LEARNING FROM THE STATISTICIAN'S LAB NOTEBOOK

Deborah Nolan and Duncan Temple Lang
University of California, United States of America
nolan@stat.berkeley.edu

An essential component of statistics education is to provide first-hand experience with applications of statistics where students learn how to analyze data in the context of addressing a scientific question. The approach we present brings the work of statistics researchers and data analysts to the community of educators so that they can utilize their expertise, data, problems, and solutions in teaching statistics. We hope to accomplish this sharing of ideas and materials through a new type of "document", an electronic lab notebook that captures the research process and acts as a database of the statistician's activities and analysis. These documents can be explored in rich new ways: they can have interactive controls that allow students to modify computations, they can be projected into different views for different audiences, and they can contain different branches of analysis for exploration.

INTRODUCTION

Most statisticians would agree that students need to know how to apply statistical methods, and that statistical experience gives students the skills needed to become collaborators in scientific endeavors (Cobb & Moore, 1997). Yet, many courses teach methodology—the theory or heuristics behind methods and their characteristics—and do not focus on teaching the skills and practice of approaching a scientific problem from a statistical perspective. The intuition and experience needed for data analysis are the hardest things to learn, and for instructors it can be the hardest thing to teach. As Wild (2007, p. 325) notes, "the biggest holes in our educational fabric, limiting the ability of graduates to apply statistics, occur where methodology meets context (i.e. the real world)."

Wild and Pfannkuch (1999, p. 224) noted that to teach statistical thinking instructors often simply "let them do projects." Although a valuable exercise, as a single, unguided encounter with statistical thinking in a real setting, it is far from adequate. Another approach is to leave the statistical experience until after they have learned the basic, traditional tools of statistics, such as probability, hypothesis testing, estimation, and inference. This experience might be in a capstone course for undergraduate majors. Again, this single exposure pales in comparison to opportunities appearing in multiple courses earlier in their studies. Furthermore, in our experience, case studies of data analyses often hide much of the thought process that is required. In a case study, an analysis is typically presented as a completed work, but the thought process that led to the conclusions and choice of approaches and methods is usually omitted. There is rarely mention of alternative approaches that were not fruitful or that were not pursued and the reasons why. Also not typically identified are the alternative approaches that led, or would lead, to almost equivalent solutions to the one presented in the final analysis.

While there is much variability in how statisticians operate, a statistician often approaches a consulting problem or scientific collaboration in ways that can be abstracted. We offer a list of the steps in a typical data analysis that captures the various aspects of statistical thinking involved in the process: decompose the problem and identify key components; abstract and formulate different strategies; connect the original problem to the statistical framework; choose and apply methods; reconcile the limitations of the solution; communicate findings. There are many nuances that we have omitted; and it is a subjective, informal process. Nolan and Temple Lang (2009) discuss each of these aspects in more detail. We point out here that it would be valuable for students to be exposed to this dynamic process. For example, identifying the key components in a problem typically involves a conversation between statistician and scientist that is hard to imitate in the classroom. Also, the high-level work of identifying potential statistical approaches and trying them out tends to involve multiple trials that students would benefit from seeing. Applications in courses are typically merely examples where students have to identify the inputs and plug them into the method. Such examples do not have the same rich, complicated context, extraneous information, and decision-making issues as real applications. Unfortunately, they focus on

In C. Reading (Ed.), *Data and context in statistics education: Towards an evidence-based society. Proceedings of the Eighth International Conference on Teaching Statistics (ICOTS8, July, 2010), Ljubljana, Slovenia*. Voorburg, The Netherlands: International Statistical Institute. www.stat.auckland.ac.nz/~iase/publications.php [© 2010 ISI/IASE]

methodology and ignore the many other dimensions needed to apply statistical ideas to a real problem. At other times, applications are derived from data collected from students and, again, typically lack a real question and context. Applications should involve uncovering the relevant information, understanding the needs of the problem, drawing conclusions and understanding their limitations, incorporating less quantitative considerations, refining the goal, and communicating the essential findings.

