usually a text box where they can write any comments or let you know any issues they had while completing the survey.

You also need to set certain options regarding the appearance of the survey. There are two menus you should always go through when finalising your survey: `Look and Feel` (where you determine the appearance) and `Survey Options`. To find out more about any of the options you see in these menus, you should consult the Qualtrics Support website (see section 7.2).

Exercise 27 - Finish off the survey and preview it.

Once your survey is ready (and you have previewed it to make sure – this is not an optional step!), you can launch it by selecting the `Distributions` tab. If you choose to distribute your survey by email and click `Get a single reusable link`, you'll get your anonymous survey link (if you don't get this, you may need to go back to `Survey Options` and check your settings). Once you start collecting data, you really should avoid making changes to your survey, otherwise the data may not be valid.

Exercise 28 - Launch the survey and send the link to a friend to pilot it.

Once you have some data you can go to the `Data & Analysis` tab to take a look at it. Here you can look at individual responses, aggregated reports, or you can download the data in a range of file formats. It can be useful to give the survey a test run yourself and look at the data that it produces, in case you need to make any last minute changes to the survey before you collect real responses.

7.2 More resources for using Qualtrics

There are more resources available to help you get the most out of Qualtrics, including the Qualtrics Support website. There are many more aspects of Qualtrics that we have not covered in this tutorial - for example, you can have more control over things like randomisation and counterbalancing by using the `Survey Flow` functionality. You can also skip over questions that aren't relevant: for example, if a participant answers "no" to "Are you a student?" you can ask Qualtrics to skip over other questions, such as "What is your study programme?" (if you feel like exploring this, look for the `Add Skip Logic` and `Add Display Logic` buttons at the bottom of the right-hand panel).

8 E-prime basics

LEARNING GOALS

- script a basic priming experiment in E-Prime
- handle experiment output created by E-Prime scripts

E-prime is stimulus presentation software that is widely used. In our labs, especially in the EEG Lab, you'll need it to program your experiments. The tutorial in this chapter is largely following E-prime's own *Starting Guide* (Zuccolotto et al., 2012), though we'll be gearing towards using the scripts in our labs. We have also added some handy tricks. Should you find the tutorial in this reader going too fast you could try E-prime's original guide which can be found under the help button in the the top menu of every E-Prime application. Steps in the tutorial that match steps in the *Starting Guide* are marked by a number between square brackets (e.g. [15]). The number refers to the page number in E-Prime's guide.

Whichever stimulus presentation program you use, they all provide you tools to show on-screen instructions, to present trials – usually running in loops – and to properly end the task with a thank you screen. As a rule of thumb, before you start programming your script, you need to have ready (1) a finalised stimuli list in Excel and materials (for this tutorial, you can find the materials on https://doi.org/10.5281/zenodo.1339334) and (2) a schema of the way you want to script your experiment. While it's tempting to just figure out a program by fiddling around with the Graphic User Interface, it's much better to approach it in a structured way.

The example experiment described in the tutorial can be used to examine automatic and controlled processes in gender stereotype priming.[16] In the experiment participants will be presented with a prime word for a second. The prime word will be either stereotypically masculine or feminine (e.g. sports or flowers) and will have either a positive or negative connotation (e.g. sports or cigars). The prime word will be replaced by a fixation (+) for a second, after which a target word will be presented. The target word will be either a male or female name (e.g. Henry or Mary). The task for the participants will be to respond to the target word – as quickly as possible. If the target word is a male name the participant should press "1", if the target word

[16] And it has been used by researchers in 1996: Blair, I.V. & Banjani, M. (1996). Automatic and controlled processes in stereotype priming. *Journal of Personality and Social Psychology*, 70, 1142-1163.

is a female name the participant should press "2". As dependent variables reaction time and accuracy will be measured. The hypothesis is that automatic processing of primes yield faster reaction times for stereotypic prime-target pairs as compared to counterstereotypic prime-target pairs. Accuracy can be used to filter out false responses.

As we build up the script you'll learn how to create pauses and how to present more than one block while counterbalancing the order of blocks. We're using the schema in figure 15. While the experiment in this tutorial is a social psychology experiment, you might already know that there is a large literature on priming effects in psycholinguistics. The details differ of course and not all experiments are priming experiments, but an experiment flow like this can also be found in psycholinguistics.

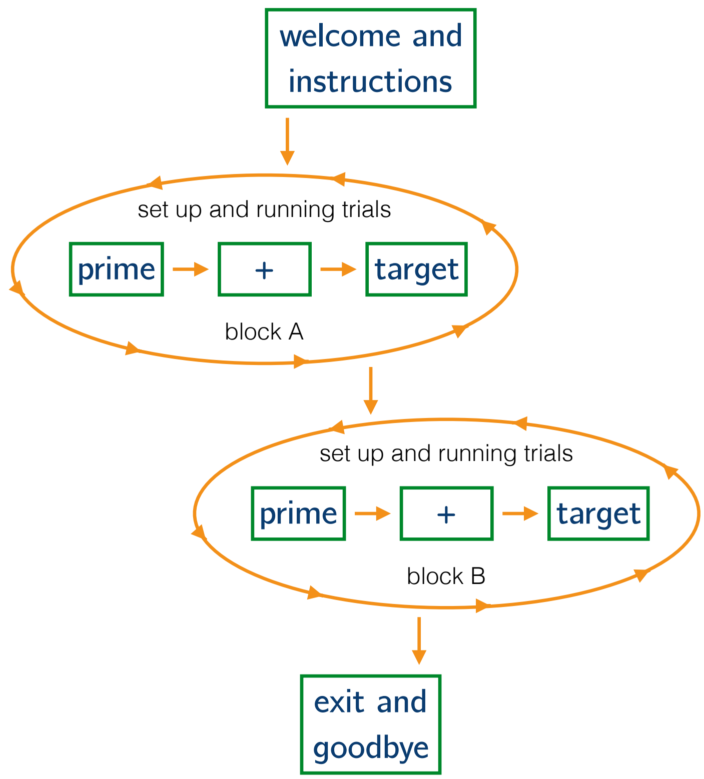


Figure 15: Flow of the experiment. Time line is in orange. Green boxes represent displays to be created.

8.1 Navigating through E-prime programs

If you go to `Windows Start > All Programs > E-Prime 2.0` you'll find several applications. In this chapter we are mainly concerned with E-Studio which will enable you to design a script. We'll go through the basics in section 8.2. During your real experiment you'll use E-Run to run your final version. This makes sure that every participant will get exactly the same script. After you've collected data from all your participants you can merge the output files using E-merge, which will be explained in section 8.4. In the last section (8.3) you'll learn how to rescue data from interrupted scripts. The following sections are written in the form of a tutorial, so be seated at a workstation with E-Prime installed.

8.2 E-studio: setting up a basic script

Launch the application E-Studio and select a Blank (Professional) script.

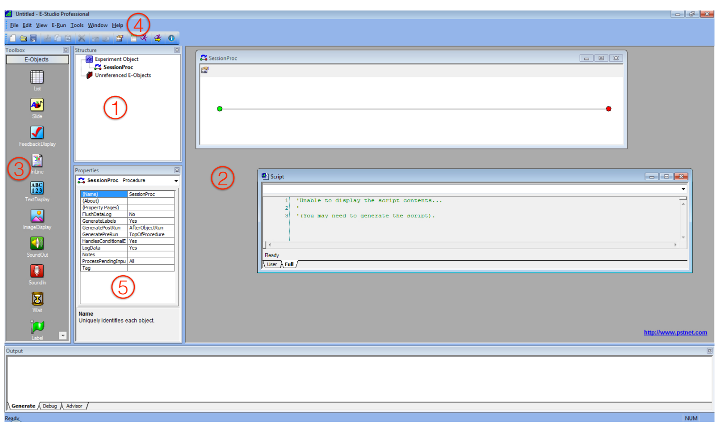


Figure 16: Basic layout of E-Prime's GUI.

1. Experiment structure (flow)
2. Workspace showing the selected object/window
3. Objects to add to the experiment
4. Top menu
5. Properties window

In the window you have open you'll see at the far left the Toolbox which contains E-Objects that you can use to create an experiment (3). The objects can be added to the experiment by dragging and dropping them onto the Structure window (1). This window shows the layout of your experiment and works similarly to a Windows Explorer folder. When you highlight an object in the Structure window, information about the object will be displayed in the Properties window (5). Now, save the file to a folder on your P-drive giving it the name `PictureTutorial-<your last name>.es2` as we're about to drop the first E-Object into your first E-Prime script.

8.2.1 Presenting welcome, instruction and goodbye text

Our script should start with a welcome text, so that's what we'll do first. Throughout scripting **save your changes regularly** by using the shortcut key combination `Ctrl+S`. It's good practice to generate your script `Ctrl+F7` after completing individual parts of an experiment in order to catch potential errors. If errors are found, the Output window will open at the bottom of the screen to display the errors. So, let's put some things together so that you can actually use these shortcuts.

1. To begin, we'll determine the resolution of the screen. Open the experiment's properties by double clicking the top node of the experiment structure and go to `Devices`. Here you'll see the devices that are attached to your script. Double click Display and set Match Desktop Resolution at Runtime to `Yes`. Click OK (both windows).

2. Double click the `SessionProc` object in the Structure window. A window version of the object will appear in the Workspace at the right. `SessionProc` is a Procedure object automatically created in each new Experiment Specification (.es2) file and is representing the time line of your experiment.[15]

3. Click and drag the `TextDisplay` object from the Toolbox, and drop it on the beginning of the `SessionProc` timeline. As soon as you add things in the Workspace you'll see adjustments reflected in the Structure window. You can close an item in the Workspace simply by clicking "x" at the top right.[15]

4. Highlight the newly placed `TextDisplay` object (by clicking on it) to view its properties in the Properties window. You can highlight the object in either the Structure window or on the `SessionProc`. Every object has a set of associated properties which are displayed in the Properties window when the object is selected.[15]

5. Click in the box to the right of `(Name)` and change the value from "TextDisplay" to "Welcome". Press the Enter key to accept the new name. Change the font name into "Verdana".[16]

6. Go to the Workspace and click on the Property Pages button (the little hand holding a piece of paper) in the upper left corner of the Welcome window. This is another way for changing object properties – especially helpful in the beginning phase. On the `General` tab of the Properties menu, locate the `Text` field

and type: "Welcome to the experiment. <Enter> Use the spacebar to go to the instructions." You use <Enter> to put the next sentence on a new line. <Enter><Enter> between two lines will result in a white line.[18]

7. Click on the `Duration/Input` tab. Specify the duration of the display to be `(infinite)` (the numbers under `(infinite)` refer to milliseconds). This setting results in the object running until some condition is met, for example, a keypress: we'll tell the participant to press the spacebar to terminate the presentation of this object.[18]

8. On the `Duration/Input` tab, click the `Add` button under the Device(s) window. You'll see which devices may be added. Click the Keyboard device to select it, and click `OK`. The keyboard will now be enabled as an input device, and the Response Options fields will become available.[19]

9. Select the `Duration/Input` tab and type `{SPACE}` in the Allowable field. Only those keys specified in the Allowable field will be considered valid responses. All other keys will be ignored. The curly brackets and all capital letters tell E-Prime that the word "SPACE" refers to a special character. For plain numbers or letters you don't need any bracketing. If you want to add a mouse, left mouse click is coded as `1`, the right as `2`. If it doesn't matter which input has to be clicked you could choose for `{ANY}` as Allowable setting. EndAction field should be set to `Terminate`: with the button press the display disappears.[20]

10. On the `Frame` tab edit the Width and Height to `100%`. Click `OK` to exit the Property Pages.[20]

11. Now you're ready to have a look at how the welcome display looks in the actual test. It's good use to check whether a script still runs after making changes. If you make a lot of changes and then find that something goes wrong, it's very difficult to find out *what* is going wrong. Save the script and run it by clicking the purple running guy in the top menu bar. Use `0` as `Subject Number` since it is not yet necessary to save any output. If you want to abort a session press `Ctrl+Alt+Shift`.

Exercise 29 - Add a display that provides an instruction for the task (and lets the participant know what to press in order to continue). Also add an additional display at the end in which you thank the participant. This display should terminate by itself after 5 seconds, so the participant

should not be able to terminate it themselves. Note: don't give an object the name "End" as it will interfere with E-prime's settings under the hood.

8.2.2 Presenting text as stimuli

As suggested by E-prime's Starting Guide you should think of your experiment in layers. The top layer works like a receptionist controlling all of the greetings and administrative events in the experiment we've just added. The SessionProc object is doing this. The next layer can be understood as a manager, because it will control all of the experimental tasks that are presented to the participant. In order to create this layer, we'll add a `List` object to the SessionProc. A List object can be used to present different stimuli (text, pictures, sound and movies) at the same point in the experiment. Using a List object we'll employ a manager to run a block of trials. Within this block a we'll add a "worker layer" which will actually run the trials. The worker layer will be represented by a secondary List object. We're going to build a block of trials in which we're presenting text stimuli in a random order. Each stimulus will be presented once. Have ready the Excel file *E-primeBasics.xlsx* containing the stimuli for the priming task. They're split up in two lists each to be presented in a separate block. We'll also take care of counterbalancing the order of blocks between the participants.

1. Locate the `List` object in the Toolbox. Drag it to the `SessionProc` and drop it between the instruction and goodbye displays. Give it the name "BlockList".[23]

2. Double click the `BlockList` object to open it in the Workspace.[23]

3. By default "ID" (stimulus number), "Weight" (the amount of times you want to present the level), "Nested" (forget this for now, coming soon!) and "Procedure" (the name of the trial procedure) are set as `Attributes`. Think of `Attributes` as the list of factors, conditions, independent variables – anything important to your stimuli and design. For the BlockList we don't need more attributes. We'll only use it to create a time line for the stimuli lists. Type "BlockA" under `Procedure`. This doesn't need to be set as default value when asked.[23]

4. Add a new `List` object to the newly created `BlockA`. To rename it into "TrialListA" you could alternatively press `F2`. After you've renamed the list, you'll be able to view its properties in the Properties window.

5. Double click the `TrialListA` object to open it in the Workspace.

6. The trial list should contain more than three attributes. The Excel file contains all this information. Except for the attribute names (so, the top row in the Excel file and `TrialListA`) everything can be copy-pasted right away. Start with filling in the attribute names. Click the Add Attribute tool button: one horizontal arrow will add one, the button with two horizontal arrows will give you the opportunity to add more than one. Add the Attributes that are listed in the Excel file.

7. In the first cell of the Procedure column of the `TrialListA`, type "TrialProc" and press Enter. A dialog will indicate that the `TrialProc` procedure does not exist. Click Yes to create the Procedure object `TrialProc`. Click "No" when you are prompted to make this procedure the default value for newly created levels. This `TrialProc` represents the timeline of trial presentation.

8. Select everything that needs to be copied in the Excel file (except for the attribute names!), copy it (use `Ctrl+C`) and click the topmost empty cell of attribute `[name]` in your .ebs2 file. Paste the stuff from the clipboard by using `Ctrl+V`. You'll see that E-Prime automatically adds the extra necessary rows!

9. Now, it's time to specify the amount and order of stimulus presentation. As with all E-Prime objects you'll be able to set such properties in `Properties Pages`. Click its button (next to the `Add Attributes` buttons). We want a randomised presentation order. Click on the `Selection` tab and set the `Order` of selection to `Random` from the dropdown box. Select the `Reset/Exit` Tab and verify that the option to reset after "All samples `[number]`" is selected. This means that the stimuli of TrialListA will be reshuffled after all have been shown. Verify the option to exit after "After 1 cycles `[number]` samples" is selected. Click `OK`. So, now the List will run for one cycle of `[number]` samples and then exit: exactly what we had in mind.[30]

10. Now we're going to define the stimuli presentation display. We've prepared a timeline `TrialProc`. This timeline will be the presentation cycle of one trial which is to be repeated for every stimulus: yes, it's a loop! A stimulus should

be preceded by a fixation cross and a prime. So, we'll need three separate displays, one for the prime, one for the fixation cross, and one for the target (stimulus). Double click the `TrialProc` object in the Structure window and drag a `TextDisplay` object from the Toolbox to the `TrialProc`. Rename this object "Prime". Drag a second `TextDisplay` object from the Toolbox and to the `TrialProc`. Rename this object "Fixation". The third object should be called "Target". Be sure that newly added displays use the same font(size) as the other displays already used in the experiment.[31]

11. Double click the `Fixation` object to open it in the Workspace. Click the `Properties` button to open the `Fixation` objects `Properties Pages`. On the `General` tab of the `Property Pages`, specify the Text to be "+" and the AlignVertical and AlignHorizontal properties to be set to `center` (default). On the `Duration/Input` tab, set the Duration to be `1000` (i.e. 1 second). Click `OK` to accept the settings.[33]

12. Double click the `Target` object to view it in the Workspace. Click the `Properties` button to open the `Target` objects `Properties Pages`. On the `General` tab of the `Property Pages`, specify the Text to be "[Stimulus]". We use square brackets to tell E-Prime that it is a variable. In this case it refers to the Attribute "Stimulus" in `TrialListA`. Just like the fixation cross, set the AlignVertical and AlignHorizontal properties to `center`. On the `Duration/Input` tab, set the Duration to be `(infinite)`. It will be terminated by the participant. So, add a keyboard and specify the keys to be used. Leave the window open, we're nearly done.[34]

13. To record the accuracy of the participant, E-Prime can compare the response given by a participant to the value it should be. In `TrialListA` this is listed under CorrectAnswer. In the field next to Correct, add [CorrectAnswer] – a variable that refers to this Attribute. Check that the `Data Logging` is set to `Standard`. This means that you'll be able to see the responses in the output file. Accuracy of the target object can be found as `Target.ACC`. In such labels, the expression before the full stop – "target", here – refers to the Object. Likewise, reaction time can be found as `Target.RT`. Click `OK` to accept the settings.[35]

Exercise 30 - The properties of the `Prime` object still need to be done. We want to present a prime for 150ms. Each target has a designated prime

as you can see in the Excel file, so, in the object you should refer to an attribute.

It's time again to test the script by running it yourself. At this point you should also assign a subject number, so that the output gets saved.

Exercise 31 - In the Excel file we've prepared a second list to be presented. It should be run in its designated procedure: `BlockB`. Add a second trial list "TrialListB" containing the information of the second list. Put it in `BlockB`. Note that you don't need to create new presentation Objects, since you can reuse `TrialProc` as procedure: all connected E-Objects will be shown with it!

So, it's pretty easy to clone procedures. Cloning also works by dragging an E-Object to another place in the structure. But beware: if you change something in the clone the original will also be changed! If you don't want this, then make sure to right click the object, copy it and paste it. Click `Yes` if you also want the child objects (nodes under the object that you're copying – if there are any) to be copied rather than being cloned.

8.3 E-DataAid: checking the output

The last test run that you performed resulted in three output files: a .txt, an .xml and an .edat file. The .edat can be opened with E-DataAid. During scripting it is good to check the output to make sure that you've saved what you need and to check the timing of presented E-Objects.

1. Click on the Windows Start menu: `All Programs > E-Prime 2.0 > E-DataAid`. The application launches without opening a specific data file. A file must be opened within the application. Click the Open tool button, or press Ctrl+O and choose the file you'd like to open.[47]

2. As you can see, a lot of data has been saved. Don't worry, you don't need to check every single number! Also, you can filter the data so you can have a quick view of the data you're interested in. On the Toolbar, click the `Filter` tool button to display the Filter dialog.[49]

3. We're interested in the accuracy data that are collected at the point the target was presented. Select `Target.ACC` in the Column name field. Accuracy has been recorded since we've logged the data collection for the `Target` display. If you cannot find it, you forgot to save this data. Sh*t happens. Just make sure you learn your lesson: always check the output before running it for real![49]

4. Click the Checklist button and click the checkbox next to the numbers that you still want to be able to see. Click `OK` to apply the filter and close the Filter dialog.[49]

5. You can see that some rows are not shown anymore. Additionally, you could hide some of the columns with the Arrange Columns button in the toolbar. Click it and select subject, accuracy (.ACC) and reaction time (.RT).

6. Inspect the result of your settings and close the program without saving the changes.

8.3.1 Timing issues

As you can imagine, the timing of displays that you present is of great importance. You might think that if you tell E-Prime (or any other program for that matter): "show this display for 50 ms" that it is actually shown for 50 ms. Well, not always. It depends on two things:

- ...the time the computer needs to prepare the object to be presented. Text is quite easy, but audio, pictures and movies may need some preparatory time. For a great part it is taken care of by E-Prime since the Pre-release in the Duration/Input tab of the Properties window is set to `(same as duration)`. Pre-release refers to the 'pre-release' of the material to be displayed: in other words, the computer prepares the presentation of the stimulus some time ahead. For audio and movies you normally would use a buffering time (we'll talk about that later).

- ...the display refresh rate. This basically refers to how frequently the display gets updated by the video hardware. The refresh rate is measured in Hertz (Hz). If you run your experiment on a 60 Hz display (as you will in the lab) this means that the screen is being refreshed 60 times per second. So, how long is each refresh cycle? 1000 ms/60 cycles = 16.667 ms
  This means that you should use durations that are multiples of 16.667 ms,

especially if you are relying on very precise or short presentation times (such as with masked priming). Certainly, durations cannot be shorter than 16.667 ms because then nothing will be presented at all! Note also that if you mess up the timing of an initial object in a loop, it may also have consequences for the onset of objects presented afterwards.

Exercise 32 - The Prime object has a duration of 150 ms. Run it (make sure you collect data!). Reset the duration to 140 and run it. Do a third run with 130 and check the differences in the output of three different runs. Which columns in the .edat file where most informative? What is the most appropriate duration in this case?

8.3.2 Presenting counterbalanced lists

From the last exercise you should have ended up with a script that presents both lists to the participant. That's not exactly what we want for now. Rather, participant 1 should get `TrialListA`, participant 2 `TrialListB`, participant 3 `TrialListA`, etc. A way of achieving this would be to create two separate scripts. But imagine you'd like to change something. Then you would have to do the adjustment twice. To prevent mistakes it's better to keep the experiment in one script. And it's possible and easy to do! We need to adjust the `Selection` settings in the `BlockList`.

1. The trick to counterbalance on the basis of subject number is this: go to the `BlockList` properties menu (use the little hand holding a piece of paper). Click `Selection` and choose as order `Counterbalance` by `Subject`. Click `OK`.

2. Check whether it works by checking the output files of two participants (by running it yourself of course).

Exercise 33 - So now we've taken care of the presentation of one stimulus list in a counterbalanced way. We actually want to counterbalance the order of lists while presenting them all. So participant sees first list A and then list B, participant 2 sees B then A, participant 3 sees A then B, etc. One way to achieve this is by selecting `Permutation` under the `Selection` button. Change the current counterbalance method and make sure to adjust the `Reset/Exit` procedure: all samples should be run.

8.3.3 Tweaking: using InLine scripts

Almost everything can be programmed by using the GUI. Under the hood, a script will be generated. The `Script` window will be shown as soon as you generate a script or by pressing `Alt+5`. Although we can't directly edit the script itself, we can add so-called "inline scripts". Inline scripts add information to the script, and we insert them using an InLine object at the point we want to add this information. Also, there is a `User` tab in the `Script` window, in which you could define global variables or rules – to be used anywhere in the script. We're going to build a break for the participants: after they've seen 5 trials in a block they will get the opportunity to pause.

1. Create a text display called "Pause" with pause instruction. Something along the lines of "Please, take a break and continue by pressing any key". Put this display at the end of the `TrialProc` timeline (it doesn't matter which of the two as they're clones). You might have already spotted that like this, every trial will finish with a pause display – don't worry, we'll fix this with the `InLine` object!

2. Add a Label object from the Toolbox right before the pause display. Give it the name "Break". A label is used to mark a certain position in the timeline. (In a moment, you'll see how we jump to this label while not showing the pause display.) Add a second label after the pause display and name it "Continue".

3. We'll need to count the number of times a certain trial has been presented, so, we'll define a counter. The counter will increase by one every time a trial is presented. We will define this counter at a *global* level – i.e. so that it applies to the whole experiment. Then we can refer to it from any part of the experiment. Click the `User` tab in the `Script` window (which should be visible in the Workspace as soon as a script has been compiled). Type the following:

   ```
   Dim iCounter as Integer
   ```

   This asks E-Prime to define ('Dim') a variable called "iCounter"[17], and we want it to be a number ('integer') variable, so that we can use it for counting.

[17] FYI – starting the variable name with the small letter 'i' is a programming convention, indicating that it is a variable we've defined ourselves (rather than a variable already defined by the program).

4. Now we're going to add rules at a *local* level. Drag an `InLine` object to a `TrialProc` and drop it right after the target display. Name it "TrialCounter". Type the following *if-else* rule in the InLine:

```
iCounter = iCounter + 1

If iCounter = 5 Then
    GoTo Break
    Else GoTo Continue
End If
```

   This code tells E-prime to add 1 to the counter count during every TrialProc. Since it is placed right after the `Target` object it will count the presentation times of this object. If the counter is at 5, the procedure follows the expected order: here we've made this specific by jumping to `Break` (the script would also work without this label, by simply continuing along the line). The participant will see the pause display. But in all other cases the procedure jumps to `Continue` and proceeds while not showing the pause display.

5. We're nearly there! The way it is scripted now, there will be just one break in the first block. The counter counts target displays at every stage in the experiment and there is only one fifth time. There are several ways to solve this, yet most people would choose to reset the counter after the break. Drag a secondary `InLine` object to a `TrialProc` and drop it right after the `Break` object and call it "ResetCounter". Type the following in the inline script:

```
iCounter = 0
```

Exercise 34 - Run your experiment a few times yourself, with different subject numbers, in order to generate some data files that you can use in the next section.

