MATH 5750/6820 Assignment Instructions

Curtis Miller

08/25/2016

Introduction

Welcome to MATH 5750 (or MATH 6820, for graduate students)! In this course, we will be discussing statistical analysis of time series. In this course, we will be describing both the mathematical theory and framework behind time series analysis as well as how this analysis is actually performed in practice, using the R statistical software. Thus, you can expect assignments to cover both aspects of time series analysis, as well as a final project requiring you to use R to complete.

Installation

The University of Utah math department computers include installations of R, as well as the R IDE, RStudio. We recommend that you have an installation of R on your personal computers (the department installations do not always work well). You can download R from the Comprehensive R Archive Network (CRAN), at https://cran.r-project.org/. While the CRAN installation often comes with a minimalist integrated development environment (IDE), we *highly* recommend using RStudio, which can be downloaded at https://www.rstudio.com/. RStudio allows for much easier interaction with R and provides many convenience features such as syntax highlighting, tab completion, a plot viewer, and a manual file viewer. Both R and RStudio are free of charge and open source.

Our textbook, *Time Series Analysis and Its Applications with R Examples*, includes several examples of R code examples used for time series analysis and analyzes a number of different data sets. They also use functions written by the authors for performing common tasks or looking at common diagnostic plots. The authors have written an R package called **astsa** that includes all data sets used in the book, along with their functions. **astsa** is available on CRAN and can be easily installed directly from the R prompt with the command:

```
install.packages("astsa")
```

You can then load the package and all its contents into working memory with the command:

```
library(astsa)
```

Other packages are used throughout the book. Usually the command `install.packages("SuperAwesomePackageName")` will handle downloading and installing these packages.

Learning R

While no class that teaches R is explicitly required to be enrolled in this course, knowledge of R is assumed, and most of the textbook assumes some knowledge of R. If you feel your R knowledge is weak, there are resources available to you.

- Appendix R in the textbook is an R supplement with some R basics. Start there first.

- Most of R involves working with functions that accomplish certain tasks. If these functions are from a package (including functions that come installed with R), they are usually well documented. Say, for example, you wanted to see how to use the `arima()` function. You can access the documentation for `arima()` with the R commands `?arima` or `help("arima")`, and a general search can be done with `??arima`.
- Curtis Miller has used R in many contexts, including time series analysis. You are welcome to meet with him either during his office hours (Tuesdays and Thursdays from 2:00 to 3:25) or by appointment. Also, he is teaching (and has taught) the MATH 3070 R lab, which teaches R basics. He will gladly make his lecture notes available to you.
- The University of Utah has a mailing list where questions about R can be asked to a community of mailing list subscribers, called `Splus-users`. You can subscribe to this mailing list at http://mail.math.utah.edu/mailman/listinfo/splus-users.
- Consider asking some of the instructors of the MATH 3070 R lab if they will allow you to sit in on their class. Curtis Miller teaches a lab every Thursday from 5:30-7:00, and is willing to allow you to sit in so long as there is space available and every enrolled student has a seat. (Priority is given to enrolled students.)
- Google + StackExchange = A++ (usually)

Homework Submission

You are required to submit your R homework in a clean, typed format. The best way to do so is to use R Markdown or Sweave/knitr. Sweave/knitr allows you to embed R code directly into a LaTeX, similar to the following:

```
\section{Example}
This is \LaTeX code. I will follow this with an awesome \textbf{R} chunk!

<<chunk_name>>=
# This is R code
print("Hello, world!")
rnorm(10, mean = 10, sd = 100)
@

And now I am back to typing in \LaTeX! And here is the definition of the
mean: $\bar{x} = \frac{1}{n}\sum_{i = 1}^{n}x_i$, and the variance is:

\[\sigma^2 = \frac{1}{n-1}\sum_{i = 1}^{n}(x_i - \bar{x})^2\]
```

The code chunk `chunk_name` between `<<chunk_name>>=` and `@` is R code, while outside of the chunk is LaTeX. When the document is compiled (using knitr and pandoc), the R code will be run and the results will be inserted directly into the document. This means that you can simultaneously write the R code for your assignments along with the write-up in LaTeX, simplifying your life.

R Markdown is like Sweave/knitr but uses the Markdown language to format text, which is very easy to use (far easier to use than LaTeX, maybe even easier than a word processor like Microsoft Word!). Here is an example of what an R Markdown document would look like.

```
# Example
This is Markdown code. I will follow this with an awesome **R** chunk!

 ```{r chunk_name}
# This is R code
```
```

```
print("Hello, world!")
rnorm(10, mean = 10, sd = 100)
```

```
And now I'm back to typing in Markdown! And here is the definition of the
mean: $\bar{x} = \frac{1}{n}\sum_{i = 1}^{n}x_i$, and the variance is:

$$\sigma^2 = \frac{1}{n-1}\sum_{i = 1}^{n}(x_i - \bar{x})^2$$
```

(Notice that the code for math is the same? R Markdown supports MathJax, which allows you to write math statements almost exactly like you would in LaTeX!) The idea of R Markdown is similar to that of Sweave/LaTeX: a human-readable document is created with R code and its output embedded cleanly into the result. An R Markdown document can then be compiled and exported as an HTML page or as a PDF. (This document, for example, was written using R Markdown and compiled as a PDF.)

Sweave/knitr documents have the extension `.rnw` while R Markdown documents have the extension `.rmd`. RStudio makes creating, editing, and compiling these documents easy, with the "Knit PDF" or "Knit HTML" buttons built right into the editor (though some programs and packages may need to be installed first before this works). We will require that you type up your homework using either of these options, and we will even provide templates (in either format) for you to submit your homework; all you will need to do is fill out the code blocks with the comments `# Your code here`, or provide your written response to questions where marked. If you want to learn more about using knitr, visit http://yihui.name/knitr/demo/rstudio/ and http://yihui.name/knitr/demo/minimal/ and http://yihui.name/knitr/options/ . You can learn more about R Markdown by visiting http://rmarkdown.rstudio.com/ and a cheat sheet is available at http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf.

Here are a few reasons why we require that you use either knitr or R Markdown for your homework:

- It makes grading your assignments easier, since input, output, and commentary are all included in one well-formatted, human-readable document.
- The emphasis on a human-readable document will encourage you to explain your code and methods while you write the code (this is called literate programming).
- Many users of R in the real world use R Markdown, LaTeX, and knitr for communicating methodology and results.

As for problems that are not necessarily R problems, you can submit handwritten solutions, but we would prefer all your solutions be typed up in one document.

We look forward to having you in the class and seeing your assignments, and hope you have an excellent semester.