23rd Australasian Transport Research Forum
Perth, Western Australia 29 September – 1 October 1999

Auditing of Expanded Survey Data

Keith Long
Long Technical Pty Ltd, New South Wales

Abstract

Auditing of expanded survey data to ensure that the expansion process is computationally correct is only one of many audits that may be required to independently verify the survey process in any transport project. Array based software is a cost-effective platform to undertake the audit. It takes the unexpanded survey data and seeks to replicate the expanded survey data by converting the methodology in the working papers into sequential, efficient and self-documenting code. An independent and complete sign-off of the expanded survey data can be achieved.

Contact Author

Keith Long
Long Technical Pty Ltd
31 Pavilion Street
Queenscliff NSW 2096

Phone: +61 2 9939 6452 Fax: +61 2 9939 6452
e-mail: klong@zeta.org.au

Introduction

This is a narrowly focussed paper which discusses the auditing of expanded survey data It will be of interest to those, including project stakeholders, who have ever been handed some Excel spreadsheets, some Access databases and some working papers and then been asked to sign off the expanded survey data as correct!

The paper briefly discusses auditing of the complete survey process, ranging from audits of the survey design through to the sampling rates, the edit/logic checks and the expansion methodology. However, the focus of the paper is the final audit - whether the process to expand the survey data is computationally correct

This paper suggests that an array based software platform is a cost-effective way to undertake this final audit. It takes the unexpanded survey data and seeks to replicate the expanded survey data by converting the methodology in the working papers into sequential, efficient and self-documenting code. Array based software is particularly suited to the multi-dimensional arrays which are typical of transport projects

The first section describes the preparation of survey data for a typical project The second section discusses a number of areas where the survey process can be audited, but concentrates on whether the process to expand the survey data is computationally correct.

Finally, the third section demonstrates how an array based software platform would replicate the survey data obtained from typical postcard, roadside interview or telephone surveys.

Project survey data - coding, editing, expansion and processing

Consider a project which undertakes an extensive survey program to quantify existing travel in a corridor.

The questionnaires obtained from people in the course of their journeys (such as by car, air, coach and rail) in the corridor form the intercept survey data. Also included were telephone and registration number surveys which were undertaken for some car journeys where permission to undertake intercept surveys was denied.

In addition, various population count data were also collected (such as road vehicle and public transport passenger counts).

The raw questionnaires were put through a variety of editing, range and logic checks, and the data were either corrected or rejected For each questionnaire, a minimum acceptable information content was specified and failure to meet this requirement lead to rejection of the questionnaire.

Expansion factors were then attached to the edited questionnaires such that the expanded questionnaire data was an unbiased representation of the journeys in the corridor

The project survey data was developed using a widely available software platform comprising Access databases and Excel spreadsheets.

Auditing the project survey data

At some stage of the project, one or more of the project stakeholders (such as bankers or the government) will usually have the opportunity to independently verify that the expanded questionnaire data is indeed an unbiased representation of the journeys in the corridor. This process is sometimes referred to as auditing the expanded survey data.

A thorough audit might consider:

1 whether the surveys asked the right questions to the right people at the right time in the right places;

2 whether the sampling rates were appropriate;

3 whether the editing, range and logic checks were appropriate;

4 whether the methodology to expand the survey questionairres was appropriate; and

5 whether the process to expand the survey data was computationally correct

The first four audit checks are outside the scope of this paper. Assuming these checks were satisfactory, the remaining task is to ensure that the process to expand the survey data is computationally correct and this paper addresses this task.

An auditing approach which simply checks a sample of spreadsheet cells or database enquiries cannot fully sign off that the process to expand the survey data is computationally correct. However, an array based software platform can cost-effectively replicate the expansion process and therefore an independent and complete sign-off of the expanded survey data can be achieved

In general, an array based software platform:

- separates the input data;
- converts the methodology from the working papers into sequential code;
- manipulates and analyses multi-dimensional arrays; and
- is elegant, efficient and self-documenting.

The array based software accepts the unexpanded project survey data as the input data and the working papers provide the survey expansion methodology.

Expansion of postcard, roadside interview and telephone surveys using an array based software platform

This is a technical section which describes how the array software works using a simple example. The approach is similar for postcard, roadside interview and telephone surveys, however some of the steps discussed below are only required for postcard surveys.

Postcard surveys

In postcard surveys, the distribution details of the survey cards are known. In our example, the *direction*, the *day* and the *hour* that the survey cards were distributed is recorded, as shown in Table 1. For example, survey cards 10-13 inclusive were distributed in the *NB-Fri-8* period.

