Teaching computer and data science with literate programming tools:

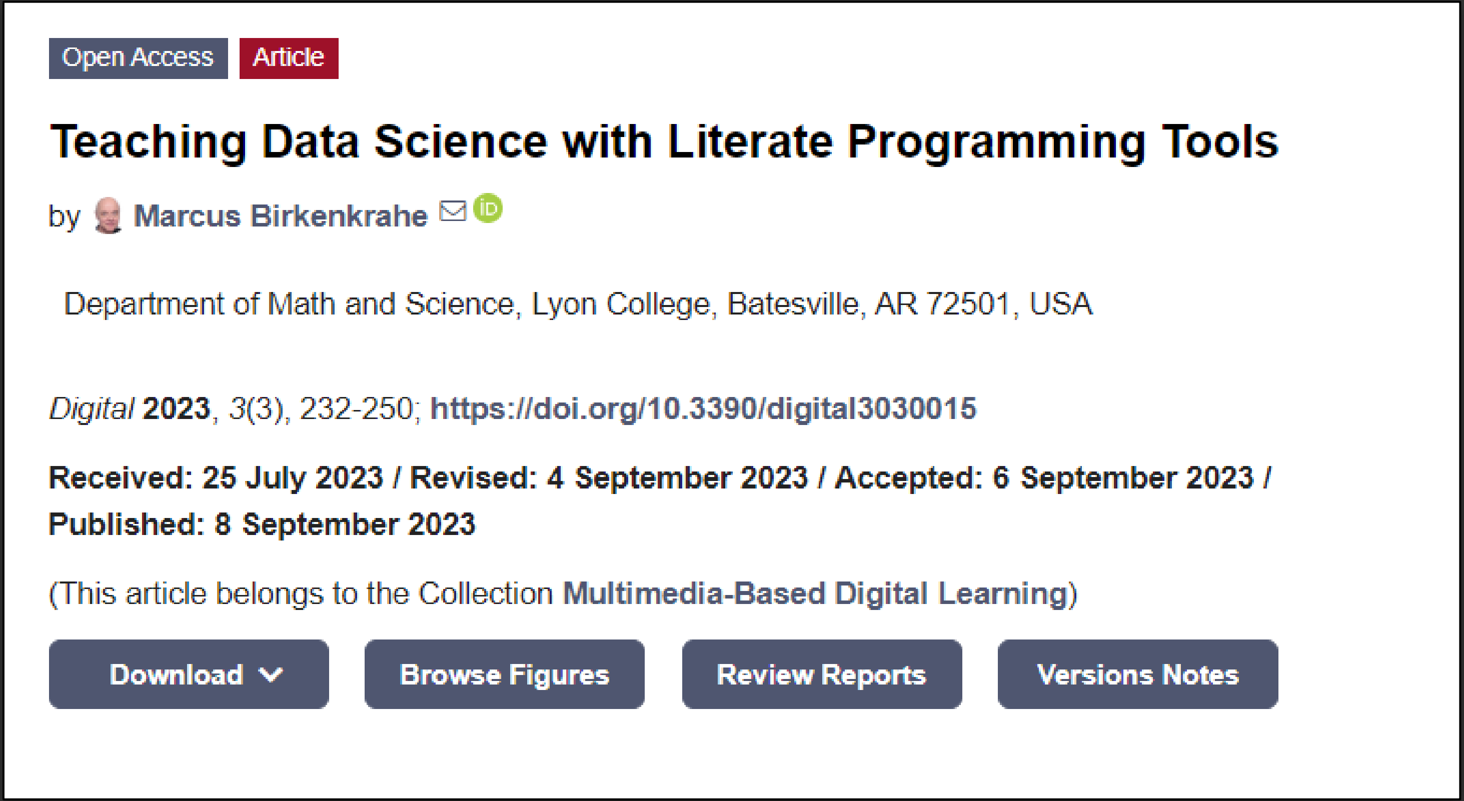
How I made Emacs + Org-mode mandatory in all my courses

Marcus Birkenkrahe - Lyon College

EmacsConf 2023

https://emacsconf.org/2021/talks/teach/

https://doi.org/10.3390/digital3030015



Teaching Data Science with Literate Programming Tools

by Marcus Birkenkrahe

Department of Math and Science, Lyon College, Batesville, AR 72501, USA

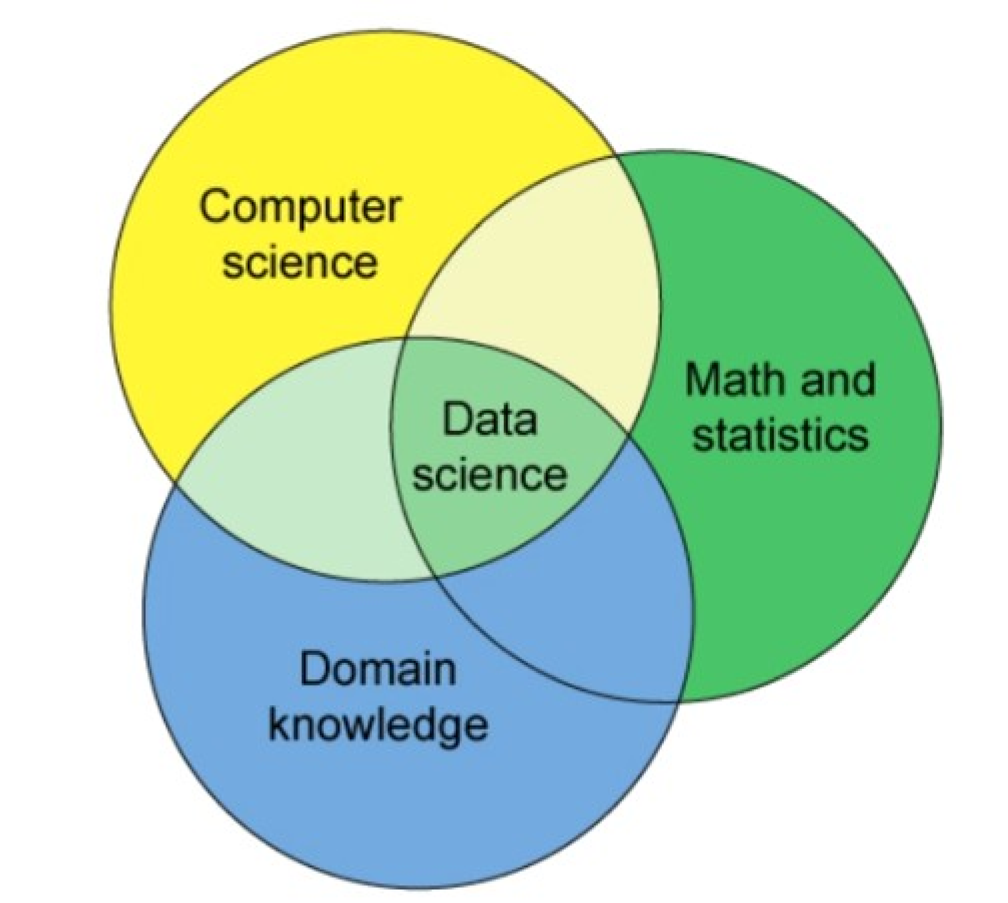
Digital **2023**, *3*(3), 232-250; https://doi.org/10.3390/digital3030015

Received: 25 July 2023 / Revised: 4 September 2023 / Accepted: 6 September 2023 / Published: 8 September 2023

(This article belongs to the Collection Multimedia-Based Digital Learning)

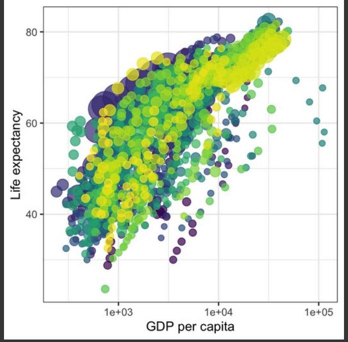
What is data science?