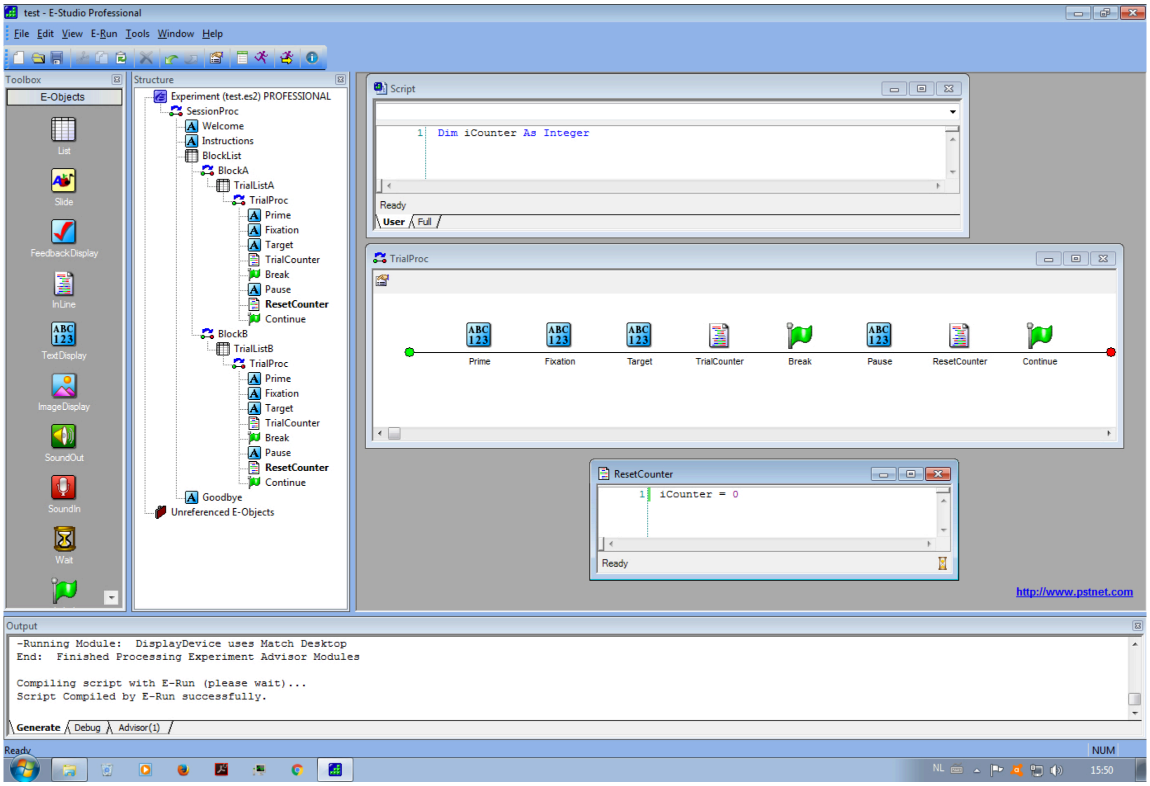


Figure 17: The script so far.

Question 15 - Given the current script, what would happen if you change the inline to present a break after every four trials?

As said there are several ways to script multiple breaks. Most importantly, you want to avoid to have a pause screen right before the end of the experiment – this looks stupid. The most straightforward way to accomplish this is to spell out every trial count number of an object that requires a pause screen to follow without resetting the counter. Since the counter keeps track on the target display in both blocks in the the current script, you'd get something like this in the inline script that's placed right after the `Target` object:

```
iCounter = iCounter + 1

If iCounter = 4 Then
        GoTo Break
            ElseIf iCounter = 8 Then
                GoTo Break
            ElseIf iCounter = 12 Then
                GoTo Break
    Else GoTo Continue
End If
```

Needless to say, if you want to use this option, you should get rid of the inline script that resets the counter!

8.4 E-merge: merge multiple output files

By now you are able to do experiments in which you present text. Once you've tested several participants you can merge all .edat files into one .emrg file. It works as follows:

1. Choose in the E-Prime 2.0 folder E-Merge to launch it. A `Quick Reference` dialog is displayed, detailing the basic steps necessary to merge files. Click `Close` to dismiss the `Quick 1 Reference` dialog.[42]

2. In the Folder Tree, navigate to the folder that contains your .edat files.[43]

3. Click the `Select Unmerged` tool button to select all E-Prime data files (*.edat or *.edat2) that have not yet been merged into another file.[43]

4. Click the `Merge` tool button on the toolbar. A dialog appears requesting the type of merge operation to perform. `Standard Merge` is the default operation. Click `Next`.[44]

5. Give a reasonable name for the merged data. This is the `Target File`. Click `OK` to launch the merge operation. Click `Yes` to create the target file. Review the `Merge Results` dialog displayed after the merge operation is complete. Click `OK` to dismiss the `Merge Results` dialog.[45]

Earlier you used E-DataAid to check your output, which was in an .edat file. Merged files can also be opened with E-DataAid. There's an option to do basic statistics but we strongly recommend you export data to be able to analyse and manage it in better-suited programs, such as Excel.

1. Open an .emrg file. From the File menu in E-DataAid, select the Export command. The Export dialog is displayed offering various options for export. Note that only the displayed (i.e. filtered) data will be exported.[54]

2. In the Export dialog, set the "Export to" field to `Excel` and select `OK`.[54]

3. In the Export As dialog, enter the filename and click the Save button. The saved .txt file can be imported in Excel using `File > Import...`[54]

8.5 More E-studio: pictures, sound and movies

On the basis of the script that you've written so far we'll explore some more advanced presentation options. It's best to open the original script and save a new version of it for each new section of the tutorial below, in which we'll go through handling of pictures, sound and movies. Note, that for audio and video to work with E-Prime in Lipsius rooms 1.26 and 1.27 you'll need to plug in a headphone into the workstation!

8.5.1 The Slide object

A Slide Object allows presentation of text, sound, images, and movies concurrently. Furthermore, you'll be more flexible as to the position of stimuli on the screen. Let's start presenting two different pieces of text at once.

1. In a newly saved version of your script delete the Target object. Make sure that it is deleted completely.

2. Drag a Slide object from the Toolbox to the TrialProc. Rename this object "Target".[31]

3. Double click the Target object to open it in the Workspace. Click the `SlideText` button on the Slide toolbar, then click anywhere within the active Slide window. A text sub-object will appear in the Slide window.[34]

4. Use the dropdown menu on the Slide toolbar to select the text sub-object. Then click the green sub-object property pages button on the Slide toolbar. There may be many sub-objects composing a Slide. Select the sub-object that you want to set the properties for.[34]

5. On the `General` tab, enter `[Stimulus]` in the Text field just as you've done earlier.[34]

6. On the Frame tab for the text sub-object, set the `X- and Y- Position` properties to `center`. This will centre the text in the display and click `OK` to accept the settings.[34]

Exercise 35 - Add the keyboard as response device. Use the same properties as in the former script (allowed buttons, duration etc.). Note: the properties button is used to set the properties of the entire object (e.g., duration, input masks, etc.) while the green sub-object properties button is used to set the properties of individual sub-objects. Now add a second sub-object text display and place it just below the first. Just for the purpose of the exercise: let the second text display be the same prime as shown during the trial, but in a different colour. Test your script.

8.5.2 Presenting pictures

Since we have a Slide object in place, it's pretty easy to change the content of the slide. We'll start with pictures. Whenever you are using any kind of external files (pictures, movies, whatever), make sure to put them in the folder where your script is located. This makes life a whole lot easier if you need to move your script to other locations (even between different computers in the lab!). So, move all materials – without using separate folders – and put everything in the folder of your script. As recommended above, save a new version of your script from 8.5.1 to work on in this section.

1. Open the `Target` object in the Workspace. Use the dropdown menu on the Slide toolbar to select the text sub-objects. Right click and select `Delete` to delete the Text sub-objects from the Slide.[59]

2. Click the `SlideImage` button on the toolbar, then click anywhere within the active `SlideState` window. An image sub-object will appear in the Slide window. The picture will be gray because no file has been selected yet.[59]

3. Click the green sub-object property pages button. On the General tab, enter [Picture] (i.e. the attribute that you can find in the trial lists) in the Filename field, for Stretch select Yes, and for Stretch Mode select UpDown.[59]

4. On the Frame tab for the image, set the Width to 50% and the Height to 75%. Set the X and Y Position properties to center and click OK to accept the settings.[59]

5. Click the white property pages button to view the properties of the Target object. On the the Duration/Input tab, verify the following:[60]

   - Duration is set to 3000ms
   - Data Logging is set to Standard
   - The Keyboard is enabled as the Input Device
   - The Allowable field is set to 12
   - The Correct field references [CorrectAnswer]
   - Time Limit is set to (same as duration)
   - End Action is set to Terminate

6. Click the OK button to close the property pages.[60]

7. Adjust the instructions and perform a test run.

Exercise 36 - Present the prime text on the picture, so that the participant sees both the picture and prime at the same time. Let the the text be in blue and slightly bigger than before.

8.5.3 Recording sound

Again, re-save your script from section 8.5.2 as a new file. In this section we're going to record voice responses.

1. Double click the Experiment object at the top of the Structure window and select the Devices tab from the property pages.

2. Click Add and choose SoundCapture and click OK to dismiss the Experiment Object property pages. Note, that for any device to actually work with E-prime, it needs to be physically there and attached to the computer you are using!

3. Open the Slide object and click the `SlideSoundCapture` button on the toolbar, then click anywhere within the active SlideState window. A sound sub-object will appear in the Slide window. The sound sub-object will not be visible when the experiment is running, and can be placed anywhere on the Slide itself.

4. Select the SoundIn1 sub-object and click the green sub-object property pages button. The `@Auto` in the Capture Filename means that E-Prime will automatically count and name the recorded files. You will see them in the folder in which your script is being saved. With the default settings the recording will last for the duration of the slide object.

5. Adjust the instructions: participants need to respond by saying male or female.

6. It's time to test! Did the recording work?

If you have an SRBOX available (all booth setups, in the Baby Lab it's the newest version called "Chronos") you could use the participant's voice to terminate the presentation of a Slide. To do so you'll need to add the SRBOX in the Experiment Object Property Pages (you've probably just seen it when you added the SoundCapture device). Then you'll be able to add it as Device in the Duration/Input tab. Allowable should be set to `6`, meaning that number 6 on the SRBOX is coded as Voice Response. You might need to adjust the voice input level (on the sound card). We recommend that you take the time to book a slot in one of the labs to try this out.

8.5.4 Presenting sound

Instead of the visual prime, we'll now create an auditory prime. Save your script from section 8.5.3 as a new file.

1. Remove the prime text slide. Delete any unreferenced objects.

2. Drag a new Slide object to the beginning of the trial procedure. Rename it to "SoundPrime".[65]

3. Double click the Experiment object at the top of the Structure window and select the `Devices` tab from the property pages.[67]

4. Confirm the checkbox next to Sound is checked. Make sure you can actually play (listen to) a sound file on the computer you are using.

5. Click OK to dismiss the Experiment Object property pages.[67]

6. Open SoundPrime in the Workspace and click the SlideSoundOut button on the toolbar, then click anywhere within the active SlideState window. A sound sub-object will appear in the Slide window. The sound sub-object will not be visible when the experiment is running, and can be placed anywhere on the Slide itself.[68]

7. Select the SoundOut1 sub-object and click the green sub-object property pages button. Open the sub-properties on the General tab, set the Filename field to [SoundFile] – the Attribute already prepared in the trial list. The filename for the sound sub-object will be resolved by this Attribute when you run your experiment.[68]

8. Click OK to close the property pages.[68]

9. Click the white property pages button to set the properties of the SoundPrime object.[69]

10. On the Duration/Input tab, set the Duration to 1500ms.[69]

11. When you are finished, click the OK button.[69]

12. Adjust the instructions and reset the target response to keyboard input, and test.

Exercise 37 - Present the text of the prime along with the auditory prime. Centre the text: in the Frame tab in the sub-properties, set the Width to 25%, Height to 25%, X to center and the Y to 25%.

8.5.5 Presenting movies

As a last step, we'll present a movie together with the sound file. Save your script from section 8.5.4 as a new file.

1. Click the SoundPrime object in the Structure window, then press F2 to rename the object to "MoviePrime" reflecting the fact that the prime will now be presented as a movie, along with text and sound.[76]

2. Click the SlideMovie button on the toolbar, then click anywhere within the active SlideState window. A movie sub-object will appear in the Slide window.[76]

3. Select the movie sub-object and click the green sub-object property pages button on the Slide toolbar.[76]

4. On the `General` tab, set the Filename field of the movie sub-object to refer to `[MovieFile]`, and set the Stretch property to `Yes`. Again, the filename for the sub-object will be determined at runtime.[76]

5. On the `Frame` tab, set the Width and Height to `50%`. Set the X-Position to `center`, and the Y-Position to `55%` and click `OK` to accept the settings.[76]

6. Click the white property pages button to set the properties of the `MoviePrime` object. Note: the sound icon may disappear behind the slide movie, but this will not affect the experiments functioning.[77]

7. On the `Duration/Input` tab, set the Duration to `3000`ms.[77]

8. When you are finished, close the property pages by clicking the `OK`.[77]

9. Adjust the instructions: participants will see and hear a priming word along with a movie. Picture determination is still the same.

10. Test it!

8.5.6 Feedback display

In some designs it is useful to give participants feedback showing them their performance after a trial. E-Prime has a built-in function to facilitate this. In the following steps we'll add a feedback display after the object Target. Go back and open your first script `PictureTutorial-<your last name>.es2` and save a duplicate that you will make changes in.

1. Drag a `FeedbackDisplay` object from the Toolbox to the TrialProc right after the `Target` object. Rename this object `Feedback`.[31]

2. Double click the `Feedback` object to open it in the Workspace.[36]

3. Click the white property pages button to view the properties of the `Feedback` object.[36]

4. On the `General` tab, specify the Input Object Name to be `Target`. You need to specify the name of the object that is collecting the participants response. In this experiment, the participants response is collected in the Slide object named Target.[36]

5. Click `OK` when you are finished.[36]

6. Open the `Duration/Input` tab of the Target object and verify the Time Limit is set to `(until feedback)`, and the EndAction is set to `Terminate`. Click `OK`.[35]

7. Run the script to test it.

Exercise 38 - Now add a practice block to this script, consisting of 4 trials. Use different primes and names.

8.5.7 Troubleshooting

Programming a script doesn't happen without making mistakes. Don't panic! Save and compile regularly, working step by step. The `Help` button in the top menu is a useful resource.

A common problem is the codification of sound and video files. In the case that E-Prime can find a media file but won't play it, you may need to use Codec Config. Follow these steps:

1. Select in the menu of E-Studio `Tools > Codecs Config`
2. At "Would you like to create the default filters and mappings?" – click `no`
3. Select in the left panel Codecs - select in the toolbar `Update Cache`
4. Select in the left panel Codecs and select `WMV Screen decoder DMO`
5. Select in the right panel the *.dll codecs
6. Select in the menu `File > Render File` – select the video file and click `yes`
7. Click in the menu `File > Close` and save the configuration

Making mistakes is part of getting to know the program. For the final exercise of this chapter, we've created a script with some common mistakes included on purpose for you to try and solve.

Exercise 39 - **From https://doi.org/10.5281/zenodo.1339334 download error.zip. Open the script error.es2 and run it. As you will see, there are some errors in the script. Try to solve them!**

9 Preparing to analyse experimental data

LEARNING GOALS

- recognise which steps need to be taken before data can be entered into a statistical analysis
- understand the issues concerning the screening and cleaning of data

So far you've learned how to research the necessary background literature for a research question, how to conceptualise an experiment, how to prepare a stimuli set and how a script works in E-Prime. You've also got an overview of the lab setups that are on offer at LUCL.

In the second half of this course, you'll learn how to actually collect the data by working through an experimental script in each of the labs and put it into practice in an independent project. For now, we are going to skip ahead a little bit and give you a preview of issues that come up when you are processing data that you have collected. Why do this now? Because knowing in advance the kind of issues that come up will always help you plan ahead – in terms of experimental design, in terms of how you run your experiment and in terms of how you organise and manage your materials and data.

9.1 Preparing and inspecting a dataset

Datasets nearly always require some pre-processing before you can start analysing them. This could mean cleaning the data to reduce unnecessary noise (as with EEG), screening the data for participants who did not fulfil the criteria or stimuli which participants could not understand, or it could be adding variables to the data file which you need for analysis. You may even find that you need to rearrange the format of the dataset in order for it to be successfully imported by the program you're using for statistical analysis. The exact steps you need to take will differ depending on the type of experiment you've done and the way you plan to analyse it. Here are a couple of typical examples:

- When you learned about the EEG lab, you learned that eye movements and blinks cause artefacts in the signal. During the pre-processing stage, you apply techniques to remove them. Pre-processing is a vital part of making your EEG data analysable – because the differences in potentials are so small, it's important that the data is as clean as possible. We use the program Brain Vision

Analyzer for pre-processing EEG data (available on the computers PC1 & PC2 in the EEG lab).

- Language production experiments require a lot of pre-processing. In Chapter 8 you learned how to record voice responses in E-Prime. This is the way that we capture responses when we do production experiments. However, before we can analyse the data, we need to listen to every response recorded, log the reaction time (the time from the start of the recording to the onset of the voice response) and transcribe any errors (and if you are doing a sentence production experiment, you'll have to transcribe every sentence).[18] Luckily there are some handy Praat scripts for helping you with this, or you can try using CheckFiles, a program designed for reviewing vocal response files (available as part of the CheckVocal software download by Athanassios Protopapas).

In the rest of this chapter we are going to use a data set adapted from a real experiment to look at some of the steps that need to be taken into account for any experiment you do.

Exercise 40 - Download and open the Excel file data_handling_tutorial.xlsx

This dataset is adapted from a priming experiment, in which 50 participants took part. Participants saw pictures of objects on the screen had to name them as quickly as possible. Each picture was preceded by a word prime. Each participant completed 36 trials. Before this, they had five practice trials to get used to the experimental task.

Question 16 - Look at the worksheet called 'Results' in the Excel file.

- **What kind of priming was tested in this experiment? Can you guess what the hypothesis was?**
- **How many counterbalancing lists were there in this experiment? Which variable was counterbalanced?**

[18] Although it's possible to record vocal reaction times automatically using a 'voice key', the voice key can't tell the difference between a real vocal response and some other vocal sound or cough, so it's not fully reliable and therefore the data always needs to be checked.

9.1.1 The layout of the dataset

The way you set up your data depends on the kind of analysis you want to do, but also on the program you're going to use to analyse it. The bottom line is, if your data isn't formatted in the right way, the program you use for statistical analysis won't be able to run the tests you want.

Typically, in the kind of research you'll do, you will have a number of participants, who each respond to a number of stimuli. So your total number of data points will be *(number of participants) x (number of trials)*. But there are different ways to arrange all these data points in a file.

The two main types of format to be aware of are wide format and long format. The **wide** format has one row for each participant. All of the measurements (observations) for one participant are listed in different columns, along one row. The 'measurements' might be the reaction time for each individual trial, or it might be an averaged reaction time for each condition in a within-subjects experiment. The **long** format gives a new row for each measurement for a participant. This means that information that doesn't change between measurements gets repeated (for example, the participant's name). Figure 18 illustrates the difference between wide and long data.

↓ wide data

participant	RT_picture1	RT_picture2	RT_picture3	RT_picture4	RT_picture5	RT_picture6
Pim	645	674	732	707	659	684
Joris	646	734	749	707	775	501
Eddy	674	576	533	772	617	771
Milosc	667	763	770	776	684	682

long data →

participant	picture	RT
Pim	1	645
Pim	2	674
Pim	3	732
Pim	4	707
Pim	5	659
Pim	6	684
Joris	1	646
Joris	2	734
Joris	3	749
Joris	4	707
Joris	5	775
Joris	6	501
Eddy	1	674
Eddy	2	576
Eddy	3	533
Eddy	4	772
Eddy	5	617
Eddy	6	771
Milosc	1	667
Milosc	2	763
Milosc	3	770
Milosc	4	776
Milosc	5	684
Milosc	6	682

Figure 18: The same data set in wide and long format.

Question 17 - Is the dataset in the Excel file in wide or long format?

9.1.2 Screening participants

You should be familiar by now with the need to gather data about participants' characteristics (e.g. age, linguistic background) as part of the process of conducting a controlled experiment. As we mentioned in Chapter 7, we usually do this with a background questionnaire, often set up in Qualtrics.

The background questionnaire is also useful for screening participants. For example, you may have tried to recruit only "native speakers", but that's no guarantee that all the people that took part fulfil your idea of native speaker. In fact, what defines a native speaker anyway? Is it the age you started speaking the language? Is it growing up in the language environment? Does it have to be the first language you were exposed to as a child? There's no strict definition of a native speaker.

The best way to resolve this is to reflect on the issues raised in Chapter 1 – both when you design your background questionnaire and when you review the responses. Are your participants representative of the population you are aiming to research? Are your participants more or less similar regarding characteristics which you need to control?

Question 18 - The Excel file has two sheets relating to participant characteristics: 'Biodata', which contains the responses to the background questionnaire, and 'Key to biodata', which explains the coding of the responses. Examine the participant characteristics.

- **Are there any participants you would exclude? Why?**
- **Do you think that the background data here is appropriate for screening the participants? (Why/why not?)**

9.1.3 Errors and outliers

It's pretty much expected that not *everything* in your experiment will go exactly to plan (if it does, you have our sincere congratulations). Sometimes it's down to the unpredictable behaviour of participants – for example, not following the instructions, or figuring out a strategy. Sometimes it's down to poor stimulus choice, or an unknown variable. And even when the experiment is well-designed to avoid all that, participants will sometimes make errors anyway. It's also important to realise that what counts as an error really depends on your experiment.

Question 19 - Looking again at the 'Results' sheet in the Excel file, what do you think counts as an error in this experiment? How are the errors in this dataset coded?

The good news is that the rate of errors in your experiment can sometimes be used as another kind of dependent variable, usually to support the outcome on the main dependent variable(s). For example, in a picture naming experiment (like this one) we might find that primed pictures are less prone to errors than unprimed pictures. This would support the idea that priming makes picture naming easier (or better, priming 'facilitates' picture naming). We should of course be careful when interpreting the distribution of errors if the error rate is very low – there may not be enough data to be confident about drawing conclusions.

The bad news is that if a participant or a stimulus is associated with a lot of errors, it may be advisable to exclude it completely from analysis. Participants and stimuli should be homogeneous with regard to non-experimental variables; an exceptional error rate for a stimulus or a participant suggests unexpected behaviour that doesn't fit in with the rest of the group. Exclusions are up to you in the end and it can be a difficult decision, but the more experience you get the better your judgement will become.

Exercise 41 - Investigate the rate of errors in the 'Results' dataset. (Tip: check back to Chapter 5 for Excel functions that could help you do this.)

- **Can you find any participants or stimuli that have particularly high numbers of errors? If so, do you think it's appropriate to exclude them?**
- **How many errors are there per condition? Do you think the error rate in this experiment is useful for evaluating the hypothesis?**

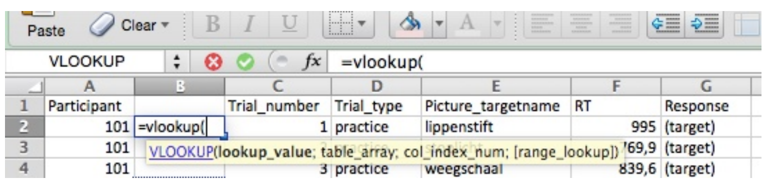
Sometimes there are data points that are not really 'errors' per se, but somehow don't fit well with the rest of the data. For example, if a participant named a picture absolutely fine but took three seconds to actually start speaking. This kind of data point is known as an 'outlier' or extreme value. Outliers can have a strong influence on the data (as you study statistics you will learn more about this issue), but they should be dealt with very carefully. It may be tempting to just exclude outliers (after all, who needs three seconds to name a picture of a banana?) but if you aren't careful you will soon find yourself trimming away the data according to your underlying bias: clearly not ethical! When you read published studies, you'll see that researchers sometimes simply cut off any data points that are a certain distance away from the mean. Whether this is reasonable is down to the situation in question – and whether you agree with it is up to you.

In conclusion, it's sometimes appropriate to exclude data points, stimuli or participants – but you better have concrete and justifiable reasons for doing so.

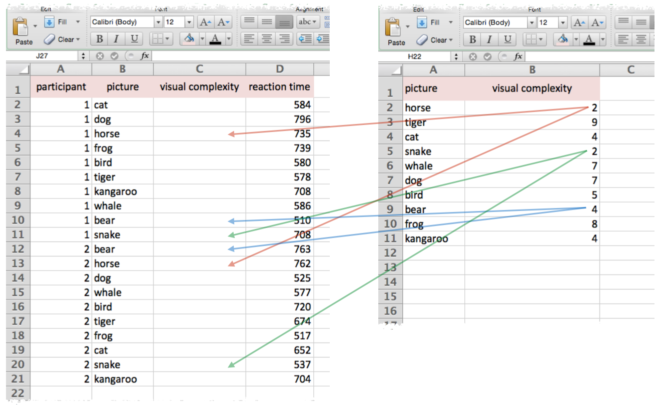
9.1.4 Making your data set complete

Sometimes you may find that you need to add some variables to your dataset in Excel. For example, you may have a data file that's exported from E-Prime, but you want to also include the visual complexity scores of the pictures (from your norming survey) or the age of your participants (from your background questionnaire) in order to be able to check the influence of these variables when you do your analysis.

If you try to do this by copy-pasting the values into the dataset, you will either make a lot of mistakes or drive yourself mad, most probably both. Fortunately, there is a very useful function in Excel, called VLOOKUP, that can make this process a lot simpler! The VLOOKUP function has the following structure:



To explain it, let's take an example. Imagine that in one worksheet you have your experiment data, and you want to add the visual complexity scores for all the pictures. In another worksheet you have a list of the pictures with their average visual complexity scores. Of course, these two lists are not in the same format: one is your entire experimental data, the other is a list of the picture norming scores. You want to automatically add the correct visual complexity score for each picture into the experimental data, like so:



Doing this by hand would be tedious (to say the least). Instead you can get the VLOOKUP function to 'look up' each picture name for you in the sheet of visual complexity scores, and give you the score associated with each picture. Here's how:

1. In the sheet of experiment data, insert a column to the right of the column containing the items you want to get the scores for. So in this case, you insert a column to the right of the column that contains the picture names.

2. Select the second cell of the new column, i.e. the first cell underneath the header row. In this cell, type `=VLOOKUP(` (don't forget the opening bracket). Don't press Enter yet!

3. Now use the mouse to click on the cell on the left (i.e. the second cell in the picture names column). You'll see the cell reference appear in the formula. Then type a semi-colon `;`.[19]

4. Now use the mouse to switch to the worksheet where your source list is (in this example, the visual complexity scores). You'll see the name of the spreadsheet appear in the formula bar (e.g. `!normingscores`). *Note: for VLOOKUP to work, the items you want to look up (e.g. your picture names) must be in the leftmost column of the source list. Otherwise, Excel won't be able to find them. So if that's not the case, you'll need to make it the case, and start again!*

5. Still using the mouse, click the top left cell of the table and drag to the bottom right cell (i.e. selecting the whole table). You'll see the table range appearing in the formula bar. Now type a semi-colon `;`.

[19] If you are using a UK/US regional settings, you will need to use a comma wherever we have used a semi-colon. See Chapter 5 for an explanation of this issue.

6. Next you have to tell Excel which column of the source list contains the scores you want to insert into your first worksheet. You do this by telling it how many columns from the left it needs to go to find the values you want. In our example, the visual complexity scores are in the second column – so, we would type `2`. Then type a semi-colon `;`.

7. To finish off the formula, type zero and close the brackets `0)`, and then press Enter.

8. Excel will automatically switch back to the first worksheet. You should find that Excel has put the correct norming score for the first picture into the column you created.

Before you copy the formula all the way down the column, there is one more thing to take care of. Remember in Chapter 5 we said that Excel references are relative? This means that when you copy-paste a formula, the cell references in the formula are 'translated' to be relative to the new cell. With VLOOKUP, you want the formula to always look up the value in the cell to the left – that's fine. What isn't fine is that VLOOKUP will also shift the cell references that relate to the source list! (To test this out, try copying the formula into the cell below, and see how the references are updated.) You need to make the references to the source list into 'absolute' references, which means that they stay the same no matter where you paste the formula:

9. Select the cell where you have your `VLOOKUP` formula.

10. In the formula bar, find the cell references for the source list – e.g. `!normingscoresA2:M27`.

11. Tell Excel you want the these references to be absolute. You do this by typing a dollar sign $ in front of **each part** of the cell reference. So your table reference should look something like this: `!normingscores$A$2:$M$27`.

