Stat 290:
Introduction to Unix and Emacs

Objectives

1. To introduce basic emacs commands
2. To introduce some basic unix commands

Notation

The notation `C-g` means to hold down the "Control" key and hit the "g" key, while `M-a` means to invoke the Meta key (either hold down the "Alt" key, or press and release the "Esc" key, depending on your keyboard - I use the latter) and hit the "a" key.

Reminders

If something starts to go horribly wrong, use `C-g` to cancel the current command (infrequently, you may need to use it several times). You can also undo commands successively with either `C-x u` or `C-_` (that's Control-underscore). Recall that `C-x C-c` will quit emacs, and `C-z` will suspend it, shrinking the window into an icon. The help menus in emacs are triggered with `C-h` and are quite extensive.

Introduction to Emacs

Start up Emacs by typing at the unix command prompt:

```
emacs &
```

Note that the ampersand runs the process in the background, so that you can continue to use the unix shell window while emacs is running. If you forgot the ampersand, at the unix prompt type C-z, then enter

```
bg
```

to put emacs in the background. Now type the following into your emacs window: Your name (on the first line), your department on the second line, followed by a blank line, for example:

```
Merlise Clyde
Institute of Statistics and Decision Sciences
```

Cursor Movement

Now we'll walk through a variety of cursor movement commands. Try the following to get accustomed to moving the cursor without using the arrows or the mouse. Hit `M-<` to move to the beginning of the buffer. `C-f` moves you forward one character, `C-b` moves you back. `M-f` moves you forward one word, `M-b` moves you back one word. `C-n` moves you down to the next line, `C-p` moves you up to the previous line. `C-e` moves you to the end of the line, `C-a` moves you to the beginning of the line. If your document scrolled over more than one screen, you can use `C-v` to scroll down and `M-v` to scroll up.

Files and Buffers

Suppose you now want to save this unnamed file (or if it already had a name, supposed you wanted to rename it). To write a file to disk, use `C-x C-w` and then type in the name of the file, `myname.txt` and hit enter, as in:

```
C-x C-w merlise.txt
```

Now we'll update the file. Move to the last line of the file, and add the following information to your file: previous applied and theoretical statistics courses that you have taken at Duke and/or other institutions (please name), experience with S-Plus or R, experience with LaTeX, familiarity with Unix. (Depending on how your defaults are set up, emacs may or may not automatically insert line breaks when you type too close to the right hand margin. If it does not, you can get it to "fix" your paragraph By typing `M-q`. The technical term for this is *auto-fill.*) Now save your updated file by typing `C-x C-s`. This is the standard way of saving your file. Emacs does have an auto-save feature, but you should get into the habit of saving your work very frequently as you work, just in case something bad should happen.

Without closing emacs, open a new file (the previous file, `myname.txt,` will still be open, but in a different buffer) by typing

```
C-x C-f lab1hw.txt
```

Tell me a little bit about what type of applications interest you or topics that you would like to learn about in this course. Save this file with `C-x C-s`.

Since `myname.txt` is still in another buffer, we can still access it. To switch to another buffer, type in

```
C-x b
```

and it will ask you which buffer you wish to switch to. The default will be `myname.txt`, so you can just hit enter (otherwise you can type in the name of the buffer you want). In this way, you can have many files open in the same emacs process, and just switch between them as need be. You can also have more than one displayed at the same time. Split the screen by typing

```
C-x 2
```

You will now have two buffers, each half the size of the emacs window, each containing the same file. Switch one of them to `lab1hw.txt` using `C-x b`. Now you have easy access to both files. To move the cursor between the buffers, use `C-x o`. To go back to a single window (just the window containing the cursor) use `C-x 1`. You can close a file with `C-x k` (this kills the buffer), but leave the two files open, as we'll continue to use them in the next section.

Cut, Copy, and Paste

Bring `myname.txt` to the front of your emacs window (if it isn't already, use `C-x b` to switch buffers, or `C-x C-f` to open it from the disk if you've closed it). Suppose we want to move the two sentences name and department to the file lab1hw.txt. Move your cursor to the first letter of the first sentence and then type Control-space. This sets the "mark" at that character. Now move past the end of the second sentence and type `C-w`. This cuts the marked section. (If you miss, you can use `C-_` to undo.) If we just wanted to delete this part, we have done so and can just move on, but we want to move it, so we need to paste it back in. Switch to the lab1hw.txt buffer, move to the beginning of the document (`M-<` is the shortcut) and type in `C-y` (yank is the mnemonic) to paste in the text. Move to the end of the file (`M->`) and paste another copy with `C-y`. This is just to practice a different deletion technique. Move your cursor back to the beginning of the section that was just pasted in and use `C-k` to kill all the text on the same line to the right of the cursor. Use `C-k` repeatedly until all the freshly pasted text has been removed. Finally, you should know that `M-w` will copy text to the kill buffer (like `C-w`) but without deleting it, so that you can paste it elsewhere without removing the original. You can either save your work with `C-x C-s`, or kill all the recent changes by killing the buffer `C-x k` and entering `yes` when it asks for confirmation of closing without saving. We're done with emacs for now, so you can close it with `C-x C-c`.

Basic Unix Commands

Back in your unix shell window, type

```
ls
```

After hitting enter, you will get a list of all the files in the current directory. You may have started in your root directory (∼), which probably isn't where you want these files. Make a new directory for this class by typing

```
mkdir sta290
```

Now move the new file into this directory, first use

```
mv myname.txt sta290/.
```

Note that the period at the end says to keep the name of the file the same. For the second file, use tab-completion so that you don't have to type in the whole file name, i.e., instead of typing in `mv lab1hw.txt sta290/.`, type in (without hitting enter) `mv la` and hit the tab key so that unix will fill in the rest of the name (or as much as it can if there are multiple files that all start the same way), and it should fill in the rest of that file name, and then you can complete the line with `290/.` and then finally hit enter. Now that you've moved the files, change to this subdirectory with

```
cd sta290
```

Typing `ls` should show that the only files in this directory are the two that you just moved in. You can copy files with the syntax `cp originalname copyname`, for example

```
cp lab1hw.txt newlab1hw.txt
```

If you have an old, incorrect, or otherwise unnecessary file, delete it with `rm`, as in

```
rm newlab1hw.txt
```

To go back to the next higher directory in the hierarchy, use

```
cd ..
```

If you do an `ls` here, you will likely find files like `lab1hw.txt`∼. These are old versions that emacs automatically keeps for you in case you later want to revert to an older version (e.g., if you accidentally erase a large section of your work and then accidentally save it, this way you can recover an earlier version). Most of the time, these files just take up a lot of space unnecessarily. In ISDS, we have a shortcut `pu` that will automatically clean up all of these old files in the current directory (you need to run it separately in each subdirectory). If you do not have this shortcut, you can delete them manually with

```
rm lab1hw.txt˜*
```

The asterisk is a wild card character. Be careful with it!

Printing

Print out the file `lab1hw.txt`. The print command in unix is `lpr`, but you typically need to specify which printer you want to use to print, which is done with the `-P` switch. For example, to print this file on the ISDS printer in room 222, make sure you are in the directory with `lab1hw.txt` and type

```
lpr -P222 lab1hw.txt
```