Computing in the Statistics Curriculum:
Lessons Learned from the Educational Sciences

Amelia McNamara, Andrew Zieffler, Matthew Beckman, Chelsey Legacy, Elle Butler Basner, Robert delMas, & V.N. Vimal Rao

PennState

Computing is fundamental to contemporary statistical practice and scientific inquiry and should be explicitly taught

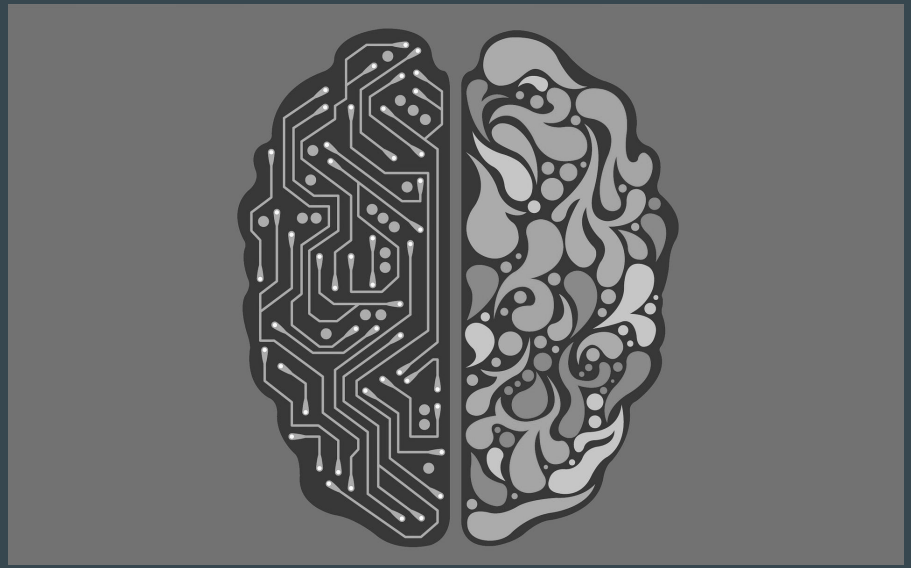
(ASA, 2017; Horton, 2015; NASEM, 2018; Nolan & Temple Lang, 2010)

Cognitive Load: Impediment to Learning

There is a finite amount of information that can be processed or stored in working memory at a time (cognitive load).

Different types of cognitive load (Hermans, 2021):

- *Intrinsic:* characteristics of the information being learned.
- *Extraneous:* the way information is presented.



Statistical Computing and Cognitive Load

- Statistical computing adds cognitive load to the learning process in statistics (e.g., Woodard & Lee, 2021)
    - Computational considerations
        - Syntax
        - Syntactical structure
        - Debugging
        - Computational thinking
    - Coding seems to be difficult for many students

- We can work to manage and lessen cognitive load by thoughtfully considering the specific coding content we teach and how we teach it.

Cognitive *Un*-loading: Make Purposeful Pedagogical Decisions

Pedagogical decisions need to be made about coding content.

- What logistical considerations do you need to account for?
- What will be taught (scope)?
- How will it be sequenced?

Example Pedagogical Decisions

- How will students compute in the course?
  - Desktop / Cloud / Both
- Where and when do students need practice with code?
  - In-class / out-of-class
  - Individual / group
  - Templates / Blank documents (RMD, R script)
- How will coding be introduced in class?
  - Live coding/ Worked examples / Group activities
- What code content do you start the course with?
  - Data structures (e.g., vectors, data frames) / “Cake” (data visualization, EDA)

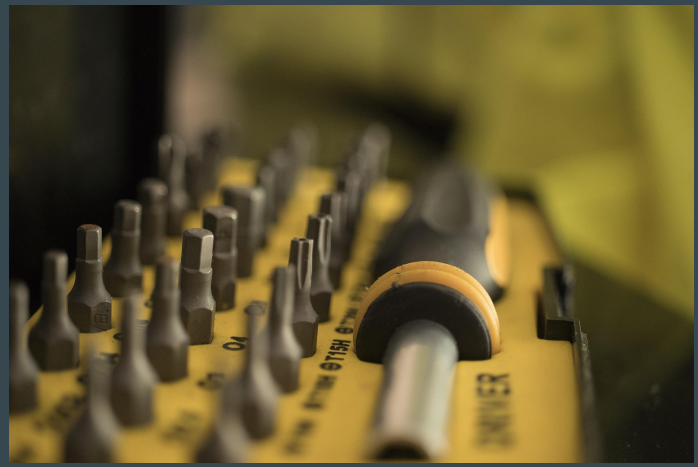
Cognitive *Un*-loading: Use Consistent Syntactic Structure

Using code with the same syntactic structure (common grammar) lessens cognitive load

- Can focus on learning new functions (verbs) and their purpose

Some syntactic structures can emphasize the relationship between syntax and concepts

- E.g., Roles of variables



```r
library(psych)
library(BHH2)
nhanes2017= read.csv("nhanes2017.csv", as.is = F)
table(nhanes2017$exerciseGT60)
par(mfrow = c(1, 2))
hist(nhanes2017$pulse[nhanes2017$exerciseGT60 == "YES"], xlim = c(min(nhanes2017$pulse), max(nhanes2017$pulse)))
hist(nhanes2017$pulse[nhanes2017$exerciseGT60 == "NO"], xlim = c(min(nhanes2017$pulse), max(nhanes2017$pulse)))
par(mfrow = c(1, 1))
tapply(nhanes2017$pulse, nhanes2017$exerciseGT60, summary)
tapply(nhanes2017$pulse, nhanes2017$exerciseGT60, describe)
t.test(pulse ~ exerciseGT60, data = nhanes2017)
```

