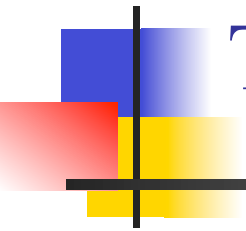


Text Processing using Perl

Why learn programming?

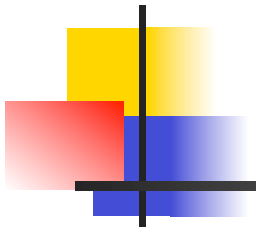
- It is at the guts of all of the programs you will be using anyway, so it helps you figure them out.
- It gives you vastly more flexibility than you would otherwise have, particularly dealing with text. Things can be done very easily with a program that are difficult with a search-and-replace or statistical transformations
- 10-year-olds program; and 16-year-olds can cover the basics in about 10 weeks (albeit in BASIC or Pascal)
  - 20-year-old hackers in developing countries can write and deploy viruses for the Windows OS that cause billions of dollars of damage across the planet in a few hours!
- It is easy to learn, though to get it down well, you need to practice, practice, practice.

Why learn programming?, continued

- Moore's Law—computer capacity doubles every 18 months. You *don't* want to use this??
  - *Economist*'s law—every discussion of computing must start by mentioning Moore's Law
- Otherwise you are at the mercy of computer programmers
  - See also: plumbing, automobile repair, landscaping, remodeling

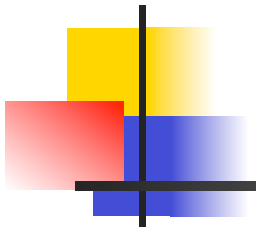
The wrong reasons to learn programming (despite what you have heard)

- Instant access to fantastic jobs earning zillion-dollar salaries
  - See *Micro-smurfs*
  - See NASDAQ technology index, 1998-present
  - If you don't enjoy it, you don't want to do it for a living
  - Academic salaries are quite competitive
- Only opportunity to meet, and possibly mate with, other individuals with severe personality disorders and zero social skills
  - Only at M.I.T...



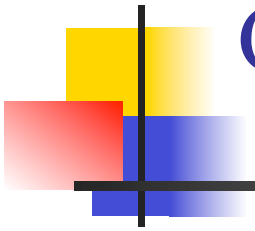
Advantages of Perl

- Most of the control structures and syntax of Perl are the same as in Python, C++ and Java.
- Perl does not require any of the headers and variable declarations used in C and Java.
- Perl contains a large number of additional string-oriented functions and data structures not available in C.
- The pattern matching and substitution options are incredibly rich: regular expressions
- Perl transparently interfaces with the operating system — in other words, a Perl program can easily move, delete or rename files, fetch web pages, and the like.



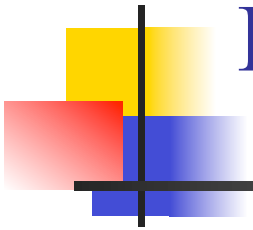
Advantages of Perl, continued

- Perl is open-source and freely available for Unix, Linux, Windows, and Macintosh. It runs as part of the operating system on many Unix machines, in Linux, and in the Macintosh OS X operating system.
- There is extensive documentation and source code available on the Web.
- "Perl is the glue that holds the web together"—much of what you download from the web will have been generated from Perl and is therefore easily processed with Perl



Caveat:

Perl comes out of the Unix community and a lot of the most powerful features of the language are based on Unix models, which will seem obscure until you become familiar with them. But once you've learned the "regular expression" syntax for Perl, you can also use it in Unix.



Disadvantages of Perl

- Perl is an interpreted language, rather than a compiled language, so it is probably too slow for writing large programs. The speed seems fine on both Unix and the Mac, however—a simple program for count event types in a WEIS file runs through a 30,000 line data file in less than a second on a Mac G3.
- This is a text-processing language, not a general purpose language.

Methodology for Dummies™

Kids, use perl programs to select information from your Stata log files and put it into tab-delimited format to create charts and tables!

