Psych 390

Britt Anderson

2025-01-01

Table of contents

Preface **4**

1 Introduction **5**

2 The role of the computer as a tool in psychological research **6**
2.1 Research Project Stages 6
2.1.1 Preliminaries 6
2.1.2 Do the research 7
2.1.3 Analyzing the Research 7
2.1.4 Diseminate the Research 7
2.1.5 Make Your Research Open 8

3 Open Research Practices Continued **9**
3.1 Fair Principles 9
3.2 Lab Notebooks 9
3.3 Bibliographic Tools and Exercises 9
3.4 IDEs and Writing up your Research 10
3.4.1 VSCode 10
3.4.2 Overleaf 10

4 Coding **12**
4.0.1 IDEs 12

5 Quarto **14**
5.0.1 Scientific Publishing 14

6 In Lab Experiments **15**
6.0.1 Online Experiments 15

7 What are some of the challenges that computational tools pose for reproducibility? **16**

8 Knowledge Management ([[https://en.wikipedia.org/wiki/Zettelkasten][Zettlekasten)]] **17**

9 Reference Management **18**
9.1 Using References With Quatro and VS Code 18
9.2 Using References With Overleaf 18

9.3 Poster Presentations ? (MOVE THIS SECTION) 18

10 Testing Your Javascript **19**
10.1 Run your own server locally 19
10.1.1 XAMPP 19
10.1.2 In Lab Experiments 19
10.1.3 Online Experiments 19

11 Databases and management **20**
11.0.1 Datascience 20
11.0.2 Grant Funding 20

12 Summary **21**

Appendices **22**

A Version Control **22**
A.1 Getting Started - Some old videos on git from PSYCH363/310 22
A.2 An overview of git and Github 22
A.3 You have choices - other version control system 22
A.4 Git is not Github 23
A.5 Things you should do 23
A.6 But first 23
A.7 Understanding Git and the Workflow in Pictures 24
A.8 Principal Terms to Learn First 24

Preface

This is a Quarto book. Quarto can be seen as a successor to RStudio and Rmd files. If you have not heard of those do not worry. If you have you will see that most things are quite similar.

I will be using Quarto in this class (PSYCH390 W2025) to help us practice writing reproducible code and papers. There are many different systems that can accomplish the same goal and we will look at a few of them so that you are not anchored to any one system. But in order to have one shared experience and to move beyond the basics, at least a bit, we will use this tool.

One of the key lessons for learning to use your computer effectively as a research tool is learning how to do things *generally*; rather than mastering one particular variety of software we want to be able to figure out how to use whatever is the right tool for a particular application. Our needs may change as our science evolves, and the tools available to us will definitely change. You don't want to be stuck 20 years from now using the software you learned in University. If you are, your science will suffer, and you will be less able to tackle the problems that interest you. Instead of doing the key experiment, you will be figuring out how to do the experiment that is most like, but not quite, the right one, but is as well as you can do with the old software you know how to use.

To learn more about Quarto books from the Quarto people visit https://quarto.org/docs/books.

To learn more about the specifics of how we are going to use Quarto, and what else we will experiment with along the way, stay tuned.

1 Introduction

This is a book created for PSYCH390 at the the University of Waterloo

We will try to write this book together with each week or so getting its own chapter. You should be able to edit the files in this book yourself to add and correct things. We will learn more about that in the weeks ahead. As a first step see if you can clone the git repository for course course.

You might find some of this older material in appendix A on using *git* useful for getting started.

2 The role of the computer as a tool in psychological research

To begin learning how to use the computer as a tool for our research we have to first be clear on what that means. Before we go into specific tools let's discuss the research process so we can better detect where there are opportunities for us to use computers and software to aid our work.

> Class Question
>
> Ignore for the moment the specific content of a psychological research project. That is whether it is a study on attention or memory. Let's consider abstractly what are the components of a psychological research project? And then how can computers help us to do our research better? For right now we will focus on the stages of research and the tools we are now using, if any, and then we will return to the particular research stages for developing our experience with particular research tools.

2.1 Research Project Stages

2.1.1 Preliminaries

How do you decide what it is you are going to do? Look up and read the prior literature.

1. How do you find what to read?
2. How do you keep track of what you have read?
3. How do you organize your notes on what you have read?
4. How do you integrate and connect your notes and readings so as to discover new connections or ideas?
5. How to you make sure you can correctly cite the articles you have read when it comes to writing up your research proposal or research result.

How do you protect yourself against a change in databases or death of a harddrive?

