

# Integrated Development Environments

Emacs and Eclipse

Adrian Waddell

University of Waterloo  
Department of Statistics and Actuarial Science

September 8, 2010

## About these slides

These slides were written on behalf of the Department of Statistics and Actuarial Science at the University of Waterloo, Canada.

At the time of writing, the current software versions are

- ▶ GNU Emacs 23.1.1
- ▶ Eclipse SDK Version: 3.5.2

There are more slides like these on our [homepage](#).

# What is an Integrated Development Environment (IDE)?

- ▶ IDEs are source code editors with additional functionality to maximize the programmers productivity.
- ▶ this functionality might be: compiler, debugger, syntax highlighting, code completion, object browser, spell check etc..
- ▶ IDEs often require a lengthy learning process because of their customizing flexibility.
- ▶ For statisticians Eclipse and Emacs are great because they offer much functionality for  $\text{\LaTeX}$ , R and C.
- ▶ Emacs also supports SAS.

## More on IDEs

$\LaTeX$ , R and C have in common that you need to write source code in order to use them.  $\LaTeX$  is a document markup language and R and C are programming languages. Each of them has its own **syntax**, however their source code could be written in any text editor (such as Notepad, vi, emacs, etc...).

It is important to understand that source code is usually saved as **plain text** and hence Microsoft Word is always the wrong choice for editing source code. In fact, Microsoft Word and its equivalents are word processors, not text editors.

There are **many** text editors. Because it takes a lot of effort to learn a powerful text editor (and/or IDE) and once the editor is mastered the user interface (UI) is often quite elegant, people tend to build an almost emotional relationship to their editor of choice. Hence if you mention that you use emacs, someone will probably tell you that vi is much better –if you live in a certain environment– (see **Editor War**).

## About Emacs

- ▶ **Richard Stallman** wrote the first Emacs editor in 1975 at MIT. Nowadays **GNU Emacs** is the most popular Emacs version.
- ▶ Emacs is an extensible and customizable text editor. Much of Emacs popularity is due to all the extensions written to support different programming languages.
- ▶ There is a GUI and a console version of Emacs.
- ▶ Emacs offers a complete environment including extensions like email, calendar, RSS, UNIX shell, file browser and a **psychotherapist** to name a few.

# The Emacs Keystroke Combinations

*Learning to use an editor is basically a matter of learning finger habits.*

*Debra Cameron, Bill Rosenblatt & Eric Raymond*

- ▶ Much of Emacs philosophy is that your fingers should always stay on the keyboard, you do not have to use your mouse at all
  - ▶ this results in a surprisingly high efficiency gain
  - ▶ therefore Emacs has a large set of keystroke combinations (commands)
- ▶ because of Emacs's age, none of the "standard" text editor commands like `Ctrl-c` for copying, `Ctrl-v` for pasting, `Ctrl-s` for saving, etc..., will work with Emacs.



## **Learning GNU Emacs**

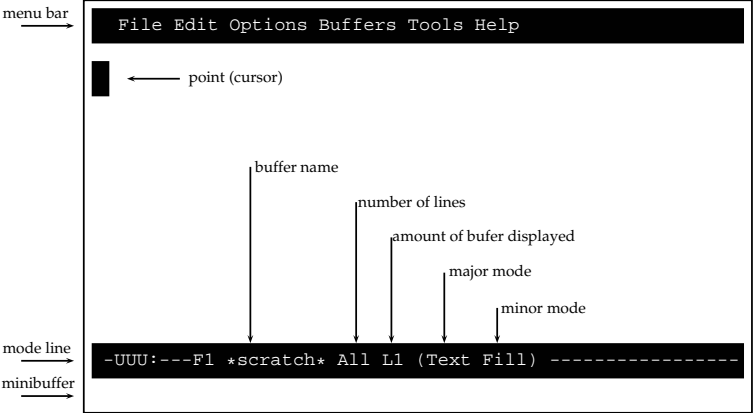
Debra Cameron, Bill Rosenblatt & Eric Raymond (O'Reilly)

# The Emacs Keystroke Combinations

Parts of the text that follows is copied from the [Emacs tutorial](#) (c-h t)

- ▶ Emacs commands generally involve the *CONTROL* key (sometimes labeled *CTRL* or *CTL*) or the *META* key (often labeled *ALT*).
- ▶ Rather than write that in full each time, we'll use the following abbreviations:
  - C-`<chr>`** means hold the *CONTROL* key while typing the character `<chr>`. Thus, **c-f** would be: hold the *CONTROL* key and type f.
  - M-`<chr>`** means hold the *META* (*ALT*) key down while typing `<chr>`. If there is no *META*, *EDIT* or *ALT* key, instead press and release the *ESC* key and then type `<chr>`. We write `<ESC>` for the *ESC* key.
- ▶ Commands come often in pairs, e.g. **c-x c-c** for closing Emacs. This would mean: hold the *CTRL* key down and press successively the x and then the c key.