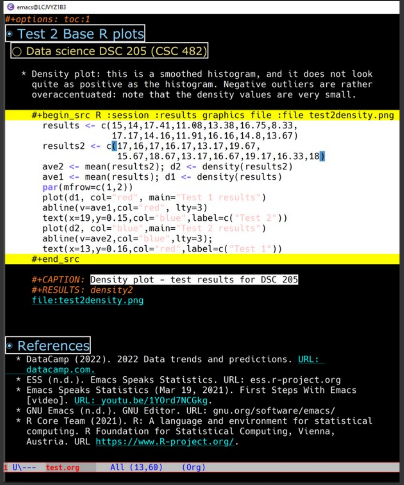
Computing + Math/Stats + Your Stuff



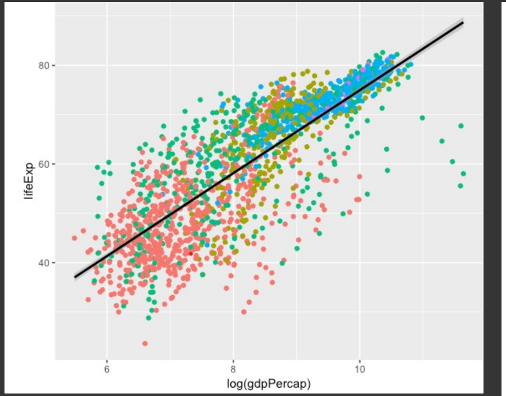
What is data science?

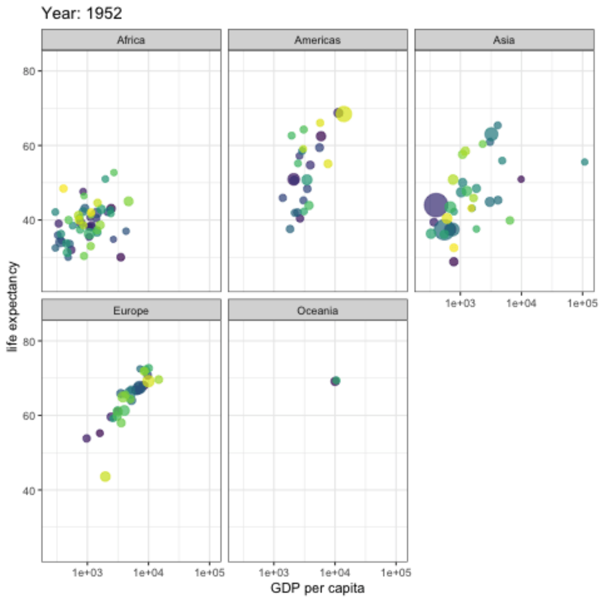
data + code + stats = story





```
#+options: toc:1
• Test 2 Base R plots
 ○ Data science DSC 205 (CSC 482)

  * Density plot: this is a smoothed histogram, and it does not look
    quite as positive as the histogram. Negative outliers are rather
    overaccentuated: note that the density values are very small.

    #+begin_src R :session :results graphics file :file test2density.png
      results <- c(15,14,17.41,11.08,13.38,16.75,8.33,
                   17.17,14.16,11.91,16.16,14.8,13.67)
      results2 <- c(17,16,17,16.17,13.17,19.67,
                    15.67,18.67,13.17,16.67,19.17,16.33,18)
      ave2 <- mean(results2); d2 <- density(results2)
      ave1 <- mean(results); d1 <- density(results)
      par(mfrow=c(1,2))
      plot(d1, col="red", main="Test 1 results")
      abline(v=ave1,col="red", lty=3)
      text(x=19,y=0.15,col="blue",label=c("Test 2"))
      plot(d2, col="blue",main="Test 2 results")
      abline(v=ave2,col="blue",lty=3);
      text(x=13,y=0.16,col="red",label=c("Test 1"))
    #+end_src

    #+CAPTION: Density plot - test results for DSC 205
    #+RESULTS: density2
    file:test2density.png


• References
  * DataCamp (2022). 2022 Data trends and predictions. URL:
    datacamp.com.
  * ESS (n.d.). Emacs Speaks Statistics. URL: ess.r-project.org
  * Emacs Speaks Statistics (Mar 19, 2021). First Steps With Emacs
    [video]. URL: youtu.be/1YOrd7NCGkg.
  * GNU Emacs (n.d.). GNU Editor. URL: gnu.org/software/emacs/
  * R Core Team (2021). R: A language and environment for statistical
    computing. R Foundation for Statistical Computing, Vienna,
    Austria. URL https://www.R-project.org/.

1 U\---  test.org     All (13,60)    (Org)
```





Computer science is a craft

- Take it apart
- Learn tools
- Fix many cars
- Mechanical literacy
- Inferential thinking

Image source: The Atlanta Journal-Constitution, Dec 23, 2012

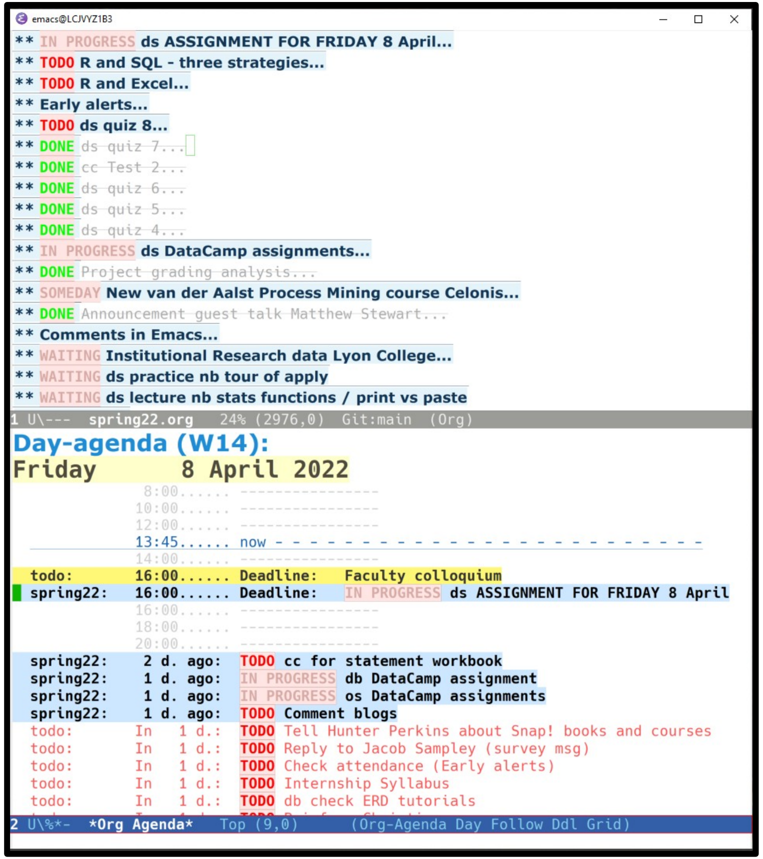
The problem

- Computers are seen as bricks with buttons
- Students cannot find downloaded files
- Cannot distinguish between browser, PC, network, cloud, client, server …
- Convenience, not customization, rules
- Machines have all the power