In the next section, we describe a project that aims plug to these holes in our educational fabric. The project provides a mechanism to enable the flow of materials from statistics researchers involved in scientific inquiry to the pedagogical community via an electronic lab notebook that captures the ideas and activities carried out by the statistician in his or her analysis and makes them available to the instructor and student.

THE ELECTRONIC LAB NOTEBOOK

We envisage a system for dynamic documents that acts much like an electronic laboratory notebook. The notebook captures the researchers' computations, analyses, notes, and thought process so that their findings can be reproduced for both themselves and others (e.g., peers, reviewers, editors, managers). In essence, the notebook is a database of all the activities within the data analysis (or simulation study); and it can be projected into different views to make the information it contains available for different audiences, e.g. a paper that describes the conclusions of the work for a journal, a technical report that provides more extensive details about the work, and an interactive document which reviewers could explore at different levels of detail, i.e., "drill down". These *database-documents* would provide resources to instructors to assist them in teaching in new ways because they would open up the thought process and experience behind a data analysis both to the instructor and the students. This technological approach would support a model for cooperation between statisticians active in research and consulting and the community of statistics educators. Instructors would then have libraries of real case studies that include data analysis projects and current research methodologies that show how statisticians think and work.

Importantly, instructors can take such a document and know that it has all the details involved in an analysis. They can annotate the material with links to explanations of the science and the statistical terms. They can annotate the computations (either programmatically or manually) to identify the inputs and outputs of the different subtasks. Such annotations can be used to display interactive controls for students who can then control various aspects of the computations—set nuisance parameters to different values, remove subsets of the data, introduce alternative datasets, create new plots, or introduce entirely different ways of analyzing data. The document provides a semi-guided exploration of the details in a data analysis that can go on to delve deeper and eventually go to free-form analysis. Furthermore, these documents can be programmatically manipulated in very rich ways where instructors don't necessarily need to work at the level of the "raw" document (described in the Infrastructure section).

With this approach, students experience the thought process of the *masters* in context, seeing their choices, approaches, and explorations. We avoid simplifying the scientific–data problems, and instead simplify how students see these details initially, allowing them to gradually open up the document to its full extent. As Wild (2007) noted, these documents give instructors a mechanism to:

- Control complexity and other environmental settings
- Provide multiple pathways to explore
- Focus attention on what is new and accelerate through what has already been mastered
- Allow students to efficiently "unlock the stories in data"
- Encourage students to "just try things out".

Example

As an example, we consider an article in *Statistical Science* on traffic analysis (Bickel et al. 2007). A schematic of a few of the tasks involved in the analysis is displayed in Figure 1, where the schematic shows the document broken into tasks which can be labeled and threaded together for creating alternative views. The tasks that have double borders have been selected for viewing, and those that are shaded belong to the same thread of “nonparametric modeling”.

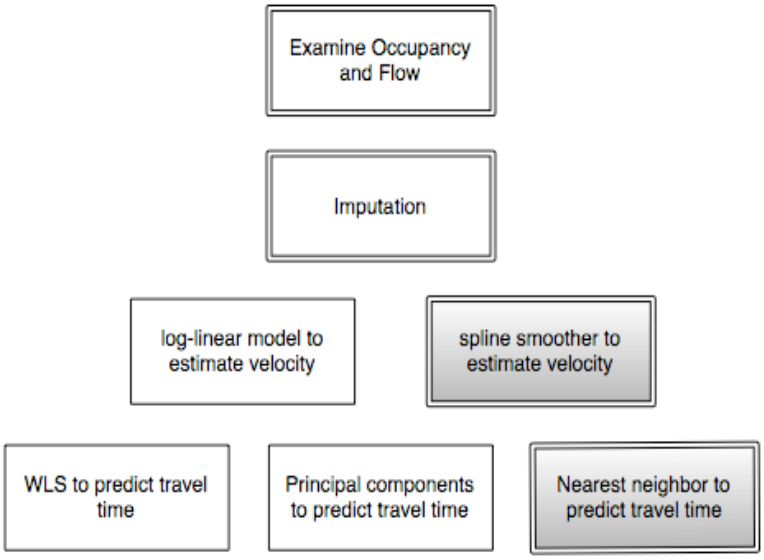
Examine Occupancy and Flow
Imputation
log-linear model to estimate velocity
spline smoother to estimate velocity
WLS to predict travel time
Principal components to predict travel time
Nearest neighbor to predict travel time

