Computing, technology & Data Analysis in the Graduate Curriculum

Duncan Temple Lang
UC Davis
Dept. of Statistics

- Statistics is much broader than we represent in our educational programs
  - Context of Scientific Discovery, not statistical methods!
  - Data analysis and "problems"
- Technology has significantly altered science And hence statistics. We must respond!
- Computing & technology are essential elements of our practice, research and education

- Extend stat. curricula with
  - computing & technology
- Mix with introduction to modern statistical methods and "real" applications of data analysis
- This combination makes our students into more valuable contributors to scientific inquiry.

Calls for Change in view of statistics

1962 Annals

1977

Analysis of Large Complex Data

http://www.stat.fi/isi99/proceedings/arkisto/varasto/frie0060.pdf

1997

Resistance

- Nay-sayers often prioritize stat. topics over computing
- defend the "mathematical" foundations of our discipline based on conservatism,
- Frequently people who don't understand technology and computation and its role in practice and in reshaping opportunities for statistical thinking and research.
- But <u>not</u> competition between computing <u>or</u> mathematics

  both are tools for statistical concepts & practice.

- After 46, 31, 11 years, it is time for action & change, not just talk.
- We must do the best we can and strive to get computing, technology and data analysis adequately into our curricula.
- Need to attract/retain researchers with modern, different perspective on data analysis & its impact.

Outline

- Why?
- What?
- For whom?
- By whom?
- How to achieve this?

- Need our students to be computationally literate to be able to
  - Do computations for their own research (simulations, implement methodology)
  - to help with our research
  - interact with scientists using complex data from diverse sources
  - disseminate new statistical methods as software
  - understand, critically appreciate and exploit new technologies as they emerge to allow us to do new types of data analysis.

- Opportunity to teach statistical methodology that the students wouldn't necessarily see in a heuristic manner
- Improve their (exploratory) data analysis skills and intuition.
- Expose them to research by implementing computations within a paper.

- Omitting computing & technology from our curricula means we are "playing with one hand tied behind our back"
- We cannot provide complete solutions to scientific problems, but merely prescriptions (not functioning/tangible tools) for how others can approach these problems.
- Software for "doing" statistics in the analytic pipeline

What - Broad Topics

- Fundamentals of scientific programming - ....
- Computing for Research - profiling, C, basic parallel computing, object-oriented computing, "R packages"
- Computational Statistics - Lin. Alg, Numeric optimization, RNG, MCMC, EM, resampling, numerical integration,
- Data Technologies - Databases (SQL), Regular Expressions, XML, Web services.
- Visualization technologies - graphical techniques & software; dynamic, interactive & Web displays

Intro to Stat. Computing

- Operating system concepts - commonalities & differences
  file system (files, folders, binary/text), editors, ...
- Types of languages - compiled/interpreted, vectorized/scalar, task-specific languages, Perl, Python, R, MATLAB, SAS
- Language elements - data types, subsetting, function calls, vectorized looping (apply()), control flow
- Input and Output (I/O)
- Writing functions - mechanics, design,
- *Debugging - tools, technique and philosophy/art,
- Efficiency - alg. complexity, idioms, profiling, interface to C/FORTRAN/...
- Batch computing & remote "shells"

- Vital to avoid teaching just the syntax of a particular language, or how to cut-and-paste & modify templates
- Need to teach concepts of computing, how to understand other languages, approach a computational task & abstract the ideas.

For whom?

- Different types of students - different courses
- Each student should take ≥ 2 computing classes
- Required class - "Scientific" Programming
  - teach how to think & reason about computing and express stat. tasks as computations.
  - ideally also cover R/MATLAB fundamentals, interface to C, efficiency, parallel computing (in context of data analysis).
- And one class in either Data Technologies, Computational Statistics, Advanced Computing.

Types of students & second course

- Student studying methodology research (theory)
  - simulation, software development (e.g. R packages), efficiency, algorithmic complexity, numerical algorithms, parallel computing, streaming data, visualization
- Probability - simulation, RNG, efficiency, visualization.
- Applications - data technologies for accessing data, additional languages, efficiency, parallel programming, visualization.

- For me, programming and the basics of data technologies
  - I/O for complex data
  - text manipulation & regular expressions
  - databases
  - XML
- are vital for all students working with data.

Masters Students

- What do they end up doing?
  - Data manipulation and processing - data technologies
  - Exploratory Data Analysis & Reports - visualization
  - Simulations - programming
  - Modeling - R, MATLAB, SAS.
- First class in stat. computing & then mix of visualization, data technologies, SAS
- Data, data, data....

Can we weave topics into existing classes?

- Not the programming class!
  Starting from nothing
- We need programming to be a fundamental class to
  - emphasize its importance & establish culture of computing.
  - provide solid, rich foundation for other topics,
  - allow the students to absorb the concepts/reasoning over a quarter/semester,
  - put in the context of data analysis/math. stat.

By whom?

- Rarely in our graduate programs
- More senior faculty haven't been exposed to this, so can't teach it, so students aren't exposed to it, so ...
- Students left to learn it on their own with little encouragement or priority
  - empirically results are poor with major misconceptions
- So very few instructors who can teach computing and technology

Computer Science Classes

- Can we send our students to Computer Science classes on programming? databases? text manipulation?
- to Applied Math for numerical analysis? optimization? ...

- No
  - we do a different type of programming (vectorized, interpreted languages versus compiled scalar languages)
  - a class in databases teaches internal details of database not how to use it.
  - importantly, don't put these methods in the context of statistical data analysis.

How?

- Have to train instructors or train themselves?
- NSF grant (Nolan, Hansen, Temple Lang) to
  - develop potential syllabi & topics for computing
  - create resources for teaching - lecture notes, exercises/homeworks/projects/case-studies, text book
  - teach instructors how to teach computing
  - evangelize computing, technologies & data analysis within the community via papers, talks, etc.

- May 2007 - Workshop for syllabi
- July 2008 - Workshop for teaching instructors
- 2009 - final workshop. What form?
  - teach additional instructors (same as 2008)?
  - summer school on technology, computing & data analysis for recent graduates, like New Researchers Conf.?
  - summer school for PhD students starting research?
  - small working group to complete materials for others to pick up?

- Materials at
  - Workshop 1
    http://www.stat.berkeley.edu/twiki/Workshop/CompCurric
  - Workshop 2
    http://www.stat.berkeley.edu/~statcur/

Summary

- It is time to step up and do something about computing & technology & data analysis in our curricula.
- Must have dedicated statistical/scientific programming course
- Data technologies, advanced/research computing, numerical algorithms, visualization classes or individual topics

Summary ctd.

- Grow pool of potential instructors by teaching these classes now
- and teaching existing instructors via workshops & developing class materials
- What form for next workshop?
  Seek funding for additional workshops?
- Strategic Initiative from ASA, ISI, ...

Actions

- Time for action on technolog & computing.
- Departments should introduce computing into graduate & undergraduate classes.
  - Explicit "computing" classes
  - Introductory programming, data technologies for scientific computing with data
  - Second class or integrate topics into classes from: Data technologies, advanced/research computing, numerical algorithms, visualization classes or individual topics