The Valet Score:

The Program

November 2015
Søren Hein
soren.hein@gmail.com

Table of Contents

1 Usage

The Valet program can be compiled as explained in the INSTALL file that accompanies the distribution. I'm not sure how many people will actually want to run the program, as this requires input files in a certain format. Perhaps the program is best viewed as the formal definition of the Valet score. I hope that authors of scoring programs will include the Valet score by porting it to their programs.

The key functions can also be used in a DLL / library form. Please refer to the DLL documentation for this.

Once Valet is compiled (or if you obtained a binary version), you invoke it according to the system you are on. For me it looks like this:

```
valet.exe –d ../data/2008-Euro/Men –v imps –l > result.txt
```

The program has a number of options. It reads from two input files, one containing player names and the other containing results. It writes to the terminal output. In the above line the output is redirected to a text file.

The options are as follows. They also appear if Valet is invoked with no arguments. The default argument of choice is shown in **bold**.

Short name	Long name	Meaning
-v	--valettype	Use this type of scoring. One of datum, **imps** and matchpoints
-d	--directory	Read the input files from this directory.
-n	--names	Use this names file (**names.txt**).
-s	--scores	Use this scores files (**scores.txt**).
-r	--rounds	Use only these rounds, even if there are more rounds present in the scores file. Example: 1,3-5,7 (with no spaces). Default: all rounds
-m	--minhands	Show the output tables separately for players who have played at least this number of hands, and the remaining players. Default: 0
-l	--lead	If present (without a value), calculate the Valet lead component and not just the overall defense component. Default: No separate lead
-e	--extremes	When calculating datum IMPs, throw out the minimum and maximum scores. Default: Use all scores
-h	--hardround	When calculating datum IMPs, round down and not to the nearest multiple of 10. So 379 rounds to 370, not to 380.
-a	-adjustopps	If present, adjust the overall Valet score according to the opponents' strength and then spread the adjustment onto the individual scoring compoments. Default: No adjustment
-o	--order	Sort the output tables in this order: **overall**, bidding, play, defense, lead, bidoverplay (bidding minus play), defoverplay (defense minus play), leadoverplay (lead minus play). Case-insensitive
-f	--format	Write the output in plain text format or as comma-separated values: **text** or csv
-c	--char	Use the character as the field separator for the csv format. Default: , (comma)

2 Input File Format

2.1 NAMES FILE

The names file consists of lines. Empty lines and lines starting with the symbol '#' are ignored.

Other lines must be of the form 'tag|Some string'.

The tag may be a number, a BBO handle such as shein, or something else. I use number starting with 1 in all my own files, and I haven't actually tested it for anything else ☺.

'Some string' is the full name associated with the tag. So a valid line would be

26|HEIN, Soren

2.2 SCORES FILE

The scores file also consists of lines. Empty lines and lines starting with the symbol '#' are ignored.

Other lines must be of the form 'a|b|c|…'. There must be exactly either 9 or 10 fields in each line, so 8 or 9 vertical bars. It is acceptable to mix such lines, so there can be lines of both types in a single file.

The meaning of the fields is as follows, in order:

- Round number. An integer, minimum value 1.
- Board number. An integer, minimum value 1. The number is used to derive the vulnerability according to normal bridge rules.
- The tag for the North player.
- The tag for the East player.
- The tag for the South player.
- The tag for the West player.
- The contract, which must be exactly one of the following: P, 1C, 1CX, 1CXX, 1D, 1DX, 1DXX, … 7N, 7NX, 7NXX.
- The declarer, which must be exactly one of N, E, S, W.
- The number of tricks taken, which must be 0, 1, …, 13.
- Optionally the opening lead, which must consist of a suit (S, H, D, C) followed by a card (2, 3, …, 9, T, J, Q, K, A).

3 Directory structure

The distribution contains the following directories:

- src, the source code together with Makefiles for various systems.
- doc, the documentation.
- data, the files for various tournaments.
- scripts, some scripts that I found useful during the collection and processing of data.

In addition to tournament data, the data directory contains the sub-directories “examples” and “results”. The first one contains the five examples that are covered in the Valet Principles document. The second one contains some summaries of the tournament data.

4 Code structure

The code consists of the following C++ files (in addition to header files):

- valet.cpp, the main function of the program.
- args.cpp, which parses command-line arguments and sets the global structure "options" accordingly.
- files.cpp, which reads the two input files.
- parse.cpp, which is used by files.cpp to parse the input files.
- misc.cpp, a few print functions that are mostly useful for debugging.
- scoring.cpp, functions for calculating bridge scores and IMPs.
- Pairs.cpp, a class that keeps track of the pairs that are playing.
- Hand.cpp, a class that keeps track of the results on a given hand.
- Scores.cpp, a class that aggregates data for all hands.

A lot of the code is concerned with shuffling data around. The real Valet logic is mainly in Hands.cpp and to small extent in Scores.cpp (for the adjustment of opponents' strength).

The code makes liberal use of C++ maps for simplicity. If you're going to port the code to a language or system without associative arrays, you'll have to change this for yourself. If you already have scoring program, you probably don't need a lot of the data-shuffling code anyway, and you already have your own data structures.

The index into the Pairs map is the concatenation of the two player tags (in lexigraphic order).

In Hand.cpp we have to make sure that we assign scores to the right individuals within the pair, so sometimes we have to swap the scores around.

There are functions for all the three kinds of scoring offered by Valet: Datum scoring (regular Butler), IMPs-across-the-field scoring, and matchpoints.