Quarto Basics

Britt Anderson

2025-01-01

Table of contents

Pı	eface	9			5											
1	Intr	oductio	on		6											
2	The	The role of the computer as a tool in psychological research														
	2.1	Resea	arch Project Stages		7											
		2.1.1	Preliminaries		7											
		2.1.2	Do the research		8											
		2.1.3	Analyzing the Research		8											
		2.1.4	Diseminate the Research		8											
		2.1.5	Make Your Research Open		9											
3	Оре	n Rese	earch Practices Continued		10											
	3.1	Fair F	Principles		10											
	3.2	Lab N	Notebooks		10											
	3.3 Bibliographic Tools and Exercises															
	3.4	IDEs	and Writing up your Research		11											
		3.4.1	VSCode		11											
		3.4.2	Overleaf		11											
4	Cod	ling			13											
		4.0.1	IDEs		13											
		4.0.2	Languages		14											
5	Wri	ting Sc	cientific Documents		20											
		5.0.1	Scientific Publishing Tools		20											
6	In L	ab Exp	periments		21											
		6.0.1	Online Experiments		21											
7	Wh	at are s	some of the challenges that computational tools pose for reproducib	ility?	22											
8	Kno	wledge	e Management (Zettlekasten)		23											
9	Refe	erence	Management		24											
	9.1	Using	g References With Quatro and VS Code		24											

	9.2 9.3	Using References With Overleaf	
10	Find	ing Citations	25
11	Tost	ing Your Javascript	26
		•	26
	11.1		26
			26
		•	26
12	Larg	e Language Models	27
	12.1	Run locally	27
	12.2	How to?	27
		12.2.1 Key Features of GPT4ALL	27
	12.3	How-to	28
	12.4	Using LLMs for Dynamic Research	28
	12.5	Why Might You Want To Do This?	29
	12.6	Next steps	29
13	Data	•	30
			30
		13.0.2 Grant Funding	30
14	Sum	mary	31
Αŗ	pend	dices	32
Δ	Vers	ion Control	32
•		Getting Started - Some old videos on git from PSYCH363/310	
	A.2	9	32
		9	32
		· ·	33
	A.5	Things you should do	33
	A.6	But first	33
	A.7		34
	-	Principal Terms to Learn First	34
В	Just	a place for some notes for installing things on Mac	35
C	Ema		36
	C.1	init files	36
	C.2	pdf-mode	36

D	Qua	rto																	38
	C.5	emacs					 					 •		•			•		37
	C.4	orgmode r	related.				 												36
	C.3	keys and l	keyboard	ds .			 												36

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 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 AD-DLINKHERE.

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/authorinstructions/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?

4.0.2 Languages

4.0.2.1 R

4.0.2.1.1 Installation

- OSX
- Windows
- Linux

Download the CORRECT R

Follow the appropriate instructions for your operating system.

Test your installation by opening a TERMINAL and typing the capital letter R. You should end up in an interpreter. You can quit with quit().

Restart R in a TERMINAL and install a package. A package is a collection of code, often much of it written in R, that is used for doing things in R. For example try:

```
install.packages("tidyverse")
```

The tidyverse is a very popular collection of R code that itself will depend on many additional pieces of R code and other packages, some R and some not.

If that went well, you need to make sure you restart VS Code and then click on the little set of squares at the left to install the R extension for VS Code.

If all goes well check out the book on using R that is itself written in Quarto, by the programmer who authored the tidyverse. And make sure you can create a file in VSCode as a quarto document, cut and paste in some R code, and see the whole thing compile to a web page. This after all will be a homework.

```
if(!require(tidyverse)){
   install.packages("tidyverse")
}
Loading required package: tidyverse
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr 1.1.4
                v readr
                            2.1.5
v forcats 1.0.0 v stringr
                            1.5.1
v ggplot2 3.5.1 v tibble 3.2.1
                            1.3.1
v lubridate 1.9.4
                  v tidyr
v purrr
        1.0.2
-- Conflicts ----- tidyverse conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()
              masks stats::lag()
```

i Use the conflicted package (http://conflicted.r-lib.org/) to force all conflicts to become

```
if(!require(palmerpenguins)){
   install.packages("palmerpenguins")
}
```