2.1.2 Do the research

What skills are needed for this?

1. Program an experiment
2. Language? Python; Javascript; other
    1. How do you make stuff appear on a computer screen?
    2. How do you verify the timing?
3. How do you make sure you can find the version of the experiment if you need to share it or re-use it.
4. There is a new tool in the lab with drivers and hardware. Can you download and install the necessary software?
5. What happens to the data?
    1. How do you store it?
    2. How do you search it?
    3. What if it is big?
    4. What if it needs to be shared?

2.1.3 Analyzing the Research

1. How do you organize the data?
2. Working on a big data project?
    1. How do you get the data?
    2. Can you set up a database?
    3. Search it?
3. How do you do statistics?
    1. What is the tool?
    2. Could you write your own analysis?

2.1.4 Diseminate the Research

1. What is the tool you use for writing your research?
2. What is reproducibility?
3. Does your writing tool support it?
4. How do you make sure others can re-run your analyses?
5. How do you make sure that you can easily adapt your analyses and make sure the right figures are included when you update the data? Or that the p-values in the text are also adapted?

6. How do you change the citation format when you submit to a journal that does not want APA or an edit changes which of your citations is the first (so it changes when you use et al)?

2.1.5 Make Your Research Open

Class Question

- What is "open research"?
- Should research be open?
- How many of your labs are following these practices?

What are Open Science practices?

A recommended reading: 10 strategies for open science

And that is it for today.

3 Open Research Practices Continued

> Class Question
>
> What are the **Fair** principles for research?

Take a few minutes and see if you can determine what "Fair" stands. Then we can see if we agree. We can then use those principles to evaluate our research tools.

3.1 Fair Principles

- Fair principles
- Fair guiding principles
-

3.2 Lab Notebooks

I will not actually be spending much time on this, but we could. It is a neglected practice in psychology to keep a lab notebook that is updated. Here are some options if you are interested in exploring this topic.

- Jupyter notebooks in Psychology
- 10 rules for lab notebooks (electronic)
- How to keep a lab notebook
- Pick an electronic lab notebook
- Lab notebook 2023 guide

3.3 Bibliographic Tools and Exercises

This would make sense to be discussed next since it fits with our outline of the process that commences with reading and cross referencing knowledge. However, we do not yet have available the tools that we will need to manage this, so we will defer this for now until ADDLINKHERE.

3.4 IDEs and Writing up your Research

> Class Question
>
> What does IDE stand for and what are examples of IDEs?

3.4.1 VSCode

There are many powerful IDEs available. It will be useful for you to try more than one. However, at this moment in time the clear winner of the IDE popularity contest is Microsoft's Visual Studio Code.

Since I want you to learn to use your computer as a tool, and to continue to be able to do so when tooling changes, you need to be able to:

1. Locate code/programs on the internet
2. Download and install that program to your computer
3. Locate the help to begin to use the tool without formal instruction.

> In-class Exercise
>
> Locate, download, install, and verify you can start VS Code.

3.4.1.1 Some helpful links to get you started

- Basics Video
- Using VSCode with Python
- Using VSCode with R
- Text Editor Page from Quarto Site
    - What is "pip"? And why will you need to know?

3.4.2 Overleaf

VS Code is far from your only option. For technical writing and collaboration many scientists rely on Overleaf.

Nature will [let you submit your manuscript](https://www.nature.com/srep/author-instructions/submission-guidelines) from an Overleaf template.

Don't forget to do the overleaf homework assignment.

But know that you always have choices. You can [write LaTeX with Quarto](https://github.com/James-Yu/LaTeX-Workshop?tab=readme-ov-file) (see also).

Classroom Discussion

What is single source publishing? How does this idea impact your preference for tools like Overleaf and Quarto?

4 Coding

For a lot of the above we need to know how to code. In order to write code we need minimally to answer two questions. What language? What tooling?

4.0.0.1 Languages

What should I consider when selecting a programming language? Will it do what I need it to do now and tomorrow.

- Is SPSS a good language for statistics?
- Is R a good language for statistics?
- Is Python a good language for statistics?
- Is R a good language for coding a web app?
- Is R a good language for coding an in-lab visual experiment?
- Should you use Julia? Common Lisp? Haskell? Lean? OCaml? Rust? Go?

How do you *future proof*? - If languages go in and out of fashion what is it you should really be learning about programming? What are good coding practices?

4.0.1 IDEs

- Who are you writing code for? Human or Machine?
- What is an IDE? What makes for a good IDE?