Table 1 Postcard survey - distribution details

HOUR (commence)	NB (NorthBound)		SB (Southbound)	
	Fri	Sat	Fri	Sat
7	1-9	22-29	101-108	137-148
8	10-13	37-40	109-120	149-150 500-502
9	14-21	41-43	121-136	153-158

The survey card numbers are not necessarily sequential and there may be more than one range of survey cards distributed in any period, such as *SB-Sat-8*. Although Table 1 is visually appealing, Table 2 is computationally preferable.

Table 2 Postcard survey - distribution details

Num	Dir	Day	Hour (com)	Range_lo (>=)	Range_hi (<=)
1	NB	Fri	7	1	9
2	NB	Fri	8	10	13
3	NB	Fri	9	14	21
4	NB	Sat	7	22	29
5	NB	Sat	8	37	40
6	NB	Sat	9	41	43
7	SB	Fri	7	101	108
8	SB	Fri	8	109	120
9	SB	Fri	9	121	136
10	SB	Sat	7	137	148
11	SB	Sat	8	149	150
12	SB	Sat	8	500	502
13	SB	Sat	9	153	158

Table 2 becomes an input data file. A simple instruction set attached to this file constructs five one-dimensional arrays (or *vectors*), each having a dimension of NUM. The first three vectors are *character* arrays and the last two vectors are *numeric* arrays.

Array DDIR		Array DDAY		Array HHOUR		Array RANGE LO		Array RANGE HI	
NUM	Data	NUM	Data	NUM	Data	NUM	Data	NUM	Data
1	NB	1	Fri	1	7	1	1	1	9
2	NB	2	Fri	2	8	2	10	2	13
3	NB	3	Fri	3	9	3	14	3	21
4	NB	4	Sat	4	7	4	22	4	29
5	NB	5	Sat	5	8	5	37	5	40
6	NB	6	Sat	6	9	6	41	6	43
7	SB	7	Fri	7	7	7	101	7	108
8	SB	8	Fri	8	8	8	109	8	120
9	SB	9	Fri	9	9	9	121	9	136
10	SB	10	Sat	10	7	10	137	10	148
11	SB	11	Sat	11	8	11	149	11	150
12	SB	12	Sat	12	8	12	500	12	502
13	SB	13	Sat	13	9	13	153	13	158

The data values in each array can be used to "point" to the tags of the dimensions DIR (direction), DAY (day) and HOUR (hour). These dimensions, along with the dimension NUM, are stored in an input dimension data file.

Dimension DIR		Dimension DAY		Dimension HOUR	
Index	Tag	Index	Tag	Index	Tag
1	NB	1	Fri	1	7
2	SB	2	Sat	2	8
				3	9

Control counts

Control counts, which represent all the relevant traffic at the site, are generally undertaken at the same time that the survey cards are distributed. These counts are used later to calculate expansion factors. Our example assumes control counts as shown in Table 3, which also becomes an input data file. This time, however, the instruction set which is attached to the file constructs a three-dimensional array called COUNT, which has dimensions DAY, DIR and HOUR.

Table 3 Postcard survey - control counts

HOUR	NB		SB	
(com)	Fri	Sat	Fri	Sat
7	80	60	50	36
8	80	40	27	51
9	50	45	19	27

Completed surveys

Only a percentage of the distributed postcard surveys are returned and these become the completed survey records. Survey respondents have answered a series of questions. For our example, we asked the purpose, origin postcode and destination postcode of their trip. This is the final input data file and is shown in Table 4 for two of the periods (*NB-Fri-7* and *SB-Sat-8*).

Roadside interview and telephone surveys will generally be able to directly append the direction, the day and the hour to each survey record.

Table 4 Postcard survey - completed surveys

Rec	Dir	Day	Hour (com)	Purp	Origin Pcode	Destn Pcode
2				Work	2450	2550
4				Work	2450	2500
7				Other	2350	2550
9				Work	2450	2550
...				...	...	...
149				Other	2350	2500
150				Other	2350	2500
501				Other	2350	2550
...				...	...	...

A simple instruction set attached to this file can construct four one-dimensional *character* arrays called RREC, PPURP, OOPCODE and DDPCODE, where each array has a dimension of REC.

Array RREC

REC	Data
2	2
4	4
7	7
9	9
...	...
149	149
150	150
501	501
...	...