12. Press Enter to finish editing the formula.

13. Copy the cell that has the formula in it, highlight the rest of the column, and paste.

14. Always do a spot-check to be sure that you have the right scores (if not, there may be a problem with your formula).

Exercise 42 - Use VLOOKUP to insert the participants' ages from the 'Biodata' sheet into the 'Results' dataset. If you're finding it easy, add one or two other columns too – for example, sex and handedness.

9.2 Proceeding with the analysis

After pre-processing your data, you're ready to analyse it. First you'll need to get your dataset into the analysis program. In statistics class you'll learn how to use SPSS, and in the second methods course you'll learn about R. When migrating your data between programs, it's good practice to always look over it or spot check it to make sure that all the values have been correctly imported. Checking data may seem really tedious but trust us, you don't want to get halfway through your analysis before realising that all your decimal points have disappeared (see Chapter 5), or discover later that the spectacular effect you've been bragging about is actually the result of a coding error.

Usually, we first **explore** the data (descriptive statistics), **visualise** it using plots and graphs, and finally **run statistical tests** (inferential statistics). It's tempting, once you have your data, to rush into trying to analyse it, desperate to see if your experiment 'worked'. However, without a clear plan for your preprocessing and analysis, you will quickly get in a mess with the data and find yourself HARKing like crazy (see Chapter 2). Therefore:

- review your hypotheses and plan the analysis steps before you start
- budget enough time for your pre-processing and analysis
- keep your data well-organised and make notes as you go

We've talked a bit about 'excluding' data (and the rights and wrongs of doing that). If you do decide to exclude some observations, stimuli or participants, it's not a good idea to actually delete this data entirely. You never know when you might need to reorganise or refer to your raw data. Most statistics programs will allow you to run tests on a subset of the data (you can add an extra column that indicates which datapoints should not be included and exclude or filter them out when running the analysis). If you *do* need to create a trimmed-down data set, make sure it is a separate file from your master data set so that you can always go back.

Lastly, in this chapter we've kept saying that choices "depend on your experiment" or are "up to you". Especially when you don't have so much experience, this can be quite disorientating. The golden rule is this: whatever decisions you make, make sure they are justifiable – often there is no right answer, so it comes down to making educated choices. Be able to explain to people why you excluded certain data, or why you chose a particular statistical test.

10 Scientific writing

LEARNING GOALS

- recognise the standard framework for writing up experiments
- integrate acquired skills by means of a research proposal

When you write up your experiments, you'll need to follow the standard format that all scientific articles and journal papers follow. This format is followed for very good reasons: it makes life more straightforward for both the writer and the reader. In this chapter we'll introduce the scientific writing framework, and then relate it to the writing of a research proposal - which is the final assignment for this course.

10.1 The point of scientific writing

Straightforwardness, or clarity, really is an important aspect of scientific articles. Obviously, one reason for writing a scientific paper is to inform readers about the work you've done and the conclusions you've drawn. However, scientific papers should also be written clearly enough so that someone else could take your paper, run your experiment again and compare the results. Re-running an experiment like this is known as **replication**. Replication is a vital part of experimental research. If the same outcome is found by different researchers working in different labs, we can be much more confident that the result is a real discovery about the language system, and not just due to a coincidence – or an unknown confound.

10.2 Elements of a scientific article

As mentioned above, scientific articles follow an agreed framework. When you search the literature for yourself, you'll discover there are variations on this structure – for example, where multiple studies are combined in one paper, or when the paper is only concerned with reviewing the literature. However, the basic format for writing up an experiment is always the same, and this is what we'll focus on here.

10.2.1 Abstract

Although the abstract is often the first thing you see, it's normally the last thing you write. It's basically a summary paragraph, so that readers can find out the main details at a glance and decide whether they need to read the paper in more detail. One way to approach this is to write one or two summary sentences for each of the main sections of your paper in turn. To keep it short, journals usually impose a strict word limit for the abstract (see section 10.3 for more about journal requirements).

10.2.2 Background

The first part of the paper is an introductory section which gives the background to your study. In this part, you explain your research question and how you're going to test it. This usually involves reviewing the background literature in order to inform the reader about the current state of the theory, followed by an explanation of the issue your research will address (and how!). If you're dealing with a debated issue, you'll need to make sure you summarise both sides of the debate and how your experiment fits into that.

Usually the background section concludes with a specific discussion of the hypotheses for the experiment you are reporting, where you explore the possible outcomes and how they would line up (or not) with existing theory. This part may include a brief overview of your experimental design, as far as is necessary for the readers to understand your hypotheses.

10.2.3 Methods

Next comes the method section. In the method you report the following:

- Participants
  How many participants? What was their mean age, and how many male/female? What criteria were used to choose/reject participants (e.g. linguistic background)? Were they rewarded for participating (money, credits)?

- Materials and design
  What were the stimuli, how were they chosen? How many trials per participant (critical, practice, fillers)? Any counterbalancing?

- Procedure
  How was the session conducted? How was the data recorded? What equipment was used?

- Preprocessing (if appropriate)
  Did you preprocess the data in some way that is important for readers to know?

- Ethics
  Show that you followed the relevant ethical procedures that would be required for your study. Studies that must be approved by an ethical committee usually have an 'ethics statement'.

10.2.4 Results

After the methods, you report the results of your experiment. An important rule to follow is that *you should not include any theoretical interpretation or judgement about your results* in the results section. The results section is where you report the objective outcomes of your experiment – so, just numbers and figures. This involves both describing the data that you got (means, frequencies, graphs) and statistical analyses (which test was used, and what were the resulting statistics).

10.2.5 Discussion

Lastly, you finally get to interpret your results in the discussion section. The key word here is *interpret* – remember from Chapter 1 why drawing strong conclusions is a difficult task! This means that we do not claim to have proven anything. Therefore in discussion sections you'll often see phrases like...

- *In this study we have found evidence for...*
- *The results suggest that...*
- *Our findings support the hypothesis that...*
- *This outcome indicates...*

The discussion should begin with a statement of whether your hypotheses (at the end of the background section) were supported or not by the results. Then, you can proceed to explore how the results you've obtained fit in (or don't!) with the theory. If your results are unexpected, you should try to offer explanations for why. Sometimes unexpected results can give rise to interesting proposals... and sometimes you just have to admit that your design had some flaws. Don't give in to the temptation of going back to change your hypothesis to fit your results – this is unethical!

The discussion section can be a difficult balance. On the one hand, you should recognise the limitations of your study – don't over-generalise or make grand claims that go beyond the evidence. On the other hand, you shouldn't let the limitations stop you from exploring the contribution that your data could make to the field (no experiment is perfect!).

Often, but not always, papers end with an additional **conclusion** section, summarising the main findings. There may also be an **acknowledgements** section where the authors state their funding and acknowledge important contributions from other people.

10.2.6 References

Following the main body of the paper, you should have your list of references. This should contain all the references in your paper, **no more and no less**! Exactly how to present your references – and the citations in the text – is decided by the journal where you submit your paper.

Exercise 43 - Find and download the paper *Altmann, G. T. (2004). Language-mediated eye movements in the absence of a visual world: the 'blank screen paradigm'. Cognition, 93(2), B79B87*. **Make notes on the following points:**

- **The research question (or goal) of this study.**
- **The experimental technique or paradigm used.**
- **The experimental hypotheses.**
- **Any ethical procedures that are mentioned.**
- **The number of different stimuli the researcher used, and how many trials there were for each participant. Are these the same number? If not, why not?**
- **The objective results of the experiment (e.g. which conditions differed and by how much?).**
- **The main theoretical conclusion(s) of the study.**

10.3 Journal styles

For publication in any field, there are requirements about how you present your written work - these range from very general (e.g. how to structure the content) to very specific (e.g. the font size, margin size, number of spaces between sections, etc). In psycho and neurolinguistics, the commonly used standard is APA style, i.e.

following the *Publication Manual of the American Psychological Association* (which we introduced in section 3.3).

Whenever you submit work to be published, there will be guidelines provided by the journal (usually called a 'style sheet'), and you may be referred to the APA Publication Manual. If you are simply writing up your work without knowing if or where you will submit it for publication, it's good to follow the APA style anyway, to keep things consistent and avoid extra work later.

Of course, the APA Publication Manual is an entire book, which we don't expect you to sit down and read cover to cover! For now, the main things we want you to focus on are the way that the content should be structured (discussed in this chapter) and the way that citations and references should be presented (which should be straightforward if you use Zotero, which you learned about in Chapter 3). The APA website has an online introduction if you want to get a quick overview of the basics. The book itself can be found in the university library.

10.4 Elements of a research proposal

Writing good research proposals is an essential skill in academia! Any time you do your own project, you'll need to propose it to whoever is going to support you (your supervisors or, later, the people who will give you money!). The better the proposal, the better your chances of getting to do the research.

A research proposal has similarities to a research article, in that it contains as much of the same content as possible (particularly background, methods and references). But of course, you can't talk about data that you don't have! Therefore, while a research article tends to focus on the interpretation of data that's been collected, a research proposal should focus on exploring the possible outcomes and their different consequences for the theory. This means you need to carefully and methodically consider *all* the possible outcomes associated with each hypothesis and how they would relate to the theory – not just the outcomes you hope for, or expect.

A proposal should also convince the reader that the proposed experiment needs to be carried out and that the data collected can give important insight into the theory. Therefore, you should still acknowledge limitations – in fact, you should discuss how you will deal with them, or why the experiment is still worth doing. After all, the people reading your proposal are going to be critical, academic readers, who may have years of experience running experiments. They will understand that setting up experiments can be difficult, but they won't be happy if you're sweeping obvious problems under the rug. You should also bear ethics in mind – think about whether your research will require ethical approval and how you will be able to demonstrate that you are following ethical procedures.

10.5 Presenting your work

Although the idea may make some of you want to crawl out of your own skin, it's important to get into the habit of presenting your work – both proposals and outcomes – to different audiences. Not only will you get feedback, you'll also find that explaining your work to others helps you to organise your thoughts and see solutions to problems.

You could give a talk or present a poster at a student conference, or just ask someone to have a chat over coffee (doesn't hurt to offer them a cookie too). It doesn't even need to be people with a lot of experience – even just talking to your colleagues can really help you to get your thoughts straight. Bear in mind that your research doesn't need to be perfect to be presentable, in fact, the whole point is to learn from other people's insights, not to impress them with your awesomeness (even if you are awesome).

You'll learn more about presenting work in the second part of this course, which concludes with a practical project and poster session.

10.6 Your final assignment: write a research proposal

So, this is it: your final assignment for the first part of this course. We would like you to apply everything you have learned so far in writing a research proposal.

In addition to your research proposal, you need to create a **background questionnaire in Qualtrics**. This is the questionnaire you will use to gather information that you need to know about your participants when you run the experiment.

The requirements for the research proposal are as follows:

- You should propose an experiment that you could carry out (and analyse) using the facilities available at LUCL. It should be an experiment that you could carry out by yourself (under supervision), in a limited period of time (i.e. as the practical project in the second half of the course).
- Your written proposal should contain: a brief review of relevant papers; your research question and specific hypotheses; a clear explanation of the experimental design (including a few example stimuli); details of any norming data that

needs to be collected; exploration of possible outcomes and discussion of how your research can help advance the theory (see section 10.4).

- If possible, indicate how you would analyse the results.
- It should be minimally 700 words and maximum 1200 words long (excluding references).
- Follow APA style for the formatting and references. Text should be double spaced.

The idea behind this research proposal should be that you could come back to it in the second part of the course and actually carry it out, as your practical project – so it needs to be realistic! This means thinking in real, practical terms about things like how you will recruit participants, where you will get your stimuli from, and how long the testing will take. Above all, **KEEP THE DESIGN SIMPLE!** We cannot stress this enough; even experiments that appear simple at first can quickly turn out to be complex beasts. Try to stick to one independent variable, if possible (and if it is categorical, not too many different levels).

Apart from keeping it simple, you need to **BE SPECIFIC.** One thing that students find difficult the first time they propose an experiment is that they often do not go into enough detail about exactly how they will operationalise their research question. Let's take an example to illustrate. Imagine you read the following description:

> "The independent variable is priming. The dependent variable is reaction time. We need to control native language. We expect reaction time to be quicker."

After reading this, could you go off and set up this experiment? No! You still do not know: what kind of priming is involved (semantic priming, orthographic priming, masked priming, etc)? What kind of responses do you need to measure (button press, vocal response, etc)? Who are the participants (which native language, etc)? When exactly will reaction times will be "quicker" – what comparison are we going to draw between conditions?

Now compare it with this description:

> "In this experiment, the independent variable will be whether a target word is preceded by a semantically related prime or by a control pseudoword prime. The experiment will be carried out in Dutch, with Dutch native

> speakers. As the dependent variable, I will measure in milliseconds how long it takes from the presentation of the target word on screen to the point when the participant starts speaking. I will compare the reaction times in the priming condition with reaction times in the control condition. I expect reaction times to be quicker in the priming condition than the control condition ... ”

This time you could probably make a pretty good attempt at setting up the experiment.

Remember, this is just an illustration, not a template! The message we want you to take away is this: you can’t carry out the experiment until it the details are clear, and this applies to your own proposals too. As long as you are vague about what you are planning to do, you will not be able to proceed. So, think about your own design carefully. If it helps, imagine that you are writing it so that someone else would be able to set up the experiment.

If you’re stuck for ideas for your project proposal, you could start by looking back through the course materials for inspiration. It’s fine if you want to use or develop something that’s already come up during the course – the best proposals are the ones that find the right balance between using what’s already available and trying out something new.

Good luck!

Part III
Workflow of data collection & analysis

11 General lab rules

LEARNING GOALS

- know the Golden Rules of the Labs!

11.1 A lab is...

- **a tidy place**
  Imagine you're going to a restaurant to have dinner with a friend. You made a reservation in advance and when you arrive you see that the only table free indeed has a reserved sign on it – but the table is a total mess. Pieces of food everywhere, dirty napkins. Worst of all, nobody working at the restaurant seems to be bothered. Is this somewhere you want to spend your evening? Working in a lab means that you have to take into account the others who work there too. It is therefore very important that everyone leaves the lab clean and tidy, ready for those coming to work there after them. Leave materials and equipment in the state you (should have) found them, don't change the configuration of software or hardware without permission and be sure to clear up any messy tables!

- **a tranquil place**
  You and your friend decide to go to see a movie instead. As the lights dim and the movie starts, the people behind you refuse to end the conversation that they were having before the movie started. After 10 minutes you've no idea what the movie is about, and when you turn around for the tenth time to ask them to be quiet, you accidentally throw popcorn all over your friend. In a lab, conversation should be kept to a minimum. It is distracting, and when people are distracted they make mistakes that could have been avoided. Maintain a professional attitude and respect the working environment of your colleagues – don't disrupt their concentration.

Note that your own procedures will also benefit from tidiness and tranquillity. That's why we encourage you to write READMEs as they force you to be clear at all times – to make it understandable for yourself and for others. Just as with participant handling (we'll discuss that in Chapter 17) it is important to follow well-defined procedures. Handle the equipment in the same manner. For example, always use the same order of starting up computers if you use more than one.

11.2 Lab bookings and access

We make use of a Sharepoint environment to communicate lab reservations, lab issues and to share useful documents. A Sharepoint is a website containing documents, calendars and info pages, that only authorised users have access to. Our Sharepoint site can be found here:

https://intranet.universiteitleiden.nl/sites/ExperimentalLinguistics

Once you enter the address in your browser, you'll see the login page. Sharepoint access only works with university login details (i.e. a student number and accompanying password). It may be the case that access is denied because you haven't yet been given permission. If so, a message box will appear to allow you to request access. Normally, your request will be processed within one working day.

When you are planning to carry out an experiment you should notify the Lab Manager(s) in advance as they keep an eye on the overall occupancy. In principle, you can make reservations yourself using Sharepoint. However, only make a reservation for *real* occupancy when you are sure that you'll have a participant. It is absolutely not allowed to just book a lab for a whole day and then wait and see how many participants will actually turn up. Also, while you can do behavioural experiments in, for example, an EEG booth, bear in mind that EEG researchers will still have priority.

To make a reservation, log into Sharepoint. Then you can select the correct lab agenda on the left, and add a reservation. Figure 19 shows the reservation pop-up window in the lab agenda of the Phonetics Lab. Be informative as to what you are going to do – give understandable titles (and select the right facility).

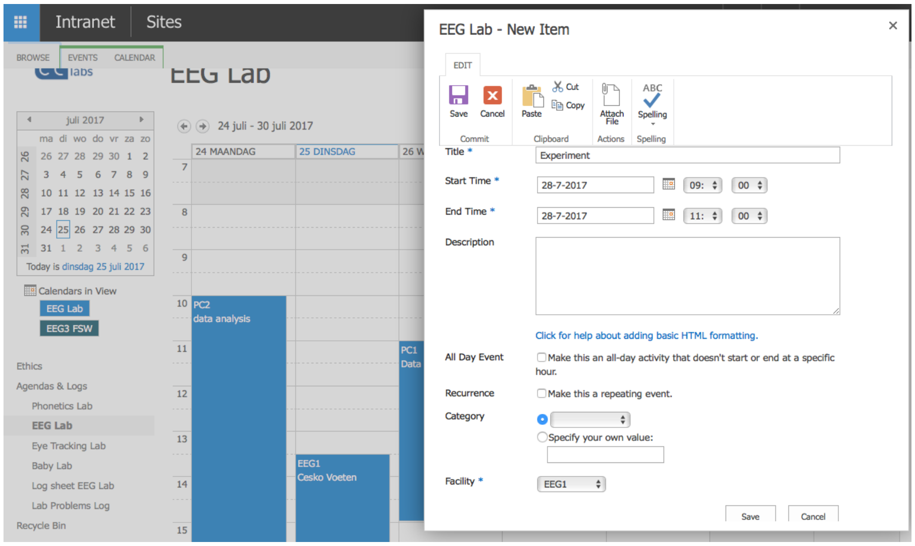


Figure 19: Pop-up window for making a booking in Sharepoint.

Often, your participants will be students, so it may be best to plan sessions that fit with the university class schedule (9:00 - 11:00, 11:00 - 13:00 etc.), especially in the case of longer sessions (such as for EEG testing).

Keys to the lab can be collected at the service desk downstairs in the Lipsius hall. You'll need to leave some kind of identification behind. Only take keys at the moments when you are actually using the lab, as there are others who might need those keys to access the labs! Likewise, return the key to the desk as soon as you're done so it's free for others to pick up.

There is a lot of expensive equipment in the labs. If you are the only one using the lab, make sure you lock the door even if you are just nipping to the loo.

11.3 Problem solving

Our experience is that many problems simply won't occur if everybody sticks to the two principles this chapter started with: **tidiness** and **tranquillity**. However, it's possible that you do run into issues. Firstly, stay calm! There are a few things you can do:

- If it's an urgent issue, contact the Lab Manager(s) immediately (lucl-labs@hum.leidenuniv.nl), or look for Jos Pacilly who has an office near the labs.

- If it's less urgent, like an error in your script, review everything that you've done step by step, and consult the relevant chapters of the reader (in some labs you will also find some useful information sheets).

- If you suspect a hardware or software issue, you should go to the "who you gonna call... ghostbusters!!" page (more officially known as Lab Problems Log) in Sharepoint. Here you can see if your problem is a known issue. If you're pretty sure that your problem is down to a hardware or software issue, you could add a new thread on the Lab Problems Log. Other lab users will be able to see, when they check the Log, that there is a problem.

In the case of issues that can affect others, **do not simply walk away and leave a known problem for someone else to discover**. This is not fair on other lab users. Problems (and mistakes) happen, but disruption can be reduced by quickly notifying the Lab Manager(s) and using the Lab Problems Log.

Use the flowchart below as a general problem solving procedure.

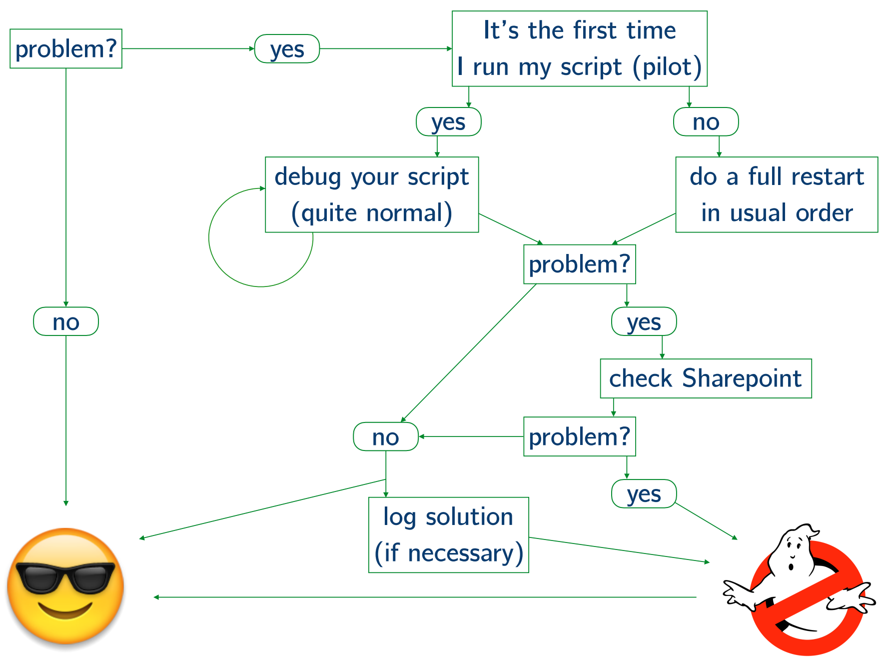


Figure 20: General problem solving flowchart

11.4 Borrowing equipment

Jos Pacilly or a Lab Manager may grant permission to use (recording) equipment outside the Labs. Send an email to lucl-labs@hum.leidenuniv.nl with your request. The following details will need to be logged:

- the name and contact information of the borrower
- the date of loan and estimated period of loan
- the materials taken
- the location and purpose of the equipment during the lending period

It is extremely important that you handle borrowed equipment responsibly. However difficult it is in the field, try your best to maintain tidiness and tranquillity in workspaces outside the lab too!

11.5 Dongles

For some software, dongles are required. These are little USB sticks which carry the licenses that allow you to use the software. Just as with borrowing equipment, we keep an agenda who has which dongle, so you need to contact us to borrow one (and when you have one, don't pass it on to anyone else!). As with all equipment, dongles must stay in the labs unless there is an agreement that you can borrow them (except for the Experiment Builder & Data Viewer dongle, which may only be used in the Eye-tracking Lab and EEG Lab, and must always be returned to the Eye-tracking Lab after use in the EEG Lab).

11.6 Saving data and disk usage

As a student (or employee) you can use and save data on the so called **P-drive** of the university network. This is your set of personal folders you're used to working with. Use this space as usual when creating your stimuli, experimental script and so forth - but be sure to make back-ups on an external hard drive or USB stick!

As you know, your space on the P-drive can be accessed wherever you are in the university: it's part of the 'network'. However, on every computer you'll also find a **C-drive**: this is the hard disk of that individual computer.

When you are ready to start testing we strongly advise you to transfer the script and stimuli onto the C-drive of the computer that you're going to use for your experiment. The reason for this is that if the computer has to constantly access everything on the university network, sometimes problems and delays can occur. If everything is stored on the hard disk of the computer, these problems won't crop up.

You can reach the C-drive if you go to `Computer > OSDisk (C:) > Users` and look for the folder which has your username as its name (usually `s` followed by your student number).

After testing a participant, please move all your data to your P-drive (and back up on your USB stick!). This is to keep the C-drive free from unnecessary memory usage. Imagine if everyone working on an EEG lab computer left all their data on the C-drive... it would be full after a few months or so! So again, keep a tidy work space!

11.7 Supplies

Although the status of supplies in the labs is monitored by the Lab Manager(s), it might happen that things run out. If you are doing an EEG experiment and you run out of towels, you can find more in the kitchenette. Other extra materials are kept in the brown cabinet next to the EEG 1 setup. If you lack some supplies or if there are no more towels, contact the Lab Manager(s). You're not allowed to use the washing machine yourself.

Supplies for the Baby Lab can be found in the drawer cabinet in that room. Again, if you have the impression that materials are running out, contact the Lab Manager(s).

It's always a good idea to check you have everything you need before you start, so that you can try to solve it before the participant arrives.

11.8 First Aid and Emergencies

On rare occasions, participants faint – though we've never experienced this ourselves. In the event that a participant faints, lay the person down flat on the ground, and call the First Aid Service (telephone 071 527 2200 or, if it's after 6pm, 071 527 4444). In case of a fire alarm (you'll hear a fire alarm which is referred to by Dutch speakers as "slow whoop"), leave the room immediately, making sure that nobody stays behind. Exit the building via the shortest route. At this link you'll see where you have to go once you're outside the building.

11.9 Golden Rules

- Be tidy: leave the place as you (should have) found it!
- Be tranquil: bear in mind other lab users.
- Only book the lab when you have a participant.
- If you have a key, you should be in the lab. Return it when you leave!
- Don't leave a lab unlocked and unattended.
- Do not borrow equipment unless you have permission.
- Notify Lab Manager(s) if there's a problem, don't leave it for someone else to discover.
- If in doubt, get in touch! lucl-labs@hum.leidenuniv.nl

12 EEG Lab: E-prime & ActiView

LEARNING GOALS

- set up and carry out an EEG experiment using E-prime and BioSemi's active system
- establish the communication between E-prime and ActiView
- tweak a configuration file for Actiview

12.1 Programming an EEG experiment

Look back to Chapter 6 to refresh your memory about the EEG lab.

12.1.1 General considerations

In EEG experiments we want participants to move as little as possible during recording of brain activity. Therefore, tasks should be designed to minimise this. Give the participant adequate information (before the experiment and during the task) and include enough breaks in between trials. Critical conditions should be presented at ideal moments. You also need to avoid having participants stare at the screen for a long time as this may generate an overload of alpha waves. Just like in reaction time experiments, it's also good to give the participant some warm up trials.

12.1.2 How does EEG recording actually work?

In Chapter 6 you learned the basics of how the EEG lab system is set up. Now that we're going to learn about the technique in more detail, it's important to understand a bit more about how exactly we are able to record electrical activity in the brain.

Electrical activity in the brain can be measured in microvolts (μV). Because the signal is so weak, it's also very susceptible to interference from outside sources such as computer monitors, lights, that kind of thing. Therefore, an amplifier is included as part of the hardware; in our system the signal is amplified at the source of recording, so that the influence of electrical 'noise' is minimised.

The μV we measure is the variation in the signal, compared to a 'reference point'.[20] During recording, this reference point is the CMS electrode (more about this in section 12.4). During analysis, the reference point we take is from the average of signal measured at the mastoids – these are the bony bits behind each ear. Because the electrical activity that relates to language processing does not tend to arise from the mastoid areas, these areas provide an optimal reference when averaged together.

