Some Issues in the Design of Statistical Languages

Luke Tierney
Department of Statistics & Actuarial Science
University of Iowa
Introduction

- Two basic designs for statistical systems:
  - Menu/dialog based
  - Language-based

  There are intermediate positions (e.g. R’s Windows GUI)

- The designs are geared towards different target audience roles:
  - data owners
  - data analysts

  One individual may play different roles at different times.
Language-Based Data Analysis Systems

● Most high-end systems have a language component:
  – S language family (R, S-PLUS)
  – SAS

● A language-based design facilitates
  – arbitrary data transformation
  – applying one analysis to results from another
  – abstraction of repeated complex analysis steps
  – development of new methodology

● Language based systems have a steeper learning curve
Language-Based Systems (cont.)

• Many language-based systems started with modest goals:
  – relatively simple analyses
  – relatively small data sets
  – relatively simple data sets

• Original languages were designed to manage FORTRAN/C code

• Many languages have become more sophisticated

• Very sophisticated programs are written in these languages
Challenges for Statistical Languages

• managing long-running computations
• large data sets
• distributed data and meta data
• insuring correctness and efficiency
• taking advantage of hardware (e.g. dual processor, 3D)
• communicating analyses to data owners
• easing the transition to data analyst
Managing Long-Running Computations

• Many procedures lead to long running computations

• A statistical environment needs to allow
  – other concurrent activity
  – monitoring of progress
  – intervention or termination

• Within a language this requires some form of thread support
Managing Long-Running Computations (cont.)

• Parallel computation may reduce computation time
  – shared memory parallel linear algebra (e.g. threaded BLAS)
  – distributed memory libraries (e.g. SCALAPACK)
  – message-passing parallel computing (e.g. based on PVM or MPI)

• Dual processor workstations are becoming more common

• Networks of workstations are very common
Large Data Set Issues

• Data too large to fit easily in memory may require special methods

• Streaming date definitely requires special methods

• Data base integration can be very useful

• More work is needed on primitives for algorithms for large data

• $2^{32} = 4$Gb—temporary speed bump or longer term issue?
Distributed Data and Meta-Data

• Distributed data is becoming important for many analyses
  – data stored at acquisition sites/instruments
  – meta-data on genes stored in on-line repositories

• Tools to access and use these data need to be integrated

• Support for handling errors and managing recovery strategies are essential

• Related issue: Persistent, often distributed, storage of
  – computed results results
  – code
Support for Programming and Extension

- Assessing and insuring correctness:
  - Well-designed language
  - Infrastructure for testing
  - Code analysis tools
  - Debugging tools

- Module system
  - Allows separate development of extension code
  - Protects against clashes
Support for Programming (cont.)

• Aids in obtaining good performance:
  – predictable performance for basic operations
  – performance model for compound operations
  – profiling tools for monitoring performance
  – compilation to simpler interpreted or compiled language
  – gradual specialization to important cases

• High level language features can interact badly with performance
  – immutable vectors in R
Communicating Analyses to Data Owners

• Results provided to a collaborator may take many forms:
  – static report (with numerical and graphical results)
  – dynamic report (e.g. including interactive graphics)
  – menu/dialog-based analysis framework for re-use

• Tools that can help:
  – Literate data analysis frameworks (e.g. Sweave)
  – Programmable interactive and 3D graphics
  – Programmable user interface

• Need to be well integrated with the statistical language
Easing the Transition to Data Analyst

- Well-designed high-level language
- Familiar language or language features?
- Good and/or familiar Object-Oriented programming framework?
- Programming by direct manipulation
- Programming by example
- Graphical language
- Can the language and menu/dialog designs coexist smoothly?
Conclusions

• This talk has outlined some issues and goals for language-based statistical systems.

• Many of the goals can (and some have) been achieved by adapting and evolving existing languages.

• Can all important goals be achieved from existing languages?

• Or is it time to start anew
  – building on what has been learned
  – but not trapped by the need for compatibility