

## Using Java in introductory statistics

NANCY E. BRIGGS and CHING-FAN SHEU  
*DePaul University, Chicago, Illinois*

In this paper, we discuss the use of Java, a programming language for the Internet, in introductory statistics courses. Monte Carlo estimation and simple linear regression are used as tutorial examples of the versatility of Java. Advantages over traditional instructional methods and other programming languages are examined.

As popular interest in the Internet has increased recently, the possibility of using the Internet as a classroom tool has become more viable. A variety of educational web-sites are available to any person with access to the Internet, and an instructor can easily use these resources in lectures. Numerous sites containing information in psychology exist on the World-Wide Web (WWW); for example, Rosen and Petty (1997) identify a number of sites that instructors may find helpful in incorporating Internet resources into their classes.

In 1995, Sun Microsystems released Java, a computer language designed for use on the Internet. Java has allowed web designers to incorporate new graphics, animation, and interactivity into their sites. Designers for educational websites have benefited from this advance as well, enhancing the informational value of their sites. This development may become one of the most important aspects of using the Internet as an instructional tool.

Java has many useful features for classroom demonstration. Its best feature is its platform independence. A program in Java is first compiled into byte code before it is interpreted by an Internet browser. The output of the program can be viewed by anyone connected to the Internet on his or her native machine, provided the browser is Java enabled. This means that once the code is written and available on the WWW, the Java program can be viewed and used everywhere. Many other programming languages, such as C and BASIC, have been used in classrooms. However, these languages require that applications written in them be compiled on the native machine in order to be used. This entails more work for the user in copying the source code (assuming it is available) and compiling it on the computer, and it is more expensive because a compiler for that specific language must be purchased and installed. Java does not require these steps, and this makes Java programs much more accessible than programs in other languages.

Second, Java comes with a variety of tools useful to educators. Abstract window toolkits and built-in support for

animation enable programmers to create multimedia and interactive programs. These packages bring audiovisual presentations within the reach of instructors who are not experienced programmers. With Java, we do not have to hire professional programmers to express our pedagogical ideas in multimedia form. We can learn to do it ourselves.

Lastly, Java is an object-oriented language. This feature makes it easy to recycle portions of previously written programs. This enables novice programmers to learn more quickly by adapting existing programs to their needs.

The most common Java applications on the Internet are applets, or little applications, embedded in web pages. Gamelan (see the Appendix for web address) has a large collection of applets for educational purposes, in subjects ranging from foreign languages to chemistry. Presently, most of the applets illustrate principles of physics and mathematics. Unfortunately, there are only a few statistical programs in Java, and even fewer applets demonstrate psychological principles. However, we have listed in the Appendix some sites on the WWW by name and address that may provide psychologists information useful in the classroom, and we have included a brief description of the sites.

Unfortunately, instructors cannot simply link up these sites as on-line resources for courses and ask students to visit them. Traffic on the Internet is often congested; the sites are not always available, and they can disappear without notice. Also, most sites do not make the source code for the applets public. This means that even if the instructor finds an applet appropriate for his or her class, it cannot be modified to fit the instructor's needs. These considerations suggest that psychologists might consider learning to program in Java themselves, not only to write their own applets but also to modify and customize ones for which they do have the source code.

Although there are many on-line Java tutorials (see Appendix), most users will find a reference book helpful. Winston and Narasimhan (1996), Deitel and Deitel (1997), and Horton (1997) are good introductory books. To plug and play Java immediately, we recommend Hoff, Shaio, and Starbuck (1996) and Pew (1997). For experienced programmers, Linden (1997), Flanagan (1997), and Niemeyer and Peck (1996) provide a more complete description of the language.

---

Correspondence should be addressed to N. E. Briggs, Department of Psychology, DePaul University, 2219 N. Kenmore Ave., Chicago, IL 60614 (e-mail: nbriggs@condor.depaul.edu).

The goal of this paper is to illustrate the use of Java in introductory statistics courses. To make our listings of codes useful to everyone, we first provide information on how to gather the necessary pieces to write, compile, and execute a Java program. We then give tutorial examples illustrating the implementation of Java on Monte Carlo estimation and simple linear regression.

## USING JAVA

Java programs can be used as complete stand-alone applications. For our purposes, however, it is the applets that are of interest.

A simple Java program, `LineDown.java`, illustrates how applets and web browsers work together. This applet will display a red line from the upper left-hand corner (the point 0, 0) to the lower right-hand corner (the point 100, 100) of a square on a Java enabled web browser. We choose this example to show you how easy it is to create a picture using Java. The code syntax is explained in any Java book:

```
//LineDown.java
public class LineDown extends java.applet.Applet {
public void paint (java.awt.Graphics g) {
g.setColor (java.awt.Color.red);
g.drawLine(0,0,100,100);
}
} //End of class
```

To see what the program does, you need to embed it in a document written in Hypertext Markup Language (HTML) like the one below, `LineDown.html`. Java-capable browsers read HTML and display the contents. The browser recognizes the opening tag `<applet>` and the closing tag `</applet>` and executes the code:

```
<html>
<head>
<title> A First Example </title>
</head>
<body bgcolor=white>
<applet code=LineDown.class width=100
height=100>
</applet>
</body>
</html>
```

In the following, we provide step-by-step instructions on how to compile and run the program, assuming that your personal computer is connected to the Internet, has a Java enabled web browser, and runs on Windows 95. The Java Developer's Kit is rather large, so a minimum of 22 MB are needed for the program itself and 6 MB for the documentation.