The solution: Emacs + Org-mode

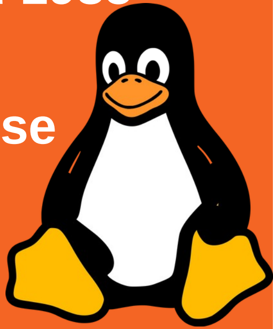
+

GNU Emacs



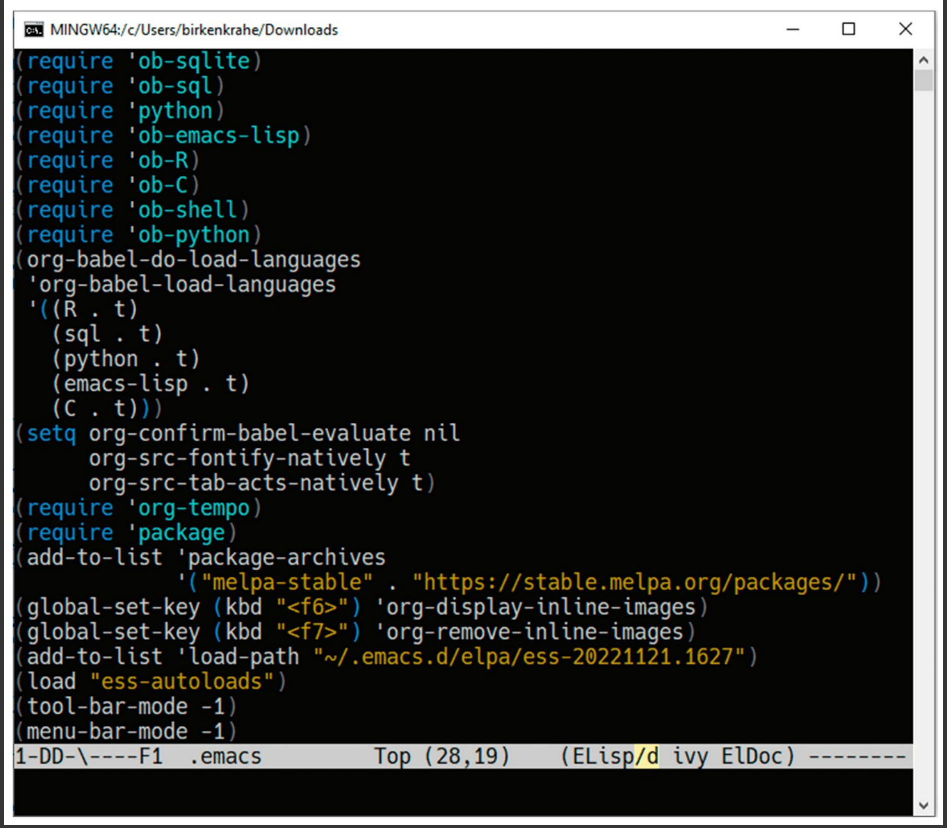
```
emacs@LCJVYZ1B3
** IN PROGRESS ds ASSIGNMENT FOR FRIDAY 8 April...
** TODO R and SQL - three strategies...
** TODO R and Excel...
** Early alerts...
** TODO ds quiz 8...
** DONE ds quiz 7...
** DONE cc Test 2...
** DONE ds quiz 6...
** DONE ds quiz 5...
** DONE ds quiz 4...
** IN PROGRESS ds DataCamp assignments...
** DONE Project grading analysis...
** SOMEDAY New van der Aalst Process Mining course Celonis...
** DONE Announcement guest talk Matthew Stewart...
** Comments in Emacs...
** WAITING Institutional Research data Lyon College...
** WAITING ds practice nb tour of apply
** WAITING ds lecture nb stats functions / print vs paste
1 U\---  spring22.org   24% (2976,0)  Git:main  (Org)
Day-agenda (W14):
Friday       8 April 2022
                8:00...... ----------------
               10:00...... ----------------
               12:00...... ----------------
               13:45...... now - - - - - - - - - - - - - - - - - - - - - - - -
               14:00...... ----------------
  todo:        16:00...... Deadline:   Faculty colloquium
  spring22:    16:00...... Deadline:   IN PROGRESS ds ASSIGNMENT FOR FRIDAY 8 April
               16:00...... ----------------
               18:00...... ----------------
               20:00...... ----------------
  spring22:     2 d. ago:  TODO cc for statement workbook
  spring22:     1 d. ago:  IN PROGRESS db DataCamp assignment
  spring22:     1 d. ago:  IN PROGRESS os DataCamp assignments
  spring22:     1 d. ago:  TODO Comment blogs
  todo:        In   1 d.:  TODO Tell Hunter Perkins about Snap! books and courses
  todo:        In   1 d.:  TODO Reply to Jacob Sampley (survey msg)
  todo:        In   1 d.:  TODO Check attendance (Early alerts)
  todo:        In   1 d.:  TODO Internship Syllabus
  todo:        In   1 d.:  TODO db check ERD tutorials
2 U\%*-  *Org Agenda*   Top (9,0)      (Org-Agenda Day Follow Ddl Grid)
```

- Programmable platform
- Self-documenting
- Fully extensible & transparent
- Text editor + operating system
- Keyboard-heavy
- Lisp machine
- Free software
- UNIX / Linux methodology
- Created 1975, launched 1985
- Used by me since 1991
- Hard to learn, easy to use



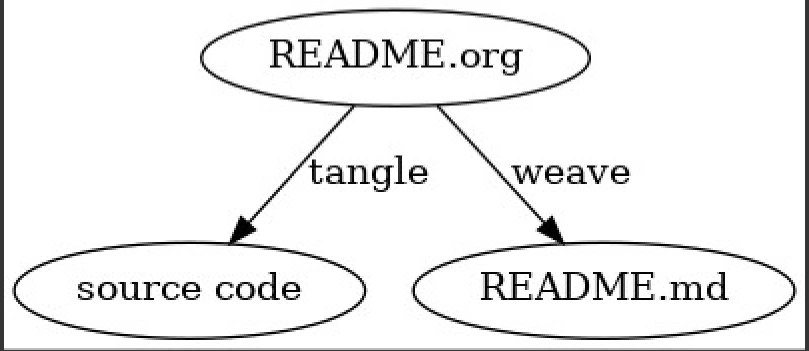
Emacs configuration file

- Run C/C++, R, SQL, SQLite, Python and bash
- Update Emacs from the melpa repository
- Create code blocks with skeleton commands
- Autoload ESS
- Disable toolbar and graphical menu bars