Array PPURP

REC	Data
2	Work
4	Work
7	Other
9	Work
...	...
149	Other
150	Other
501	Other
...	...

Array OOPCODE

REC	Data
2	2450
4	2450
7	2350
9	2450
...	...
149	2350
150	2350
501	2350
...	...

Array DDPCODE

Data	REC
2	2550
4	2500
7	2550
9	2550
...	...
149	2500
150	2500
501	2500
...	...

Again, the data values in each array can be used to "point" to the tags of the dimensions REC (record number), PURP (purpose), OPCODE (origin postcode) and DPCODE (destination postcode).

Dimension REC

Index	Tag
1	2
2	4
3	7
4	9
...	...
40	149
41	150
42	501
...	...

Dimension PURP

Index	Tag
1	Work
2	Other

Dimension OPCODE

Index	Tag
1	2350
2	2450
3	2500
4	2550

Dimension DPCODE

Index	Tag
1	2350
2	2450
3	2500
4	2550

In our example, we bypass the intermediate arrays PPURP, OOPCODE and DDPCODE and simply define an instruction set which directly creates three one-dimensional *equivalence* arrays called Q_REC_PURP, Q_REC_OPCODE and Q_REC_DPCODE which describe the equivalence between the record number and each of the purpose, origin postcode and destination postcode. Our example does however need the *numeric* array RREC and this array is simply created from the dimension REC.

RREC is convert REC [Code]

RREC is char2num RREC [Code]

Appending direction-day-hour to survey records

For the postcard surveys, this step assigns a period (direction-day-hour) to each completed survey record using a *range* function. The result is saved in a one-dimensional *numeric* array called TT.

TT is range RREC,>=RANGE_LO,<=RANGE_HI [Code]

Array TT	
REC	Data
2	1
4	1
7	1
9	1
...	...
149	11
150	11
501	12
...	...

The array TT can now be used to separately *lookup* the direction, day and hour for each of the completed survey records. The results are stored in the temporary one-dimensional *character* arrays TT1, TT2 and TT3 respectively:

TT1 is lookup TT,NUM,DDIR [Code]

TT2 is lookup TT,NUM,DDAY [Code]

TT3 is lookup TT,NUM,HHOUR [Code]

Array TT1	
REC	Data
2	NB
4	NB
7	NB
9	NB
...	...
149	SB
150	SB
501	SB
...	...

Array TT2	
REC	Data
2	Fri
4	Fri
7	Fri
9	Fri
...	...
149	Sat
150	Sat
501	Sat
...	...

Array TT3	
REC	Data
2	7
4	7
7	7
9	7
...	...
149	8
150	8
501	8
...	...

These arrays are then used to create three *equivalence* arrays called Q_REC_DIR, Q_REC_DAY and Q_REC_HOUR, which are the equivalence between the record number and each of the direction, day and hour.

```
newequiv Q_REC_DIR,REC,DIR,IT1                    [Code]

newequiv Q_REC_DAY,REC,DAY,IT2                    [Code]

newequiv Q_REC_HOUR,REC,HOUR,IT3                  [Code]
```

The final result is six one-dimensional *equivalence* arrays, which relate the record number to each of the direction, the day, the hour, the purpose, the origin postcode and the destination postcode.

Expansion factor

The next step is to attach an expansion factor to each of the completed survey records In our example, we will calculate an expansion factor for each direction, day and hour surveyed

From the completed survey records, we construct a three dimensional *numeric* array called SURV which contains the number of unexpanded survey records for each direction, day and hour surveyed.

```
SURV is accum 1,Q_REC_DIR,Q_REC_DAY,Q_REC_HOUR    [Code]
```

As the arrays COUNT and SURV both have the same shape (DIR by DAY by HOUR), the expansion factor EXP1 (which also must have the same shape) is simply calculated as:

```
EXP1 is COUNT / SURV                              [Code]
```

These results are summarised in Table 5 for our two time periods.

Table 5 Postcard survey - expansion factors

HOUR	NB		SB	
(com)	Fri	Sat	Fri	Sat
COUNT:				
7	80	60	50	36
8	80	40	27	51
9	50	45	19	27
SURV:				
7	4			
8				3
9				
EXP1:				
7	20.0			
8				17.0
9				

Long

The next step is to attach an expansion factor to each completed survey record:

EXP2 is lookup Q_REC_DIR,Q_REC_DAY,Q_REC_HOUR,EXP1 [Code]

Array EXP2	
REC	Data
2	20.0
4	20.0
7	20.0
9	20.0
...	...
149	17.0
150	17.0
501	17.0
...	...