### Obtaining and Installing a Copy of Java

This information is also available on JavaSoft's website at <http://java.sun.com/products/jdk/1.1/installation-win32-x86.html>.

1. Open your web browser to location <http://www.javasoft.com/products/jdk/index.html> and select JDK 1.1.5. This is the latest version of Java Development Kit as of January 1998. Download the software to the directory `C:\JDK1.1.5` (create one if necessary). We recommend that the documentation files be downloaded and that you read the readme file. Technical assistance is also available from JavaSoft (see Appendix).

2. When the download is finished, double click on the icon named `jdk115.exe`. This is a self-extracting installation program.

3. Add `C:\JDK1.1.5\BIN` to your path in the `autoexec.bat` file. Run `autoexec.bat` to make the path take effect.

### Compiling and Running a Java Program

1. Use your favorite editor to create the files `LineDown.java` and `LineDown.html`. Be sure to save both programs in the same directory.

2. Open the DOS prompt. To compile, type "`javac LineDown.java`".

If successful, a `LineDown.class` file is created. It should be noted that when compiling a program, you may encounter problems if that program was created with an earlier version of JDK than the one used for compiling. From the viewpoint of later versions of JDK, the early versions use a deprecated Application Programmer Interface (API), which consists of the names of library calls and the number and types of arguments they take, they may not recognize some of these functions unless notified. This is one of the problems of using an evolving technology. In fact, many of the Java books we cited are preparing second editions because of the changes in API. To avoid encountering such problems when creating Java applets, we recommend using a Java compiler of the same version as the reference book you choose.

3. Double-click on the `LineDown.html`, and the page will load in a web browser and display the Java applet and any text included in the HTML file. Or, you can type "`appletviewer LineDown.html`" at the DOS prompt. The `appletviewer` is included with the JDK and is used for testing and running applets off the WWW. If viewing the applet on a web browser, it must be Java enabled (Netscape Navigator 3.x or better, or Microsoft Internet Explorer 3.0 or better) in order to see the applet.

## JAVA FOR INTRODUCTORY STATISTICS

Many concepts in statistics can be represented using Java applets. For example, an instructor can illustrate correlation with an applet showing that, as the dispersion of the data points increases, the correlation coefficient converges on zero. We demonstrate the versatility of Java applets with the examples of Monte Carlo estimation and simple linear regression. In classroom teaching, our discussion of linear regression follows the content covered in chapters 10–12 of Freedman, Pisani, Purves, and Adhikari (1993). We supplement our presentation of chap-

ter 13, "What are the Chances?" by the classic example of Monte Carlo estimation of  $\pi$  (Beckmann, 1977). It should be noted that each of the following applets was written using Java version 1.0.2, so users may find that later versions of JDK have problems compiling. These applets and their source codes are available on the WWW at <http://condor.depaul.edu/~nbriggs>.

#### Listing 1: MonteCarlo.java

This program demonstrates by animation the Monte Carlo estimation of the value  $\pi$ . Consider a unit circle inside a square. The area of a quarter of a unit circle (shown) is  $\pi/4$ . The area of a quarter of the square is 1. At each iteration, 100 random dots are drawn within the unit square, and the number of hits (dots falling within the arc) is counted. A very crude estimate of the value of  $\pi$  is given by four times the ratio of the number of hits and total number of dots. The estimation works by appealing to the law of large numbers (Feller, 1957).

We also implemented a set of programs to illustrate the basic concept behind the regression equation:

$$\text{Response} = \text{Slope} \times \text{Predictor} + \text{Intercept} \\ + \text{Chance Error.}$$

#### Listing 2: RegLine.java

This program contrasts the "true" linear relationship between the response and predictor variables with the estimated relationship in linear regression. The user supplies the slope and size of the error in an HTML page. The applet then uses this information to determine the relationship between the variables without error, as well as the relationship including the error. Due to the error being random, reloading the applet will generate different regression lines. This shows how a regression line depends on the size of the chance error (Freedman et al., 1993).

#### Listing 3: RnxReg.java

To demonstrate linear regression when X (predictor) is random, this applet generates the corresponding response by the above regression equation. The chance error is also assumed to be normal with a mean of zero. The resulting display shows a collection of regression lines fitted to the data generated. Unlike the previous example, the resulting picture shows the range of regression lines possible when considering both a random predictor and a random error. The user can change the correlation coefficient in the HTML file to see how this affects the confidence bands. Also, this applet can easily be modified to show the effect of a fixed predictor and a random error on the regression line. For example, replacing the line "xDat[i] = rN.nextGaussian();" with "xDat[i] = (double)I;" will fix X (predictor).