Authentic example: Analyze the difference in pulse by exerciseGT60 from NHANES

```r
# Load libraries
library(psych)

# Import data
nhanes2017= read.csv("nhanes2017.csv", as.is = FALSE)

# Get levels and sample sizes
table(nhanes2017$exerciseGT60)

# Plot histograms
par(mfrow = c(1, 2))
hist(nhanes2017$pulse[nhanes2017$exerciseGT60 == "YES"], xlim = c(min(nhanes2017$pulse), max(nhanes2017$pulse)))
hist(nhanes2017$pulse[nhanes2017$exerciseGT60 == "NO"], xlim = c(min(nhanes2017$pulse), max(nhanes2017$pulse)))
par(mfrow = c(1, 1))

# Compute summary statistics
tapply(nhanes2017$pulse, nhanes2017$exerciseGT60, describe)

# Carry out two-sample t-test
t.test(pulse ~ exerciseGT60, data = nhanes2017)
```

```r
# Load libraries
library(mosaic)

# Import data
nhanes2017 = read.csv("nhanes2017.csv", as.is = FALSE)

# Get levels and sample sizes
tally(~ exerciseGT60, data = nhanes2017)

# Plot histograms
histogram(~ pulse | exerciseGT60, data = nhanes2017)

# Compute summary statistics
favstats(~ pulse | exerciseGT60, data = nhanes2017)

# Carry out two-sample t-test
t_test(~ pulse | exerciseGT60, data = nhanes2017)
```

Mosaic Package
http://www.mosaic-web.org/mosaic/

The Most Important Template

The following template is important because we can do so much with it.

(~ , data =)

It is useful to name the components of the template:

goal (y ~ x , data = mydata)

We're hiding a bit of complexity in the template, and there will be times that we will want to gussy things up a bit. We'll indicate that by adding ... to the end of the template. Just don't let ... become a distractor early on.

goal (y ~ x , data = mydata , ...)

Questions to help you revise how you teach coding:

- What syntactic structure makes the most sense for my students/course/goals?
- Is the code being presented to students consistent in its structure?
- How does new code connect with previous content?
- Will students see/use this code more than once?
    - **“Stitch in time saves 9”**
- How will students encounter code?
    - Live coding, scaffolded documents, cheatsheet?

Resources and Places to Start

- Introductory statistics labs in R, Amelia McNamara (formula or tidyverse)
- Speaking R, Amelia McNamara (guidance for live coding and reading code)
- Statistical Modeling and Computation for Educational Scientists, Andrew Zieffler (tidyverse)
- Simulation Based Inference, Randy Pruim (formula)
- Data Science in a Box, Mine Çetinkaya-Rundel (tidyverse)

Chelsey Legacy
legac006@umn.edu

Amelia McNamara
mcna6887@stthomas.edu

Elle Butler Basner
esb5324@psu.edu

Thank you!

Andrew Zieffler
zief0002@umn.edu

Robert delMas
delma001@umn.edu

V.N. Vimal Rao
rao00013@umn.edu

Matthew Beckman
mdb268@psu.edu

References

American Statistical Association. (2017). Curriculum guidelines for undergraduate programs in statistical science. American Statistical Association. https://www.amstat.org/asa/education/home.aspx/curriculumguidelines.cfm

Ayres, P. (2016). Impact of reducing intrinsic cognitive load on learning in a mathematical domain. Cognitive Psychology, 20(3), 287-298. https://onlinelibrary.wiley.com/doi/abs/10.1002/acp.1245

Hermans, F. (2021). The Programmer's Brain. https://www.manning.com/books/the-programmers-brain

Horton, N. J. (2015). Challenges and opportunities for statistics and statistical education: looking back, looking forward. *The American Statistician*, *69*(2), 138–145. https://doi.org/10.1080/00031305.2015.1032435

National Academies of Sciences, Engineering, and Medicine. (2018). Data science for undergraduates: Opportunities and options. The National Academies Press. https://doi.org/10.17226/25104.

Nolan, D., & Temple Lang, D. (2010). Computing in the statistics curricula. *The American Statistician*, *64*(2), 97–107. https://doi.org/10.1198/tast.2010.09132

Sweller, J. (2011). Cognitive load theory. In J. P. Mestre & B. H. Ross (Eds.), *Psychology of Learning and Motivation*, 55, 37-76. Academic Press. https://doi.org/10.1016/B978-0-12-387691-1.00002-8

Woodard, V., and Lee, H. (2021). How students use statistical computing in problem solving. *Journal of Statistics and Data Science Education, 29*, 1–18. https://doi.org/10.1080/10691898.2020.1847007

Photo Credits

https://edtechmagazine.com/k12/article/2020/04/how-teachers-can-better-manage-classroom-computing

https://pixabay.com/illustrations/artificial-intelligence-ai-robot-2228610/

https://pixabay.com/illustrations/start-beginning-arrows-direction-5896294/

https://bangaloremirror.indiatimes.com/bangalore/cover-story/starting-from-scratch/articleshow/74016629.cms

https://pixabay.com/photos/hardware-screwdriver-kit-4649058/