Aggregating dimensions

Dimensions such as OPCODE (origin postcode) and DPCODE (destination postcode) can often have too many values for analysis. The solution is to aggregate postcodes into broader areas. In our example, we define two equivalence arrays called Q_OPCODE_OAREA and Q_DPCODE_DAREA. The dimensions OPCODE, OAREA, DPCODE and DAREA and the equivalences Q_OPCODE_OAREA and Q_DPCODE_DAREA are stored in the input dimension data file

It is now possible to create equivalence arrays called Q_REC_OAREA and Q_REC_DAREA by linking each survey record to a postcode and then linking that postcode to an area:

newequiv Q_REC_OAREA,Q_REC_OPCODE,Q_OPCODE_OAREA [Code]

newequiv Q_REC_DAREA,Q_REC_DPCODE,Q_DPCODE_DAREA [Code]

Constructing a six dimensional array

This step creates a six dimensional *numeric* array called DEMAND from the expanded survey records

DEMAND is accum EXP2,Q_REC_DIR,Q_REC_DAY, [Code]

Q_REC_HOUR,Q_REC_PURP,Q_REC_OAREA,Q_REC_DAREA

An array such as DEMAND will generally satisfy around 90 percent of our data analysis In our example, we could have replaced Q_REC_OAREA with Q_REC_OPCODE and Q_REC_DAREA with Q_REC_DPCODE. This would have created a six dimensional array which would have satisfied 100 percent of the data analysis - the tradeoff is degraded performance as the size of the array increases.

The code

The code is sequential, efficient and easily audited. The input dimension and data files plus the code are easily transferable The project file contains the programs with comments, all dimension and equivalence arrays, input data arrays as well as all temporary and output arrays.

1) batch in dimension/equivalence data file
2) batch in input data files
3) RREC is convert REC
4) RREC is char2num RREC
5) TT is range RREC,>=RANGE_LO,<=RANGE_HI
6) TT1 is lookup TT,NUM,DDIR
7) TT2 is lookup TT,NUM,DDAY
8) TT3 is lookup TT,NUM,HHOUR
9) newequiv Q_REC_DIR,REC,DIR,TT1
10) newequiv Q_REC_DAY,REC,DAY,TT2
11) newequiv Q_REC_HOUR,REC,HOUR,TT3
12) SURV is accum 1,Q_REC_DIR,Q_REC_DAY,Q_REC_HOUR
13) EXP1 is COUNT / SURV
14) EXP2 is lookup Q_REC_DIR,Q_REC_DAY,Q_REC_HOUR,EXP1
15) newequiv Q_REC_OAREA,Q_REC_OPCODE,Q_OPCODE_OAREA
16) newequiv Q_REC_DAREA,Q_REC_DPCODE,Q_DPCODE_DAREA
17) DEMAND is accum EXP2,Q_REC_DIR,Q_REC_DAY,

Q_REC_HOUR,Q_REC_PURP,Q_REC_OAREA,Q_REC_DAREA

Viewing multi dimensional arrays

It is easy to visualise either a one or a two dimensional array, a spreadsheet table being a simple two-dimensional array. At a pinch, a three dimensional array can be visualised as a cube of data Thereafter, it becomes difficult ...

A viewer has been designed to manipulate multi dimensional arrays. Let us manipulate the COUNT array which has dimensions DIR, DAY and HOUR.

Long

To reproduce Table 3 we would use:

	Down	Across	Total
DAY		X2	
DIR		X1	
HOUR	X		

Hour	NB Fri	NB Sat	SB Fri	SB Sat
7	80	60	50	36
8	80	40	27	51
9	50	45	19	27

Or we might want:

	Down	Across	Total
DAY	X2		
DIR	X1		
HOUR		X	

		7	8	9
NB	Fri	80	80	50
NB	Sat	60	40	45
SB	Fri	50	27	19
SB	Sat	36	51	27

Or perhaps:

	Down	Across	Total
DAY	X		
DIR			X
HOUR		X	

		7	8	9
Fri		130	107	69
Sat		96	91	72

And so on ....

Conclusion

Array based software is a cost-effective platform which can be used to verify that the process to expand survey data is computationally correct. It takes the unexpanded survey data and seeks to replicate the expanded survey data by converting the methodology in the working papers into sequential, efficient and self-documenting code. An independent and complete sign-off of the expanded survey data can be achieved.