Figure 1. Schematic of an electronic notebook

Notice in the figure that the document includes multiple approaches to a task in the analysis, e.g. travel time is predicted using weighted least squares, principal components, and nearest neighbor methods. An instructor might create a projection of the document with a focus on nonparametric approaches to the analysis, and thus select for viewing the spline smoother for estimating velocity and the nearest neighbor method for predicting travel time, or she might leave it to the students to dynamically move between the parametric and nonparametric approaches to the tasks. Further, an instructor could add to the document a tutorial on nearest neighbor methods, additional plots to examine occupancy and flow, or select a subset of the data for in-depth investigation. The instructor’s additions can include user interface controls for students who can then view the notebook in a Web browser such as Firefox and interactively control and modify various aspects of the computations to try out alternatives. See Nolan and Temple Lang (2007) for a description of interactive possibilities for these documents.

INFRASTRUCTURE

The infrastructure for these dynamic documents is a combination of a markup language (XML), a transformation language (XSL) and a computational engine used to evaluate code in the document, in our case the R computational environment (R, 2006). The document is a collection of text, code, output, and other metadata and is written in the extensible Markup Language—XML syntax is similar to HTML, having elements or nodes of a hierarchical form. XML is a natural choice as the model relies on being able to *mark up* the different elements to provide structured documents. XML is very widely used in modern software and allows us to easily connect the notebook with new Web formats and modern publishing tools. Their structured nature also allows us to synchronize and update the documents and the software they reference, verify code, check table and figure captions, dynamically construct content, spell check diagrams, and so on.

We build on the XML-based vocabulary Docbook (Walsh & Muellner, 1999) for writing structured text documents such as books, articles and software documentation. Docbook has extensive documentation, including two online books that cover all major aspects of its use. With only about 15 Docbook elements and an understanding of the basic structure of an XML document, and author can create a Docbook document. The author might start by creating an article that looks something like the following (adapted from Bickel et al., 2007).

```
<article xmlns:r="http://www.r-project.org">
<title>Measuring Traffic</title>
<section><title>Introduction</title>
<para>
As vehicular traffic congestion has increased, especially in urban areas, so have
efforts at data collection, analysis and modeling. This paper discusses the
statistical aspects of a particular effort, the freeway Performance Measurement
System (PeMs)...
</para>
</section>
</article>
```

Hopefully the meaning of the XML elements such as <section>, <title> and <para> (paragraph) are self-explanatory. The key thing to note is that the document must be legitimate, well-formed XML. All elements are of the form <name>...</name>, i.e. with opening and closing named tags, and properly nested. Elements can have child elements, e.g. <article> has <title> and <section>, and <section> has <title> and <para>. The resulting document is a hierarchical tree structure. Other Docbook elements used frequently include <ulink> for hyperlinks, <emphasis>, <table>, <figure>, <xref> for cross-references within and between documents, <itemizedlist> and <listitem>.

XML permits extending a vocabulary and we have added new elements to the Docbook vocabulary to introduce new concepts. These are <r:code> , <r:plot>, <r:function> and <r:expr> which are used to represent R commands/code with different types of output. We would use these something like:

```
<para>The figure below illustrates detector failure. It shows scatter plots of
occupancy readings in <r:expr>length(levels(lanes))</r:expr> lanes at a particular
location.
<r:plot width="12cm">
 xyplot(Flow ~ Occupancy | lane, rtraffic)
</r:plot>...
<r:altApproaches>
<r:altApproach thread = "LinearModel">
<para> We have observed an empirical fact: that there exists linear relationships
between...</para>
</r:altApproach>
<r:altApproach thread = "Nonparametric">
<para> As an alternative, we now consider using information based on nearest
neighbors. </para>...
```

The author can display output from R using the <r:output> element, either nested within <r:code> elements or immediately following it. Note that we have used the prefix r: for all of the R elements. This is a name space in XML to avoid conflicts with other vocabularies that we might want to mix in the same document. The name space is declared via the xmlns:r="http://www.r-project.org" content in the <article> element, with the prefix “r” being the author’s choice.

