

## Homework 1: The Emacs Text Editor

This homework will give you practice with the emacs text editor. When doing these exercises, please use the commands on the reference sheet at

<http://www.stat.wisc.edu/~jgillett/DSCP/emacs/emacs.html>

The goal is not just to get the specified tasks done, but rather to do so using the functionality available in emacs, so try to do the tasks elegantly, using the tools provided by the editor.

You may ask your classmates, TA or instructor for help with these exercises. If you collaborate with classmates, please include their names and their primary @wisc.edu email addresses on the line after your name in your submission files.

### Editing text in emacs

1. Start emacs on your Linux machine.
2. Use emacs to make a new directory, 1 (for “homework 1”), in your ~/DSCP directory. That is, make the new directory ~/DSCP/1 (make ~/DSCP first).
3. There are about 65 commands listed on the reference sheet (linked above). Try every one of them on tiny files you create.

This is an important step. It doesn’t take long. If you do it, you have a good chance of knowing which commands to consider for each of the exercises below. If you skip it, you may find the exercises below unnecessarily difficult and frustrating.

Delete the tiny files you created in this step.

4. Download `baby_T_Test.R` and move it to your 1 directory.
  - (a) Read `baby_T_Test.R` into emacs.
  - (b) The code has four problems with parentheses. Indent the buffer. The first indenting irregularity indicates a parentheses problem on the previous line. Fix it by adding a parenthesis. Repeat until the whole file is indented correctly.
  - (c) Save the file as `baby_t_test.R` (note “T\_Test” changed to “t\_test”).
  - (d) Make these replacements in `baby_t_test.R`.
    - Inside the `baby.t.test()` function, replace each `r` (where `r` indicates the list being set up as a return value) with `return.list`.
    - Replace each occurrence of three newlines with two newlines. Hint: We can’t use **Enter** to type a newline in the minibuffer, as it ends the minibuffer’s input. Use **C-q C-j** to type a literal newline into the minibuffer. **C-q** runs `quoted-insert`, which allows inserting a literal newline in Linux (newline is **C-j**, the decimal ASCII

code 10), a control character, etc. There are still sequences of three newlines remaining, so jump to the top of the buffer and do it again. (We will see a way to do these two steps in one step after we study regular expressions soon).

(e) Use the emacs R buffer to:

- i. Run the chunk of code consisting of the `baby.t.test()` function definition.
- ii. Run the “test case” code one line at a time. Note that the `p.value` test fails. The bug is that I mistakenly used `df=n` instead of `df=n-1` in my call to `pt()` in `baby.t.test()`. Fix this bug, run the `baby.t.test()` chunk again, and run the test code (one line at a time) again.
- iii. Run the entire buffer. Save the file.

(f) Kill the R buffer. Start it again.

- i. Find `3 + 4` (just type it at the R command prompt).
- ii. Run `source("baby_t_test.R")` (just type it at the R command prompt).

(g) Kill the R buffer again and exit emacs.

5. Download `gettysburg1.txt` and `gettysburg2.txt` to your 1 directory.

- (a) Use Multiple Windows (in a single emacs session) to split emacs vertically (into halves) and vertically again (now you have one half and two quarters).
- (b) Resize the windows so they all have the same size.
- (c) Open `gettysburg1.txt` in the first window and `gettysburg2.txt` in the second. Open (the new empty file) `gettysburg_emacs.txt` in the third window.
- (d) Use cut, copy, and paste commands to reassemble the Gettysburg Address from its segments in the first two windows into a whole in the third window. Handle each block of lines in the downloaded files separately (e.g. lines 1-2 are a block, as is line 3, as are lines 4-6). Reassemble the Address’s paragraphs, by switching among the three windows as needed. Save `gettysburg_emacs.txt` when you are done. Close its buffer.
- (e) Return to a single window. Open the Directory Editor on `.` (“dot”), the current directory.
  - i. Rename `gettysburg_emacs.txt` to `GettysburgAddress_emacs.txt`.
  - ii. Find (open) `GettysburgAddress_emacs.txt` again.
- (f) Kill the rectangle consisting of the line numbers and spaces preceding each line of the address. Save `GettysburgAddress_emacs.txt` again.

6. Download `roster.txt` to your 1 directory. Open it in emacs. Save it as `password.txt`. Convert `password.txt` so that each line contains the last four digits of a student’s ID number and each student’s NetID, separated by a space. For example, the first line,

```
"123456789      Brown,Joe      jbrown@wisc.edu  "
```

should become

```
6789 jbrown
```

Use the following steps.

- (a) Remove the rectangle consisting of the first 5 columns of digits. (Now the first line is  
"6789 Brown,Joe jbrown@wisc.edu ".)

- (b) Use a regular expression to trim the remaining text to the desired result. (The first line becomes “6789 jbrown”.)
- (c) Save `password.txt` again.

Hint: `roster.txt` is a confusing file because it contains three whitespace characters, each invisible: SPACE, TAB, and NEWLINE. (To understand the file better, you might make a throw-away copy like `roster2.txt` in which you replace each TAB with a hyphen, `-`, and each SPACE with an underscore, `_`.) One way to handle whitespace within a line is to use a character class with square brackets, `[]`, in which you type both a TAB and a SPACE.

## Prepare your files for submission

Start emacs and open the (new empty) file `hw1.txt`.

1. Type your name followed by your primary `@wisc.edu` email address and two blank lines.
2. If you discussed this homework with any of your classmates, please list their names and primary `@wisc.edu` email addresses and two blank lines.
3. Insert your `baby_t_test.R` file followed by two blank lines.
4. Insert your `GettysburgAddress_emacs.txt` file followed by two blank lines.
5. Insert your `password.txt` file followed by two blank lines.
6. Search online to find something interesting emacs can do. Summarize it briefly (no more than 100 words) for consideration by the class for a demonstration.
7. Save `hw1.txt` and exit emacs.

Hint: Please don't worry about tiny variations in interpretation of these instructions. We will give credit to reasonable interpretations of the instructions.

Upload your `hw1.txt` file via the Canvas “HW1” grades item.