The pedagogical aim of these Java programs is to strengthen the students' understanding of the variability of response measures as well as that of the regression parameters and their dependence on the size of random error. These programs demonstrate how the true relationship is

"masked" by noise and how statistical procedures use the observed relationship to estimate the true relationship.

## DISCUSSION

As educators, we know the value of incorporating new and interesting instructional techniques into our classrooms. Java is excellent for this purpose; it provides a great opportunity to include new technology in the teaching process, and it can illustrate many concepts in ways that traditional teaching methods cannot. No longer does the instructor have to rely solely on the chalkboard to illustrate statistical principles; programming in Java enables educators to more effectively reach students. With a customized applet, instructors can show students graphical animations of the concepts they are discussing. Because it is accessible to anyone on the Internet, Java offers a distinct advantage for use in the classroom over other programming languages.

Furthermore, using Java applets in the classroom is economical. Instructors need spend nothing but time in creating or modifying useful programs for their classes, and it is not necessary to install the language itself onto each native machine in the classroom. Indeed, Internet-based educational activities can be a more efficient use of institutional resources than current methods (Bailey & Cotlar, 1994).

Java also allows for a more interactive approach to learning. Applets can be modified so that the students themselves can influence the final result. As in the applet RegLine.java, information included in the HTML file can be used by the applet to show how changing certain parameters also changes statistical results. This type of interactive learning not only helps the students to understand the concepts but also increases the students' interest and enjoyment (Carlson & Falk, 1990; Petty & Rosen, 1987). Presenting class material in Java also gives students experience in using Java and HTML documents. Using the Internet in the classroom enhances presentation of material, and exposure to the technology itself can only be beneficial to the students.

## REFERENCES

- BAILEY, E. K., & COTLAR, M. (1994). Teaching via the Internet. *Communication Education*, 43, 184-193.
- BECKMANN, P. (1977). *A history of  $\pi$*  (4th ed.). Boulder, CO: Golem.
- CARLSON, H. L., & FALK, D. R. (1990). Interactive learning models using videodiscs in college and inservice instruction. *Computers in Human Services*, 7, 277-293.
- DEITEL, H. M., & DEITEL, P. J. (1997). *Java: How to program*. Upper Saddle River, NJ: Prentice-Hall.
- FELLER, W. (1957). *An introduction to probability theory and its applications* (Vol. 1). New York: Wiley.
- FLANAGAN, D. (1997). *Java in a nutshell* (2nd ed.). Sebastopol, CA: O'Reilly.
- FREEDMAN, D., PISANI, R., PURVES, R., & ADHIKARI, A. (1993). *Statistics* (3rd ed.). New York: W. W. Norton.
- HOFF, A. V., SHAIQ, S., & STARBUCK, O. (1996). *Hooked on Java*. Reading, MA: Addison-Wesley.
- HORTON, I. (1997). *Beginning Java*. Chicago, IL: Wrox.

LINDEN, P. V. D. (1997). *Just Java* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

NIEMEYER, P., & PECK, J. (1996). *Exploring Java*. Sebastopol, CA: O'Reilly.

PETTY, L. C., & ROSEN, E. F. (1987). Computer-based interactive video systems. *Behavior Research Methods, Instruments, & Computers*, **19**, 160-166.

PEW, J. A. (1997). *Instant Java* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

ROSEN, E. F. & PETTY, L. C. (1997). Using Internet resources in a research methods course. *Behavior Research Methods, Instruments, & Computers*, **29**, 222-227.

WINSTON, P. H., & NARASIMHAN, S. (1996). *On to Java*. Reading, MA: Addison-Wesley.

**APPENDIX**  
**Java and Educational Resources on the World Wide Web**

Name	Address	Description
Bruno Herreros	suzy.unl.edu/bruno/java/linreg.html	Linear regression applet
Cognitive Psychology Lab at Purdue Univeristy	www.psy.purdue.edu/~coglab	Collection of applets demon- strating cognitive principles
Eric Krider	javaboutique.internet.com/BallDrop	Normal approximation for binomial distribution
Gamelan	www.developer.com/directories/pages /dir.java.html	Extensive collection of Java applets
JavaSoft	www.javasoft.com/products/jdk/index.html	Sun Microsystem's Java website
JavaSoft Assistance	www.javasoft.com:80/feedback/faq /assistance.html	Technical assistance for JDK
Marty Hall at Johns Hopkins University	www.apl.jhu.edu/~hall/java/	Java programming resources
Psychology 240 Bulletin Board at DePaul	condor.depaul.edu/~csheu/ps240hw.html	Resources for introductory statistics
STATLETS	www.statlets.com/#Access	Collection of statistical analysis applets
University of Illinois, Urbana-Champaign	kahuna.cogsci.uiuc.edu/ipl/	Internet Psychology Laboratory
WebStat	www.stat.sc.edu/~west/webstat	Java-based statistics program

Note—The leading indications for hypertext transfer protocol (<http://>) for the addresses have been omitted. This list is also available on line at <http://condor.depaul.edu/~nbriggs/resources.html>.

(Manuscript received October 21, 1997;  
revision accepted for publication January 9, 1998.)