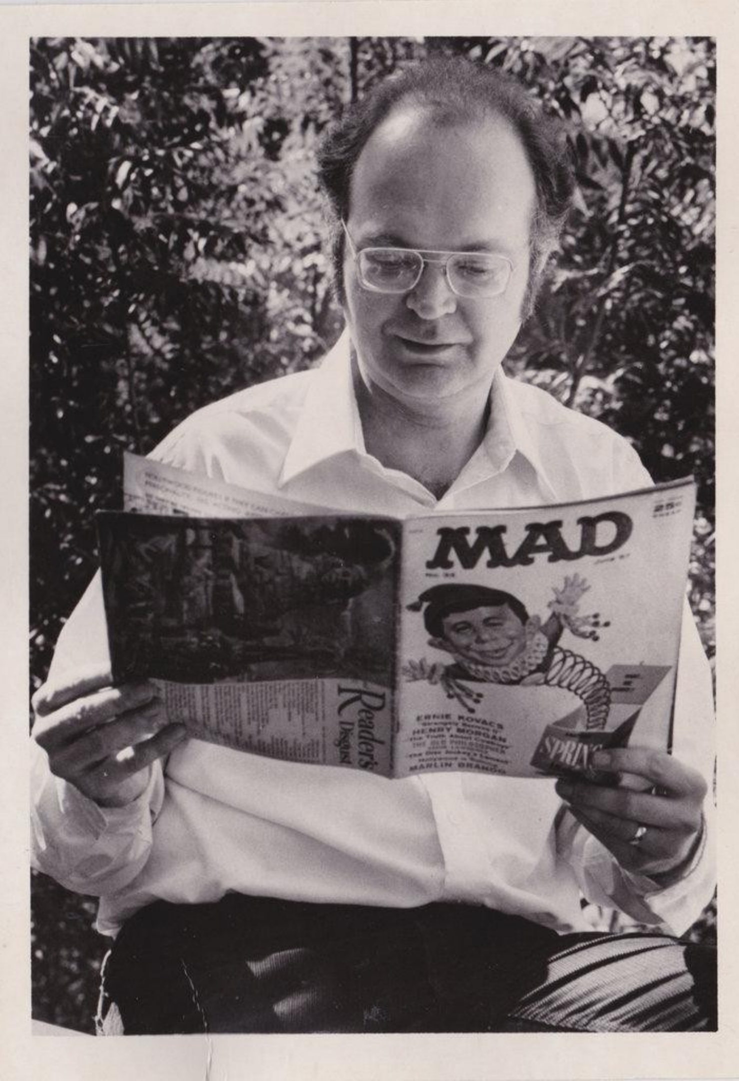
```
(require 'ob-sqlite)
(require 'ob-sql)
(require 'python)
(require 'ob-emacs-lisp)
(require 'ob-R)
(require 'ob-C)
(require 'ob-shell)
(require 'ob-python)
(org-babel-do-load-languages
 'org-babel-load-languages
 '((R . t)
   (sql . t)
   (python . t)
   (emacs-lisp . t)
   (C . t)))
(setq org-confirm-babel-evaluate nil
      org-src-fontify-natively t
      org-src-tab-acts-natively t)
(require 'org-tempo)
(require 'package)
(add-to-list 'package-archives
             '("melpa-stable" . "https://stable.melpa.org/packages/"))
(global-set-key (kbd "<f6>") 'org-display-inline-images)
(global-set-key (kbd "<f7>") 'org-remove-inline-images)
(add-to-list 'load-path "~/.emacs.d/elpa/ess-20221121.1627")
(load "ess-autoloads")
(tool-bar-mode -1)
(menu-bar-mode -1)
1-DD-\----F1  .emacs         Top (28,19)    (ELisp/d ivy ElDoc) --------
```

Story + code =

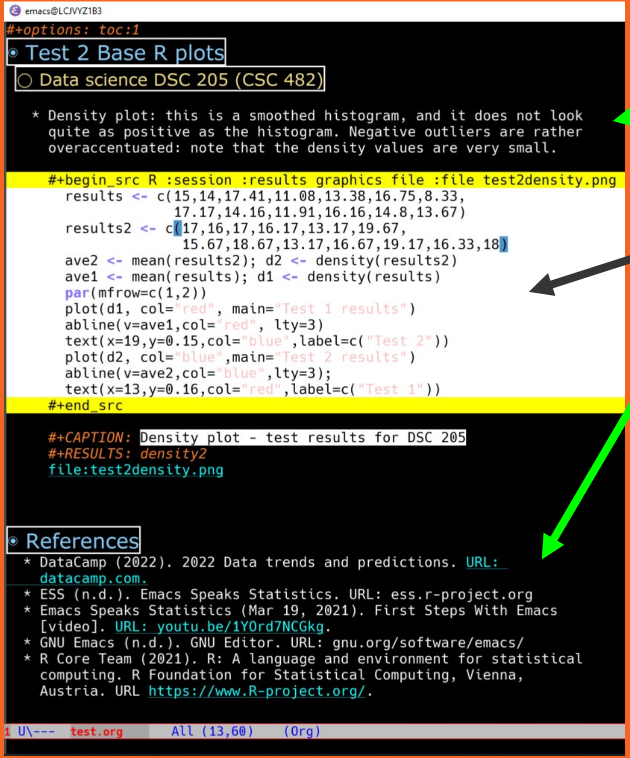


source + documentation

What is literate programming?



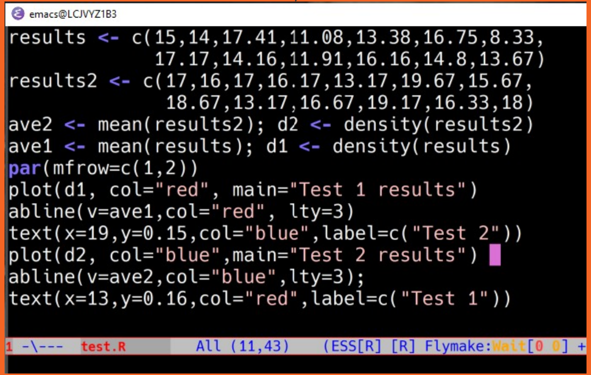
What is literate programming?



```
emacs@LCJVYZ1B3
#+options: toc:1
◉ Test 2 Base R plots
 ○ Data science DSC 205 (CSC 482)

   * Density plot: this is a smoothed histogram, and it does not look
     quite as positive as the histogram. Negative outliers are rather
     overaccentuated: note that the density values are very small.

     #+begin_src R :session :results graphics file :file test2density.png
       results <- c(15,14,17.41,11.08,13.38,16.75,8.33,
                    17.17,14.16,11.91,16.16,14.8,13.67)
       results2 <- c(17,16,17,16.17,13.17,19.67,
                     15.67,18.67,13.17,16.67,19.17,16.33,18)
       ave2 <- mean(results2); d2 <- density(results2)
       ave1 <- mean(results); d1 <- density(results)
       par(mfrow=c(1,2))
       plot(d1, col="red", main="Test 1 results")
       abline(v=ave1,col="red", lty=3)
       text(x=19,y=0.15,col="blue",label=c("Test 2"))
       plot(d2, col="blue",main="Test 2 results")
       abline(v=ave2,col="blue",lty=3);
       text(x=13,y=0.16,col="red",label=c("Test 1"))
     #+end_src

     #+CAPTION: Density plot - test results for DSC 205
     #+RESULTS: density2
     file:test2density.png


◉ References
  * DataCamp (2022). 2022 Data trends and predictions. URL:
    datacamp.com.
  * ESS (n.d.). Emacs Speaks Statistics. URL: ess.r-project.org
  * Emacs Speaks Statistics (Mar 19, 2021). First Steps With Emacs
    [video]. URL: youtu.be/1YOrd7NCGkg.
  * GNU Emacs (n.d.). GNU Editor. URL: gnu.org/software/emacs/
  * R Core Team (2021). R: A language and environment for statistical
    computing. R Foundation for Statistical Computing, Vienna,
    Austria. URL https://www.R-project.org/.

1 U\---  test.org      All (13,60)    (Org)
```

