Hidden Curriculum

INT 93

Michael Topper

What is Hidden Curriculum?

Hidden Curriculum:

- Anything useful to research, but never taught
- Time saving - work smarter, not harder
- Reproducible

Main Topics:

- File Organization
  - Research projects are huge tasks
  - Time consuming
  - Likely to forget what you did along the way


Your Future Self:

Your Future Self:

Make your future-self happy:

- Your future-self is lazy.
- Your future-self is forgetful.
  - Key decisions?
  - Anomalies in the data?

Your future-self does not want to:

- Figure out which order to run files
- Understand what each line of code does
- Look for files across different systems
- Input new results if the data changes

File Organization:

How Should I Organize My Files?

- An unfamiliar person should be able to:
  - Understand your structure
  - Know where the data is
  - Know how the data was created
  - Know which files determine which results
  - Reproduce the results exactly

Important Methods:

- Folders - large tasks/smaller tasks
- Naming Convention - Be consistent!
- README
- Version Control (beyond this course)

Organizing with Folders

SEE EXAMPLE

Personal Preferred Structure:

- analysis_data
- created_data
- raw_data
- figures
- tables
- code
- presentations
- paper
- literature_review

You can use any structure, but keep it simple!

File Naming

File Naming:

Naming Needs to Be:

- Machine Readable
  - Sometimes, spaces matter (Linux/Mac vs. Windows)
- Human Readable
  - Understand your own file quickly
- Default Ordering (less important)
  - Your files work well with the default ordering scheme

Typical Naming Conventions:

- snake_case (preferred!)
  - Spaces are underscores
  - All text is lowercase
- camelCase (also good!)
  - No spaces
  - Separate words by CapitalLetters

Examples:

snake_case:

- Cleaning raw Data -> clean_raw_data
- Table 1 analysis -> table_1_analysis

CamelCase:

- Cleaning raw Data -> CleaningRawData
- Table 1 analysis -> Table1Analysis

Human Readable:

The more information you give in the title, the easier it will be for your future-self and others.

Less Clear:

snake_case:

- Cleaning raw Data -> clean_raw_data
  - Which data?
- Table 1 analysis -> table_1_analysis
  - I don't know what Table 1 does.
  - Would need to open file

More Clear:

snake_case:

- Clean NLSY 1996 Cohort -> clean_nlsy_1996_cohort
- Table 1 Main Results Regressions -> table_1_main_results_regressions

README

README

Definition:

A special file that gives a "run-down" of the other files in the folder.

Elements:

- .txt file (so anyone can open this)
- One in every folder with multiple files
- Describes the main purpose of each file/folder contained
- Clear - everyone should know what files do what after reading
- Update these frequently:
  - Hard to do, realistically, before you take a long break from the project.

Coding Files

Coding Files:

Coding Files (Surface)

- Number when ordering needed
- Make informative file names (as discussed)

Coding Files (Within)

- Use comments to explain code to readers
- Make informative names!!!!!!
  - Bad: r1
  - Good: regression_income_race_1
- Use Whitespace Effectively
- Keep files short (~100-150 lines)
  - One task per-coding file

Coding: Bad

```
x=1
y=2
if(x≠y){
  print("They are not equal")
}else{print("They are equal")}
```

- No whitespaces: hard on the eyes
- No comments: time-consuming to read
- Uninformative naming of variables

Coding: Good

```
## Defining Variables
first_number = 1
second_number  = 2

## Checking if numbers are equivalent
if (first_number == second_number) {
  print("They are equal")
}
else{
  print("They are not equal")
}
```

Case Study: Michael's Work

Does Michael practices what he preaches?

Link to Michael's job market paper Github repository

Assignment: Redo a Project

Make an Old Project Reproducile

- Take an old project and use the tools we talked about to make it easy for an outsider to understand.
  - Rename, restructure
  - READMEs
- If you do not have one of these, create a pseudo-file structure for your current project with some fake files.

Further Reading/Reference

- Slides from Jenny Bryan on Naming
- Slides on Code Quality