# The Emacs Screen





# The Emacs Minibuffer

- ▶ The minibuffer is how Emacs communicates with you.
- ▶ Always look at the minibuffer for messages when executing a command.
- ▶ If you entered a wrong command sequence, e.g. you pressed `M-x` instead of `C-x` (the minibuffer shows the state of your command) you can always cancel your command sequence with `C-g`.
- ▶ If Emacs does not do what you want, try first `C-g`.

## Further Resources for Learning Emacs

The [Emacs tutorial](#) (`C-h t`) is a good starting point to learn Emacs.

The official [GNU Emacs](#) website has a [guided tour](#).

A GNU Emacs reference card can be found [here](#).

Also the [Emacs Wiki](#) has some [suggestions](#).

Xah Lee made some tutorials for [Emacs](#) and [Emacs LISP](#).

Finally, do not get discouraged after your first few Emacs sessions nor expect that you learn all these key combinations within a week or two. Start modestly with the basics, open files, saving files, copy-paste and so on. Once you are fluent with the basic commands you will see, that you will be curious to explore other commands. E.g. instead of using the arrow key to move one character back you could also use `C-b` which lets you keep your hand in the standard writing position. It is really all about finger habits.

# Installing Emacs with ESS and AUCTeX

**Windows:** [Vincent Goulet](#) maintains a modified Emacs version for Windows. This will save you a lot of time compared to setting up Emacs when downloading it from the [gnu servers](#).

**Mac OSX:** [Aquamacs](#) is an Emacs version with the common look and feel of an OSX program. Aquamacs provides ESS and AUCTeX out of the box. Alternatively, [Vincent Goulet](#) also maintains a modified Emacs for OSX.

**Linux:** Use your package manager to install Emacs, ESS and AUCTeX. On a debian based system (like ubuntu) this would be done with the bash command

```
sudo apt-get install emacs ess auctex
```

# Installing ESS on your UW UNIX account

Some of the UW math servers like `student.math` and `cpu05.math` have Emacs installed. You can add **ESS** as follows.

- ▶ Log in into your account (via the net client or via ssh)
- ▶ Download the ESS package and extract it: type in your terminal

```
wget http://ess.r-project.org/downloads/ess/ess-5.8.tgz
gzip -dc ess-5.8.tgz | tar -xf -
```

- ▶ Link Emacs to load the ESS module.
  - ▶ Start Emacs by typing `emacs -nw` in your terminal
  - ▶ Press **C-x C-f** (*Ctrl* and `<x>` together, then *Ctrl* and `<f>` together) to open the `.emacs` file
  - ▶ then enter `.emacs` (check the minibuffer)
  - ▶ Write including the brackets:  

```
(load "~/ess-5.8/lisp/ess-site")
```
  - ▶ Press **C-x C-s** to save the `.emacs` file
  - ▶ Press **C-x C-c** to close Emacs
- ▶ That's it. You can test if ESS works by typing `emacs -nw test.R` into your terminal and check whether the **major mode** is `ESS[S]`.

# Emacs: First Steps

- ▶ I can not stress enough to run the Emacs tutorial within Emacs. Press `C-h t` (`CTRL` and `<h>` together and then `<t>`).
- ▶ Cancel some command sequence with `C-g`.
- ▶ Close Emacs with `C-x C-c`.  $\Rightarrow$  Read the minibuffer.
- ▶ Get help on the meaning of a keystroke combination with `C-h k` for describe-key
  - ▶ `C-h k C-x C-s` gives the description of the `C-x C-s` command.

# Emacs: Open, Save and Close a File

- ▶ `C-x C-f` open a file (find-file)
  - ▶ enter the path into the minibuffer
  - ▶ use `TAB` for auto completion, `TAB TAB` splits the screen for a file list
  - ▶ if you enter a file name that does not exist, Emacs creates this file
  - ▶ if you entered the wrong path use `C-x C-v` to open an alternative path
  
- ▶ `C-x C-s` saves the file at the current path.
  
- ▶ `C-x k` closes the current file, see [buffers](#).

## About `~ /`

`~ /` is a short notation linking to your home directory. The location of your home directory is different under Linux, OSX and Windows. Read the [Wikipedia article](#) for more information.

## Emacs: Marking/Selecting Text

Selecting text is important if you want to copy, move or delete some text.

You are probably used to select text with your mouse resulting in some visual effect like `selected text = region`. Emacs behaves the same way if you run a GUI version of Emacs. In the terminal version Emacs will not react to mouse events.

For Emacs, the selected region always extends from the *mark* to the *point* (cursor). The *mark* is set using `C-SPACE`, you will see “Mark set” in the minibuffer.

You can swap your *mark* and *point* using `C-x C-x`. This helps to see whether you placed your *mark* in the right position or not.