Loading required package: palmerpenguins

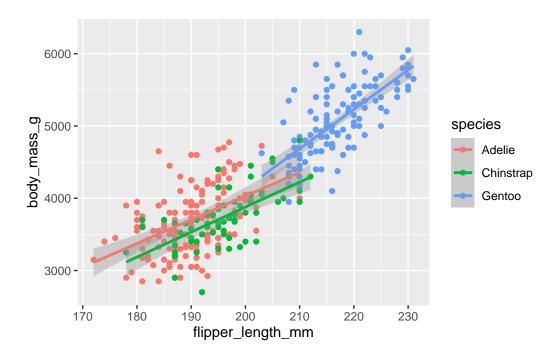
```
library(tidyverse)
library(palmerpenguins)

ggplot(
   data = penguins,
   mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)
) +
   geom_point() +
   geom_smooth(method = "lm")
```

[`]geom_smooth()` using formula = 'y ~ x'

Warning: Removed 2 rows containing non-finite outside the scale range (`stat_smooth()`).

Warning: Removed 2 rows containing missing values or values outside the scale range (`geom_point()`).



4.0.2.2 Python

4.0.2.2.1 Installation

Installation Instructions

Install the mathplotlib package.



setting up a venv in vscode and python

4.0.2.2.2 Testing

⚠ Warning

You may need to install the "reticulate" package for R if you want to run both python and R code in the same document as I am trying to do here.

```
import matplotlib.pyplot as plt
import numpy as np
# fake data
np.random.seed(19680801)
data = np.random.lognormal(size=(37, 4), mean=1.5, sigma=1.75)
labels = list('ABCD')
fs = 10 # fontsize
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(6, 6), sharey=True)
axs[0, 0].boxplot(data, tick_labels=labels)
{'whiskers': [<matplotlib.lines.Line2D object at 0x110e5b610>, <matplotlib.lines.Line2D object
axs[0, 0].set_title('Default', fontsize=fs)
axs[0, 1].boxplot(data, tick_labels=labels, showmeans=True)
{'whiskers': [<matplotlib.lines.Line2D object at 0x110ea5950>, <matplotlib.lines.Line2D object
axs[0, 1].set_title('showmeans=True', fontsize=fs)
axs[0, 2].boxplot(data, tick_labels=labels, showmeans=True, meanline=True)
{'whiskers': [<matplotlib.lines.Line2D object at 0x110f14190>, <matplotlib.lines.Line2D object
axs[0, 2].set_title('showmeans=True,\nmeanline=True', fontsize=fs)
axs[1, 0].boxplot(data, tick_labels=labels, showbox=False, showcaps=False)
{'whiskers': [<matplotlib.lines.Line2D object at 0x110f16ad0>, <matplotlib.lines.Line2D object
tufte_title = 'Tufte Style \n(showbox=False,\nshowcaps=False)'
axs[1, 0].set_title(tufte_title, fontsize=fs)
axs[1, 1].boxplot(data, tick_labels=labels, notch=True, bootstrap=10000)
```

```
{'whiskers': [\text{matplotlib.lines.Line2D object at 0x110f84050>, \text{matplotlib.lines.Line2D obje}

axs[1, 1].set_title('notch=True,\nbootstrap=10000', fontsize=fs)

axs[1, 2].boxplot(data, tick_labels=labels, showfliers=False)

{'whiskers': [\text{matplotlib.lines.Line2D object at 0x110f86350>, \text{matplotlib.lines.Line2D obje}

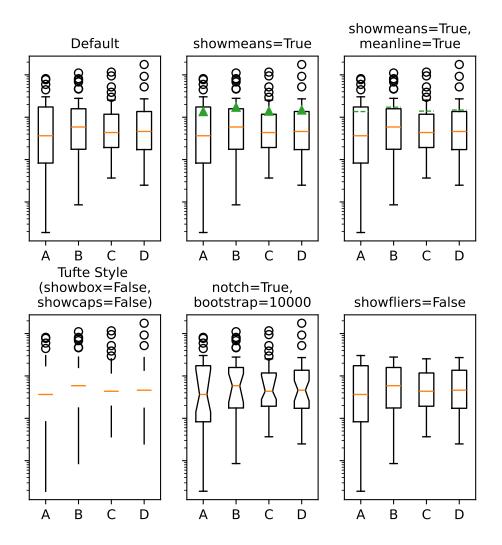
axs[1, 2].set_title('showfliers=False', fontsize=fs)

for ax in axs.flat:
    ax.set_yscale('log')
    ax.set_yticklabels([])

[Text(0, 0.001, ''), Text(0, 0.01, ''), Text(0, 0.1, ''), Text(0, 1.0, ''), Text(0, 10.0, '')]

[]
[]
[Text(0, 0.001, ''), Text(0, 0.01, ''), Text(0, 0.1, ''), Text(0, 1.0, ''), Text(0, 10.0, '')]

fig.subplots_adjust(hspace=0.4)
```