It's crucial to understand that when we measure electrical brain activity, we don't just measure a signal in isolation. Think about it: what does it tell us if the scalp brain signal at a certain moment is 10 μV? Nothing, really. In addition, the brain is constantly doing all kinds of stuff (not just your experiment!) and generating all kinds of signals. What we want to know is how the signal *changes* in response to experimental stimuli. So what we have to do is to measure the variation in the signal of one condition (say a control condition), compared to another condition. The averaged difference in signal between these two conditions at some time point may be called an ERP – event related potential (see section 6.3). Note that you'll need lots of trials to get proper averages, so EEG testing sessions are usually pretty long!

By now you should have already seen during an EEG lab demonstration that eye movements affect the signal a lot. As explained in section 12.1.1, some types of movement can be minimised with good instructions and task design, but eye movement will always be problematic (also partly because the eyes are so close to the scalp and therefore the muscle activity is picked up and distorting very strongly in the EEG signal). So rather than stressing participants out by trying to get them to not move their eyes (!) we instead get around this by actually measuring the eye movements (also called 'ocular movements') in order to be able to isolate them from the data later on. For this, we place electrodes around the eyes – these capture the necessary movement signals to process later.

As this is a practical course, we're going to focus on learning by doing. However, if you plan to work more extensively with EEG or just want to understand it a bit better, we recommend you get hold of a copy of Stephen's Luck's book *An Introduction to Event Related Potential Technique* (Luck, 2014).

[20] It is always compared to a reference point because *voltage* is actually a measure of *potential difference*. If you don't remember this from physics class at school, don't worry too much! But if you want to brush up on your electrical principles, there's a handy appendix in Luck (2014).

12.1.3 Programming a task script in E-prime

In this section we are going to program a script in E-prime. It will be a dummy script on the basis of of E-prime's own PictureTutorial, which you can build on in the future if you like. The script is basically the same as the script you've made in Chapter 8. Use `Save As...` to save a new version of PictureTutorial, leaving the original version. (In case you mess up, you can always find a copy in E-prime's Tutorial folder.) If you are unfamiliar with E-prime, you'll need to take a look through Chapter 8 before you proceed.

Exercise 44 From https://doi.org/10.5281/zenodo.1339334 download the tutorial file <CrashCourse.zip>. Open it and put the items (a set of pictures, script, an .xlsx file and a configuration file) in a new folder with an appropriate name. Open the script with E-studio.

12.1.4 Prerequisites

At the top level of your E-prime script in the Structure window you can view and add devices. By default, scripts will have a display, a keyboard and a mouse added. If you do an experiment where you present sound stimuli you should add "Sound". Recording sound would require the device "SoundCapture". If you need to use the response buttons in the booth and/or the 'voice key' you need to add "SRBOX".

Exercise 45 - Go to `Experiment Object Properties` > `Devices` and add the SRBOX. In the labs it is attached to port number 3, so check this in the SRBOX properties window. Also check whether the desktop resolution is set to the values of your choice (match to desktop resolution will be fine).

Note that you can program your script without an SRBOX attached to the computer you are working on. Just uncheck the choice box, otherwise you cannot compile the script: you would get warnings that the SRBOX cannot be found. As soon as you are ready to test your script on a lab (booth) computer you simply check the box.

On the lab computer, the Display Index should be set to `2`: this will present the stimuli on the screen to the right as well as the screen in the booth.

When you do behavioural experiments, you get one response for each trial. However, when you record EEG signals, you're recording a continuous stream of data, which is stored in a bdf file (a BioSemi Data Format). Therefore, you will need to mark the moments of interest somehow in the recording, so that you can find them later. We do this by sending "triggers" to the recording program (Actiview) which is installed on a secondary computer. The triggers get logged in the data so that you can later pick out the brain response associated with each trial.

Open the .xlsx file in the folder you just created. You'll see the different stimuli listed. Every row contains all the info of one stimulus that may be used during a trial. The last column "Triggers" lists the the code that belongs to a certain condition – in this case trigger number "10" relates to the condition where the target "male" is presented after a positive the male prime.

In total, 255 different triggers are possible, by using a number between 0 and 256 – where 0 is used to reset to baseline. It is advised **not** to use numbers 1–9 as they may cause communication problems with E-prime (don't ask why). Before you script a task, you should always make a comprehensible code list for your triggers. The triggers should be added as Attribute in the relevant TrialList. You should always have decided on the design of your trigger system **before** programming your script!

12.1.5 The trial procedure

As you know, a trial procedure can be understood as a loop instruction. In the script that you have at hand it consists of a prime, followed by a fixation, target picture and concludes with a feedback screen. Under the header `Procedure` you'll see TrialProc as procedure. The advantage of working with self-defined procedures is that you can reuse those snippets of timeline at different moments in your experiment without the need to copy the contents one-by-one. For example, when you want to add a practice session, or when you'd like to use counterbalanced trial lists.

Exercise 46 - Double click "TrialList". Add the information of the .xlsx file to the trial list. Since you cannot copy the header bar you will need to specify the heading(s) in E-prime separately. Other cells can be copied and pasted in one go.

Exercise 47 - Change the font in the experiment to Verdana. Compile and save the script and run it to check whether it works. This is good practice: save frequently and every time you've changed something (major), do a test run. Is the font large enough?

You can use participant number `0`, which results in not saving any respondent data. This is handy while testing, though at some point you should check how an output file looks, to confirm that you haven't forgotten anything. Optionally, you can click the purple running guy with the yellow arrow in the top bar to choose your way of fast testing. You'll probably get a message about allowable values. You might have to change this at the point where you've assigned response choices.

12.1.6 Response buttons

The buttons on the chairs in the booth have specific numbers that are linked to the SRBOX. Left is '4' and right is '5'. In the trial list, 'CorrectAnswer' is assigned to the keyboard numbers '1' and '2'. For testing this is fine of course (you will do so many times!), but in the actual experiment the participant will need to use the response buttons.

Exercise 48 - Add the SRBOX locally as device to collect response data, assigning the right numbers. Also change TrialList and the instructions object text appropriately.

In case you want to use the voice-key as response, this has SRBOX number '6'.

12.1.7 Sending triggers: communication between E-prime task script and ActiView EEG recording software

In order to let E-prime communicate with ActiView, which is installed on a secondary computer, you'll need to specify the (parallel) port which E-prime should use to send triggers, and for which script objects this will be the case (Slides, Displays, etc.). You'll provide this information using an inline script (so, not at the top level of your script as you did with devices). It is advisable to start your experiment with an inline script in which you have something like:

```
Target.OnsetSignalEnabled = True
Target.OnsetSignalPort = &HC010
Target.OffsetSignalEnabled = True
Target.OffsetSignalPort = &HC010
Target.OffsetSignalData = 0
```

This will ensure that, once you present the object *Target*, the port, which has a specific number and might differ between setups, is known. In our labs the port through which triggers are sent has the code *&HC010*. Locally – i.e. at the point in the script when you present a stimulus – you tell E-prime which trigger should be sent by adding an inline containing the following:

```
writeport &HC010,0
Target.OnsetSignalData = c.GetAttrib("TarTrigger")
```

In this example, a stimulus presented in the Slide called *Target* should have an accompanying trigger listed in the Attribute column *TarTrigger*. The inline should precede the Slide. At the beginning of the inline the port is set to zero, which changes once the presentation object appears.

Exercise 49 - Add the inlines explained above into the script that you're working on.

12.1.8 Adding condition independent triggers

In some designs, you are only interested in the correct responses. To make this easier, you can add this info in the EEG recording by sending a trigger based on whether the participant's button press was correct or not. Then it's easy to filter out the incorrect responses later. To do this, an inline can be added containing something along the lines: If the response is correct send trigger such-and-such, and if not, send something else. In E-prime language this would translate to the following (replacing *trigger* with the relevant numbers):

```
writeport &HC010,0
If YourObject.ACC = "1" Then
    writeport &HC010, trigger
Else
    writeport &HC010, trigger
End If
```

E-prime calculates the accuracy (ACC) on the basis of the button press, which is compared to the value that you have listed under CorrectAnswer in the trial list. So, to get this to work you need to have assigned this Attribute to the properties window of the appropriate object.

Exercise 50 - Add an inline after the Target object containing a correct response trigger. If correct send 55, if incorrect send 66. Check the properties window of the target object. Where is Correct Response assigned?

Exercise 51 - Adding a trigger to mark the onset of a trial may be handy too. Add one telling E-prime to send trigger number 98 at the beginning of each trial (ask yourself which object the inline refers to). Don't

forget to adjust the inline right at the beginning of your experiment! When you're done, run the script to test it.

Finally, the structure of your experiment should look like the figure 21.

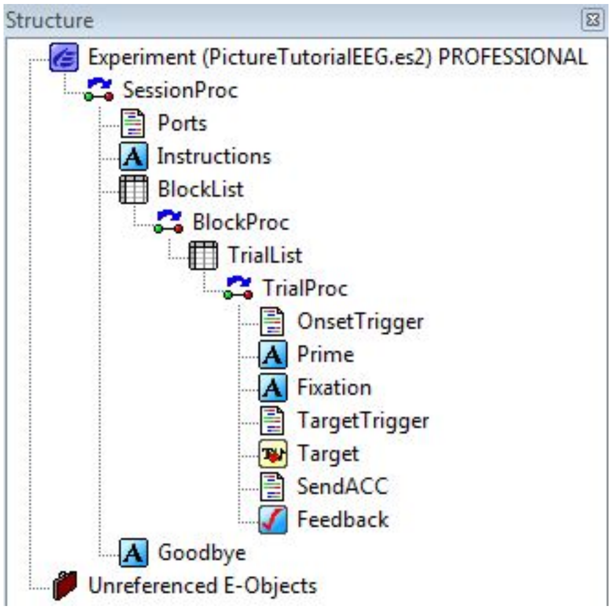


Figure 21: The script containing inlines.

12.2 Tricks in ActiView

Now it's time to check whether the communication with the secondary computer works. In the zip file you'll find a erp-32.cfg file, which is a configuration file to be used by ActiView. We strongly recommend you to use a configuration file other than the default provided by ActiView. We have predefined configuration files in Sharepoint – suitable for 32 and 64 setups. You can customise these by opening them in Notepad++ (Windows) or TextEdit (Mac). You could also design a configuration file from scratch. In any case, be aware that you should only save the channels that you need. The standard option is to save *all* 256 possible channels, and this ends up with an unnecessary memory load, which will be especially annoying during data analysis (and we've been there, done that...!).

Exercise 52 - Open erp-32.cfg with a text editor and scroll down below until you see `SavePath=` and change its setting to the appropriate directory

to save your bdf file (i.e. the EEG output). Leave everything the same. Save and close the file.

Once you start the recording, by hitting the 'Pause Save' button, ActiView will automatically write data to this directory (the one you have specified by `SavePath=`). You can also see this within ActiView, above the green 'Save' button (if you're not saving anything this button is called 'Start File').

Exercise 53 - Turn on the EEG amplifier in the booth. Open ActiView on the secondary computer and load the configuration file: click `About ActiView > Load Config File` and press `Start`. What happens if you turn off the amplifier?

Because there are no electrodes attached, you'll see flat lines in the `Monopolar Display & Triggers` tab and maximum values on the `Electrode Offset` tab which measures the connection strength (ideally this should be between -15 and +15).

Exercise 54 - Start your E-prime script and check whether the coloured lines in the trigger window change at the moment you present a stimulus.

The coloured lines in the trigger window will give you a colour code of your trigger number. When you open a saved ActiView file (extension .bdf) for analysis in Brain Vision Analyzer (see section 12.3) you'll see the actual numbers again.

If you want to change the configuration file in ActiView, make the changes, stop the file and go to `About ActiView` to click `Save Config File`. This will be your personalised configuration file.

12.3 Preview of the EEG data

We'll leave data analysis to a later stage – when you're going to run a real experiment, for example, as project later in this course. However, it is wise to check the output generated by ActiView in a program that can read bdf files. Brain Vision Analyzer (BVA) is the program we use for this, and it is installed on PC1 and PC2 in the EEG lab. To check whether the EEG and triggers were actually recorded do the following:

1. Open BVA
2. Select `FileNew > Workspace...`

3. Create (or select) the folders where 'Raw data' (your bdf files) can be found, and where 'History files' and 'Exported data' should be saved
4. Save the workspace under a meaningful name

Click OK and you'll see the EEG output (and triggers underneath) if all went well!

12.4 Carrying out the experiment

12.4.1 Electrode types

In our lab, we use the BioSemi ActiveTwo system.[21] We use two types of electrodes:

- Flat electrodes: to record ocular movements (EOG) and reference potentials at the mastoids (see section 12.1.2).
- Electrodes with conductive pins: to be plugged in an electrode cap. They are either bundled as 32 electrode "electrode bands", or as CMS/DRL duplets.

The surface of BioSemi electrodes contains silver/silver chloride (Ag/AgC), which has excellent conductive properties. As noted in 12.1.2, the electrodes contain a micro amplifier to reduce the impact of electromagnetic interference that might appear on the way between the electrode and computer. This explains the name "Active System" – the system actively amplifies the signal.

Please **treat the electrodes and their wires with great care** as they are very expensive and fragile:

- Never scratch the electrode surface.
- Hold the electrodes by their plastic body when you attach or remove them.
- Never let electrode surfaces come into contact with metal: during cleaning, always place them directly the plastic wash basin (and use only a bit of Ivory soap)!

The actual connection between skin and electrode is achieved by using a special electrode gel. This gel is called 'Signa gel' (you'll see the green tubes in the lab). It is a highly conductive water-based gel. It provides an optimal contact between the skin and the electrode, while still being easy to wash off after the experiment.

[21] This system is also used in the Psychology department of Leiden University. The remainder of this chapter is partly based on the reader "Syllabus for the research master course Experimentation II: Neuroscientific Research Methods 2013" written by Guido Band.

12.4.2 Headcaps

There are different caps for different head circumferences. The size is indicated by the colour of the cap and a tag inside the cap. The tighter the cap the better the signal, but keep in mind that the participant needs to be reasonably comfortable with it at all times. In the table below you'll find the cap colours and corresponding sizes.

Colour	Size in cm
green	46-50
yellow/green	48-52
yellow	50-54
red/yellow	52-56
red	54-58
red/blue	56-60
blue	58-62
blue/brown	60-64

You measure the head circumference with a measuring tape. To get the right measuring point, wrap the tape around the head so that it crosses over the 'inion' (the most prominent bone at the back of the skull) and the 'nasion' (the point between the eyebrows).

All caps have 64 recording electrode positions and two extra holders for the CMS/DRL electrodes. In figure 22 you see the electrode layout for 64 electrodes. Note that odd numbers refer to the left hemisphere! Except for electrode name Fp, which stands for frontal polar, all other abbreviations are self-explanatory.

Question 20 - So what do F, C, P and O stand for in the electrode position abbreviations?

12.4.3 Attaching flat electrodes

After you have measured the head circumference it is time to attach the flat electrodes – you do this **before** putting the cap on the head. For monitoring eye movements we attach flat electrodes according to the scheme in figure 23.
Additionally we attach one flat electrode on each of the mastoids, to give us the reference point (see 12.1.2).

Before attaching separate electrodes, you should reduce the electrical resistance (impedance) of the skin by removing grease and dead skin cells from the upper skin

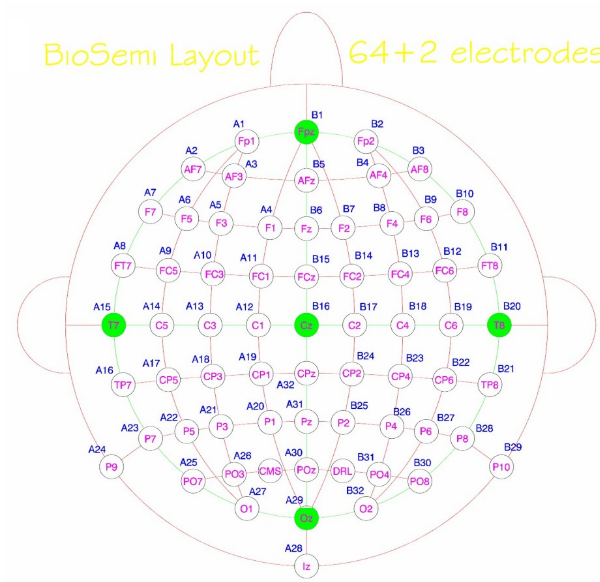


Figure 22: Layout 64 electrodes.

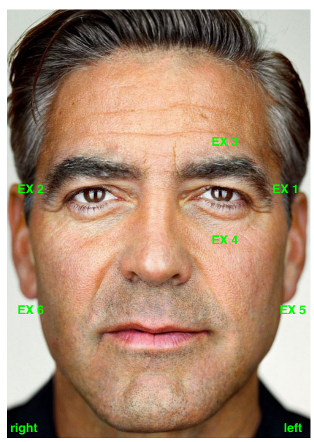


Figure 23: Layout flat electrodes for recording eye movements and mastoid potentials. Looking straight use the participant's pupils as reference for horizontal and vertical midlines. EX-numbers correspond with EX-numbers on the Biosemi box (see 12.4.5).

layer. This can be done with alcohol swabs, or NuPrep gel, which is an abrasive substance. As the skin around the eyes is very sensitive it is best to use an alcohol swab around the eyes. However, you can use NuPrep on the mastoids. Eye movement recordings have a strong signal-to-noise ratio, so they can be obtained without scrubbing the skin. However, the EEG signal behind the ears is weak and needs to be recorded with precision to serve as a good reference for other EEG electrodes. Rub NuPrep gently over the skin surface of the mastoids with one of the cotton tipped sticks.

Flat electrodes are attached with a round sticker. Place it around the grey conductive area, and let the brown lip face outward to facilitate removing the white sticker cover. If the lip is too near the electrode wire, it'll be hard to get to it once it's on the participant's skin!

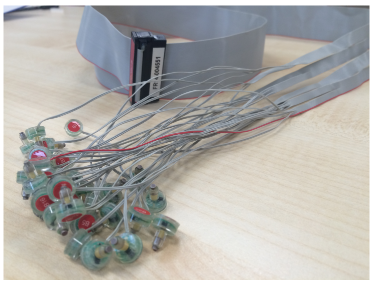


Figure 24: Electrode bundle no. #4551

Figure 25: Flat electrodes and CMS-DRL cable

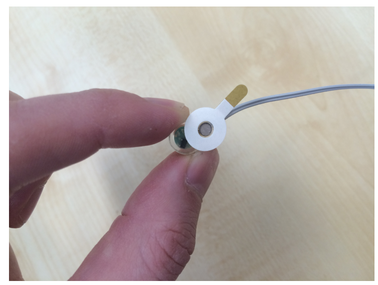


Figure 26: Flat electrode with sticker

Figure 27: Caps in different sizes and velcro straps

Figure 28: Conduction preps

You'll need to fill a syringe with Signa gel. Inject a single drop of Signa gel on the conductive electrode cup, within the sticker ring. **Do not touch the electrode surface with the syringe tip**: it will cause damage and reduce the quality of the signal. Attach the flat electrode, add a piece of Micropore tape to keep the electrode in place and plug in the connector to the Biosemi box which is placed at the back of the participant's chair. Attach the electrodes in numerical order.

12.4.4 Attaching electrode bundles

Participants can help with positioning caps on their heads. The best way is if the participant puts their thumbs inside the inner front side of the cap while you pull it backwards over the head. The front of the cap should land just above the upper eye electrode. Let the participant fasten the Velcro chin strap. Use the Cz electrode position as a reference point to centre the cap on the head: it should be in the middle from inion to nasion, and in the middle from upper right ear and upper left ear. Use a measuring tape to check this!

For preparing the skin underneath each of the electrode positions, you use the tip of the syringe. Insert it through the hole and move it gently through the hair until it reaches the skin. To dislodge dead skin cells and improve the connection, you should move it back and forth over the skin surface, without adding too much pressure (however, while you move it around over the skin, you should feel some resistance to know that you are actually doing something!). After you've gently scraped the skin, you add gel. This should be just enough to make good contact by bridging the space between the skin and the electrode. **While you inject the gel in the electrode holder, press the the holder to the skin.** If you dont do this, the gel will splurge all over the place under the cap and it will create unintended connections between adjacent electrode positions, which will mess up the location and voltage information. While filling the holder, slowly bring the syringe back out of the hole, ending with small blob of gel sticking out of the hole.

Once the electrode holders are filled, you are ready to plug in electrodes CMS ("common mode sense") and DRL ("driven right leg"). These are together used as an initial reference and as ground for recording voltages (as you know from section 12.1.2). So, the potential you will see in ActiView of electrode Cz for example, is the potential difference beteen Cz and CMS.

Once the CMS-DRL cable is plugged in, you can turn on the amplifier and start ActiView to check the activity of the flat electrodes. The blue light on the amplifier should be lit continuously. A flickering light means that one or more electrodes are malfunctioning or the impedance of the CMS-DRL connection may be too high. Also check the offset (`Electrode Offset` tab) especially of the mastoid electrodes. Try to keep it between -15 and +15! If it is too high, take away the electrode, cleanse the skin and attach a new electrode.

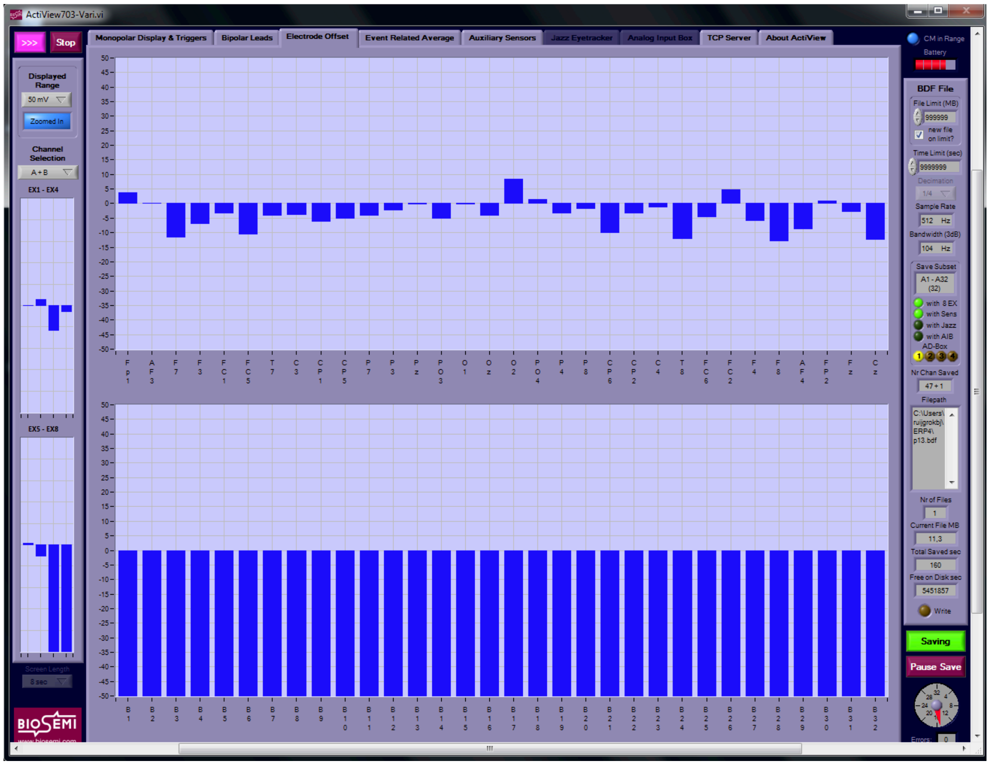


Figure 29: ActiView window showing electrode offset. Full blue bars represent unconnected channels.

Now you are ready to attach the electrode bundle(s). For a 32-channel setup, you should use one electrode bundle and plug it in socket A. In a 64-channel setup, you can use up to two bundles of 32 channels each and plug them into socket A (left hemisphere) and B (right hemisphere). Since the holders are filled with gel, plugging in electrodes in the cap holders should be going smoothly – if not, it is a sign that you forgot to fill a holder.

Now, check the overall signal quality. If all the electrodes are rather noisy (when ActiView is showing dense lines instead of tiny scribbly lines), you can normally solve it by putting some more gel in the CMS-DRL sockets. Also check the individual offsets of the electrodes and try to keep it between -15 and +15.

At this point it's good to show your participant how movements of the eyes, head, and jaw muscles influence the signal (using the `Monopolar Display & Triggers` tab). This way they can fully understand why it's important to follow instructions regarding minimising movement / waiting for breaks in order to move.

12.4.5 The recording setup

You'll be using two computers. On the left computer (which has two screens) you'll run the experiment. On the right computer the EEG data will be saved. In the booth, you can have the screen show either of these displays, by using the screen switch (check back to 6.3). However, make sure that when you turn on the computers, the screen switch is set to 'B', which shows the right-hand computer display. On the one hand, this prevents malfunctioning of screen presentation; on the other hand it is the screen that you'll need first in the booth – the presentation of ActiView to check the connection of the electrodes and to show the participant how their eye movements etc. affect the signal. Remember to then switch it to 'A' to present your experiment.

The signal from the booth to the computer is sent via the BioSemi box. It runs on rechargeable batteries that can be recharged by connecting the box to the charger (these are black boxes which live on the long table by the window). Check the BioSemi battery and change it if necessary. You can see the status of the battery in the upper right corner of a window in ActiView, but also on the box itself. The meaning of the indicators on the BioSemi box is as follows:

- Battery empty: the power light can not be switched on
- Battery low: the indicator is lit
- Battery full: power light can be switched on and the battery low indicator is off
- Charger power is on: green light

The charger also has different indicators:

- Battery is full: green light
- Battery is almost full: yellow light
- Battery is too empty: red light

12.4.6 Cleaning

Once you've finished the experiment, you need to take care of properly cleaning the equipment.

After checking that the participant is ready, remove the electrodes from the cap first, then remove the cap and lastly the flat electrodes. Do this gently! Place the materials to one side in the booth: your participant's comfort comes first. Invite them to stand up and leave the booth. Back in the main lab, offer them the opportunity to wash and dry their hair (there is shampoo, a sink, towels and a hairdryer). Many participants prefer not to go through the hassle of washing their hair in the sink, so offer them a towel anyway that they can use to rub the gel out of their hair. Place the towel in the laundry basket when they have finished.

If necessary, use the Labs meeting point for a debrief. When the participant has left you can start cleaning. Put the electrode bodies (i.e. one end of the wires) and the cap (except for the Velcro straps!) in the plastic washing bowl. Empty the syringe of any remaining gel, and put it into the washing bowl too. **Do not put electrodes on the metal surfaces or directly into the sink.** Fill the bowl with lukewarm (not hot) water and a single drop of Ivory soap (see figure 30).

Figure 30: Cleaning materials: Ivory soap, toothbrushes and interdental brushes.

Use a toothbrush to gently remove gel from the electrodes one by one. Remember that you should hold the electrodes by their plastic body! Use an interdental brush for cleaning the electrode holders on the cap – it fits exactly through the holes. Remove the gel from the electrode holders one by one. Rinse all the materials under cool running water.

You will also need to disinfect the materials, in order to maintain proper sanitary standards. To do this, put them in the prepared Sekusept solution for **exactly** 15 minutes (see figure 31). Put on gloves before taking out the materials from the Sekusept solution and rinse the materials under cool running water before hanging them up to dry. You must handle the Sekusept solution with extreme care as it is a very aggressive chemical solution. Preparing the solution is taken care of by the Lab Manager(s). Before you use Sekusept, make sure you have read the Appendices A, B and C.