Emacs highlights your region if enter `M-x transient-mark-mode`. This mode can be loaded automatically, see [customizing Emacs](#).

# Emacs: Delete, Copy, Cut and Paste Text

You can delete single characters using `DEL`, `Backspace` and `C-d`.

To *kill* something means that Emacs copies the killed (deleted) text into the *kill ring* such that you can later access it. You can kill regions, words and lines as follows

- `M-d` kill from *point* to the end of current word
- `M-DEL` kill from *point* to the beginning of current word
- `C-k` kill from *point* to the end of the line (one line break)
- `M-k` kill from *point* to the end of a sentence (two line breaks)

## Copy a region

Copy a region using `M-w`.

## Cut a region

Cut/Kill a region using `C-w`.

## Paste the region

Paste the last entry in the *kill ring* using `C-y`. In Emacs language, pasting is called *yanking*.



# Emacs: Rectangle Editing

When you wish to copy and paste rectangular text blocks you will find the `C-x r k`, `C-x r y`, `C-x r c` and `C-x r o` commands useful.

1. Set the *mark* on the upper left corner with `C-Space`

```
File Edit Options Buffers Tools Help
Owner  Brand   Color  Age
John   VW         silver 4y
Bob    Porsche   green  2y
Lisa   Toyota    red    6y
-uuu:*--F1 EmacsRectangle.txt Top L1 (Text)-----
Mark set
```

2. Go to the lower right corner and kill the region with `C-x r k`

```
File Edit Options Buffers Tools Help
Owner  Color  Age
John   silver 4y
Bob    green  2y
Lisa   red    6y
-uuu:*--F1 EmacsRectangle.txt Top L5 (Text)-----
```

3. Paste the region using `C-x r y`

```
File Edit Options Buffers Tools Help
Owner  Color  Age  Brand
John   silver 4y   VW
Bob    green  2y   Porsche
Lisa   red    6y   Toyota
-uuu:*--F1 EmacsRectangle.txt Top L5 (Text)-----
```

4. Add an additional blank rectangle of the region size with `C-x r o`.

# Emacs: Buffers

- ▶ Emacs manages multiple files in *buffers*.
- ▶ Use the following commands to work with buffers
  - `C-x C-b` show all buffers
  - `C-x b buffername` switch to buffer “buffername”
  - `C-x C-s` save active buffer
  - `C-x s` save all buffers
  - `C-x k` kill buffer

# Emacs: Windows

- ▶ A *window* is an area on the screen through which a *buffer* gets displayed.
- ▶ You will most commonly use
  - `C-x 2` to split the window horizontally
  - `C-x 3` to split the window vertically
  - `C-o` to move between windows
  - `C-x 1` to show only the active window
  - `C-x 0` to close the active window
  - `C-x ^` to make the window taller
  - `C-x }` to make the window wider
  - `M-x shrink-window` to shrink window horizontally
  - `C-x {` to shrink the window vertically

## Emacs: Search and Replace

Emacs provides several variations of search and replace operations, see the [GNU Emacs Manual](#).

For now, if you are looking for some *search string*, use `c-s` and enter the *search string* into your minibuffer. Emacs will jump to the next occurrence between your current position and the end of your file. Press repeatedly `c-s` to find the next occurrence. Once you reached the end of the file Emacs will once not respond to `c-s` and then on the second execution of `c-s`, Emacs will jump to the beginning of the file.

Use `c-g` to abort the search and return to where you were.

Use `M-x M-%` to search and replace a string. Follow the guidance on the minibuffer.

More advanced search and replace techniques will be discussed later in the chapter Regular Expressions.



## About Eclipse

Eclipse began as an IBM Canada project but is these days an open source project maintained by the [Eclipse Foundation](#) with Headquarters in Ottawa.

Eclipse was written in Java and primary thought as a Java IDE. However Eclipse was designed as an universal tool platform (that is, language neutral) and many plug-ins have been written to support R, C, C++ and Python to name a few.

Eclipse is less exotic in its behavior than Emacs. In fact you won't have to learn too many new key combinations in order to get started. However Eclipse is a large (ca. 160MB) program which needs an installed Java Runtime Environment (JRE). You also have to get familiar how projects are organized in Eclipse.

For many tasks Emacs still remains the more productive environment. Once you learned all the Emacs keystroke combinations, you can change Eclipse's key scheme to be the one of Emacs. Use the Eclipse menu (Window, Preferences, General, Keys, Scheme).

# Installing Eclipse

Jeromy Anglim has written a blog about [Getting Started with Sweave: R, LaTeX, Eclipse, StatET, & TeXlipse](#).

Either follow his instructions or

1. Download and install [Eclipse Classic](#).
2. Install the [StatET](#) plugin.
3. Configure StatET using [Longhow Lam's Eclipse and the R plug-in StatET](#) documentation.
4. Install the [TeXclipse plug in](#) following the [configuration page](#).