4.0.2.3 Javascript

5 Writing Scientific Documents

Papers Posters Websites Books Math Formulas Images Graphs References Analyses/Code

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 Tools

What is a markup language?

What are the advantages/disadvantages of using a tool other that Word for writing up your research?

Write once use everywhere.

Some of the available options.

- LaTeX This was the standard for several decades, and used more by mathematicians than others, but today it seems limited in that it produces mostly pdfs, and many people also want their files to export to html (webpages)
- A guide to LaTex for Word Users (pdf)
- The descendants of markdown
- Quarto (we will skip Rmd and since we are using its successor qmd. Both are related to the people who wrote RStudio,knitr, and are now known as Posit).

Here are some of the helper links for things we can do with Quarto: - journal formats try the Elsevier format for this course. A lot of psych journals are Elsevier owned. - presentation - Talking to databases. - Make a website/blog for your work or lab - And more ...

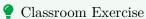
6 In Lab Experiments

- 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?



Read and discuss: A toolkit for transparency

8 Knowledge Management (Zettlekasten)

- 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 Finding Citations

- Semantic Scholar
- OpenAlex
- Google Scholar
- Consensus
- Undermind

11 Testing Your Javascript

11.1 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.

11.1.1 XAMPP

XAMMP

11.1.1.1 Downloading

Download link

11.1.2 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

11.1.3 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

12 Large Language Models

12.1 Run locally

Why? Own your own data.

Security and Privacy.

Research applications you may not want participants' data fed to openAI or other proprietary vendors.

12.2 How to?

There are many methods to running LLMs locally on your own hardward. I have chosen GPT4All. It can run under OSX, Windows, and Ubuntu (linux).

12.2.1 Key Features of GPT4ALL

According to the website https://getstream.io/blog/best-local-llm-tools/:

GPT4All can run LLMs on major consumer hardware such as Mac M-Series chips, AMD and NVIDIA GPUs. The following are its key features.

- Privacy First: Keep private and sensitive chat information and prompts only on your machine.
- No Internet Required: It works completely offline.
- Models Exploration: This feature allows developers to browse and download different kinds of LLMs to experiment with. You can select about 1000 open-source language models from popular options like LLama, Mistral, and more.
- Local Documents: You can let your local LLM access your sensitive data with local documents like .pdf and .txt without data leaving your device and without a network.
- Customization options: It provides several chatbot adjustment options like temperature, batch size, context length, etc.

• Enterprise Edition: GPT4ALL provides an enterprise package with security, support, and per-device licenses to bring local AI to businesses.

It is also a very popular LLM for self-hosting (second only to Ollama, which you should also try out).

12.3 How-to

From the GPT4all download page download the version for your operating system.

Follow the download instructions appropriate to your system and pay particular attention to any warnings and make sure that you have sufficient freespace. This is a big application and the model parameters are also quite large.

After you have successfully downloaded and installed the program checkout the quickstart guide.

- 1. Start Chatting
- 2. Install a Model
- 3. I tried "Llama 3.2 1b Instruct"
- 4. Load the model
- 5. Start chatting

12.4 Using LLMs for Dynamic Research

What are the mechanics we need for this? We need our model to accept input and generate output and we need a method to transfer the elements of the conversation back and forth. The first stage is to think of the components we have used so far and how we might stitch them together for this task.