4.0.1.1 Using an IDE

For this course we will default to VSCode, because it is currently very popular and becoming somewhat of a standard. Everything said above about not getting to attached to the flavor of the month applies to IDEs. Especially since VSCode is a tool tied to Microsoft. However, there is an opensource build of VSCode that you can use instead. You can also use anyother tool you want as long as you can figure out how to make it do the things I will ask you to do. I, for one, live in Emacs.

4.0.1.2 VSCode

https://www.youtube.com/embed/B-s71n0dHUk?si=y3fy80M0mGxGLwr5

- Basics video
- Using VSCode with R
- Using VSCode with Python

Exercise Install VSCode

4.0.1.3 Jupyter Notebooks

What are jupyter notebooks? Are jupyter notebooks ide's? What are their purpose? What languages to they support?

5 Quarto

What is RStudio? What happened to RStudio? How does Quarto relate to RStudio? What languages does it support? Can you write code in it? Can you write text in it? Can you write an article in it?

5.0.1 Scientific Publishing

- Quarto (the former RStudio people)
  - journal formats try the Elsevier format for this course. A lot of psych journals are Elsevier owned.
- a presentation
- Someone showing how to connect an sql database to quarto.
- A curated list of Quarto related stuff
- Make a website/blog for your work or lab?
  - Quarto provides an option. For example: here

6 In Lab Experiments

- Making stuff appear on monitors.
- What is OpenGL?
- Use pygame to make some simple visual experiments.
- Have a beauty contest the following week to see what people have been able to make?
- Make sure that people are using venv

6.0.1 Online Experiments

Running a server for testing and more? XAMPP Running a lab now seems to require some familiarity with servers. And people who want to write their experiments in javascript often want to try things out first so it seems something like XAMPP might be a good resource. *Exercise* to download and get the XAMPP server running? Need to expand this. An exercise with JavaScript? This site has a simple bit of code for throwing an image on the screen. Then use the XAMPP server to test it? Require changing the image? Animate a button to toggle or get a random image?

7 What are some of the challenges that computational tools pose for reproducibility?

Classroom Exercise

Read and discuss: A toolkit for transparency ….

8 Knowledge Management ([[https://en.wikipedia.org/wiki/Zettelkasten][Zettlek

1. Org-roam
2. Dendron - works with VS Code
3. Obsidian

9 Reference Management

- What is Crossref?
- What is a doi?
- What is an orcid?
- What is zotero?
- What is csl?

> Classroom Exercise
>
> Locate and download the csl file for the current APA style and also for one at least one other non-apa style.

9.1 Using References With Quatro and VS Code

9.2 Using References With Overleaf

VS Code is not your only option. Overleaf also use ".bib" files to hold references, though some of the other syntax is different.

Here are some links to get you started with references in overleaf. 1. Linking Overleaf and Zotero 2. [Connecting Zotero and Quarto](https://quarto.org/docs/visual-editor/technical.html#citations-from-zotero)

9.3 Poster Presentations ? (MOVE THIS SECTION)

[Scientific Posters with Quarto](https://github.com/quarto-ext/typst-templates/tree/main/poster)

LaTeX and Knitr in Overleaf][knitr]] with overleaf. A scientific poster with Overleaf?

10 Testing Your Javascript

10.1 Run your own server locally

Running a lab now seems to require some familiarity with servers. And people who want to write their experiments in javascript often want to try things out first so it seems something like XAMMP is a good tool to know about.

10.1.1 XAMPP

XAMMP

10.1.1.1 Downloading

Download link

10.1.2 In Lab Experiments

- Making stuff appear on monitors.
- What is OpenGL?
- Use pygame to make some simple visual experiments.
- Have a beauty contest the following week to see what people have been able to make?
- Make sure that people are using venv

10.1.3 Online Experiments

An exercise with JavaScript? This site has a simple bit of code for throwing an image on the screen.

Can you use the XAMPP server to test it? Require changing the image? Animate a button to toggle or get a random image? WIP HERE

11 Databases and management

- What is it
- A book on DuckDB - database stuff
- MariaDB This SQL like, and open source. Might be easier to get started with and still be SQL enough to give them some professional benefits. I was thinking we could get some data online, often they come as CSV's and read it into the database? This is one example how. A blog that compares MariaDB to SQL. A quickie tutorial.
- Why would I want to use a relational database over a csv file (or R data frame or similar)? This could be a class exercise and discussion.
- *Exercise* Download MariaDB

11.0.1 Datascience

- one person's roadmap for non-cs grads

11.0.2 Grant Funding

At the moment I am not sure if will have time for this. But thinking of having the students review the peer review manual for NSERC Discovery Grants. Then have them each write a minimal proposal. Assign the proposals to members of the class, and then hold our own reviewers meeting to decide which projects to fund. Top grants get performed for experiments? Get extra-credit points?