Humans

Machines



```
emacs@LCJVYZ1B3
results <- c(15,14,17.41,11.08,13.38,16.75,8.33,
             17.17,14.16,11.91,16.16,14.8,13.67)
results2 <- c(17,16,17,16.17,13.17,19.67,15.67,
              18.67,13.17,16.67,19.17,16.33,18)
ave2 <- mean(results2); d2 <- density(results2)
ave1 <- mean(results); d1 <- density(results)
par(mfrow=c(1,2))
plot(d1, col="red", main="Test 1 results")
abline(v=ave1,col="red", lty=3)
text(x=19,y=0.15,col="blue",label=c("Test 2"))
plot(d2, col="blue",main="Test 2 results")
abline(v=ave2,col="blue",lty=3);
text(x=13,y=0.16,col="red",label=c("Test 1"))

1 -\---  test.R      All (11,43)    (ESS[R] [R] Flymake:Wait[0 0] +
```

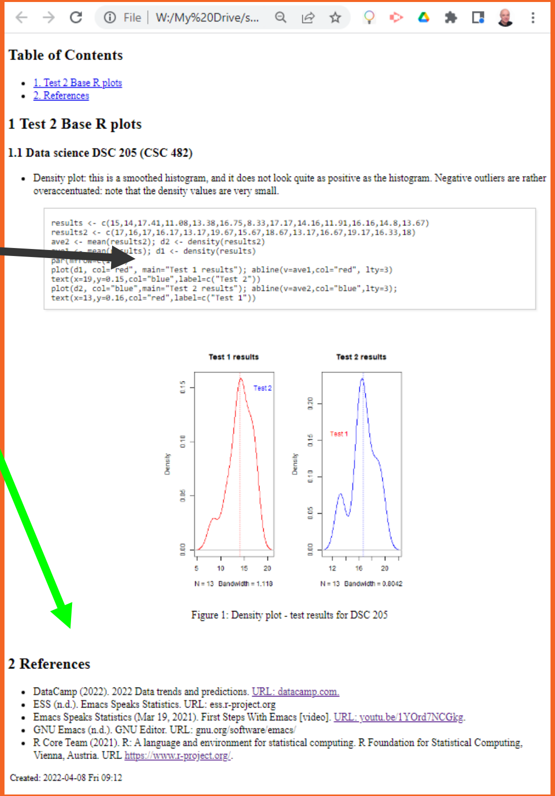


Table of Contents

- 1. Test 2 Base R plots
- 2. References

1 Test 2 Base R plots

1.1 Data science DSC 205 (CSC 482)

- Density plot: this is a smoothed histogram, and it does not look quite as positive as the histogram. Negative outliers are rather overaccentuated: note that the density values are very small.

```
results <- c(15,14,17.41,11.08,13.38,16.75,8.33,17.17,14.16,11.91,16.16,14.8,13.67)
results2 <- c(17,16,17,16.17,13.17,19.67,15.67,18.67,13.17,16.67,19.17,16.33,18)
ave2 <- mean(results2); d2 <- density(results2)
ave1 <- mean(results); d1 <- density(results)
par(mfrow=c(1,2))
plot(d1, col="red", main="Test 1 results"); abline(v=ave1,col="red", lty=3)
text(x=19,y=0.15,col="blue",label=c("Test 2"))
plot(d2, col="blue",main="Test 2 results"); abline(v=ave2,col="blue",lty=3);
text(x=13,y=0.16,col="red",label=c("Test 1"))
```

Figure 1: Density plot - test results for DSC 205

2 References

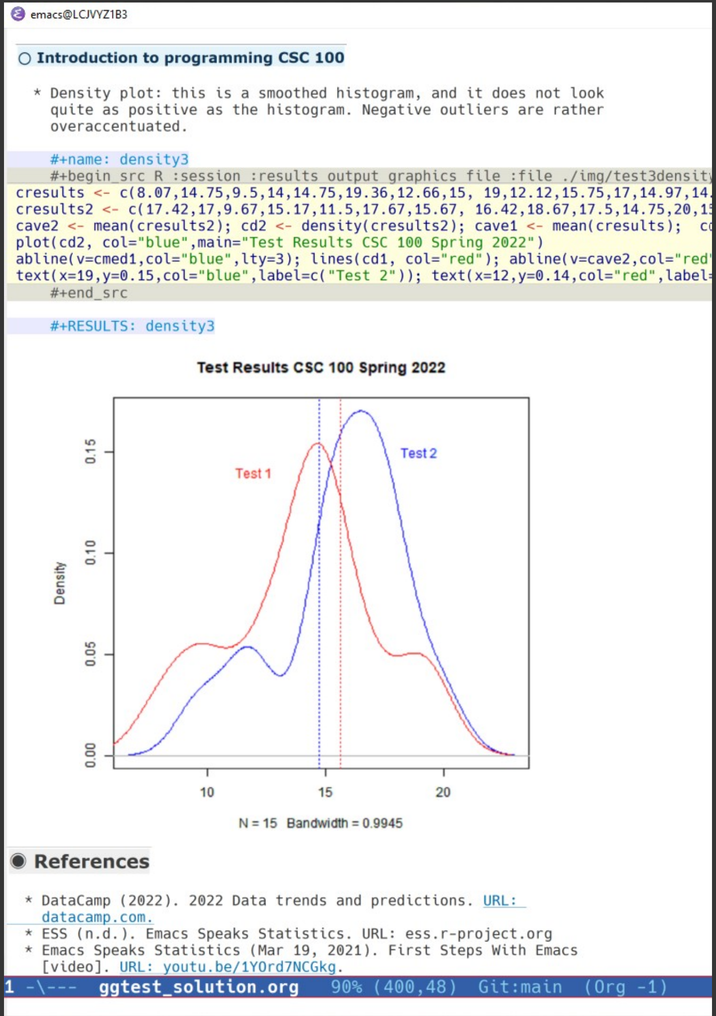
- DataCamp (2022). 2022 Data trends and predictions. URL: datacamp.com.
- ESS (n.d.). Emacs Speaks Statistics. URL: ess.r-project.org
- Emacs Speaks Statistics (Mar 19, 2021). First Steps With Emacs [video]. URL: youtu.be/1YOrd7NCGkg.
- GNU Emacs (n.d.). GNU Editor. URL: gnu.org/software/emacs/
- R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.r-project.org/.