Figure 31: Sekusept solution. Handle with care!

12.5 Checklist: Running an EEG experiment

Below you'll find a checklist for running an EEG experiment. Be aware that in order to follow this checklist, you must have carefully read and understood all the instructions in this chapter so far.

Exercise 55 - Using the script provided, work through this checklist with your group to collect and view your own data.

12.5.1 General preparation

- Have ready for the participant:
  - an information sheet and an informed consent for reading and signing
  - a log-sheet for background information of the participant (see chapter 17 section 17.2 for more on the topic of "logging").
  - money and payment administration sheet, if applicable
- Before turning on the computers, put the display switch to **B**,

- Open the task to be presented first on the stimulus PC.
- Open the program ActiView on the right-hand (EEG) PC.
- Check the BioSemi battery and change it, if necessary.

12.5.2 EEG preparation

- Get the following items and lay them on a shelf in the booth:
  - a syringe filled with Signa electrode gel (one syringe should be enough for 32 + 6 electrodes)
  - alcohol swabs
  - paper towels
  - measuring tape
- Prepare seven pieces of Micropore tape, with a length of 4 cm each.
- Collect an electrode bundle (or two for 64 electrode setup), a CMS/DRL cable and six flat electrodes.
- Hang the electrodes in the booth.
- Log the serial number of the electrode bundle.
- Attach the ring stickers onto the flat electrodes.

Arrival of participant

- Welcome the participant in the Labs meeting point. Show the way to the EEG Lab and make him/her feel comfortable by being friendly and calm and briefly explaining what will happen during the session.
- Ask the participant to read the information sheet and sign the informed consent form.

Attaching flat electrodes

- Handle all electrodes with care and do not touch the flat electrode surface with your finger or with the syringe tip as it will cause damage.
- Cleanse the necessary spots of the skin.
- Attach the electrodes (prepared with gel), secure them with a piece of Micropore tape and guide the wires over the ears (or shoulders).
- Connect the electrodes to the BioSemi amplifier box immediately after attachment.
- When attaching the lower eye electrode, take care that when it is on the face it is not intruding into the participant's field of vision.

Placing the electrode cap

- Measure the circumference of the head and select an appropriate cap.
- Attach the Velcro straps to the rings of the cap.
- Position the cap straight and centrally, using Cz as reference middle point. The front edge of the cap should be just above the upper eye electrode.

Connecting the electrode cap

- Add gel in the electrode holder: gently move the tip of the syringe through the holder until you reach the scalp, move the tip gently to prepare the skin, then release the gel while retracting the syringe from the hole.
- Fill all necessary holders in the order of position number.
- Plug in the CMS and DRL electrodes.
- Connect the CMS-DRL cable to the BioSemi box.
- Switch on the BioSemi box.
- The blue light should stay on. If it is flickering, some connection is insufficient or malfunctioning.

- If the blue light stays on, the rest of the electrodes can be plugged in; keep checking the blue light.
- Attach the remaining electrodes.

Inspecting the signal

- Clean your hands before using the computer (always do this when going from electrode handling to using the computer).
- Start ActiView.
- Choose the configuration file for inspection and recording.
- Check the signal and show the participant the effect of eye blinks, head movements and jaw contractions.
- Check the electrode offset and apply extra gel if necessary. Do channels all remain noisy? Consider replacing the CMS-DRL cable.

Recording the experiment

- Start the recording: is the green light on in the ActiView window?
- Start the task.
- Keep an eye on the signal quality and improve it between task blocks, if necessary.
- Log anything that seems important.
- Take a screen shot of the offset window and save it to your data folder.

Concluding the session

- Handle the participant with priority.
- Turn off the BioSemi box.
- Remove the electrodes and the cap.
- Provide the participant with tissues and offer them use of the sink, shampoo, towel, hairdryer and mirror.

- Pay the participant (if that is what you agreed on) and thank him/her. The participant may leave. If you need more time for a debrief, use the Labs meeting point: don't disturb other lab users!
- Back-up the recorded data.
- Fill in the EEG log sheet in Sharepoint.
- Carefully remove tape from the electrodes.
- Wash the gel from electrodes and cap.
- Soak the rinsed electrodes and electrode cap in Sekusept solution for **exactly** 15 minutes; rinse afterwards and hang the materials to dry at the left side of the table.
- Throw away any rubbish.
- Shut down PCs and turn off the light.
- Tidy the room for future use: leave it as you (should have) found it!

Figure 32: Hang wet materials on the left. Move dry materials to the right.

13 Eyetracking Lab: Experiment Builder & Data Viewer

LEARNING GOALS

- carry out an eye-tracking experiment using Experiment Builder
- calibrate and validate the Eyelink setup
- open and view data using Data Viewer

Exercise 56 - Review the material in Chapter 6 to refresh your memory about the eye-tracking lab.

13.1 Experiment Builder

13.1.1 Programming experiments with Experiment Builder

In the eye-tracking lab we don't use E-Prime to present experiments. Instead we use a special software designed to work with the particular eye-tracking camera setup that we have. This program is called Experiment Builder. Experiment Builder is similar to E-Prime in some ways, for example that every experiment has a flow, which you design by selecting elements from a range of available options - for example, a text display, or an image. However, in most ways it's quite different from E-Prime.

Exercise 57 - Watch these videos for an introduction to how Experiment Builder works: Tutorial Video 1 (6 mins); Tutorial Video 2 (12 mins)

Also unlike E-Prime, Experiment Builder is not widely available at the university. In fact, it's only available on three computers in our labs - one in the eye-tracking lab, and the two analysis PCs (PC1 and PC2) in the EEG lab. This means it's not possible for us to give the whole class a practical tutorial on how to use Experiment Builder. Instead we'll provide you with an example script. However, if you're interested in working with eye-tracking in your final project or your bachelor thesis, you can talk to us and we'll try to arrange for you to get some extra training. And of course, the script you work with in this section will come in handy as a base for building your own design.

13.1.2 Opening and finalising the script in Experiment Builder

Exercise 58 - Before you start, you need to have downloaded EyetrackingTutorialScript.zip from https://doi.org/10.5281/zenodo.1339334 containing the Eyetracking tutorial (Experiment Builder files) and extracted it into your own folder on the university network.

You can open Experiment Builder via the Start menu (`All programs... > SR Research > Experiment Builder`). However, for this tutorial you already have a script, so you can launch the program directly by double-clicking the *.ebd* file within the experiment folders. **Experiment Builder automatically creates a set of folders for every script you build. This folder structure must not be changed, renamed or deleted!** When you want to move your experiment to another computer, you'll need to take the whole set of folders with you.

The experiment flow is visible as a diagram in the main part of the window. However, in the structure window on the left, you'll see the experiment flow set out as a tree structure a bit like E-Prime. Below the structure window there are some tabs. `Experiment` shows the tree structure of the experiment, and `Devices` shows you the different pieces of hardware associated with the script. Below the structure window is the properties window. Whatever element of the experiment is currently selected, its properties are shown in this window (and can be edited).

It's important to check that all these hardware devices have the correct settings, so that everything works properly with the Eyelink system. To check the settings of the Eyelink camera, you click the `Devices` tab and then select `EYELINK` at the top of the tree structure. The properties window will now show the properties for the Eyelink system. For our setup, the following settings are required:

Tracker Version:	Eyelink 1000
Camera Mount:	Desktop
Desktop Version:	Illuminator on Right
Mount Usage:	Monocular Remote

While building your experiment, you can test run it by going to `Experiment > Test Run`. However, this still requires you to have the eye-tracker set up. Often, you just want to check the script is working OK, without using the eye-tracker. In the properties for the Eyelink system, you'll see a check box labelled 'dummy mode'. Ticking this box means that *Test Run* will just test out the script itself. Make sure you don't forget to un-tick it again to do a real test run. Whenever you're testing your script, keep an eye on the main window for any error messages!

As with E-Prime, when you've finalised your experiment, you will get the program to create a stand-alone script so that you can run it over and over again without having to open the Experiment Builder program itself. This helps you avoid accidentally changing something in the script while running your experiment. In E-Prime this is called *compiling* the script; in Experiment Builder it's called *deploying* the script. Running a deployed script also doesn't depend on the dongle being present.

When the script is ready, you need to deploy it to create the stand-alone version. You do this by going to `Experiment > Deploy`. You'll be asked to give the experiment a name and choose a location for it to be saved. This is different from the set of folders where you saved your unfinished script. The deployed script is an .exe file and can be launched by double-clicking on the file (not in Experiment Builder!)

Exercise 59 - Find and open the script (.ebd) file for this tutorial and complete the following:

- **Check the hardware settings are correct**
- **Find the display screen that participants see at the end of the whole experiment, and input your own goodbye message**
- **Test run your script in dummy mode (tip: press Escape to skip through the trials)**
- **Deploy your script**

Before the participant arrives, you can already start the experiment running (ie. open the deployed version). When the experiment runs, it will first ask you to fill in a name for the data file (as you have seen during the test runs). Naming is in principle up to you, although when running a real experiment, you should name the data file systematically with a number that is different for each participant, e.g. data01, data02, data03 and so on. The filename must in any case have .edf at the end.

13.2 Carrying out the experiment

When the participant arrives, welcome them and aim to make them feel at ease. Ask them to read the information sheet and sign the informed consent form.

Before you invite the participant into the testing booth, you need to find out their dominant eye. You already know that you have a dominant hand and a dominant leg - well, you also have a dominant eye! You can find out which of your eyes is dominant by doing this test:

1. Form a circle with the fingers and thumb of one hand.
2. Look through this circle at a particular point on the wall with both eyes open.
3. While looking through the circle, use your other hand to cover first one eye, and then the other eye. What you will find is that covering one eye doesn't make much difference to what you see through the circle, but covering the other eye makes the point you are looking at suddenly jump to the side.
4. This eye (the one that is covered when the point jumps) is your dominant eye.

Ask the participant to do this test, and tell you which eye is dominant (if in doubt, ask them to tell you when the point jumps, and watch to see which eye they have covered). This is the eye you will track. Now take one of the little black and white "target" stickers and place it above the dominant eye. It should be a little bit nearer towards the middle of the face than the pupil, and a little way above the eyebrow. If the participant's hair covers their forehead, you may need to ask them to clip it back or use the elastic hairband provided.

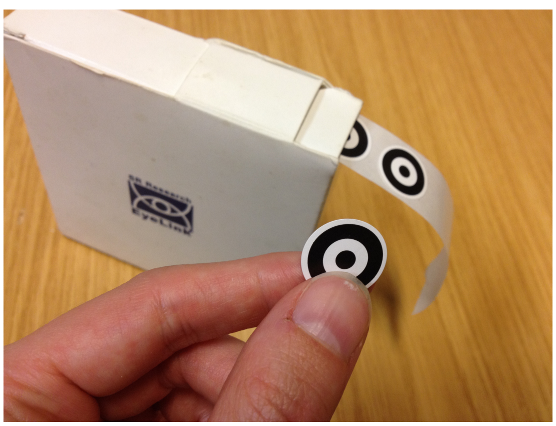


Figure 33: Eye-tracking target stickers.

Figure 34: Cleanser and hairband.

Ask the participant to sit comfortably in the booth and look directly at the screen, and rest their chin in the headrest (if you're using it). The participants' eyes should usually be about level with the middle of the top part of the display screen. Make sure they are sitting comfortably (because they are going to have to hold this position during the whole experiment!)

13.2.1 Camera setup

Now go back to the host PC and look at the Camera Setup screen (Figure 35). First, click *Set Options* and double-check that the correct camera setup has been chosen (Figure 36). If everything's ok, click *Camera setup* to return to the setup screen.

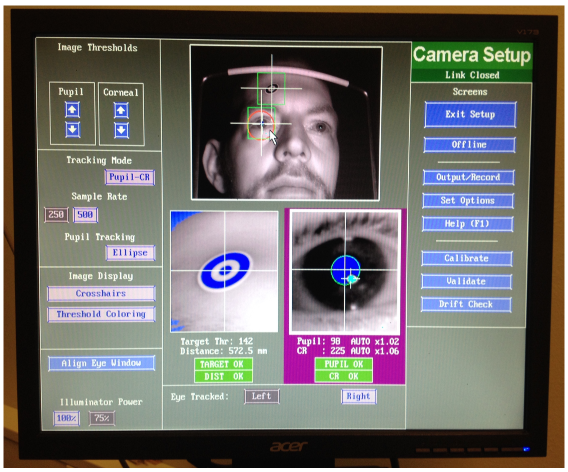


Figure 35: Camera setup screen.

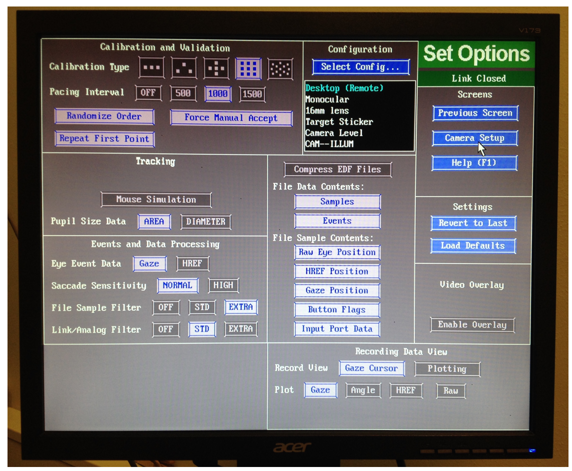


Figure 36: Set options screen.

The tracked eye should appear in the middle of the overview camera image. If the camera is having difficulty finding the pupil, you can use the mouse to move the red circle to the eye in the overview image (see Figure 35). You can also try clicking *Align Eye Window*. Now that the participant is sitting in the right position, you need to check that the camera can properly detect and track the pupil. Make sure that the correct eye is being tracked (the dominant eye with the sticker above!). In Figure 35 you can see this is the participant's right eye. If the camera is trying to track the wrong eye, select the correct eye using the buttons at the bottom of the setup screen.

If really necessary, you can even put a book or two under the camera for better tracking. But you should make sure that there is appropriate distance between the camera and the eye (about 600 mm) and the camera is not so high that it blocks the participant's view of the screen. (You should check the distance by looking on the camera setup screen for where it says *Distance* - in Figure 35 you can see it's about 591 mm.)

If the image is fuzzy, you may need to focus the camera by gently turning the top edge of the camera lens. Do not touch the actual lens, only the sides! The lens is focused when the image is crisp and you can make out the eyelashes clearly.

The way the eye-tracker works is by using the reflection of infrared light on the cornea (the turquoise colour in the camera setup images) and the absorption of infrared light in the pupil (the blue colour in the camera setup images) as measuring points, to calculate the gaze direction. Both of these points should be within the green circle that appears on the image of the tracked eye. To optimise the tracking, you can first press A on the keyboard to auto-adjust the camera image.

Now inspect the image of the eye on the Camera Setup screen. The basic aim is to get all four of the boxes under the eye and target sticker images to be green (as in Figure 35). With most participants, you'll need to spend some time making adjustments to make sure the eye is properly tracked. At first, it can be difficult to figure out how to get it to work. Figuring out what adjustments need to be made gets easier with experience. Here are some general points of advice:

- Dark eye-makeup and glasses can interfere with the reflections the eye tracker uses to calculate eye position. When we recruit participants, we advise them not to wear too much makeup to the session, and ask them to wear contact lenses rather than glasses if they have the option. Otherwise, there are some products in the lab for removing eye makeup if the participant is happy to do so (Figure 34), but with glasses it is usually necessary to just go ahead and try and track the eyes with the glasses on. As always, the priority is that your participant is comfortable - they are not obliged to continue if they are not happy with the setup.

- On the top left of the camera setup screen (see Figure 35) you will see up and down arrows for adjusting the *Image Thresholds* for the pupil and cornea. Basically this adjusts the sensitivity of the image to the two different types of reflection, which can improve the tracking. For example, if the colour for the pupil (dark blue) seems to be all over the eye, click the down arrow for pupil to reduce sensitivity. But if the dark blue doesn't even fill the pupil, click the up arrow to increase sensitivity. Likewise for the corneal reflection.

- You should be concerned if there's a large lighter-blue circle on the image of the eye: this means there is too much reflected light. If this is the case, you can try various adjustments, such reducing the illumination (see *Illuminator Power* in Figure 35). If that isn't enough, you can dim the lights in the lab using the dimmer switch, and turn off any lights that are on in the booth. You may need to do this particularly when the participant is wearing glasses.

Now, to confirm everything is well set up, ask the participant to look at the four corners of the screen. Make sure the tracker is able to keep track of the pupil while they look at each of the four corners, and that there isn't too much extra corneal reflection (watch the boxes underneath the eye and sticker images, to see whether they stay green). If this is fine you can move on to the calibration. If it is not fine then review all the adjustments and try this step again.

13.2.2 Calibration, validation and drift correction

Calibrating is the process of checking that the computer and the eye-tracking camera agree with each other about where the participant is looking. For example, if I'm looking at the top of the display screen, it's no good if the camera thinks I'm looking at the side of the screen - then if we run an experiment, the data about what picture or word I'm looking at will all be incorrect. The way the system calibrates itself is by displaying individual dots one after another on the screen, which the participant needs to look at. If the eye position calculated by the camera matches up with the location of the dot that is known from the display, everyone's happy and we can get some good data.

To start calibration, click on *Calibrate* on the host PC screen. You need to ask the participant to look at each dot as it appears on the screen (and not to look away!). For each dot, when the host PC says that the participant's fixation is stable (you'll see the bar on the right turn green and the word STABLE will appear), you should click *Accept fixation*. Then repeat for the next dot, and the next, until finished.

The points should now form a grid. You'll also get a message below the black area of the screen, telling you if the calibration was good enough. If there's a problem, you'll need to go back and adjust the setup and try again.

When calibration is successful, you can begin validation (click on Validate). Validation double-checks the accuracy of the eye position calculation. It's pretty similar to the calibration procedure: you need to click every time the fixation is stable, and again you get a message at the end telling you how successful the test was. As ever, if there's a problem, you need to go back to the setup stage (section 13.2.1), make some adjustments and then go back through the whole calibration and validation procedure again.

You may find that the most difficult points to calibrate and validate are in the corners of the screen. It's more difficult for the system to correctly calculate the eye position when looking in the corners. Due to this common issue, it's better to avoid presenting stimuli in the corners of the screen.

The way we typically design our eye-tracking experiments is that before each trial where there are eye movements being recorded, there is a *drift correction* screen. This is similar to the calibration technique - a dot appears on the display screen and the participant has to look at it, and you as experimenter have to click to accept the fixation on the host PC, before the participant can continue with the trial. The point of this is to make sure that the eye-tracker stays well calibrated throughout, so we don't find out later that things got out of alignment and the data got ruined.

13.2.3 Running the experiment

When the setup is all done, you can begin the experiment. Usually you do this by pressing the `Escape` button on the experiment PC keyboard to start the experiment. This is a good point to check the participant is happy and let them know they are about to begin the experiment. When you know they havent got any more questions, you can shut the door on them and they can start.

While running the experiment, you need to be alert throughout because you have to be ready to click to accept the participant's fixation whenever there's a drift correct screen. In fact, it's good practice in general to stay focused and pay attention while participants are doing your experiments (even if you are extremely bored after the 10th participant!), so that you're aware if participants have misunderstood the instructions or something is not going right with the experimental script.

13.2.4 Checklist for running an eye-tracking experiment

This is a checklist for carrying out an eye-tracking experimental session. For more detail on any of the points, look back over the previous sections.

Exercise 60 - Using your finalised script from Exercise 59, work through this checklist with your group to collect your own data.

General preparation

- In the booth, switch on the power on the extension cable (on the floor) to start up the booth equipment. Take the lens cap off the eye-tracking camera.

- Turn on the experiment computer.
- Turn on the host PC. During startup, press Enter to start up in Eyelink rather than Windows.
- Start the experiment by double-clicking the deployed version. Enter the name for your data (.edf) file.

When the participant arrives

- Welcome the participant and ask them to read and sign the informed consent sheets.
- Find out the participant's dominant eye and place the target sticker above that eye.
- Invite the participant to sit in the booth. Their face should be about 600 mm from the camera with eyes roughly at the height of the top of the display screen.
- Check that the participant is sitting comfortably enough to stay still for the experiment.

Camera setup

- View the Camera Setup screen on the host PC.
- Check the Set Options menu to ensure the right camera settings.
- Check the alignment of the eye image and that the correct eye is being tracked.
- Make adjustments to the image as necessary to ensure good tracking (this may involve some trial and error).
- If problems seem to be caused by heavy eye makeup or glasses (and can't be solved by adjusting the settings) explain the issue and ask if there is any possibility of removing makeup/glasses for the session.
- Confirm that tracking is reliable by asking the participant to look at each of the four corners of the screen. If there are problems, review the previous steps and try again.

Calibration and validation

- Instruct the participant about the calibration and validation procedure (i.e. about looking at the dots).

- Do the calibration procedure, then the validation procedure. If there are issues you may need to go back and make more adjustments before trying again.

Running the experiment

- Check that the participant is happy and ready to begin. Press Escape to start the experiment.

- Shut the door of the testing booth.

- Pay attention to the drift correction screens where you need to click to accept participants' fixations.

- When the experiment has finished, open the door for the participant and allow them to exit the booth. Ask them if everything was fine (i.e. for feedback about the procedure).

Closing

- Thank the participant and make sure to tell them to remove the sticker from their head before they leave!

- In the booth, put the lens cap back on the camera. Turn off the booth equipment using the switch on the extension cable.

- Shut down the experiment PC, and turn the host PC off by pressing the power button on the front of the computer.

- Make sure to tidy up, turn off the lights and lock up after yourself - the lab must be left as you (should have) found it!

13.3 Inspecting the data with Data Viewer

Once the experiment is over, the data is saved automatically as an *.edf* file with the name you chose earlier. You'll be able to find your edf file in the `results` folder which has been automatically created within the project folder. If you double-click this file, it will be opened with Data Viewer, which is the data processing software associated with the eye-tracker. In Data Viewer, you can inspect your data and export it in *.csv* format, which you can later open in a statistical analysis program.

The pre-processing you need to do and the way in which you export your data may vary depending on the kind of experiment you are doing. Therefore, in this tutorial we will not go into depth about the pre-processing and exporting of the data (extra training is possible if you wish to work with eye-tracking for your own projects). Instead, we will go over some ways to take a look at the data you've just collected, to give you a feel for the kind of eye movements that different stimuli can elicit.

When you open your edf file in Data Viewer, you will see a screen like Figure 37, showing the image that was seen by the participant, with various light blue circles scattered around on top of it. These circles represent the fixations – the larger the circle, the longer the participant looked in that area. You'll also see a lot of buttons. (Whenever you're unsure what a button is for, you just need to hover over it with the mouse and a label will pop up.)

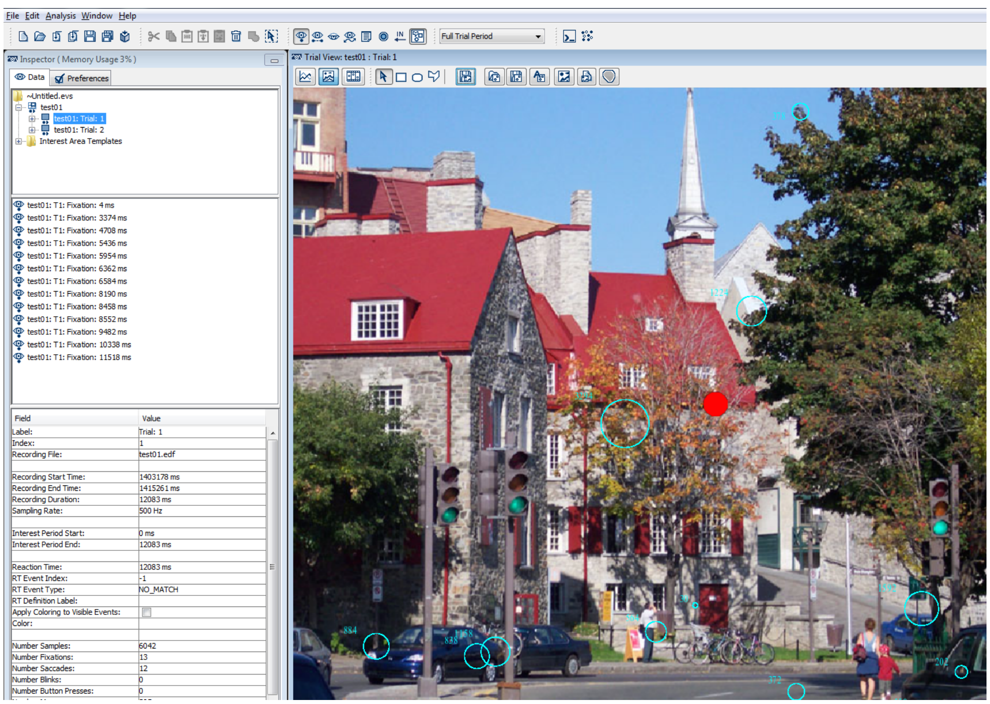


Figure 37: Taking a look at fixations in Data Viewer.

At the top left, the trials are listed, and below that the individual fixations per trial are listed. When you select a fixation on the left, the corresponding circle will be lit up on the image. Using the up and down arrows on the keyboard, you can go through the fixations in the order they occurred.

Another way to see the fixation pattern is to play back the trial as a movie. Above the main image, you see a row of buttons. Click the button third from the left (labelled `View Trial Play Back Animation`). A set of playback buttons will appear on the row – click the play button to view the trial. To return to the fixation view, click the second button from the left (*View Trial Data Spatially with Image Overlay*).

The row of buttons also gives you the option to save the image with the fixations on it (button labelled `Save Trial View as Image`), or you could even save the movie file (`Save Trial to Video File`).

Exercise 61 - Open your data file in Data Viewer and have a look at the fixations for each trial. Can you find out how to view the path of the eye movements as well as the fixations?

As we said at the beginning of this chapter, you must never change or rename the folders in the folder structure that's created automatically by Experiment Builder. Whenever you are designing, running or analysing experiments, you should *always* keep up-to-date backup copies of your work. For Experiment Builder this means you always need to copy the whole set of folders!

Exercise 62 - To conclude this section, copy and zip your Experiment Builder experiment files.

14 Baby Lab: E-prime & ELAN

LEARNING GOALS

- set up and run a testing session in the the Baby Lab
- annotate and code video data using ELAN

The Baby lab is set up to use E-Prime. You have already learned about E-prime in Chapter 8; when there are special E-prime settings related to the Baby Lab, you'll find them included in the text of this chapter. As noted in Chapter 6, researchers in the Baby Lab sometimes use a special program called LOOK (created by the Infant Lab at Lincoln University in the UK) instead of E-Prime. In this chapter you will also find an overview of how to use the LOOK program (section 14.4).

14.1 Setup and Hardware

The figure below shows the Baby Lab experimenter's desk.