It is also a useful transitional step to work incrementally. First, maybe try to set up an interface to allow someone to talk to the model via a web interface. Then you might be able to figure out how to use GET and POST from PHP to communicate between two web servers. There are definitely better ways to do that, but it is a fairly direct starting point. If you want an additional challenge perhaps try to use the tool: curl.



Classroom Exercise

Download and implement the above.

Start the GPT4All Server

See if you can get your GPT4All model to talk to another group's model.

12.5 Why Might You Want To Do This?

Durably reducing conspiracy beliefs through dialogues with AI

https://www.youtube.com/watch?v=qD1fnYDTbHM

This LinkedIn post talks about ChatGPT integration. What would you have to change to get it working with your local server?



? Class Question

What is a RAG in this context?

12.6 Next steps

Recently (Duan, Li, and Cai 2024) has published an R package to make all of this easier. The github repository has the library, a "read-me" and installation instructions. Our class activity is to see if anyone can get this up and working by next week.

13 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

13.0.1 Datascience

• one person's roadmap for non-cs grads

13.0.2 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?

14 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).

B Just a place for some notes for installing things on Mac

C Emacs

I used homebrew awhile ago for this and forgot the details.

C.1 init files

```
Seems to live at ~/.emacs.d/init.el
```

Added these lines and restarted to get the basic package set up working.

```
(require 'package)
(add-to-list 'package-archives '("melpa" . "https://melpa.org/packages/") t)
(package-initialize)
(require 'use-package)
(setq use-package-always-ensure t)
```

C.2 pdf-mode

Do a package-list-packages to get a refesh and install (mark with an "i") the pdf-tools package from melpa. Then you "x" to execute the installation. Afterwards you need to run M-x pdf-tools-install. This can take awhile.

C.3 keys and keyboards

Several emacs shortcuts conflict with defaults on mac. You can change these in the system utilities for keyboard shortcuts. Just search them out as they arise and turn off the mac ones. The ctrl key is ^ in their printed list.

C.4 orgmode related

1. for exporting html needed to install package htmlize.

C.5 emacs

There is a quarto editing mode which I installed through the packaging tools.

D Quarto

Find the download here. The demo needs matplotlib and plotly. So, you need python which I got from brew. brew install python-matplotlib will pull in a lot of other libraries as well. There are some tricky steps here. Plotly is not easily gotten via homebrew, and I did not want to mix up my system by also using pip now that I was working hard under homebrew. This brings up the idea of virtual environments. That is generally a good idea, especially with python in my opinion. While doing some searching I came across pipx that seems to try and do both. Allow you to install via homebrew (the pipx) and then use pipx to create virtual environments that can isolate the libraries, but expose the binaries.

```
brew install pipx
pipx ensurepath
sudo pipx ensurepath --global # optional to allow pipx actions with --global argument
brew update && brew upgrade pipx.
```

Then, because I could not get plotly in homebrew, I created a **venv**.

```
python3 -m venv /Users/britt/p390venv
cd p390venv
source bin/activate
pip install plotly
```

I needed to end up also doing pip install for matplotlib and jupyter from within the venv. Then I simply cut and pasted the demo document and followed their recommendation.

The file I tested as "test.qmd" is ...

For a demonstration of a line plot on a polar axis, see Figure D.1.

```
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
```

```
subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

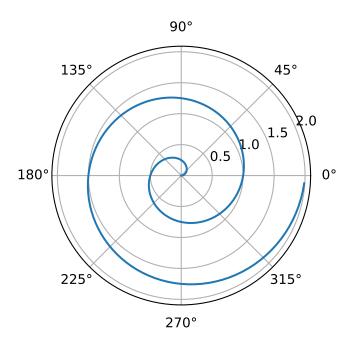


Figure D.1: A line plot on a polar axis

And the command was quarto render test.qmd --to html.

And if you brew install texlive then export to pdf works well too.

Duan, Xufeng, Shixuan Li, and Zhenguang G. Cai. 2024. "MacBehaviour: An r Package for Behavioural Experimentation on Large Language Models." *Behavior Research Methods* 57 (1). https://doi.org/10.3758/s13428-024-02524-y.