Created: 2022-04-08 Fri 09:12

Story + code = source code + documentation

Emacs as a literate programming tool

- Execute code blocks
- Code in 43 programming languages
- Display results
- Interact with shell
- Extract source code
- Render documentation
- Manage tasks & projects
- 5,000 add-on packages



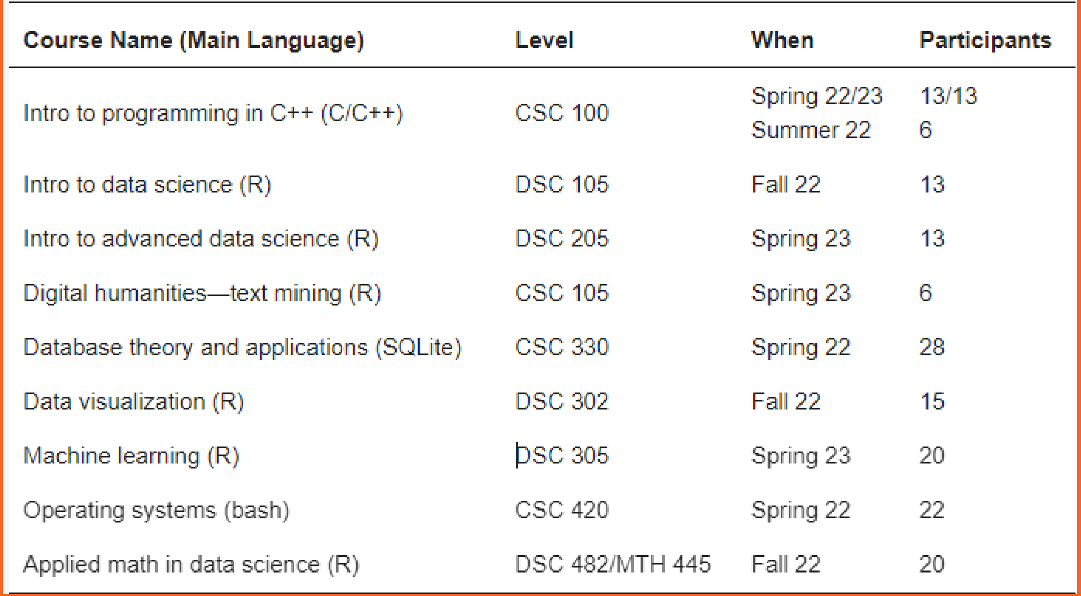
○ **Introduction to programming CSC 100**

```
  * Density plot: this is a smoothed histogram, and it does not look
    quite as positive as the histogram. Negative outliers are rather
    overaccentuated.

    #+name: density3
    #+begin_src R :session :results output graphics file :file ./img/test3density
cresults <- c(8.07,14.75,9.5,14,14.75,19.36,12.66,15, 19,12.12,15.75,17,14.97,14.
cresults2 <- c(17.42,17,9.67,15.17,11.5,17.67,15.67, 16.42,18.67,17.5,14.75,20,15
cave2 <- mean(cresults2); cd2 <- density(cresults2); cave1 <- mean(cresults);  cd
plot(cd2, col="blue",main="Test Results CSC 100 Spring 2022")
abline(v=cmed1,col="blue",lty=3); lines(cd1, col="red"); abline(v=cave2,col="red"
text(x=19,y=0.15,col="blue",label=c("Test 2")); text(x=12,y=0.14,col="red",label=
    #+end_src

    #+RESULTS: density3
```

● **References**

```
 * DataCamp (2022). 2022 Data trends and predictions. URL:
   datacamp.com.
 * ESS (n.d.). Emacs Speaks Statistics. URL: ess.r-project.org
 * Emacs Speaks Statistics (Mar 19, 2021). First Steps With Emacs
   [video]. URL: youtu.be/1YOrd7NCGkg.
1 -\---   ggtest_solution.org    90% (400,48)  Git:main  (Org -1)
```

Case study: basic setup



Course Name (Main Language)	Level	When	Participants
Intro to programming in C++ (C/C++)	CSC 100	Spring 22/23 Summer 22	13/13 6
Intro to data science (R)	DSC 105	Fall 22	13
Intro to advanced data science (R)	DSC 205	Spring 23	13
Digital humanities—text mining (R)	CSC 105	Spring 23	6
Database theory and applications (SQLite)	CSC 330	Spring 22	28
Data visualization (R)	DSC 302	Fall 22	15
Machine learning (R)	DSC 305	Spring 23	20
Operating systems (bash)	CSC 420	Spring 22	22
Applied math in data science (R)	DSC 482/MTH 445	Fall 22	20

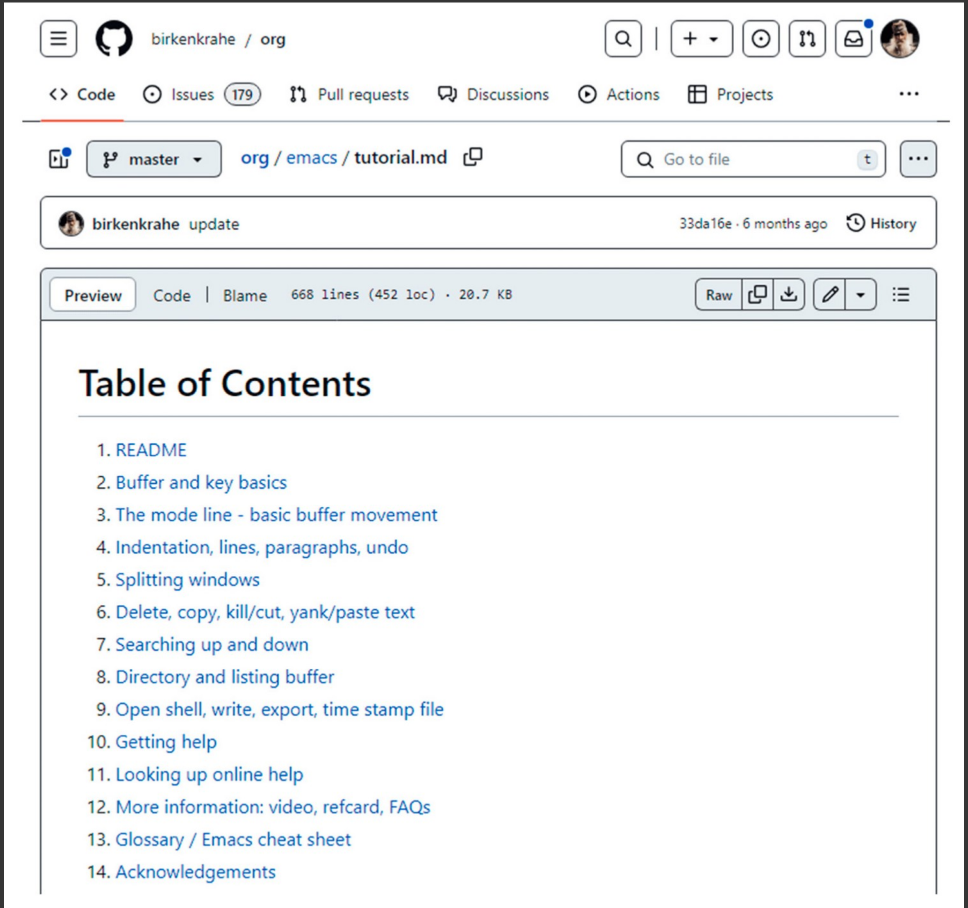
The material of all of my courses is available online at github.com/birkenkrahe

