Getting Started

John Leonard

November 11, 2012

Abstract

The purpose of this guide is to help new research team members get up to speed with our common tools and practices. Whether you are student pursuing a degree, or researcher or full-time faculty member, you'll need to need to know how to install and use the tools described in this document.

This is a living document, I welcome all team members to take part in it's evolution. If nothing else, you'll get good practice working with the tools. At best, you'll be helping everyone get better at doing what we need to do.

Tasks to accomplish

1 Verify tools and skills

1.1 Get a Good Text Editor

The first step towards programming is a basic ability to write and edit computer scripts. All operating systems have a text editor pre-installed, but it is typically inadequate for writing computer code. Good text editors will have features such as find/replace, line numbering, LATEXexecution, colored command text, and even spell check. There are many good text editors for free, so don't pay money for anything.

For Windows:

- **Notepad** The default pre-installed text editor

- **Others** Somebody else will need to fill out this section.

For Mac:

- **TextEdit** The default pre-installed text editor.

- **Emacs** The go-to text editor for Unix, developed as open-source software.

- **Aquamacs** Emacs built in a Mac OSX-native application, with some extra cool stuff (this is what I run).

1.2 Open a command line window

The first thing you need to know is how to open a command line window.

Windows   Below is a link to a youtube video that does a good job of covering the basics:

- http://www.youtube.com/watch?v=qrQMLOykSH4

Macintosh   The Macintosh command line interface is called' "Terminal", and is available in /Applications/Utilities/Terminal. Also, you can type "Terminal" in the spotlight search line.

1.3 Verify and install subversion

Before you can checkout the practice project, you'll need to install subversion client. The command below will help you determine if one is already installed on your machine.

```
>> svn --version --quiet
```

If not installed, follow the link below to download and install the appropriate version for your computer:

- http://subversion.apache.org/packages.html

For windows there are plenty of choices. I usually use the tools from SLIK:

- http://www.sliksvn.com/en/download

Install the command line client (on the left) and a visual tool like TortiseSVN (found on the right.)

Once installed add the following keyword and variable to the system environment:

```
SVN_EDITOR=notepad
```

Macintosh   Subversion is included in the default installation of Mac OSX.

1.4   Verify and install RTOOLS (Windows only)

On Linux computers most tools for creating programs are almost always installed. You can check this by bringing up a command line window and issuing the command:

```
>> make --version
```

The project team plans to develop tools on both Mac and Windows platforms at the same time. To facilitate this, we need to install a set of special linux-like tools for Windows. These have been gathered and built to work cleanly with R and Latex. You can download these tools from:

- http://www.murdoch-sutherland.com/Rtools/

During installation, RTOOLS will ask if you want to modify the PATH. The answer: YES!

If you have already installed programming tools like MAKE, GREP or others, or if you have already installed CYGWIN on your windows box, pay close attention to the machine's behavior after you install RTOOLS. These tools will be placed first in the path, overriding any other of the older tools.

1.5   Verify and install updated version of Make (Windows only)

I found in writing the tests to verify that everything would work under windows or Mac, I need to use features found in Make 3.81. At the time of this writing, make version 3.79 was shipping with rtools. You can download the latest version of Make from sourceforge:

- http://gnuwin32.sourceforge.net/packages/make.htm

Again, if asked during installation to modify the path, answer YES!

If during building of any of the scripts, you see something goofy, it may be different version of RTOOLS and other installed software conflicting.

1.6 Verify and Install Make (Linux or Mac)

On Linux computers most tools for creating programs are almost always installed. You can check this by bringing up a command line window and issuing the command:

```
>> make --version
```

On Mac computers, Make is not installed by default, but can be found in the Xtools developer package. Typically installations of Mac OSX do not include this package, though it is available on the installation CDs. If you do not have the OSX installation CDs (I didn't), you will need to join the Mac Developer Program before you can download Xcode. To join, visit:

- http://developer.apple.com/devcenter/mac/index.action

and click on "Join the Mac Developer Program." When you have created your account, you may return to the previous webpage and click "Download Xcode." When these programs are installed, you can verify make by issuing the command:

```
>> make --version
```

1.7 Verify and install perl

While you may not use perl right off the bat, it is a good idea to spend the time now to make sure it is installed and working on your machine. In the future, we may need to write quick and dirty programs to transform data from one format to another and perl is definitely the tool for the job!

```
>> perl -V:version
```

If perl isn't installed, you'll need to do a bit more work. For windows, the best version of perl availble can be found at:

- http://www.activestate.com/activeperl/downloads

Macintosh   Perl is intsalled on Mac OSX in the default installation.

1.8 Verify and install R

R is an open-source statistical package. Different versions are available from the "Download and Install R" section at:

- http://watson.nci.nih.gov/cran_mirror/

Macintosh   R is available from http://cran.cnr.berkeley.edu/, and the default download for OSX 10.6 installs both 32- and 64-bit versions (R64).

1.9 Verify and install LaTeX

LaTeX[1] is a typesetting language originating in the 1970's by Donald Knuth as Tex. It has since gone through several iterations with the current version LaTeX2. It runs on different platforms, below is the link to the Windows version.

- http://miktex.org/

For some reason, MikTex is a challenge to install. Once it is running, it seems to work like a charm. If you find cool tips or tricks that made the installation easier, please document them here.

Macintosh To use LaTeXeffectively on Mac OSX, I recommend two software packages. The first, MacTeX(http://www.tug.org/mactex/), installs the LaTeXscripts and styles that are used.

The second, TeXShop (http://pages.uoregon.edu/koch/texshop/), takes advantage of the pdf manipulation tools built into Mac OSX to create a very nice LaTeXeditor. This package is not required, but will make your life easier.

