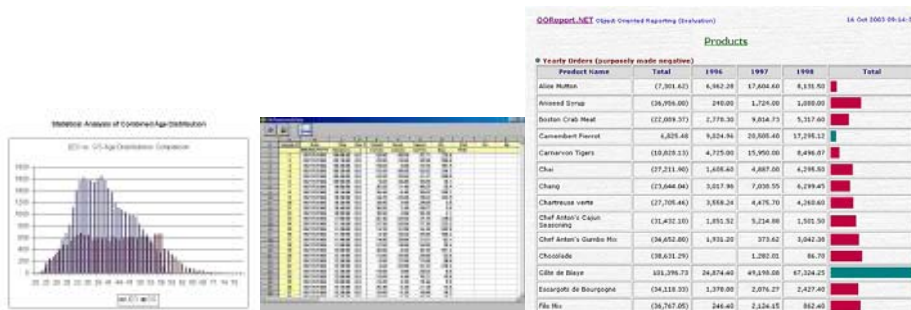


BECY

Tabular – data manipulation language



Columbia University
Fall 2005

Members:

Bong Koh
Eunchul Bae
Cesar Vichdo
Yongju Bang

bdk2109@columbia.edu
eb2263@columbia.edu
cv2139@columbia.edu
yb2149@columbia.edu

Overview

Often times, one encounters a data set that requires a variety of basic mathematical manipulations and functions while retaining the ability to output a display of the analysis in an aesthetically pleasing and useful manner. Larger and more powerful languages such as C/C++ or Java are usually not the best suited to handle tasks of a formatted data set such as tabular data while smaller languages lack the ability to graphically present the data and its analysis without further support.

Our goal is to provide a more specific language capable of providing both analytical power and presentation capability. For example, if one had a list of some students and their records and wanted to perform basic analysis on all the students such as final grade calculations or average grades across students, programming in a compiled language would probably not be appropriate. In order to facilitate the processing of such a formatted data set, Betsy would provide native support for importing a data file as a basic data structure, easily using the rows or columns of data and then outputting a more user-friendly presentation.

Language Description

The language we propose to implement is a scripting language to manipulate and present tabular data. The ultimate goal of such a language is to provide a simple and flexible scripting language tailored for numerical manipulations of lists of data.

Syntax

Its syntax will closely resemble a command line language as follows;

```
command parameter1 parameter2 ... ;
```

Conditional

Betsy will also contain support for conditionals in this manner;

```
Condition ? command1 : command2;
```

```
//If true, then command1, otherwise command2
```

Loop

Like any other languages, Betsy will provide 'Loop' as follows;
 (startrowTOendrow);

Comparative and arithmetic operators

== != > >= < <= + - * / && || ()

Mathematical functions

Most basic mathematical functions on groups of data will be supported by Betsy;
 average{}; sum{}; ...

Functions for display

print{}; table{}; ...

The language will have built-in "shortcut" variables for columns of data and also special syntax for easy row manipulation. Data files formatted in tab-separated values will have inherent access included within the language. Upon processing of a data file, a program in the language will generate a formatted HTML file for use with a typical browser.

Example Program:

```
# Input file: grades.txt
jean    10    2
peter   8     1
josh    9     3
amber   9     6
```

Structure of the program**1) Fill a list data structure from a data source**

```
list{
    jean 10 2;
    peter 8 1; // "list" contains the information to work with
```

```
josh 9 3;
amber 9 6 };
```

OR

```
list{include "grades.txt"}; // "include" associates a txt file into the list
```

2) Adding information to the list

```
list.push{bong 10 3; vivian 6 2};
```

OR

```
list.push(include file2);
```

3) Remove information from the list

```
list.remove(5;6);
```

OR

```
list.remove(5TO$1.SIZE);
```

4) Perform data processing

```
sum($1.1TO100);
```

```
average($2.1TO100);
```

5) Print

```
print($1.1TO4;" ";$2.2TO3;"...;$2.4);
```

// will displays as follows

```
jean 8...6
```

```
peter 9...
```

```
josh ...
```

```
amber ...
```

```
print($1.4-3) OR print($1.1);
```

// will output as follows;

```
jean
```