- Introductory to advanced
- Different computing applications
- Taught over 3 terms
- 6-28 participants
- Used GitHub, Canvas, DataCamp

Emacs + Org-mode notebooks used for:

- Code along lectures
- Home assignments
- Practice in class
- Student projects
- GitHub repository

Onboarding: simplified Emacs tutorial

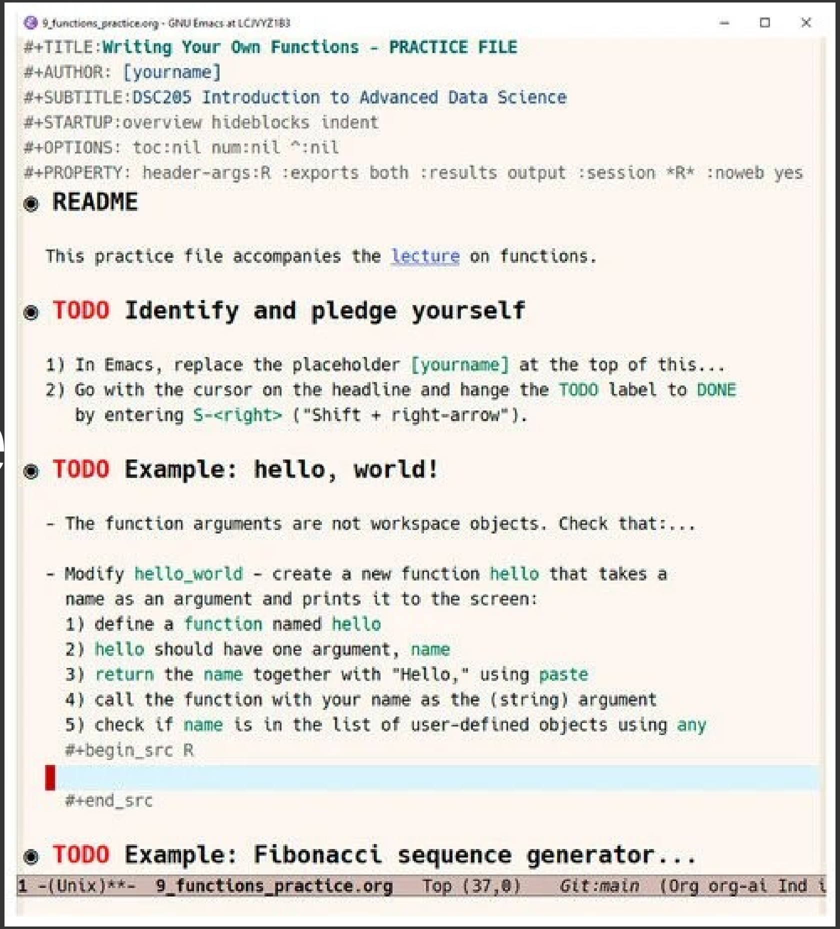


668 lines (452 loc) · 20.7 KB

Table of Contents

Instruction + Interaction

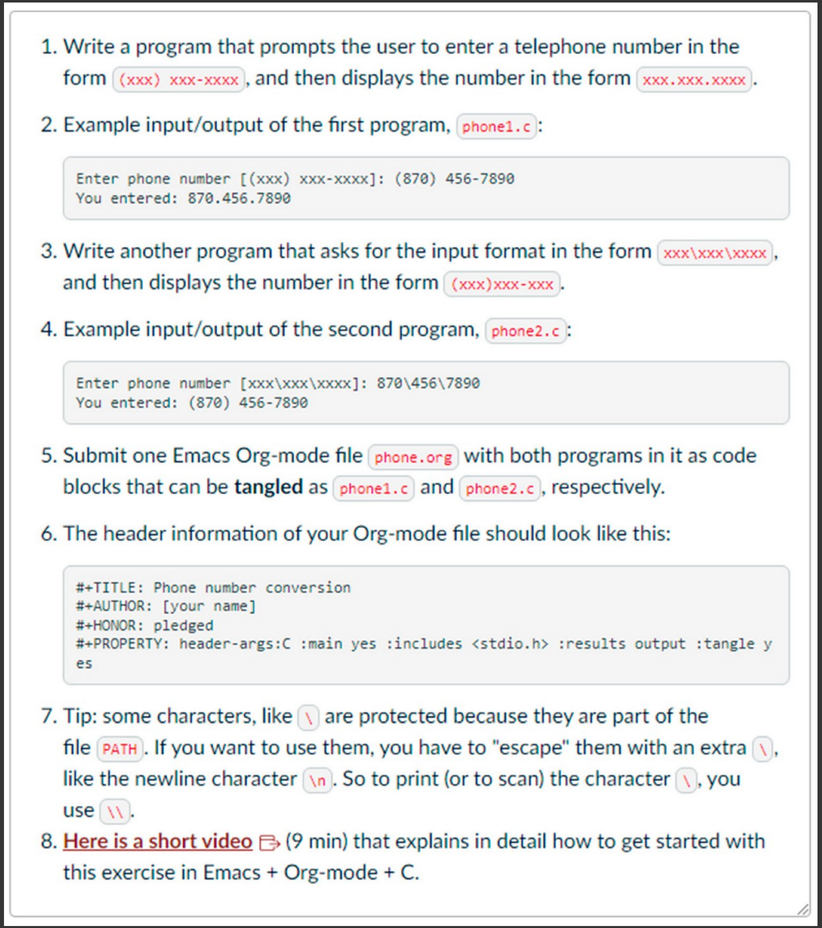
- Emacs + Org pre installed
- All lectures code-along



```
#+TITLE:Writing Your Own Functions - PRACTICE FILE
#+AUTHOR: [yourname]
#+SUBTITLE:DSC205 Introduction to Advanced Data Science
#+STARTUP:overview hideblocks indent
#+OPTIONS: toc:nil num:nil ^:nil
#+PROPERTY: header-args:R :exports both :results output :session *R* :noweb yes
◉ README

  This practice file accompanies the lecture on functions.

◉ TODO Identify and pledge yourself

  1) In Emacs, replace the placeholder [yourname] at the top of this...
  2) Go with the cursor on the headline and hange the TODO label to DONE
     by entering S-<right> ("Shift + right-arrow").

◉ TODO Example: hello, world!

  - The function arguments are not workspace objects. Check that:...

  - Modify hello_world - create a new function hello that takes a
    name as an argument and prints it to the screen:
    1) define a function named hello
    2) hello should have one argument, name
    3) return the name together with "Hello," using paste
    4) call the function with your name as the (string) argument
    5) check if name is in the list of user-defined objects using any
    #+begin_src R

    #+end_src

◉ TODO Example: Fibonacci sequence generator...
1 -(Unix)**-  9_functions_practice.org   Top (37,0)    Git:main  (Org org-ai Ind i
```