12 Summary

In summary, I have not gotten to writing a summary yet.

A Version Control

A.1 Getting Started - Some old videos on git from PSYCH363/310

A few years ago when I was teaching a more elementary version of some of this material (PSYCH363) I included some videos on git and using Github. Some components of these videos may be a bit out of date, and a few things about the interface have changed. Still they may be a reasonable starting point for those of you who feel you need a refresher on version control. Note that for PSYCH390 I am assuming you can either already use version control or figure it out on your own. I will give some class time for getting things working, but mostly you will have to pursue this on your own. The notes here are intended to be prompts and guides, but not to be comprehensive.

You will notice that in the videos I talk a lot about Linux. We are not using that for this course so you can ignore those references. However, I do demonstrate the use in the *terminal*. That is available to all of you regardless of operating system. You will just have to find your own OS version of it.

Frequenly you will see this referred to as a command line tool ("cli"). It is now easier to get this this working on Windows and OSX than it used to be. For OSX you can also check out **Homebrew**. For the adventurous Windows user you can look at WSL2.

A.2 An overview of git and Github

https://vimeo.com/456349738

A.3 You have choices - other version control system

1. Mercurial
2. Darcs
3. CVS
4. Subversion
5. Pijul

Each has their own fans. CVS and Subversion are more legacy options, but you will still see them occasionally. Darcs is more of an experiment than a broadly used system. Mercurial used to be the cool kid, but now seems eclipsed by *Pijul*. That is the one for experimental users.

A.4 Git is not Github

Git is the version control software. Github is a very popular place to host your publically accessible git repository, but it is far from your only option. You can host elsewhere.

1. OSF.io For scientists OSF.io seeks to make itself a way to host scientific projects and their data. Trivia question? Do I have any repositories on Osf.io?
2. Sourceforge An oldie, but still used.
3. Bitbucket
4. Gitlab The university provides you with a gitlab account: https://git.uwaterloo.ca
5. Codeberg If you believe in freedom and neck beards.

A.5 Things you should do

1. Install git
2. Get an account on github
3. Fork the course repository
4. Clone the course repository to your laptop
5. Set up my version as an additional **remote**.
6. See if you can make a "pull request" to me. You may find that you have to set up an **ssh key** in order to efficiently pull and push to Github. Github has very clear instructions on how to go about doing this.

A.6 But first

If you have not used git before you will have to configure git. It needs to know who you are and how to reach you at least.

You can execute commands like this in your terminal:

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

More on these configuration options can be found in a nice online book.

A.7 Understanding Git and the Workflow in Pictures

When starting to use git I found it very confusing to tell what was where and what direction things were going when I pushed and pulled. I found pictures helpful adjuncts to the prose descriptions. A few helpful illustrations to the distinction of clones, forks, pull, push, and pull request can be found here.

I also made a little video to try and illustrate some of these concepts a few years ago. Read for the gist and not for the specific examples.

https://vimeo.com/456349595

A.8 Principal Terms to Learn First

- **Fork:** A copy of one github repository to another **github** repository.
- **Clone:** A copy of one **git** repository to another **git** repository. The first repository might be hosted on github, but the second one, the *cloned* one exists on a local machine. In your case this is probably your laptop.
- **Remote:** This is a repository that you are following. You will typically *pull* from these, but your *push* permissions may be limited depending on the distinctions between forks and clones, and whether you own the remote or someone else does. You can have more than one remote.
- **Pull:** the transfer of information and changes **from** one repository incorporated into another. This is how you get the new information from a remote transported to a local repository that you control.
- **Push:** this is the transfer of information **to** a repository you control (or have permissions to push to) from another repository that you control. This is often from your local laptop version to the hosted repository (your fork) on github.
- **Pull Request:** When you have information or changes that you think would be helpful to a remote you do **not** have push permissions for then you can request that the owner of that repository pull in your changes. This is a formal process called a pull request. It is primarily a github concept and not a git concept.
- **Branch:** within a repository the development of the code may be proceeding in a few different directions at the same time. The principal production branch is conventionally called **master**. And the principal repository that is the main, shared one is called **origin**. We will not be working with branches in our course, but those terms do show up in commands.

All of these "basics" are covered in detail in the book Pro Git (available on line).