Now that R and LaTeXare installed on your Mac, you need to tell R where to find Sweave. To do this, execute the following commands in the Terminal:

```
$ mkdir -p ~/Library/texmf/tex/latex
$ cd ~/Library/texmf/tex/latex
$ ln -s /Library/Frameworks/R.framework/Resources/share/texmf Sweave
```

2 Practicing with the tools

The following sections provide brief activities to accomplish in order.

2.1 Download this practice repository

```
>> svn list https://svn.ece.gatech.edu/repos/transport
>> cd (whereever you want to keep your files)
>> svn checkout https://svn.ece.gatech.edu/repos/transport/GettingStarted ./GettingStarted
>> cd GettingStarted
>> dir  (or ls -alF on unix machines)
```

2.2 Check the environment

For instructions in the Makefile to work properly, make needs to be told where to find all the programs that we installed above. In some cases (like svn and make), the programs were installed on the command line "path" meaning that they can be referenced from any folder or directory.

The environment target lists key environment settings available to make.

```
>> make environment
```

2.3 Customize the environment for your computer

```
>> set (and look for the setting COMPUTERNAME) in the list
>> notepad src\Machines.cfg
```

Then edit the file to add a new IFEQ block for your computername, copy the various macro assignments and update them to reflect where the files can be found on your machine.

2.4 Build an existing document

```
>> cd .. (you need to be in the main GettingStarted folder)
>> make doc
```

If everything is set up properly, you should be able to build the document `GettingStarted.pdf` with this comment.

2.5 Creating your own sandbox and makefile

Once you know that everything is working properly on your computer, it's time to give a shot to creating something from scratch.

```
>> cd whereever (you need to be in the main GettingStarted folder)
>> cd users
>> mkdir jl66 (use your name)
>> cd jl66
>> notepad Makefile.  (note the period - it is necessary to fool notepad)
```

Because Mac has no command or application called `notepad` you will need to use the command `touch` instead. Then, inside the Makefile, add the following lines:

```
menu:
    @echo + hello world!
```

Be sure to use the `<TAB>` character. Make does not accept `<SPACE>` at beginning of commands. Often, this is the trickiest part of writing makefiles - those pesky hidden spaces!

2.6 Create your first PDF document

Here are a few quick steps to create a working PDF document.

1. Change into your sandbox folder,

2. Create a bare bones RNW file, let's refer to it as Sample1.rnw

3. Open the Makefile in the main folder and copy the two macros that assignments for GettingStarted.pdf and GettingStarted.tex.

4. Paste these macros into your own makefile and edit the macros, substituting Sample1 with GettingStarted in the several different places.

5. Save the Makefile file and rnw files.

6. give it a shot by issuing `make Sample1.pdf` from the command line.

2.7 Adding R to your document

Once you have a working RNW file, then you can layer the next complexity into it - adding and running R code. Most of the hard work has already been accomplished - we've already been sweaving the RNW file - we just haven't included any real R code.

Add the following code into the main body of your document:

```
<<echo=TRUE>>\=
"hello world"
@
```
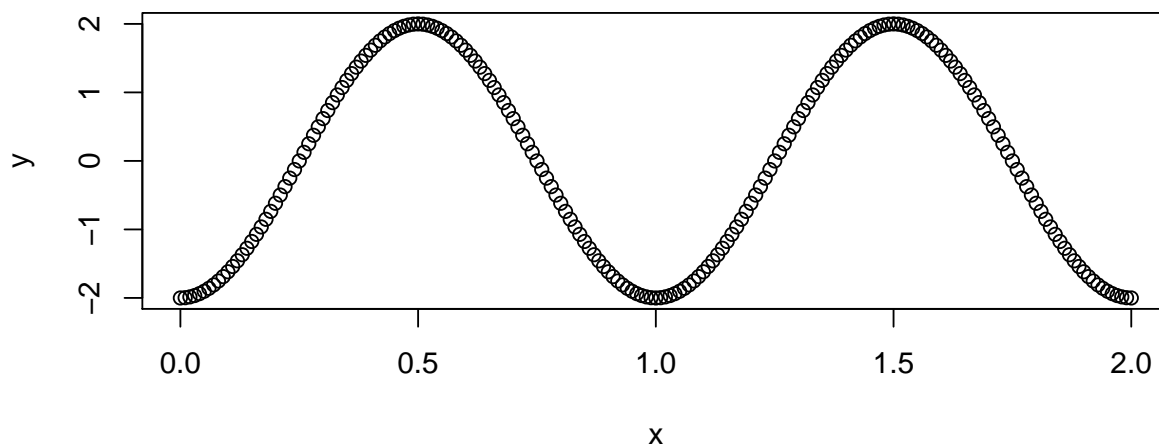
Then save the file and remake the PDF. If all works OK, you should see fix-spacing text appear in your file like this:

```
"hello world"

## [1] "hello world"
```

Now you're dangerous. Try the following code:

```
<<fig=T,echo=F,fig.width=6.5,fig.height=3.5>>\=
x=seq(0,2,by=0.01)
y=2*sin(2*pi*(x-1/4))
plot(x,y)
@
```


3   Learning more - a collection of possible next steps

The sections below provide a brief overview and a URL to a location on the web that you can learn more.

3.1 CRAN - Comprehensive R archive network

This is THE place to find already written R packages that solve specific problems.

3.2 CPAN - Comprehensive Perl archive network

This is THE place to find already-written PERL packages.

3.3 CTAN - Comprehensive Tex archive network

This is THE place to find already written LaTex package and other goodies to make your documents look even prettier.

4 Frequently asked questions

Use this section to add quick question-and-answers as you find them. As the section gets really big, we may need to reorganize things a bit.

References

[1] Leslie Lamport, *LaTeX: A Document Preparation System.* Addison Wesley, Massachusetts, 2nd Edition, 1994.