TEACHING COLLEGE LEVEL LINEAR ALGEBRA USING THE R PROGRAMMING ENVIRONMENT

Scott K. Hyde
Department of Mathematics
Brigham Young University – Hawaii
Laie, HI, 96762
hydes@byuh.edu

Abstract

Using technology in the classroom provides many benefits for enhancing the learning of concepts of many mathematical subjects. As one such technology, the R statistical programming environment provides students greater access to the concepts and material for a Linear Algebra course. The main function of the R statistical programming language is interactive analysis of statistical problems. However, because many statistical problems require the use of matrices, R is also a very powerful matrix program. Additionally, R is an open source program, which means that it is free for use in the classroom and at home, with no licensing fees or other costs. However, R does not have symbolic manipulation, which typical computer algebra systems of today offer. When symbolic manipulation is not required, then R provides an alternative free solution for universities and colleges for which licensing costs for CAS software are prohibitive. In addition, R is available for many different operating systems, including Windows, MacOS, Unix, and Linux.

This paper summary of my presentation introduces the use of R in teaching Linear Algebra concepts. Topics include instructions for obtaining and installing R on a Windows computer and how to create and manipulate vectors and matrices using various functions in R. Matrix factorizations are also included. Because of space constraints, only one example is given on using R to enhance the understanding of Linear Algebra.

When R does not provide a particular operation, R can be extended, allowing functions and operations to be added to fit the needs of the user. This is accomplished either by writing new packages, or by installing additional packages from the Comprehensive R Archive Network (CRAN). Instructions on the installation of a package for R from CRAN are given.

1 Introduction

What is R? R is a statistical programming environment, a system for statistical computation, and a system for the production of high quality graphics. R is a computer language and comes with a debugger. The R software package allows for the interactive analysis of data. Since many statistical problems use matrices, it also allows the interactive analysis of matrices. Finally, with R you can run programs which are stored in text files.

R was originally written by Ross Ihaka and Robert Gentlemen[1] as a research project. It has since become a highly regarded open source software project, with a large number of contributors.

A fair question to ask would be, "Why should R be used?". The first reason is that R is open source. What open source means is that the code is free to be developed by anyone. R has hundreds of developers, some of whom are paid by the Free Software Foundation (because R is a GNU Project), and many who are not paid. Open source also means that the software is free to use. It is free for anyone to download and install from `http://www.r-project.org`. In addition, R is available for many computer platforms, including Unix, Linux, Apple (MacOSX), Windows, and others.

2 Commands

Because of space limitations, only a short list of commands available in R is given.

`c()`	concatenate	`+ or -`	add or subtract
`:`	create vector	`det`	determinant
`seq`	create a sequence	`t`	transpose
`rep`	replicate	`diag`	create a diagonal matrix
`matrix`	create matrix	`solve`	inverse; solve a system
`[]`	access entries	`cbind`	augment matrices horizontally
`dim`	size of a matrix	`rbind`	augment matrices vertically
`length`	length of a vector	`cumsum`	cumulative sum of entries
`nrow`	no. of rows	`cumprod`	cumulative product of entries
`ncol`	no. of columns	`$`	access elements in a R object
`sum`	sum of entries	`eigen`	Eigenvalue Decomposition
`prod`	product of entries	`svd`	Singular Value Decomposition
`min`	min of entries	`lu`	LU decomposition
`max`	max of entries	`Schur`	Schur decomposition
`*`	elementwise multiply	`cholesky`	Cholesky decomposition
`%*%`	matrix multiply	`qr`	QR decomposition

Table 1: Selected list of commands for R. Note: * performs a Hadamard product[2]. Note: some of the factorizations are only available with the CRAN package `Matrix`.

3 Examples

An example that I have given my class, which was inspired by an MIT Linear Algebra problem [3], shows how orthogonal polynomials are related to the columns of an orthogonal matrix (the QR factorization).

First, construct a orthonormal basis $\{f_0, f_1, f_2\}$ by applying the Gram-Schmidt algorithm to the set $\{1, x, x^2\}$, using the inner product $\langle f, g\rangle = \int_0^1 f(x)g(x)\,dx$. Define $\boldsymbol{a} = 1$, $\boldsymbol{b} = x$, and $\boldsymbol{c} = x^2$. The algorithm proceeds as follws:

$$\boldsymbol{A} = \boldsymbol{a} = 1$$

$$\boldsymbol{B} = \boldsymbol{b} - \frac{\langle \boldsymbol{A}, \boldsymbol{b}\rangle}{\langle \boldsymbol{A}, \boldsymbol{A}\rangle}\boldsymbol{A} = x - \frac{\langle 1, x\rangle}{\langle 1, 1\rangle}1 = x - \frac{\int_0^1 x dx}{\int_0^1 dx}1 = x - \frac{1}{2}$$

$$\boldsymbol{C} = \boldsymbol{c} - \frac{\langle \boldsymbol{A}, \boldsymbol{c}\rangle}{\langle \boldsymbol{A}, \boldsymbol{A}\rangle}\boldsymbol{A} - \frac{\langle \boldsymbol{B}, \boldsymbol{c}\rangle}{\langle \boldsymbol{B}, \boldsymbol{B}\rangle}\boldsymbol{B} = x^2 - x + \frac{1}{6}$$

The length of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ is $||\boldsymbol{A}|| = 1$, $||\boldsymbol{B}|| = \frac{1}{2\sqrt{3}}$, and $||\boldsymbol{C}|| = \frac{1}{6\sqrt{5}}$. Therefore, the orthonormal basis set for polynomials of degree two or less is

$$f_1 = \frac{\boldsymbol{A}}{||\boldsymbol{A}||} = 1, \quad f_2 = \frac{\boldsymbol{B}}{||\boldsymbol{B}||} = \sqrt{3}(2x - 1), \quad \text{and } f_3 = \frac{\boldsymbol{C}}{||\boldsymbol{C}||} = \sqrt{5}(6x^2 - 6x + 1)$$