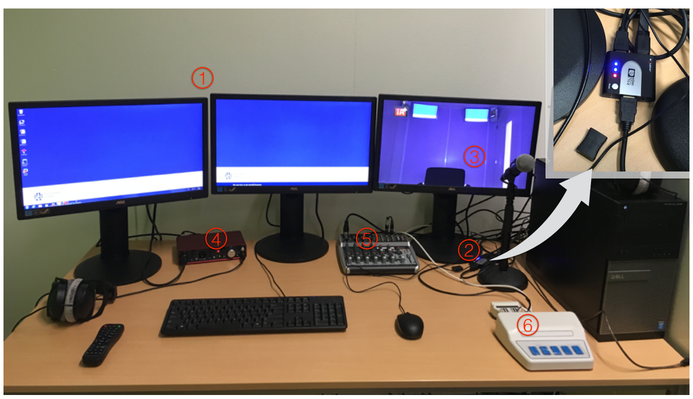


Figure 38: Layout of the Baby Lab experimenter's desk

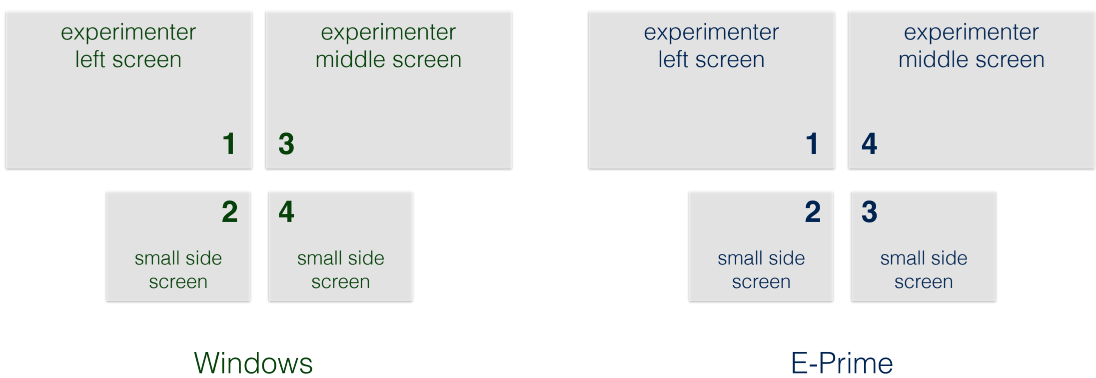
1. Experiment screens (left and middle)
2. Screen splitter
3. Booth monitoring screen
4. Booth mic audio controller
5. Audio console
6. Button box

14.1.1 Experimenter's screens

The left two computer screens (1) are connected to the computer on the desk; the right hand screen displays what's happening in the booth. A splitter (2) connects the computer with both the middle monitor and the large TV screen in the booth.

In order for the middle computer monitor to be mirrored in the booth, it should be connected to the splitter. If you wish to use the middle screen of the computer, but do not wish to mirror your screen in the booth, you can unplug the two HDMI cables from the splitter, and join them via the small adapter provided (see inset picture). This connects the middle monitor directly to the computer, ignoring the TV screen in the booth. But remember to always put the connections back as you (should have) found them before you leave the lab!

If you are using E-prime to create and run experiments, be aware that E-Prime codes the number of the screens differently from Windows! Below you'll find the key for the two screen numbering orders:



This means that you will need to make sure the displays are matched: Display 4 of Windows should be called Display 3 in E-Prime, and vice versa. You can modify this from E-Prime by going to `Edit > Experiment > Devices` and selecting the relevant display.

The screen on the right of the experimenter's desk (3) is connected to the video camera in the booth, and is not connected to the computer. Its function is to allow you to observe and monitor what's going on inside the booth.

14.1.2 Inside the booth: TV screen and camera

In order to capture how the experiment is being presented in the booth, it is important to turn on both the TV screen and the video camera. You can turn on the TV by using the remote control. Walk into the booth and press the green "on" button. If it does not mirror the middle screen, walk to the desk and press the Reset button on the HDMI splitter.

You can turn the camera on and off by using the button on the right side on top of the camera. Make sure that it makes a beeping sound when you turn it on and off to be sure the camera responded. Also make sure that the camera is capturing the back wall mirrors, which is how you see what the infant is seeing, and that you can see the infant's face well. See below on information on recording.

14.1.3 Sound

From booth:
Sound from the booth reaches the experimenter via the video camera and is output via the screen on the right (3). As mentioned above, this screen is not connected to the computer, only to the video camera. In order to hear what's happening in the booth, you can plug headphones directly into the right screen (left under-side of the screen), or you can unplug them for the audio to be heard through the screen's speakers. It is recommended to use headphones, so that the lab is quiet and you can be impartial to the sounds being played during the experiment by simply taking the headphones off.

To make high quality audio recordings, a microphone can be connected to the audio cable inside the booth. This audio cable is not connected to a microphone by default (this is to keep distractions and extra equipment to a minimum). It is rolled and hanging on the left side under the TV screen. The other end of the cable is connected to the red audio controller outside of the booth on the experimenter's desk (4), which allows for recordings to computer software such as Praat, Audacity, etc. This is comparable to the methods used in the Phonetics lab.

To booth:
There is an external microphone on the desk through which the experimenter can communicate with the baby's parent (or adult participant) via the speakers in the booth. In order to speak to the participant, use the audio console (5): slightly turn the left bottom dial up from *-infinity*.

The volume of experiment audio within the booth can be controlled in several ways:

1. Windows sound control

2. Sound control within stimulus presentation software (e.g. E-prime)

3. Audio console (5). The right-hand up/down slider should be kept at a default of 0 db, which is standard. Infinity is silent. If you want to turn up the volume slightly from the slider, it should go upwards.

There is also an amplifier in the booth (right side under monitor). It has a very sensitive dial. Left is silent, right is maximum. The dial should be fixed somewhere slightly right of silent, and should ideally not be touched! However, if you experience problems with audio in the booth, it is worth checking that this has not been tampered with.

Note: If you are using E-Prime, you need to make sure codec config is set up, such that you will not get a buffer error for sounds or videos. E-Prime normally asks you if you want to load the default settings, and you just click OK. This allows it to play audio and video files (see section 8.5.7 for further help if movies won't play).

14.1.4 LED lights

Inside the booth you will see LED lights which are used to draw the attention of the baby during the test: two at the sides and one at the front (see Figure 39). The front light is usually green and is used to draw the baby's attention between trials, to assure attention to the experiment (the "attention grabber"). The Headturn Preference paradigm relies on these three lights.[22]

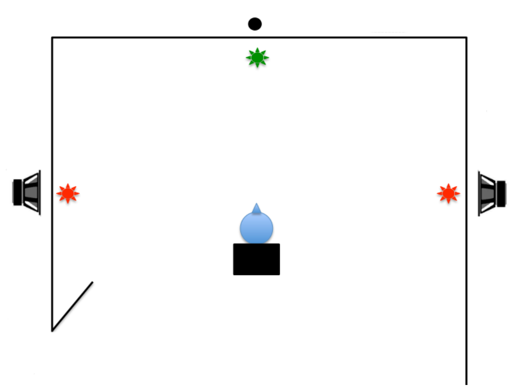


Figure 39: Diagram of the LED lights in the booth.

[22] Currently, these lights are only being used with E-Prime.

The side lights operate hand-in-hand with the speakers from which our target audio plays. During the familiarisation phase of the experiment, whatever it is that we want the baby to learn is played from both speakers for a set amount of time. During that time, the lights should play if the baby looks at them, so she/he understands there's a contingency between his/her looks and the lights. The lights should stop when the baby looks away from them, but the audio keeps playing. Before the test phase begins, the front LED light acts as attention grabber. Then, during the test phase, one of the side lights begins to play to get the baby's attention. When he/she looks, the experimenter will press a button, and continue pressing it for as long as the baby looks. While the button is pressed (indicating looking), the script will play one of the test sequences. During this time the side lamp still keeps going together with the sound, for as long as the baby looks. This ends when either:

- the baby looks away for more than two seconds. When the baby looks away, the experimenter lifts his/her finger from the keyboard and the sound and light stop.
- the maximum duration (usually 15 seconds) is reached. Then the script simply goes directly to attention grabber automatically.

The LED lights are connected to the computer via the ports of the button box and they are controlled via E-Prime. There is no on/off switch for the lights or the button box. Instead, an initiation code needs to be included at the beginning of the E-Prime script in order to turn the lights off and make them ready for use. This code needs to be included in all E-Prime scripts using the LED lights. See the initiation code below:

```
'Ensures all the LEDs are off before beginning
'
writeport &h2040,0
Chronos.DigitalOut.WriteInteger 0
```

In order for the LED lights to be recognised, the button box (Chronos box) needs also to be added to E-Prime via `Edit > Experiment > Devices > Add... > Chronos`. For the button box to be recognised, it must be connected to the computer (default). The Chronos box must also be added to each item properties box in E-Prime as an acceptable response (as is the case for keyboard, mouse, etc).

Note that if you design your experiment on another computer where this device is not connected, you will still need to come to the lab to add it and test the script including this hardware.

14.1.5 Using the video camera

Recording

Once the camera is on (see above), you can turn the recording on and off by using the round, watch-like remote control. Use the button with the red dot to start recording, and the button with the red square to stop.

Always make sure that the camera makes a beeping sound when when you start or finish recording. Only then can you be sure the camera has actually responded to the remote control! If the camera does not respond to the remote control, it is possible to start and stop the recording using buttons on the camera itself.

It is very important to ensure that the camera's battery is sufficiently charged before using it; if the battery is too low, it may not turn on at all or it may switch off during the recording. You can charge the battery by using the transparent, grey USB cord next to the computer and plugging this into the front of the computer. Make sure the camera is turned off when you charge it; when the camera is turned on and the USB cord is plugged in, the computer recognises it as an external drive (see below).

Extracting the video file

You can extract the video file from the camera as follows:

1. Plug the USB cord into the computer while the camera is turned on. You will see Removable Disk (F:) appear in Windows Explorer.

2. The video will be in the VIRB folder in the DCIM folder on the F:-drive. Copy it to the desired location (the Babylab workgroup, for example) and delete the file from the VIRB folder.

14.2 Carrying out the experiment

14.2.1 Making appointments

The protocol for contacting and recruiting infant participants is slightly different from how we recruit adult participants in the other labs. It works via a database of babies who have recently been born in Leiden, whose parents the Baby Lab researchers have been able to approach thanks to the cooperation of the Gemeente (city hall).

The database is contained in an Excel file which can be found in the Baby-Lab workgroup (under `Algemeen > deelname > Database Babylab`). **The information in this database is highly confidential and should not be copied or**

shared under any circumstances. The database is also used by researchers at the Social Sciences faculty. This means that all details about appointments must be carefully maintained in the database file! Use the file as follows:

1. The baby's information is displayed in the first few columns and the parent's information is displayed in the last few columns. Select the baby you want to contact according to your target age group.

2. Update the database when you have called parents and/or made appointments. If you add any comments in the database, always specify the date including the year. Input dates as DD-MM-YY to get the infant's age to display in the column.

3. If you are making an appointment, write down the date of the appointment in the column labeled *afspraak xx mnd LING* and the time in the column labelled *Tijd* next to it. The *xx* refers to the infant's age in months and should correspond to your target age group. When you input the experiment date in the proper format, it will calculate the exact age in the column *Leeftijd afspraak*.

14.2.2 Before the participants and their parents arrive

Setting up for an experiment involves preparing the room to make sure it is clean, presentable, and safe for infants and parents. Make sure you have water in the pitcher to prepare coffee or tea and of course, all necessary paperwork. In addition, it is important to prepare your experimental files before the participants arrive and make sure all hardware and software is functioning properly. It is advisable to arrive 30 minutes before your scheduled appointment to prepare, troubleshoot and charge necessary equipment.

1. Turn on the lights in the booth and dim them to the desired level.

2. Plug the USB cord into the computer to charge the video camera and turn it on for recording after about 15 minutes.

3. Check that the video camera is capturing the back wall mirrors, which is how you will see what the infant is seeing (necessary for running the experiment and for coding the video).

4. If using screens, make sure they are on. If using LED lamps, they are controlled via E-prime.

5. If running an auditory experiment, make sure you have some masking music ready for the parent on an external device (phone, iPod, tablet). Masking music should be non-rhythmic but contain lyrics to mask the speech of the experiment. If running a visual experiment, there are blinded sunglasses in the drawer that the parent should wear.

6. Run your script to check that there are no unexpected bugs or hardware malfunctions.

7. Make sure the room is baby-friendly: that means it should be clean and free of all unnecessary or unsafe objects!

Good preparation is essential because if something malfunctions, and you cannot fix it, or if it takes too long to fix it, you may risk losing your participant! If you find something is malfunctioning before the participants arrive and it can't be solved, you then at least have the chance to call them to reschedule.

14.2.3 Welcoming the participants and their parents

When the parent and child arrive, make them feel comfortable. Be friendly and try to interact a bit with the infant. Offer to take their coats, invite them to sit at the table, and offer them tea, coffee, or water. While you prepare their drink you can chat to them and ask them to read through the information sheet. Then, ask them to fill in the questionnaire and sign the consent form.

Next, instruct parents on how to behave during the experiment and the expectations you have of the baby's behaviour (for example, in a preference experiment, it is often the case that baby will get bored; this is normal! We want to know what the baby is and is not interested in). There is a document in the Baby-Lab workgroup `Taalverwervingslab > formulieren > Guidelines for Parents` (EN and NL), which you can use as a rough script for the information and instructions you should give parents. The bottom line is to make parents feel comfortable and secure, and to make sure that they don't interfere with their baby's behaviour in any way. A stressed parent will make their baby stressed as well! And if the parent interferes, his/her attention will always be more interesting to the baby than the stimuli, and the data will not be usable. Let the parent know when and to what extent they can interact with their infant if necessary (at certain points in the experiments, such as during the attention grabber, and only non-verbally, etc.). Let them know that you are watching the experiment (and whether the infant is unhappy or crying, etc.) via the video camera, so if at any moment they want to stop, they can do so by simply waving to you.

Before going in the booth, ask the parent if they need to feed or change their baby so that he/she is comfortable and happy before the experiment begins. Once everyone is ready, you can start.

14.2.4 Beginning the experiment

1. Lead the parent with their baby into the booth.

2. Lead them to the chair and ask them to seat their baby on their lap facing forward.

3. Place the headphones with masking music on the parent's ears, or the blinded sunglasses on the parent's face. For the music, give them a moment to adjust and make sure the volume is not too loud for them. For the glasses, also give them a moment to get comfortable, and make sure the infant is not too distracted or disturbed by them. You can instruct the parents to simply remove and hide the glasses if the infant becomes distracted by them during the experiment.

4. Leave the booth and close the door. As the booth has windows, turn off the light in the main room when you exit the booth, so that the light from the room does not illuminate the inside of the booth.

5. Walk back to the experimenter's computer and start your experiment.

It is important to proceed in a calm fashion. It's always worth the effort to take a few moments to ask yourself if you have pressed the "Record" button, and have not forgotten anything: infant participants are hard to come by and easily lost!

14.2.5 Running the experiment

A variety of experiments are possible in the Baby Lab. Here we take a typical **Visual Fixation** procedure, run in E-Prime, as our example:

1. Select and run your script. The experiment starts with attention grabber lights.

2. When the baby is looking at the light, hold your specified response key down. A timer function in the script counts to 2 seconds before beginning the stimuli. Normally it will be the familiarisation phase that begins. Depending on whether you want the familiarisation to be continuous or attention-dependent, you can design it so that you don't have to do anything during the familiarisation, or that you respond to the infant's looks.

3. After the familiarisation, an attention grabber can be used before the test phase begins.

4. The first test trial can be set to automatically begin after the attention grabber criterion has been reached. When the test trial begins:

   (a) Lift your finger from your response key quickly, and then hold it down again if the baby is looking during the trial. You must lift your finger from the key at the beginning of the trial because the timer reacts to changes in key presses per process (within the trial and within the attention grabber). So the response key should be first lifted, then pressed to indicate looking, and lifted again to indicate in the infant is not looking during the trial.

   (b) When the baby looks away during the trial, lift your finger from the response key as soon as the baby looks away. If you lift your finger for more than 2 seconds, the trial will stop, and the attention grabber will come back again. If the baby looks back before 2 seconds, hold the response key pressed again. Think of your finger as an extension of the baby's eyes.

5. The trial length should be set in the script to your desired maximum duration. When this criterion has been met, the experiment should proceed to the next attention grabber, then to the next test trial, and so forth.

6. Repeat until the end (in E-prime you can include a goodbye message for the parent in your script).

A **Headturn Preference** experiment is run in a similar way, except that in this case you can interact with the program by using multiple response keys instead of just one for looking or not looking (for example, left, right, and up arrows).

7. Once the experiment is finished, escort the parent and infant out of the booth, tell them that the infant did very well and you got enough data (even if you did not, mention that you can still use some of it so as not to let them think it was

useless!). How you compensate participants is up to you and your supervisor, but typically we reward participation with a baby book, reimburse travel expenses, and give them a certificate with a polaroid photo of the baby in the lab attached.

8. Upon completing your experiment, input your questionnaire information into your subject sheet and file the hard copies (they are usually stored in the drawer below the desk next to the experimenter's desk). Don't forget to rename your files clearly and consistently and store them on the Workgroup.

14.3 Analysing the videos in ELAN

We code the videos using ELAN, a video annotation software developed by the Max Planck Institute. This program allows us to categorise behaviour frame by frame. In the case of the Visual Fixation paradigm, it amounts to a (very) "poor man's eye-tracker", because we code the infant's eye movements to and away from the screen. In the case of the Headturn Preference paradigm, we code their whole head movements to middle, left, and right. Note that if you're going to use ELAN in Lipsius room 1.26, you'll need to plug headphones in to the desktop computer otherwise the video won't appear.

14.3.1 Annotating the video

In the image below, you can see a number of different "tiers" on the left side of the screen (*Trial*, *AttentionGrabber*, etc). Most of the tier names refer to different types of (looking) behaviour.

The annotations are the segments marked by black horizontal lines. On the *Trial* tier, you can see for example "Trial1" to mark when the trial is taking place, and on the behaviour tiers "Trial1a", "Trial1b", etc. to mark when the baby is looking where during the period of that trial.

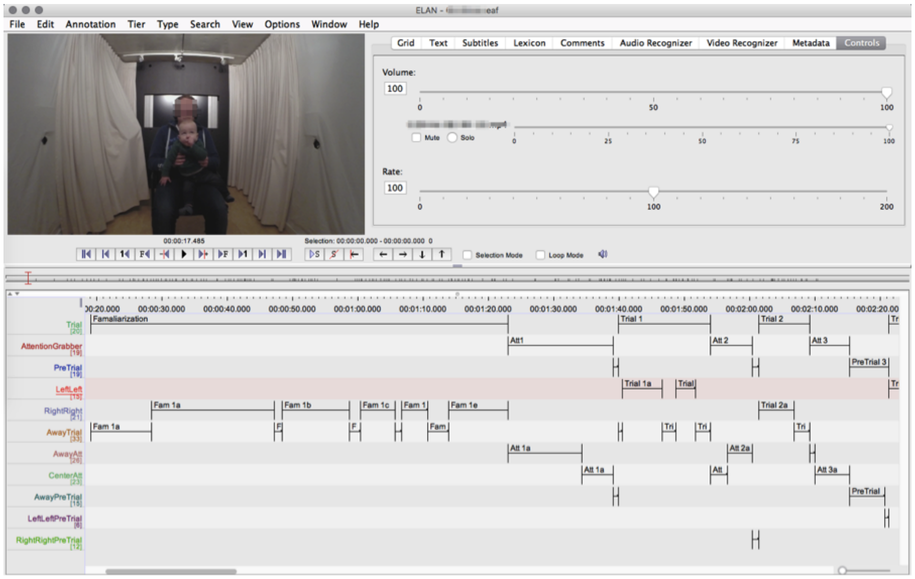


Figure 40: Annotating video in ELAN.

Example tiers for coding are available as .etf files. You can load these into ELAN and on each relevant tier, select a moment of behaviour, and create a new annotation (`Annotations > New Annotations`, or `ALT+N`). When you do this, you will be able to write in text to label it in a way that is meaningful and consistent for your work - for example, with the trial number, as in Figure 40.

In order to get a good time resolution, you can zoom in using the slider on the bottom right of the screen. Zooming in all the way gives you frame-by-frame precision. This level of precision is very important as the differences in looking times between stimuli are often small on average (meaning that you could potentially miss a significant effect if you do not code precisely).

You can find all useful keyboard shortcuts for navigation, annotation, etc. under `View > Shortcuts`.

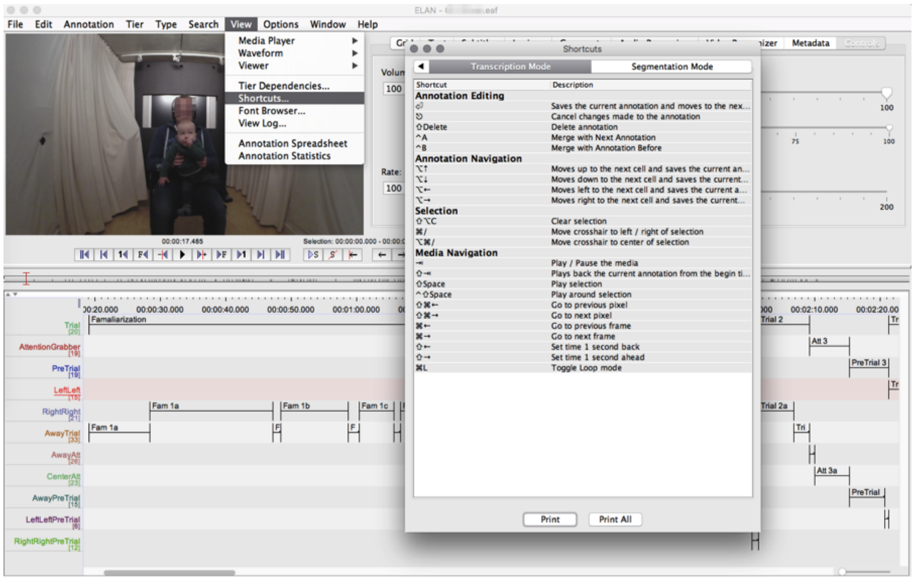


Figure 41: Keyboard shortcuts.

14.3.2 Exporting the data

Once you've finished coding your video, you can export the data as a "tab delimited" text file to be read in Excel or the software of your choice, where you can choose to have every tier as a column. Go to `File > Export As > Tab-delimited text` and select the following in the dialog box:

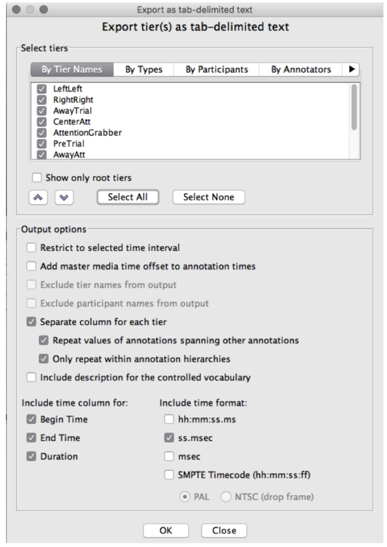


Figure 42: Data export menu.

These settings are customisable and depend on your experiment and what you want. Once you have exported your file, you can use your data points as is necessary for your analysis.

14.3.3 More resources for ELAN

There is a manual in which you can find more information about many options available to you in ELAN via `Help > Help contents`, or downloadable from the Max Planck Institute Language Archive tla.mpi.nl. The MPI site also offers a how-to guide and an online forum for ELAN. You can also reach this from ELAN via `Help > Visit the ELAN website > Forum`. If you run into bugs or have trouble with something, check there, or ask your own question. The developers typically answer within a day or two.

14.4 Using the Lincoln Infant Lab Package (LOOK)

You can find the LOOK program in the folder LOOK under Taalverwervingslab within the Babylab workgroup on the J: drive. To create an experiment, you can use the *Create.exe* program. This program works as follows:

1. When you run *Create.exe*, you will be able to create a new experiment by clicking `File > New`. You can also save and load existing experiment (*.bex*) files here.

2. When you have added the experiment to the Experiments list and have selected and opened it, you will see the Experimenter frame. In this frame, you can set the dimensions for the control screen and the screen where the stimuli are displayed (see below). You can also add blocks to your experiment here by clicking Add next to the Blocks list.

3. If you have created a block, you can double-click it to access the Block frame. In this frame, you can set looking criteria (for when a block or trial ends and for when a habituation criterion is met) and you can add trials to your block.

4. When you have added trials to the Trials list, you can double-click a trial to access the Trial frame, where you can add stimuli to the trial. If you double-click a stimulus in the Stimuli list, you will be taken to the Stimulus frame, where you can add the file and specify other settings.

5. You can also access the Attention Getter frame from the Block frame, where you can add your attention grabber file and specify where it is presented.

When you have created the experiment and have saved it as a *.bex* file, you can run it by using the *Present.exe* program. This program works as follows:

1. When you start the program, you will see a screen asking you if you want to load a preferences file or use default preferences. Select default preferences the first time and then save them to a preferences (*.bpf*) file.

2. Under `Preferences`, you can decide which keys to use to register actions such as a look. You can also save the *.bpf* file here.

3. Back on the main panel, you will see a panel where you can enter information about the participant (subject number, date of birth, gender) and where you can add experiments (*.bex* files).

4. Add your experiment file to the list and then click to select that experiment in the list. Before you run the experiment, use the `Check stimuli` button to see if the stimuli are all present and working correctly.

5. Once you have set the preferences, you can run the experiment you have selected by clicking on the `Run selected experiments` button.

6. You can use the Esc button on the keyboard to exit the experiment prematurely if necessary.

7. The output files will be saved to the same folder where the *.bex* file is located and will be named according to the experimental file and subject number. It is advisable to give it a more informative name afterwards (e.g. including name and experimental date). The video and experimental files should have the same name.

You can find a more detailed guide to LOOK here: http://www.lincoln.ac.uk/home/media/universityoflincoln/schoolofpsychology/Handbook_08-12-2009.pdf.

Note that this program was made for Windows XP, while the computers in the Labs (as in the university in general) use a much more recent version of Windows. Due to the fact that LOOK is built for a different operating system, there are some functions that do not work. As of July 2017, it has not been possible to set up a classic Habituation paradigm. A Habituation paradigm would calculate the extent of decrease in an infant's looking time across exposure (habituation) trials compared to the first (three) trials, after which the program ends the exposure phase and moves to the test phase. If you wish to implement such a paradigm in the Baby Lab, a script would need to be written for this in another stimulus presentation program such as E-Prime.

14.4.1 Screen Resolution for LOOK

Note that since this program was made for Windows XP, it needs a lower screen resolution if you plan to present stimuli on the screen(s). In order to play video stimuli, use the following settings to make the stimulus video play on the right screen (connected to the screen in the booth) and make the experimenter's console play on the left screen.