The Docbook markup is relatively simple and one learns new “words” as one needs them. While XML is generally more verbose than LaTeX and other languages, there is a close correspondence between the Docbook and LaTeX vocabularies. What XML and Docbook give us over LaTeX is an array of technologies that provide much more flexibility in constructing documents and a rich set of tools for processing them in many different, programmatic manners which significantly improve the entire document production process. The extensibility of XML allows us to introduce new markup and customize and extend the tools, which is perhaps the most important aspect for developing new paradigms for reproducible research documents.

Transforming the R-Docbook document

Once the author has created an R-Docbook document, he or she will want to project it into a form that can contain the results of the embedded code and can be given to readers. Because the document is XML, it can readily be converted into any form the author wants. The author can extract all the code segments, or just those in a particular section. The author can remove entire sections or discard the code, leaving only the text. Typically, the author wants to create either an HTML or PDF version of the document. The Docbook software contains XSL libraries for transforming regular Docbook documents to either HTML or another XML format—Formatting Objects (FO) (Pawson, 2002)—used for describing high quality printed material, similar in concept to LaTeX. FO content can then be transformed directly to PDF using fop (*www.apache.org/fop/)*.

The approach we use to provide the dynamic aspect of these documents integrates R and the XSL engine that performs the XSL transformations for Docbook (Nolan, Peng & Temple Lang, 2009). By integrating R and the XSL engine, we have the ability to call R functions from XSL templates. We can pass XML nodes from XSL template actions to R functions in order to generate content for the output document. We use this to implement the XSL templates for `<r:code>` and other XML elements. We pass the node to an R function that extracts the code, evaluates it and then converts the result(s) to the target format.

The software we describe is available in the R package XDynDocs with support from several additional packages (XML, Sxslt). These packages are available on the Omegahat website (*www.omegahat.org*) and are distributed under the Berkeley Software Distribution (BSD) license. They have been designed with extensibility and customization by others as a primary goal. As a result, they offer a platform for us and others to experiment with richer forms of dynamic documents and reproducible computational-based research techniques. They also transfer to other programming environments, e.g., MATLAB or Python, very naturally.

CONCLUSION

Teaching computing and statistical thinking is very challenging. The approach we have outlined attempts to make it easier for individual instructors to introduce and teach this material within the statistics curriculum. To aid the teaching of statistical reasoning and experience, we aim to unite the research community and instructors by providing a flow of real-world data analyses from the former to the latter. This is done by providing an infrastructure for reproducible results for the researcher that enables the capture of computational details and thought process in an electronic lab notebook that acts as a project database. While this is beneficial to individual researchers and their community of fellow researchers and reviewers, it is also useful to course instructors. These documents allow students to enter the world of the researcher and to engage in the research process.

REFERENCES

Bickel, P.J., Chen, C., Kwon, J., Rice, J., van Zwet, E., & Varaiya, P. (2007). Measuring Traffic. *Statistical Science, 22*(4), 581-597.

Cobb, G. W., & Moore, D. S. (1997). Mathematics, statistics, and teaching. *The American Mathematical Monthly, 104*, 801-823.

Nolan, D., & Temple Lang, D. (2007). Dynamic, interactive documents for teaching statistical practice. *International Statistical Review, 75*(3), 295-321.

Nolan, D., Peng, R., & Temple Lang, D. (2009). Enhanced Dynamic Documents for Reproducible Research

Nolan, D., & Temple Lang, D. (2009). Approaches to Broadening the Statistics Curricula. In M.C. Shelley, L.D. Yore & Hand, B. B. (Eds.), *Quality Research in Literacy and Science Education.*

R Development Core Team. (2006). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.

Wild, C. J. (2007). Virtual environments and the acceleration of experiential learning. *International Statistical Review, 75*(3), 322-335.

Wild, C. J., & Pfannkuch, M. (1999). Statistical thinking in empirical enquiry. *International Statistical Review, 37*(3), 223-248.

Walsh, N., & Muellner, L. (1999). *DocBook: The Definitive Guide.* O'Reilly.