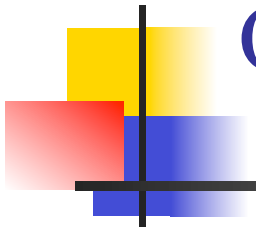
Example:

```
# extract z-scores for 'mediatn' variable
open(FIN,"stata1.log"); open(FOUT,">extract.output");
while (chop($line = <FIN>)) {
     if ($line =~ m/conflict\./) { # get data set ID
          $aout[0] .= "\t" . $';
          $kset = 0;    }
     elsif ($line =~ m/mediatn(\s)+\|/){ # get z-score
          $aout[++$kset] .= "\t" . substr($line,36,7);}
}
for ($ka = 0; $ka<=$kset; ++$ka) {
     print FOUT $aout[$ka], "\n"; }
close(FIN); close(FOUT);
```

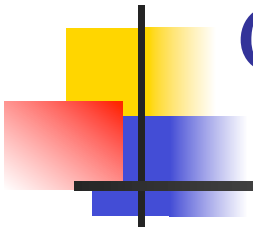
A Perl program for downloading a known set of URLS

```
open(FIN,”my.file.of.URLs”);
open(FOUT,">my.file.of.HTML.txt");
while ($theURL = <FIN>) {
   chomp($theURL);
   $theHTML = get($theURL );
   print FOUT “\n\n$theHTML”;
}
close(FIN);
close(FOUT);
```

Other languages to consider

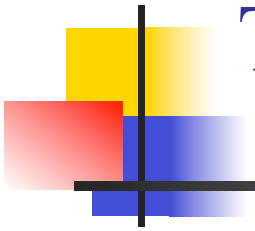


- Python: most of the capabilities of perl, but written later and generally considered more consistent, less quirky, and devoid of perl's "attitude." Web-based documentation is almost as thorough. I've heard a number of instructors recommend this over perl for beginners
- Java: this has become the standard language for undergraduate computer science instruction. It is a general-purpose language but has a rich set of string-processing functions, and is operating-system independent.



Caution:

- Don't assume that you will be able to download from a site: it may use internal scripts or other methods that get in the way. Experiment first.
- However, *most* sites can be downloaded. In particular, any site that can be indexed by Google can be downloaded using automated methods (since that is how Google works). This provides an incentive for sites that want traffic to be Perl-friendly



Text Filtering

- This is an essential step in any original automated analysis. The text that you download *will not be in a format that you can immediately analyze!*
- Filters are used to regularize the text for later processing. Perl is ideal for this task.

What a Text Filter Needs to Do

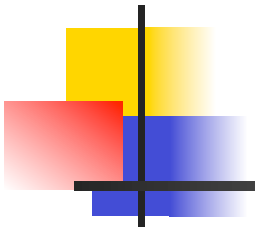
- Remove the HTML tags and other web-specific coding
- Locate the beginning and end of the document text
- Segment article into sentences
  - Problems: Periods in abbreviations
    Abbreviations at the end of sentence
- Identify quotations for separate treatment:
  - Problems: Short quoted phrases in mid-sentence,
    …Bill "Mad Dog" Jones…
    Use of double-apostrophes rather than quotation marks
- Eliminate duplicate stories—comparison of character counts seems to work for this
- Ignore everything in the file not required for the above tasks

Text File Formats

- ASCII (“text”)—this is usually what you want.
- MS-Word (or other word processing)—nearly impossible to process; convert to “text”
- HTML—downloaded from the web; this is ACSII plus tags
- RTF—”rich text format”; also ASCII with tags
- PDF—portable document format (Adobe); see “MS-Word”, though it can be converted to text fairly easily
- JPEG and other graphics formats: These are scanned images of the document and cannot be coded directly
  - OCR might work on some of these, but it is tedious

Operating System Differences

- How is a line ended?
  - Macintosh—ASCII 10 (\n)
  - Unix—ASCII 13 (\r)
  - Windows ASCII 10 + ASCII 13
- Special characters (e.g. diacriticals å, ü)—there are a wide variety of "standards";
- "Unicode"—successor to ASCII; incorporates character sets of all widely-used languages (e.g. Russian, Arabic, Hebrew, Hindu, Chinese, Korean, Japanese), though there are multiple versions of it



Filters available from Event Data Project: `http://eventdata.psu.edu`

- Reuters from NEXIS via modem (various formats; mostly in Pascal, some in C)
- Reuters Business Briefing, modem
- Dow Jones Interactive/Factiva, WWW screen-captures
- NEXIS Academic Universe WWW (Perl)