1. Set your video resolution to 960 x 720

2. Set the screen resolution in Windows to 1280 x 720 (Screens 1 and 3)

3. In the Create program, under Experimenter Console, set the image location coordinates and dimensions as:

   Left 100
   Top 0
   Width 960

Height 720

and set the Stimuli Display area:

Left 1450
Top 0
Width 969
Height 720

4. In LOOK's Present program, under the Hardware tab, set the Total screen area dimensions as 4000 x 720.

15 Working outside the lab: PsychoPy & R

LEARNING GOALS

- be prepared to gather data in non-lab settings
- script a basic sentence judgement task in PsychoPy and understand the output
- explore data and run basic statistical tests in R

In some cases you might want to do experiments outside the lab – 'in the field'. Although the term 'fieldwork' has strong connotations for most people of working in a remote area of another continent with lots of mosquitoes and no running water, we use it in a broad sense to mean any data gathering that is done outside the controlled lab environment. In this chapter we first go into the notion of experimental linguistic fieldwork in a bit more detail. After that we offer an introduction to two software programs that are free and easily installed on your own laptop.

15.1 Gathering data in the field

Gathering data outside the university environment – for example by approaching people on the street or going to live with a speaker community – is a very standard aspect of many other areas of linguistic research. However, it is not typical of psycho- and neurolinguistic research. There are probably three main reasons for this:

- when working outside the lab, it's more difficult to control extraneous variables (particularly relating to the surroundings)
- until recently, research equipment has been difficult to transport and experimental software has required expensive licenses
- students at the university are the most convenient participants to recruit

The effect of these issues is that most psycho and neurolinguistic research to date has been carried out on not just convenient samples but 'convenient languages' – languages that are spoken around the universities where the researchers are working, and languages which are familiar for those in the scientific community (predominantly English and Dutch). This works in a cycle: the more research and resources for a particular language, the more convenient it is to stick with that language, so the more research is done... and so on. An additional side-effect of this situation is that many paradigms which are used to investigate linguistic processing rely on stimuli that are written, because literacy in these convenient samples is reliably high.

Question 21 - What are potential validity issues surrounding the use of written stimuli to investigate human language processes?

The issue of control when working outside the lab is one that deserves some attention, and we'll look into this in section 15.2. However, the second and third points above do not need to stand in the way of your research.

Firstly, with advances in technology, it's increasingly possible to gather precise and rich data outside the lab: equipment is smaller and prices more reasonable. Additionally, there is now free, open-source software which allows you to run sophisticated experiments and analyses without having to be at the university or pay a lot of money for a license. In this chapter you'll learn how to use two of the most popular open-source programs.

Secondly, there are real validity issues with theories that are based entirely on data collected from young, highly-educated participants from one small part of the world. In the field of psychology, you may hear about WEIRD samples: Western, Educated, Industrialised, Rich and Democratic. There are plenty of articles online about this (for a nice intro have a look at this article by Slate magazine). As we mentioned back in Chapter 1, your sample should be representative of the population you generalise to. Let's say we find that a group of Dutch speakers are slower to process questions than statements. Can we simply assume that this will be valid for speakers of Japanese, Quechua or Swahili? The problem is clearly that language-specific properties are very likely to influence the way linguistic input and output is processed, and therefore if we wish to generalise to all languages (or all humans) we would need to sample in a way that is representative of a variety of linguistic types. Since that's difficult to do in one experiment, a rule of thumb is to always be mindful of linguistic diversity in how you frame your research questions and how you generalise your results.

15.2 Practical tips for experimental fieldwork

Let's look now at some of the practical challenges of doing experiments outside the lab, and some ways to approach them. This advice is mainly from the point of view of working on an understudied language, where you are not a member of that speaker community yourself. However, don't forget that a lot of these points apply in general to research outside the laboratory setting. If you're interested in running experiments in a more remote or unfamiliar region, you might benefit from following a course on linguistic field methods.

- **The equipment and setup**
  One of the most obvious disadvantages of fieldwork is that it's more difficult to control variables relating to the environment, which is a particular concern for experimental research. If you have to test in various locations, try to meet participants in settings that are similarly 'tranquil' such as libraries, study areas or a quiet room away from distractions and interruptions. You should take care to always test using the same computer and equipment (so that no confounds arise due to hardware differences). You could also ask participants to put on headphones to help them focus. If you are comparing your field data with data from a more accessible setting or group (for example, comparing elderly Dutch speakers with a university-age control group, or comparing data from Chukchi with data from English) try to conduct both experiments under similar conditions.

- **The experimental design**
  If you work with a language which is not widely researched, you will not have tools such as CELEX to help you create a controlled design. You can conduct norming surveys in your speaker community; or you could try to gather word frequencies by building your own corpus of texts and writing a simple script to find out how many times a word occurs. In general, choose your experimental paradigm wisely when first approaching an understudied language or unknown experimental setting. For example, to conduct an experiment measuring precise reaction times that are accurate to the millisecond level, you need to be sure you can run the experiment in an environment where there are zero distractions. If you can't ensure this, you could explore other ways to investigate your question using measures that are more robust (for example, accuracy rates, choice tasks, picture descriptions, mouse tracking and so on).

- **The materials**
  Not all stimuli will be appropriate for all settings – typical objects and situations differ around the world. At worst you might offend someone if you don't bear cultural differences in mind, but a more likely problem is that the stimuli from one experiment are unfamiliar or weird for participants in another setting, which raises issues of validity. Look for native speakers to help you with checking stimuli and to pilot your experiment.

If literacy is an issue, avoid paradigms that use written stimuli and get your task instructions recorded as audio files (you could even make a video clip). It's best to conduct as much of your interaction with participants as possible in the language you are researching, even if participants are literate in a standard or official language. Using a standard or official language may make it difficult to convince participants that you really do want them to respond in the 'oral' language, which may have much lower prestige.

- **The participants**
  There are many things to bear in mind when it comes to recruiting and interacting with your participants. Many people outside the university setting will be unfamiliar with the process and the point of research in general and computer literacy is also an important issue to bear in mind. Also be mindful of how you interact with your participants on a personal level. When testing in the lab in Leiden it's quite acceptable to make an appointment by email, have someone sit down and perform a task, hand them some money, ask them to sign a form and expect them to leave again immediately. In other settings you should be prepared for a more friendly, less businesslike approach which may ultimately mean a less tight schedule.
  It's obviously important to be familiar with the cultural norms in the community where you are going to test, especially regarding payment and personal details. For example, handing out money might not be appropriate, so you might think instead about offering a small souvenir or gift after participation. In some settings people may not be willing to give personal details or sign forms, so it may be necessary to adapt your approach while still ensuring that you fulfil ethics requirements. All in all, if you're not a member of the speaker community, you can really benefit from finding a couple of reliable contacts who are willing to help out as advisers, mediators or research assistants.

In the next two parts of this chapter we introduce Psychopy, which you can use to run experiments, and R, which you can use to analyse your data (and make some very nice plots). Both of these programs are available at the university too – they are useful wherever it is you choose to do your data collection!

15.3 Introduction to PsychoPy

PsychoPy (Peirce, 2007) is a cross-platform free program written in Python that can be used as an alternative to E-Prime. Especially when you want to carry out simple experiments using your own laptop, PsychoPy can be very handy. It has a wide community of users, which means that if you run into problems you'll have good support. In the following tutorial we'll script a sentence judgement task.[23] A sentence judgement task might be necessary when you want to get norming data for sentence stimuli; alternatively, a sentence judgement task can be an experiment in itself – perhaps you are curious about how consistently native speakers rate certain grammatical constructions... or perhaps you want to see if construction acceptability can be influenced by other factors... there are many possibilities!

We'll script the task in the "Builder View" which means that we use the GUI (Graphic User Interface). Later on, or if you're an experienced Python programmer, you could use the Coder View but we'll leave it aside here. Though it's nice to know that PsychoPy gives you access to the code!

15.3.1 Getting started

Open PsychoPy from the Windows start menu. If you'd like to install a standalone version of PsychoPy on your own computer you can go to https://github.com/psychopy/psychopy/releases. Note that in this tutorial we used version 1.83.04 (the tutorial might work with older versions but we're not entirely sure; we are however sure that it still works on newer versions up till 1.84.2). Whichever version you use, keep in mind that you shouldn't change between versions between participants during the course of an experiment!

- Mac: choose StandalonePsychoPy-1.83.04-OSX_64bit.dmg
- Windows: choose StandalonePsychoPy-1.83.04-win32.exe

Go to https://doi.org/10.5281/zenodo.1339334 to download PsychoPytutorial.zip containing the other materials we'll be using.

15.3.2 PsychoPy elements

To build up a script in PsychoPy's timeline and to automate procedures, a few things are required:

1. **Trial Lists/Input Information**
   This can be an Excel file containing all stimulus information – just as usual!

[23] We thank Bastien Boutonnet for creating some of the materials on which this chapter is based.

2. **Routine**
   A "routine" in PsychoPy is essentially "a part of an experiment". It could be an instruction screen, a blank screen, a screen presenting stimuli, as well as all the things you want to do in the background, or ask the participant to do. Compare it to a Slide object in E-Prime.

3. **Loop**
   A "loop" in PsychoPy is an element which will induce the repetition of one or a series of routines. Compare this to a procedure in E-Prime.

Have a look at the Builder View in figure 43. Note, that you don't see a "list" component. Shortly you'll discover how to implement a trial list.

1. Experiment flow or structure
2. Workspace showing the selected routine
3. Components to add to a routine
4. Top menu

Figure 43: Basic layout of PsychoPy's GUI.

15.3.3 Basic scripting

As you have learnt in earlier chapters, the syntax of an experiment script typically contains the following in some form:

- Welcome and explanation of the task
- An example for the participant of the task that they will perform ("practice")
- The experimental task: usually the same thing repeating over and over (a loop, in other words) – see a stimulus, do something, see a stimulus, do something...

- A way to collect responses from participants
- A thank you and goodbye screen

Open an empty PsychoPy **Builder View** (`Ctrl+L` on Windows, `cmd+L` on Mac) and save it using an understandable name (and keep saving regularly!). By default, a routine called "trial" is already in place, and an empty time line. You can remove it by clicking the "x" in the top left. There are three possible ways to add a new routine:

1. `Experiment > New Routine` in the Application menu
2. `shift+CMD/CTRL+N` as a keyboard shortcut
3. Insert `Routine > (new)` in the `Flow` panel. Clicking `Routine` here will also show all the routines that are already in place (and of course the ones that have been added through possibility 1 or 2). Click it if you want to implement a routine in the timeline. A • symbol indicates where on the timeline you can place that routine. Click when the • is in the right place.

In the right-hand panel containing possible components, you can select the possible elements to add to the routine that you have open. Now let's make an introduction routine.

1. Add a new routine, name it `introduction` and insert it in the time line in the flow panel.
2. Add a text element by clicking the square with red and black T's in the right panel. A properties window will open.
3. Type next to `Text` the contents of the introduction:

   "Welcome to this experiment. You will see a series of sentences. Please indicate how acceptable you think each one is on a scale from 1 to 7.

   To choose a point on the scale you can click a number. Confirm your choice by pressing ENTER.

   When you are ready to start the task, press the spacebar."

4. Since we'll wait for the participant to terminate the routine (by pressing spacebar), we'll leave the `duration (s)` empty. Make sure that the dropdown menu next to the `Text` input you choose is set to `set every repeat`. Click `OK`.

5. Add a keyboard element and rename it as `key_resp_intro`. Allowed key is ''space'' and again we'll leave the `duration (s)` empty. Click `OK` when you're finished.

6. Check the running of this routine by clicking the green button in the top menu. Once you've finished you'll find a folder automatically added to the directory where your script lives. It is called "data" and contains script info, log data and a .csv with participant data (information and responses).

Warning: sometimes the university network computers do not save .csv files and you need to check the `xlsx` box in `Experiment settings > data` to actually be able to inspect your output in a friendly way. We don't know why this is the case and recommend that you run your actual experiment on your own laptop (the point of learning PsychoPy in this chaper was so that you can run experiments outside the lab anyway)!

So, as you have seen by now, what we are aiming for in this tutorial is a simple routine that presents a series of sentences which participants have to rate. We are not going to create slides one by one – this would be really time consuming! Instead, we make use of the fact that PsychoPy can pull data from a list (similar to E-Prime and Experiment Builder).

In our trial list we need at least the following information:

- Each sentence we want to present
- An ID (code) for each sentence, to make life easier when analysing the results
- A code indicating which condition each sentence belongs to (for example, if you are testing a number of different grammatical constructions, you would code for each sentence which grammatical construction it exemplifies).

Question 22 - We'll be using a seven-point scale for participants to rate how acceptable the sentences are. Imagine you are collecting this data in order to compare the acceptability of two sentence types – in other words, comparing two conditions. Which statistical test would be appropriate here?

Now, let's try to put some things together. Once we're finished you'll have a basic script you could easily adjust to make your own experiment. You will also be able to use (and understand) sample scripts that are included with the package – you can find these in the "Demos" menu. We'll continue working on the PsychoPy file (using the "Builder View") that you already have open. Make sure that you have downloaded the materials into the same directory as your script!

1. Open *ListA.xlsx* and check the organisation of the stimuli. Once you're familiar with the contents you may close it.

2. Add a routine after the introduction. Give it the name "sentence".

3. Add a text element and type `$sent` in the text field. The dollar sign indicates a variable (similar to when you used the square brackets in E-Prime). This variable refers to the column "sent" in the trial list file. Make sure that the dropdown menu next to this is set to `set every repeat`. Set the duration to infinite.

4. To actually implement the trial list, we'll need a loop. Click `Insert Loop` (Flow panel). Choose the end point of your loop (the point where it will go back in the timeline, which is after the sentence routine) using the •. Choose the start point where you want the loop to go back to, using the other • (at the beginning of the routine). A properties window will open.

5. We will run List A in a random order only one time, so the amount of repetitions should be set to `1`. Click `Browse` to select the trial list. PsychoPy will recognise the structure of the file and will show you the names of the columns. Figure 44 shows how the script should look at this point.

6. Click `OK` to finalise the loop.

7. To add a rating scale click `Responses > Rating` in the components panel (the right-hand panel). Put in `1` and `7` as lowest and highest value.

8. Click `OK` to accept the settings and run the script. If you want to stop the script while it's runnning, press `Esc`.

9. It already looks fancy, right? But the scale could be a bit clearer. Double click the rating scale in the timeline to open its properties. As category choices type `1,2,3,4,5,6,7`. This will attach a number to every choice point.

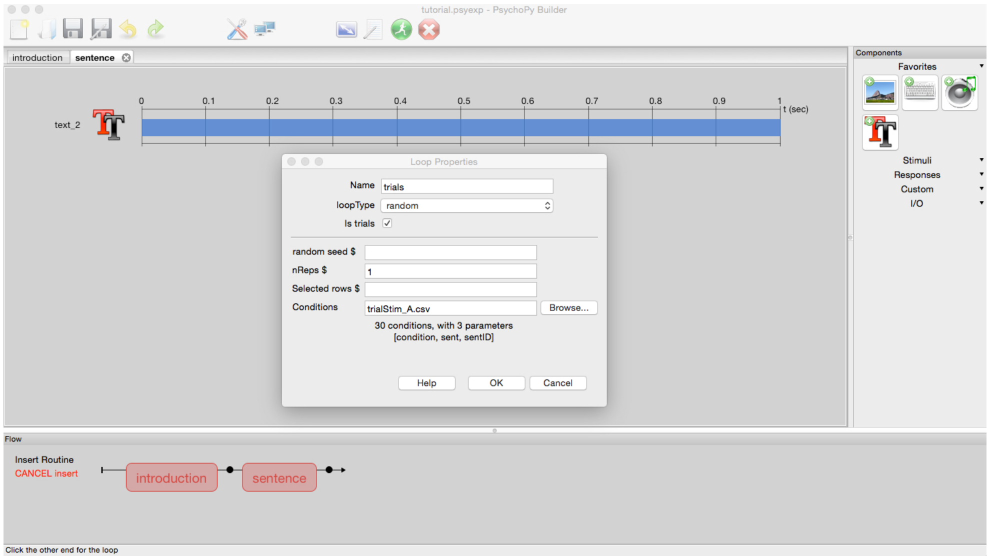


Figure 44: Properties window of a loop.

Exercise 63 - Routines in PsychoPy are like Slide Objects in E-Prime: you can display several different items in one presentation field. In our experiment we could add a description for the scale. Though this is possible in the rating properties window, you get way more flexibility if you select a secondary text component. Add a scale description underneath the Choice button using [x,y] coordinates. Use `0.05` as letter height.

Exercise 64 - Add a thank you screen at the end that will disappear after 10 seconds.

Test your script – every time after you have changed something to keep track of any bugs! PsychoPy will create in the directory of your script a folder called "data" (if there isn't one already). Here you'll find the output of the experiment. PsychoPy always saves data (also for participant number `0`) and doesn't overwrite same participants numbers.

15.3.4 Counterbalancing

In your directory you should have two stimuli lists `ListA.xlsx` and `ListB.xlsx`. Suppose we want to present List A to participant 1, List B to participant 2, List A to participant 3 and so on. The easiest way to do so is to tell PsychoPy the list you'd like to use in the start up window (in which you also type the participant number). This start up window is referred to in PsychoPy as `Experiment info`. We'll create an extra text box in which you can enter which list you want to use for that participant.

1. In the top menu click `Experiment settings` (the rectangle with two arrow points).
2. To the right of `Experiment info` add a box by clicking a `+`. Under `Field` type `counterbalance` as name. As `Default` we'll use `ListA.xlsx`. Click `OK`
3. Open the properties of the trials loop. In the field `Conditions` we're going to refer to the information you give in the `counterbalance` field we've just prepared. Instead of the file name we type `$expInfo['counterbalance']`. Click `OK` and check whether it works for both lists!

To present two or more lists in counterbalanced orders you'll need to assign these orders somehow. In our set up we could present participant 1 List A and then List B, participant 2 List B then List A, participant 3 List A then List B, and so on. We'll create separate Excel files to assign the different orders. Also, the trials (the existing loop) should be embedded in a block loop as shown in figure 45. The block loop will assign the order of the lists, which will in turn be read by the trial loop. Essentially the structure you see here is an implementation of the base structure in figure 15 from chapter 8.

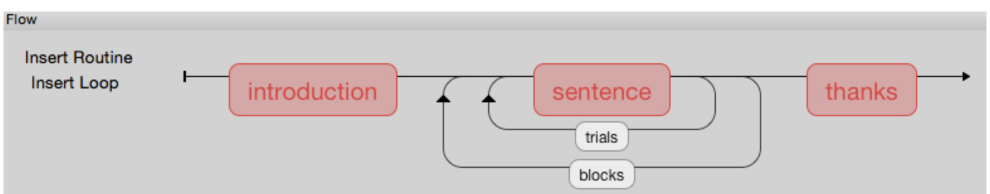


Figure 45: A loop within a loop.

Exercise 65 - Create a new Excel file. Give the first column the header `TrialList`. On the second row, type `ListA.xlsx` and on the third row type `ListB.xlsx`. Save the new file as `AB.xlsx`. Create a second file with the same column header but the reverse list order and name it `BA.xlsx`.

Now we have the order information in place, we can embed the trial loop in a block loop as in figure 45.

1. In the top menu click `Experiment settings` and change the default name of `counterbalance` into `AB.xlsx`.
2. Insert a new loop as indicated in figure 45. Name it `blocks` and choose `sequential` as loop type, since we delegated the ordering to an Excel file.
3. We'll run two trial lists for one round, so the amount of repetitions should be set to `1`.
4. In `Conditions` we refer to the Experiment Info: `$expInfo['counterbalance']` asking which list order to use. Click `OK`.
5. The last thing to change is the settings of the trial loop, which is still referring to the Experiment Info. Open the properties and type in the `Conditions` box which lists should be presented. They are to be found under the column header "TrialList", so here we can refer to the trial lists by typing `$TrialList`. Click `OK` and test!

15.3.5 Adding a pause screen in a loop

Just as with E-Prime you can add snippets of code to the script. If you're an experienced Python-er you could generate the script and make adjustments in the script right away. We won't do this! To script a pause screen in a loop we just need to insert a snippet with two lines of code.

1. Add a routine **before** the routine `sentence`, name it `blockBreak`. We'd like to have a pause after one block, so, the routine should be embraced by the block loop but **not** by the trial loop – put it between the two arrows pointing upwards.
2. Add a pause text, participants should be able to proceed by pressing the spacebar.
3. Here's the trick: in the routine `blockBreak` we'll add a code component. In the components panel (on the right), click `Custom` and select the box with the image of some functions (if you hover the mouse over it, it should have a description starting with "Code: ...").

4. In the top bar of the properties window of the code, select the tab `Each Frame`.

5. Type in the following code:

```
if blocks.thisTrialN not in [1] :
    continueRoutine=False
```

This will work as a counter for the loop. With this code, after one level of the loop called `blocks` (there are only two levels here!) the routine called `blockBreak` will be presented. Suppose that you'd have break before trials and want to have a pause after 5, 10 and 15 trials, the code should check `[5,10,15]:` as long as the trial number is not any of these numbers, the pause routine will be skipped. [24]

Exercise 66 - Add a practice routine. Use the practice trial list Prac.xlsx as item list and put it in a loop. Add routines to introduce the practice session to the participant, and to continue the experiment after the practice.

Exercise 67 - Add a second pause screen to create a break within the trial loop of the 'real' experiment. Let it go to the pause screen after every 10 sentences.

In the next part of this chapter, you will learn the basics of R, an open source statistical analysis software. For the purpose of the tutorial, you'll work with some data we've provided. However, for the exercise at the end of the tutorial, you'll work with data from the PsychoPy script you've just created (remember – data gets stored in the same directory as the script).

[24] The reason to use a counter-intuitive function `if not in [...] continue=false` has to do with the fact that we start with the break. Programming a break after a routine requires additional scripting which may be influencing other parts of your script.

15.4 Introduction to R

In a previous course, you may have worked with the IBM software SPSS ("Statistical Package for the Social Sciences") which is popular in the field of Psychology. Another piece of statistical software that's gaining popularity is R (R Development Core Team, 2008). Unlike SPSS, it's an open-source program. Because it's open source, it's a tool that's constantly growing and developing – there are more possibilities for doing things with your data in R than you'll probably ever need.

15.4.1 How does R work?

R is quite different from SPSS. One of the biggest differences between the two is that while with SPSS you have all your data sitting there in front of you in a spreadsheet a bit like Excel, in R you typically don't have your data visible all the time. Although it's possible to look at it whenever you want, it can seem a bit strange at first that it is sort of 'hidden away'.

Why is it like this? Well, it's because R is not conceptualised around the idea of a spreadsheet. Basically, it's a form of programming. So, instead of doing analyses by going to a menu and clicking options, you command R to do what you want by typing out functions. And, instead of a table of data in front of you, you have your dataset stored as an 'object' that you can call up and do things with, using those commands. Let's briefly look at each of those ideas in turn.

If you're new to programming, then think back to the chapters where we used Excel functions (chapters 5 and 9). Remember that we used functions to ask Excel to do things for us, and in the brackets after the function name we had to specify the values or cells to feed into the function. R works on a similar principle: if you want to do a t-test for example, you would use a function like `t.test()`. You type in the function, and in brackets you write the necessary inputs – so for a t-test, you need to tell R which are your two groups. When you run the command, R will produce the output for the t-test, similar to SPSS.

```
t.test(variable1~variable2)
```

Another aspect of R that will be new to you if you don't know programming is that you can give things names ('define' objects) and then recall them later using those names. For example, we do our t-test, and we want to store the analysis so that we can quickly look at it again later. To do that, we write the name we want to give it (e.g. `analysis1`) and then use this arrow sign `<-` to tell R that we want the thing called `analysis1` to be whatever comes after the arrow – in this case our t-test.

```
analysis1 <- t.test(variable1~variable2)
```

Now when we want to look at that output again we just type `analysis1`. The names you choose for things are up to you. You can define anything you want, pretty much:

```
mean_age <- 22
```

```
employee.of.the.month <- "bobby"
```

Another characteristic of R is that it is designed as a simple core program with the intention that users will add the extra functionality they need for their specific purposes. These extras come in the form of "packages", which are also open source and downloadable from within the R program – you'll learn how to load these packages in the tutorial. The great thing about open source is that everybody can contribute to the library of packages. This makes R very versatile (if a little overwhelming at times!).

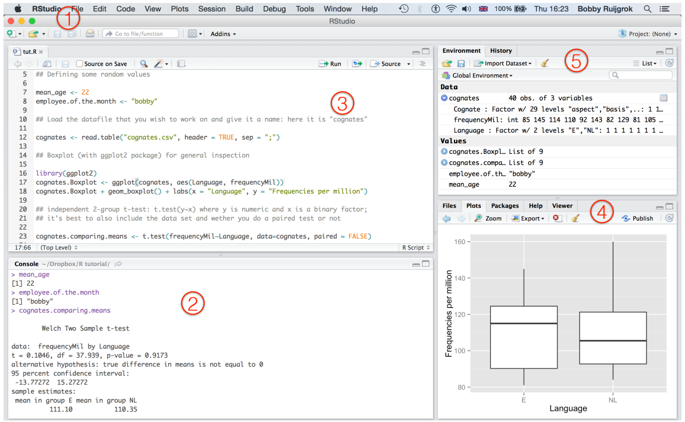
15.4.2 Using RStudio

So R takes a bit of getting used to if you're an SPSS user (but in our experience the time investment is well worth it). To help you get the hang of doing things in R, we recommend using RStudio. This is an interface written to make using R more straightforward, especially if you are a beginner. RStudio should be available at the university, or you can download it on your own computer:

1. Choose the version compatible with your operating system here:
   https://cran.rstudio.com

2. Then download the right installer for your "Platform" (i.e. operating system) here:
   https://www.rstudio.com/products/rstudio/download/

Exercise 68 - (Download and) open RStudio.



(1) Top menus

(2) Console

(3) Script editor pane

(4) Files-plot-packages-help pane

(5) Global environment and history

Figure 46: Basic layout of Rstudio.

Figure 46 shows the basic layout of the RStudio interface. The menus at the top (1) are handy so that you don't have to do EVERYTHING using commands (even though eventually you probably will). Let's first look at the window on the bottom left (2). This window is called the **Console** and is where your output will appear. You can type commands directly into the Console. However, we normally use a **script**, which appears in the Source window (3). The script is like a notebook where you have written all of the commands you want to execute, including notes so that you can keep track of what you're doing. On the bottom right there is a window which will display any graphs or plots you produce (4). This window is also useful for viewing the contents of folders, loading packages and viewing the help pages. Finally on the top right there's a window where you can view the **Environment** and **History** (5). The environment is a list of all the objects that you've defined (as you can see for yourself in figure 46). The history shows you all the past commands you've executed.

Scripts are one of the most useful aspects of working with R. Rather than doing a few analyses or making a few graphs and wondering how you did it, making a script means putting every step you take in writing, allowing you to go back and refine it and add to it however you need to. Viewed this way, data analysis in R is more like making a recipe than making a cake. Of course, the point of the process is to make the cake, but when you have the recipe written down, you can use it again to make the same cake whenever you need to. You can also copy or adapt the recipe when you have new ingredients.

The best way to learn R is to start using it, so we now move over to the R tutorial for this chapter, which is written in the R script provided. We'll get some descriptive statistics, then make some plots, and finally run a basic test. After this tutorial you will have a basis for working with your own data in R, which you'll first practice using data from the Psychopy script you made earlier.

Exercise 69 - From https://doi.org/10.5281/zenodo.1339334 download the tutorial file Rtutorial.zip, open it and save in a designated folder. In RStudio, open the script that comes in the zip file (`File` > `Open...`) and follow the instructions.