In R, this same procedure can be approximated by using 1000 points and approximating the integral with summation instead. R will do Gram-Schmidt for us via the function qr:

```
> x = seq(0,1,along=1:1000)
> A = cbind(x^0, x^1, x^2, x^3, x^4, x^5)
> QR = qr(A)
> Q = qr.Q(QR)*sqrt(1000)
```

The $\sqrt{1000}$ factor matches the approximate "integral" inner product rather than the ordinary dot product. A plot $\boldsymbol{x}$ versus the columns of $\boldsymbol{Q}$ can be seen in Figure 1. They are the orthogonal "polynomials" up to degree 5.

```
> colors=c("black","blue","green","red","purple","orange","black")
> plot(x,Q[,1],ylim=range(Q),type="l",col=colors[1],ylab="columns of Q")
> for (i in 2:6)
+   points(x,Q[,i],type="l",col=colors[i])
```

Note that the columns of $\boldsymbol{Q}$ will only match the orthogonal functions f_1, f_2, and f_3 within a sign change. R chose -1 for the first column of $\boldsymbol{Q}$ (as compared to $f_1 = 1$, and the second column matches f_2, and the third column of $\boldsymbol{Q}$ is off by a sign. Hence,

to plot the functions on top of each other, we need to change the sign of f_1 and f_3. Because of space constraints, the code is omitted.

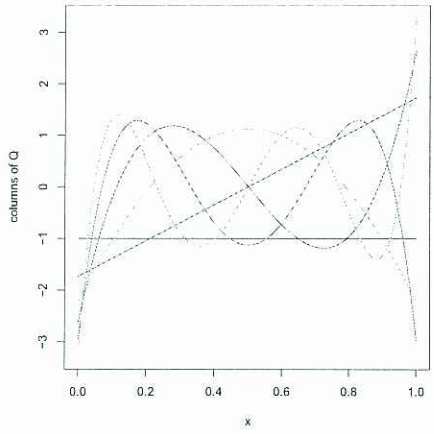

Figure 1: Plot of $\boldsymbol{x}$ versus columns of $\boldsymbol{Q}$

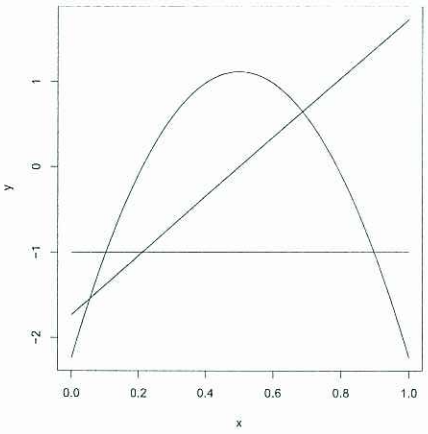

Figure 2: Plot of the orthogonal polynomials versus first three columns of $\boldsymbol{Q}$.

4 Conclusion

R is a very powerful matrix program which can be useful for teaching linear algebra. Learning how to use R is within the grasp of undergraduate students. Students will find that the program enhances their understanding of linear algebra. I received several positive comments from students in my undergraduate linear algebra course. One comment was that they were grateful for the understanding they gained by seeing the connection between orthogonal polynomials and orthogonal matrices through the use of the graphing tools of R.

5 Appendix

5.1 Installing R on a Windows System

To install R on a Windows system, follow these steps:

1. Navigate to `http://www.r-project.org`
2. Click on the word "CRAN" underneath the Download on the left.
3. Choose a mirror (preferably one that is close to your location).
4. Select the operating system you use (Windows in this example).
5. Click on "base".

6. Click on the executable "R-2.8.1-win32.exe" (or similarly named R executable).
7. To install, double click on the file "R-2.8.1-win32.exe" that you downloaded.
8. A more detailed set of instructions can be found at `http://jekyll.math.byuh.edu/other/howto/R/R.shtml`, including the instructions for the installation of a couple of very nice editors of R code.

5.2 Installing an R package

To install a package for R that resides on the Comprehensive R Archive Network (CRAN), first run R. Then use the command `install.packages()` (or click on the Packages menu, then select "Install Package(s)"). You will then be asked to

1. Select a mirror (choose one close to you) and then click OK.
2. Select a package to install (scroll through the names until you find the package you want) and then click OK.
3. To load the package you installed, type `library(<name of package>)`, where `<name of package>` is the name of the package you installed. You can also load the package by selecting the "Packages" menu, then select "Load Packages".

To install a package located outside of CRAN, you need to specify additional arguments to the `install.packages()` command. For example, to install the `m343linalg` package that was used for teaching Linear Algebra at Brigham Young University in Hawaii, you type the following commands:

```
> where="http://jekyll.math.byuh.edu/rlibs/"
> install.packages("m343linalg",contriburl=where)
```

To get help on a particular package, select the "Help" pull down menu, then select "Html help". Your default web browser should pop up. Select the "Packages" item. Then you can select from the list of packages the one you need help on. If you know the name of the command you want help on, you can simply type `help(<command>)` at the R command prompt.

References

[1] Kurt Hornik, *The R FAQ*, `http://CRAN.R-project.org/doc/FAQ/R-FAQ.html`, 2008, ISBN 3-900051-08-9.

[2] James R. Schott, *Matrix analysis for statistics*, John Wiley & Sons, 1997.

[3] MIT Staff, *Problem set 6*, `http://web.mit.edu/18.06/www/Fall07/psetsF07.html`, 2007.