Assignments + Projects

- Submit literate Org-mode files
- Communicate throughout



1. Write a program that prompts the user to enter a telephone number in the form `(xxx) xxx-xxxx`, and then displays the number in the form `xxx.xxx.xxxx`.
2. Example input/output of the first program, `phone1.c`:

```
Enter phone number [(xxx) xxx-xxxx]: (870) 456-7890
You entered: 870.456.7890
```

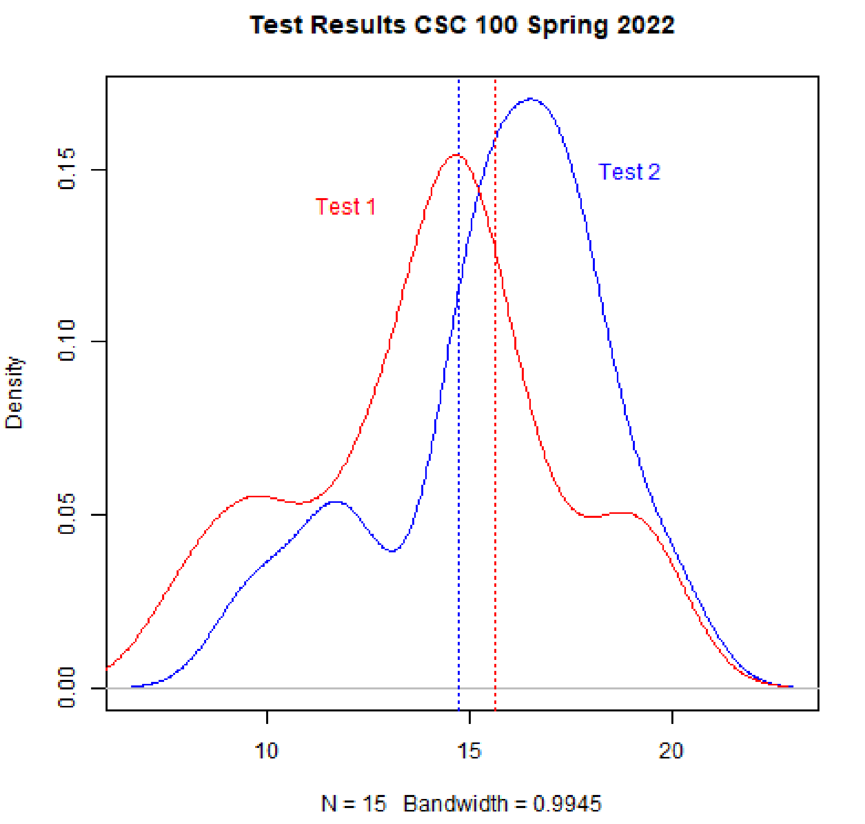
3. Write another program that asks for the input format in the form `xxx\xxx\xxxx`, and then displays the number in the form `(xxx)xxx-xxx`.
4. Example input/output of the second program, `phone2.c`:

```
Enter phone number [xxx\xxx\xxxx]: 870\456\7890
You entered: (870) 456-7890
```

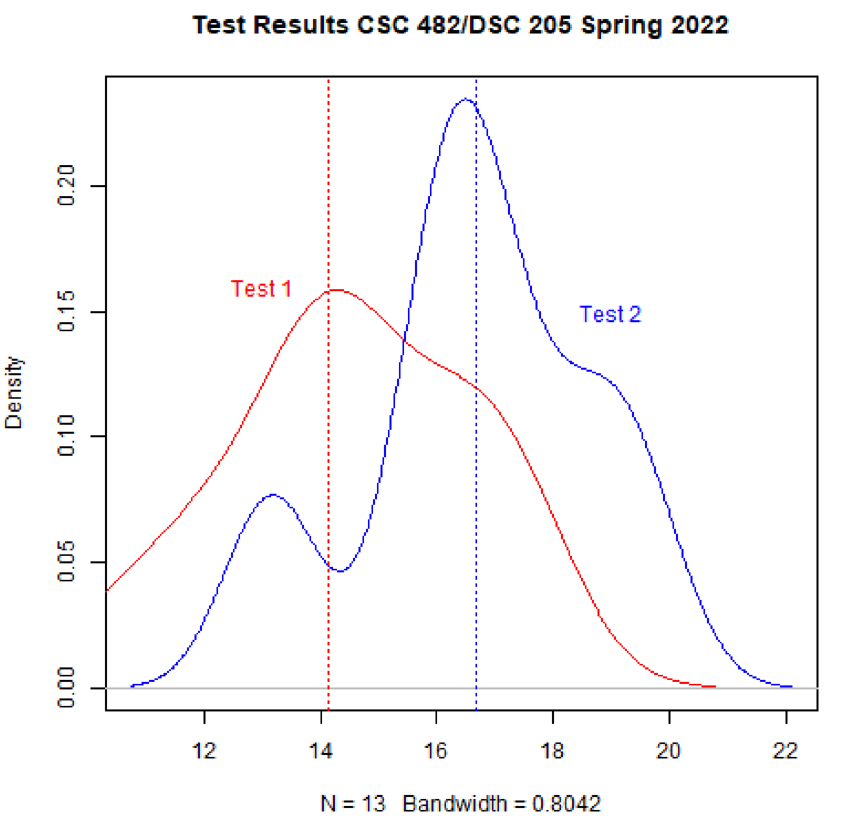
5. Submit one Emacs Org-mode file `phone.org` with both programs in it as code blocks that can be **tangled** as `phone1.c` and `phone2.c`, respectively.
6. The header information of your Org-mode file should look like this:

```
#+TITLE: Phone number conversion
#+AUTHOR: [your name]
#+HONOR: pledged
#+PROPERTY: header-args:C :main yes :includes <stdio.h> :results output :tangle y
es
```

7. Tip: some characters, like `\` are protected because they are part of the file `PATH`. If you want to use them, you have to "escape" them with an extra `\`, like the newline character `\n`. So to print (or to scan) the character `\`, you use `\\`.
8. Here is a short video (9 min) that explains in detail how to get started with this exercise in Emacs + Org-mode + C.



Before | After introducing literate programming



Before | After introducing literate programming

Overall results positive:

- Emacs hard for all but all succeeded across all courses
- Documentation results uneven but higher quality than ever
- Interactivity praised by all students
- Computing and infrastructure competences much improved

Conclusion & outlook

- Immersion and interaction is everything
- Emacs + Org-mode perform well as central literate programming platform
- Pre-configuring and onboarding are important to train students quickly