Exercise 70 - Run three sessions of the PsychoPy script that you made earlier in this chapter. Use R to merge the three output files and to run descriptive statistics (a) per sentence and (b) per condition. Save the output as a txt file.

15.5 More resources for using Psychopy and R

When a piece of open source software (such as PsychoPy and R) has many users it usually means that a lot of issues have been (or are being) discussed online. If you have a question there's a good chance that you can find some help by simply searching using google or browsing dedicated forums.

PsychoPy help:
http://www.psychopy.org/gettingStarted.html
https://discourse.psychopy.org/
http://stackoverflow.com/questions/tagged/psychopy

R general help – check also the READMEs of the packages that you're using:
https://www.r-project.org
https://groups.google.com/forum/#!forum/statforling-with-r
http://stackoverflow.com/questions/tagged/R

Again, we'd like to point your attention to Andy Field's book Discovering Statistics using R (Field et al., 2012).

Part IV
Practical project

16 Preparations for the practical project

LEARNING GOALS

- set up, conduct and analyse an experiment independently
- manage a research project (scheduling, problem solving, presenting work)

This is it: time to put into practice everything you have learned to run your own experiment, collect data and present your results to the group.

You have two options for the project:

1. Carry out the experiment you proposed back in the first part of the course (incorporating any advice we gave you back then).
2. Conduct an experiment using a template script provided to you (you need to come up with your own hypotheses and stimuli that will work in the paradigm).

You can work individually or in small groups (max. 3 people) depending on the project. In the first week the whole class will meet as usual to discuss projects and assign groups and supervision. There will also be a participant handling masterclass (read more about this in Chapter 17). For the rest of the time, you need to work independently, sticking to the plan agreed by you and your project supervisor. You may also need to contact the Lab Manager(s) by email at lucl-labs@hum.leidenuniv.nl for practical queries and troubleshooting while working in the labs.

16.1 Work schedule

Below you can find a detailed schedule you should aim to follow.

Week 1	..begin project work..
Week 2	..experiment preparation..
Week 3 **DATE TBA:**	..experiment preparation.. **participant handling "masterclass"**
Week 4	..testing participants..
Week 5	..preprocessing & analysing data..
Week 5	..finishing analysis & interpreting results..
Week 7 **DATE TBA:**	..making your poster.. **Submit poster as pdf**
DATE TBA:	**Poster presentations**

The results of your experiment are to be presented by means of a poster. You must submit your poster as a pdf by **DATE TBA**. All the posters will be printed professionally and brought to the presentation session for you. **If you miss the deadline, you will automatically have 2 points deducted from your grade** (and you will have to print out your poster yourself!). Deadlines in this course are strict to make it fair for everyone. Extensions will only be granted in exceptional circumstances and when you have demonstrated commitment during the rest of the course. Therefore, **if you're falling behind or experiencing difficulties during the project, ask for help as soon as possible.**

16.2 Frequently asked questions

How many participants do I need?
It depends on the type of experiment. Some techniques (such as EEG) are a lot more time consuming than others, so it's understandable if you can only test a couple of participants. However, if you're running a short and simple behavioural experiment, you can probably manage at least 10 participants.

Who can participate in my experiment?
It's fine for you to ask colleagues on the course to be your participants (but not people within your project group). Of course, it's better to ask people who don't know what your project is about. If someone from the course asks you to be a participant, try to make time to help them out (you might learn something too).

What if my experiment is a mess/failure/disaster?
This course is about learning the practical skills for doing independent research in the LUCL labs and beyond. Figuring out what to do when things go wrong is a fundamental part of the researcher's skill set! You will not be graded on how spectacular your results are, but on the process of doing the research. So keep calm, carry on, and make a poster at the end of it.

I worked together with a classmate. Can we submit the same poster?
No, the poster that you submit must be your own individual work. The poster serves as the write-up of your project: you make your own decision about how to present the information, and you present your own interpretation of the results and how they contribute to existing theory. Demonstrating your skills of interpretation and presentation is as important as successfully collaborating on design and data

collection.

Should my poster be in English?
Just as all other assignments in this course the language you use should be English. Maybe to your relief, text on a poster should not contain whole sentences (see also Chapter 18 for more tips on preparing your poster).

(more FAQ to come!)

17 Interacting with participants

LEARNING GOALS

- communicate effectively and appropriately with participants
- recruit participants and keep organised logs of testing sessions

We already started talking about how to act ethically towards your participants back in Chapter 2, and since then you have been building up your own ideas about how experimental sessions should and shouldn't be conducted.

Now that you're (almost) at the point of testing real participants, we're going to go over some other practicalities to do with managing participation in your experiment.

17.1 Participant recruitment and payment

17.1.1 Publicising your experiment

Ideally, we would choose a random sample of the entire population who fit our screening criteria. This just isn't what happens in reality. It's fairly normal to use a 'convenience' sample – so, asking around, putting up posters, handing out flyers...

Posters and flyers are a good way to make people aware of your experiment. However, you'll probably find that most of your participants come from personal contact. Approaching people on campus may feel very embarrassing, but it's actually a pretty successful technique (carry a flyer ready to show them!). And don't forget to ask participants to ask their friends.

If you decide to contact people by email or through social media, it's crucial to maintain a professional attitude with regard to interactions and personal details. Just like with a poster or flyer, it's good to create a standard email/Facebook message template to make everything efficient.

When designing your publicity (posters, flyers, emails) you should aim to make people informed but not confused! You don't want to overload it with text and images, but make sure people have the details they need in order to decide if they will take part. We recommend you include minimally:

- a very basic description of what the participant has to do (e.g. "plaatjes benoemen in het Nederlands")

- basic participant criteria – sometimes this is easy (e.g. "rechtshandig") but sometimes you'll have to give a general indication and take it from there (e.g. "Nederlandstalig" when you really want native speakers who acquired Dutch as first language and grew up the Netherlands)
- location of the testing (even if only general, e.g. "Lipsius")
- how long it takes to participate (e.g. "max 20 min")
- the 'payment' (see section 17.1.2)
- clear contact details! You could even make a dedicated email address (e.g. onderzoeknederlands@gmail.com)

Flyers are handy because people can take all the information with them. If using posters, think about using tear-off slips.

As you might expect, there's a different arrangement for recruiting infants for Baby Lab research. Instead of making flyers for individual experiments, there is a database of participants. Every so often, parents who have recently had babies (in Leiden) are contacted and asked if they would like to join our participant database. For this, we have a professionally produced, informative flyer that is sent out to new parents (you can find examples of this in the Baby Lab). When an experiment is planned, researchers use the participant database to invite parents with babies of the right age for the experiment.

It's important at all stages of the process to ensure your participants' privacy and manage their personal information ethically. Make sure you are keeping in line with current data management regulations. Adopt the habit of only logging and using personal data where it is absolutely necessary. For example, you must never write participant names on the bookings that you make on the lab calendars.

17.1.2 Paying participants

As a bachelor's or master's student, you are probably not going to have the budget to actually pay participants. However it's still nice to offer them something like a cup of coffee or chocolate if you can. If you do have a budget, the normal payment rate is around 5 euros per half hour. (Not bad!) Different considerations may apply to experimental fieldwork – check 15.2.

When you download the participant information sheet from Sharepoint, you'll see a part about money payment. Participants are free to stop their participation at any time, and if they do so, they either get part of the money (depending on how long they stuck around for!) or they get nothing: in either case, this must be clearly specified on the information sheet. Of course, if you aren't paying participants, you should also remember to remove references to money from the information sheet!

Payment also requires correct administration – refer to Section 2.2 if you are planning to pay your participants.

17.2 Experiment logs

At different points during the course we've emphasised that you should work in an organised manner. By now you know how to organise a bibliography and stimuli set. Hopefully, you remember from section 9.2 that for proper analysis of the data you need to keep your data well-organised and make notes as you go. Remember also how important it is to add comments to your R script? "A good recipe needs clear instructions." It'll be no surprise to you then when we tell you that you should also make notes during the running of your experiment. You'll make your life much easier if you 'log' certain things.

Which details you need to keep notes on depends a bit on the design. The main point of keeping a log is to allow you to note any problems or observations that arise during each testing session. Therefore you minimally need participant number, date and time, and a space for general notes.

However, another important function is to keep track of the testing as a whole and spot any irregularities! So, for example, if you have counterbalanced lists you should log which participant gets which order. Many details will also be acquired by the background questionnaire or logged by the experiment software, so having a bit of overlap will mean you can spot immediately if something isn't running as planned.

You'll need to keep track of which log entry goes with which participant – and with which background questionnaire! Think carefully about how you will do this, because you cannot use the participants' names, for ethical reasons. You should **always assign each participant a unique, anonymous participant number**. This can then be used in the background questionnaire (include a question where this must be filled in), in your own experiment log, and even with the experimental software if it gives you this option (e.g. PsychoPy's 'experiment info'). It's important to **avoid linking any personally identifying details with participants' experiment data** – for example, store consent forms and payment administration separately from the rest of your work.

For EEG experiments we also have a separate log book – you will find it on the

Sharepoint EEG Lab Log Sheet. Here you log the electrode band(s), cap size, battery and booth number (and any problems). It wouldn't hurt to also note those data in your own log sheet!

It's a good idea to include "debrief" time (say a couple of minutes) at the end of the session, asking the participant how they experienced the experiment. Maybe they'll tell you that they were using a strategy or had guessed the hypothesis (even if wrongly). These things can affect response behaviour, so note them down!

Finally, a log sheet is invaluable while doing your analysis as it will help you to remember things that are necessary for understanding your data. Spotted some outliers in the data, or a participant that doesn't fit? Now you can check your log! Having this kind of information can be very helpful when you're facing a difficult decision about whether to exclude an anomalous data point or participant from the analysis.

At the end of the day, your research procedure should be transparent and your results should be interpretable. Making organised notes at every stage during the empirical cycle will help you to achieve this.

17.3 Participant handling masterclass

We're going to give you a chance to practise your participant handling technique with a 'masterclass'.

There are three goals of this masterclass:

1. to give you the chance to pilot your instructions and task design before collecting 'real' data
2. to give you experience of conducting an experimental session with a participant
3. for you to get honest feedback about your performance as an experimenter – this is something you won't usually get from real life participants!

As mentioned in Chapter 16, by the time we do the masterclass, **you must have an experimental script that's ready to run**. That means testing it out *before* you come to the session! You'll also need the following:

- information sheet and consent form for your experiment (bring several copies of each!) You can download templates for these from the Sharepoint Shared Documents library.

- experiment log file, ready for you to take notes. Can be on paper or as an Excel file, whichever you prefer.

Lastly, before you come to the masterclass make sure you read Chapter 2 again to refresh your memory about how to interact appropriately with participants.

Your participants will rate the session (and your interaction) in a number of areas, and give you constructive criticism. You might find this nerve-racking, but we've run experiments ourselves and we know that sometimes things don't go to plan! Aim to be flexible and professional. However when problems arise from carelessness, disrespect or bad preparation, it's not acceptable.

18 Making a poster

LEARNING GOALS

- present experimental data in poster format
- create a high quality poster in Adobe Illustrator

18.1 Elements of a scientific poster

A poster is a way to communicate your work to others in the field. It contains the same elements as a paper, although in a bare-bones form. Chapter 10 will refresh your memory about the key elements of a scientific write-up. This chapter gives you practical advice for getting that content into a poster form.

18.2 Design

In cooperation with Olga Kepinska[25] we have made a Language and Cognition Group (LACG) poster templates that can be used to make a high quality conference poster. You can find the templates in Sharepoint. It is based on the banner which is used on the LIBC-language website: the colour green is the Humanities green as prescribed in Leiden University's style guide; the main text colour resembles the LUCL blue. The font is Arial. For both the portrait and landscape formats we have included proportional column widths. You can create your desired layout by arranging the banners (if you need more, just copy-paste) and you can alter the titles as necessary. Although you can change the text of banners you should at least have on your poster *Background*, *Methods*, *Results*, *Discussion* – adhering to the scientific writing framework.

18.3 General tips

Take your time and start at least two weeks before printing. Check the details as communicated by the conference where you are going to present. For example, is it permitted to use a landscape format? How busy will it be? Whichever format you use, it is best to arrange your content in columns and not use the whole width. Make a sketch before you start the actual editing process.

[25] Olga also helped to write this chapter.

Three important basics:

- keep it simple – don't overload your poster with tons of data
- no prose – let it be clear but don't use whole sentences
- keep it logical – be consistent and orderly

Although it is tempting to show every analysis on your poster, this often doesn't contribute to clarity. Concentrate on the main result(s) – it's even better if the title contains the main result. People will only stop by your poster if it is attractive at first glance. A good poster should read like an advertisement, so use telegram-style phrases. When you present your poster, you do the talking, using the bullet points as a guide. If you really think that extra graphs or tables are necessary you could consider bringing a tablet or printout to show this material.

Consistency can be achieved in different ways, in content and in form. A clear colour scheme helps readers quickly grasp the content. For example:

- white text for titles
- blue for main text
- grey for references and acknowledgements
- match colours of conditions mentioned in the text with the colours in tables and graphs

One font size per style is preferred.

18.4 Basic steps in Illustrator

The reason why we made a template in Adobe Illustrator (find it under `All Programs`) is that we strive for a professional result. A downside of Illustrator is that virtually anything is possible and that you can be distracted by the depth of all the menus. Try to focus on what you really need: implementing text and images. Have a look at Figure 47. Basically, you need the cursor mode (1) and the text mode (2). To change the colour of some text, select it and choose the fill button (3). You can just add images of graphs by dragging them into the poster that you've got open. If you want to select multiple text blocks, images, etc., click on the items while holding down a SHIFT key. For example, it's best to move title banners with text at the same time. **TIP**: you can group multiple items by `Object > Group`. Save your changes regularly (business as usual!). When you are done editing, save your document as a pdf file by `File > Save As`, choose `Adobe PDF (pdf)` as format, save, deselect the default check boxes in the general window and save. If you want to become a top-notch expert you may want to study the Illustrator help desk, but for our purposes it shouldn't be necessary!

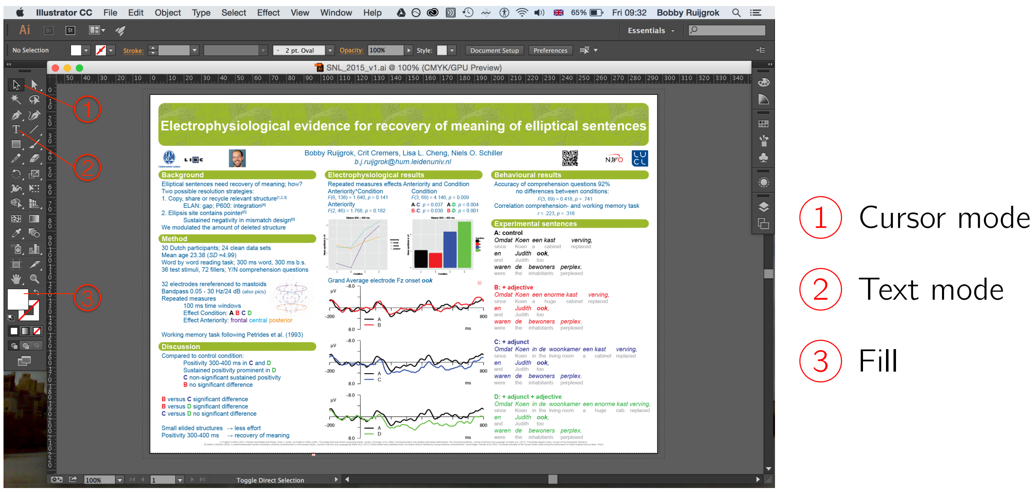


Figure 47: Illustrator work pane.

18.5 Pictures, graphs and tables

With the templates we have implemented logos that you will probably need. Sometimes Illustrator cannot load images that you use. Storing all images in the same directory as the file you are editing prevents this problem. If you want to add/change a logo, graph etc., only use files with an acceptable amount of pixels (resolution) – or, even better, use vectorised pictures. Furthermore, if you want to change the size make sure you keep the height and width in proportion. If necessary, you can export a graph in a different dimension in the statistical program you use. Pictures and text that have been stretched or squeezed look ugly and make the poster look amateurish.

It's advisable to put some time into ensuring the quality of your images as they are the eye-catchers of your poster. For example, export SPSS graphs as EPS files. **TIP**: you can use R to produce vectorised files. It is more versatile and the output is unbeatable. You can find scripts for some basic graphs here.

As said, anything is possible and sometimes Illustrator 'thinks' for you and might interpret your images slightly differently. Don't panic, stay calm, there's always a solution.

18.6 Tech specs

The actual editing size of the template is A4 with 5 mm margins all around and between banners. The way we have designed this template will produce excellent A0 and A1 size posters. When you move columns and banners around, please check from time to time whether your margins still make sense. You may want to select `View > Show Grid` to check the spacing. You can choose between CMYK or RGB as colour mode. CMYK works best for printed material like a poster, RGB for things on a screen. Table 2 lists the codes for the colours that you will probably mainly use. Note, however, that in CYMK mode very bright colours will be slightly dimmed. "Hex" is useful to know if you want to match colours of plots generated by the package "ggplot2" (see the R script in **Exercise 69**). Not enough choice? Knock yourself out here.

Colour	**C**	**M**	**Y**	**K**	**R**	**G**	**B**	**Hex**
Poster								
Banner	0.35	0	1	0.14	143	219	0	#8FDB00
Title white	0	0	0	0	255	255	255	#FFFFFF
Main text	0.9	0.53	0.08	0.12	22	105	206	#1669CE
Reference text	0.43	0.35	0.32	0.01	144	164	172	#90A4AC
Primary								
Black	0	0	0	1	0	0	0	#000000
Red	0	1	1	0	255	0	0	#FF0000
Blue	1	1	0	0	0	0	255	#0000FF
Green	1	0	1	0	0	255	0	#00FF00
Yellow	0	0	1	0	255	255	0	#FFFF00
Magenta	0	1	0	0	255	0	255	#FF00FF
Cyan	1	0	0	0	0	255	255	#00FFFF
ggplot								
group 1	0	0.5242	0.5605	0.0275	248	118	109	#F8766D
group 2	0	0.1311	1	0.2824	183	159	0	#B79F00
group 3	1	0	0.6989	0.2706	0	186	56	#00BA38
group 4	1	0.0255	0	0.2314	0	191	196	#00BFC4
group 5	0.6196	0.3882	0	0	97	156	255	#619CFF
group 6	0	0.5918	0.0735	0.0392	245	100	227	#F564E3

Table 2: CMYK, RGB and Hex colour codes for the **LACG** poster (and other presentations, papers, etc.). Primary colours are standard colours in SPSS, R, MATLAB, Latex, and so on. For convenience the colour codes for default ggplot colours are listed (you're welcome).

18.7 Printing

Has anyone else checked your final version? Your supervisor(s) will want to see it anyway, but other colleagues may also have useful input. So, again, start on time since proofreading can take a while! Be sure to send around your file as pdf.

If you're getting your poster printed at a print shop, we recommend you export your poster in pdf. It may surprise you, but not all print shops have software like Adobe Illustrator. More importantly, pdf guarantees that the print looks exactly how you designed it. Here are some printing options:

- We have good experiences with the 160 gram prints from Labor Vincit.

- Slightly more expensive, printing 180 gram, is The Printer which has two shops in Leiden.
- The University Copy and Printshop in the Lipsius is by far the most expensive, printing 200 gram.

Normally, the printing process takes one working day, although sometimes it can be done while you wait.

18.8 Presenting your poster

Posters are informal, but still professional. Therefore, you should still practise the story that you are going to tell when people ask you to "run them through it" (most people prefer to hear it from you than to have to read the poster).

Prepare a sign-up form for those who want a pdf of your poster to write down their email addresses. It's a nice way to broaden your network, too. Of course you can bring hard copies as well – it depends on your budget, as colourful prints are costly (but greyscale copies lose all the impact of your images!). While explaining your poster, be aware of your position: try to avoid standing in front of the story!

Appendix A

WORKING SAFELY WITH CHEMICAL SUBSTANCES

The term ‘Chemical substances’ applies to all substances that are used during physiological research: alcohol, acetone, electrode paste, disinfectant, collodion, etc.

1. Ensure that labels on all bottles, jars, cups, etc. are clear and accurate.

2. Always check the label of a substance before use, mainly to make sure that the right substance is being used, but also find out whether any precautions have to be taken (when in doubt, read the manual).

3. Bottles, jars, cups and boxes should be kept closed as much as possible, for both economical and health reasons, and to avoid vapour reaching the eyes.

4. Never smell the contents of bottles directly. If necessary, wave some of the odour towards the nose.

5. Always ensure good ventilation.

6. Never use a bottle or jar for substances other than those indicated on the label.

7. Whenever a substance touches the skin, immediately rinse with lots of water. It is better to be safe than sorry.

8. Whenever a substance comes into contact with the eyes, immediately flush the eyes with lots of water. Never rub! If the tap has a plastic hose, turn it upwards and use it as a shower for the eyes. Alternatively use a clean cup as an emergency eye bath. Always use cold water and control the flow with the free hand. Flush each eye in turn, pausing regularly to avoid hypothermia of the eye.

9. Never flush chemicals down the drain. The Lab Manager(s) will take care of disposal.

If in doubt, it is better to call for help. The First Aid Service can be reached on 071 527 2300, or ask at the Lipsius reception.

Appendix B

USER MANUAL FOR SEKUSEPT PLUS

General use

1. Make sure that all materials in need of disinfection are being rinsed first to remove the gel.
2. Always wear safety glasses, safety gloves and an apron or lab-coat when using Sekusept PLUS. Sekusept PLUS is corrosive and can irritate skin, eyes and clothes (S36/37/39).
3. Sekusept PLUS is suitable for 'Electrocaps', syringes and electrodes. For all other materials please make sure that Sekusept PLUS is a suitable disinfectant.
4. Do not leave any material submerged in the disinfection liquid longer than appropriate as this can corrode the materials. When diluted to 4% (as is the case in our lab), the incubation time is 15 minutes.
5. Rinse all materials that have been disinfected with Sekusept PLUS thoroughly with streaming water before they can make contact with the skin.
6. This disinfection method is appropriate against infection with HIV and Hepatitis-B.
7. Make sure that Sekusept PLUS and the disinfection bath are always stored in a closed container.
8. The efficacious ingredient in Sekusept PLUS is GLUCOPROTAMINE (25%)
9. Sekusept PLUS is in accordance with DGHM and DVV
10. Risks: skin wounds, burning wounds (R34), harmful when admission through skin (R22)
11. First aid:
    - in case of injuries: notify a medical doctor
    - in case of inhalation: inhale fresh air, when complaints endure consult a medical doctor

- in case of skin contact: immediately rinse with water for 10 minutes (S28). Take off clothes that have been contaminated (S27).
- in case of eye contact: immediately rinse with abundant water for 10 minutes and consult a medical doctor (S26).
- in case of swallowing: drink abundant water, NO MILK; medical attention is required immediately.

12. Treatment of electrocaps after each measurement:

    - Remove the Velcro straps (sticky cloth). These do not need to be disinfected.
    - Remove the electrodes.
    - First remove all remaining gel thoroughly with cotton tips and a soft toothbrush under running water. Ivory soap is appropriate in this stage.
    - Use Sekusept diluted liquid for 15 minutes to disinfect the cap, the electrodes, and syringe. Always thoroughly rinse with water afterwards to prevent skin irritation.
    - When necessary use the hair-dryer to speed-dry the materials. Make sure the temperature of the materials does not exceed 40 degrees C.

Preparation of the incubation bath

The following instructions only apply to those who are authorised to prepare a disinfection liquid. Make sure you have read instructions for general use of Sekusept above.

The numbers refer to figure 48.

(1) Collect the disposal container, Sekusept and lab coat, all stored in the brown cabinet (together with spare materials).

(2) Collect the measuring cup and funnel stored in the the kitchenette's cabinet.

(3) Using a funnel gently empty the used disinfectant into the disposal container. This container is stored in the brown cabinet together with spare materials.

(4) To prepare a 4% solution, pour 20ml of Sekusept in a measuring cup.

(5) Add water to the measuring cup to make it up to 200ml.

(6) Empty the measuring cup into the small container and add another 300ml of water to the container.

Shut the lid of the container and rinse used materials thoroughly afterwards. Log the date on the log sheet, which is kept on the shelf above the small container.

Figure 48: Steps to prepare incubation bath.

Appendix C

HAND CARE

1. Use warm water.
2. Use soap from the dispenser “Unicura”.
3. Rinse your hands with lots of water.
4. Dry them with disposable paper towels. It is important to have dry hands, as they carry less bacteria.
5. Turn off the tap using the same paper towel.
6. Keep nails clean and short.

References

Baayen, H., Piepenbrock, R., and Gulikers, L. (1995). *The {CELEX} Lexical Database. Release 2 ({CD-ROM})*. {L}inguistic {D}ata {C}onsortium, University of Pennsylvania.

Brugman, H. and Russel, A. (2004). Annotating multimedia/multi-modal resources with elan. In *Proceedings of LREC 2004, Fourth International Conference on Language Resources and Evaluation*.

Field, A., Miles, J., and Field, Z. (2012). *Discovering Statistics Using R*. SAGE Publications Ltd, London ; Thousand Oaks, Calif, 1 edition edition.

Kerr, N. L. (1998). HARKing: Hypothesizing After the Results are Known. *Personality & Social Psychology Review (Lawrence Erlbaum Associates)*, 2(3):196.

Kovalenko, L. Y., Chaumon, M., and Busch, N. A. (2012). A Pool of Pairs of Related Objects (POPORO) for Investigating Visual Semantic Integration: Behavioral and Electrophysiological Validation. *Brain Topography*, 25(3):272–284.

Luck, S. (2014). *An Introduction to the Event-Related Potential Technique (Cognitive Neuroscience)*. MIT Press, Cambridge, MA.

Oakes, L. M. (2010). Using Habituation of Looking Time to Assess Mental Processes in Infancy. *Journal of cognition and development : official journal of the Cognitive Development Society*, 11(3):255–268.

Peirce, J. W. (2007). PsychoPyPsychophysics software in Python. *Journal of Neuroscience Methods*, 162(1-2):8–13.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ruijgrok, B. (2017). Bridging theoretical and experimental linguistic research. In Reckman, H., Cheng, L. L., Hijzelendoorn, M., and Sybesma, R., editors, *Crossroads Semantics*, pages 9–20. John Benjamins Publishing Company, Amsterdam. DOI: 10.1075/z.210.01rui.

Zuccolotto, A. P., Roush, R. E., Eschman, A., and Schneider, W. (2012). E-Prime 2.0 Getting Started Guide.

A Practical Introduction to Psycholinguistic Methods provides a comprehensive training for undergraduate and postgraduate linguists at Leiden University who want to get started using experimental techniques. The course is designed around the timeline of an experimental study - from posing and operationalising a research question, to implementing and running an experiment. Learning goals are consolidated through hands-on tutorial assignments and put into practice with an independent final project.

After completing this course, students have the practical skills necessary to embark on independent research in the Leiden University Experimental Linguistics Labs.

ISBN: 978-90-816359-4-3