Statistical disclosure control and tabular data

Methods

Contents

- Redesign
    - Size reduction
    - Recoding spanning variables
- Suppression
    - Primary
    - Secondary
- Rounding
- Miscellaneous

Redesign

Size reduction

Combine rows and/or columns to increase the number of respondents to cells

	Region								number of respondents	
	A		B		C		D		Total	
Harps	58	151	47	2	36	98	89	23	230	274
Organs	71	16	124	21	24	9	31	8	250	54
Pianos	92	5	157	12	59	7	28	1	336	25
Other	800	302	934	362	651	287	742	227	3127	1178
Total	1021	474	1262	397	770	401	890	259	3943	1531

Redesign

Size reduction

Combine rows and/or columns to increase the number of respondents to cells

	Region						number of respondents	
	A		B+C		D		Total	
Harps	58	151	83	100	89	23	230	274
Organs+Pianos	163	21	364	49	59	9	586	79
Other	800	302	1585	649	742	227	3127	1178
Total	1021	474	2032	798	890	259	3943	1531

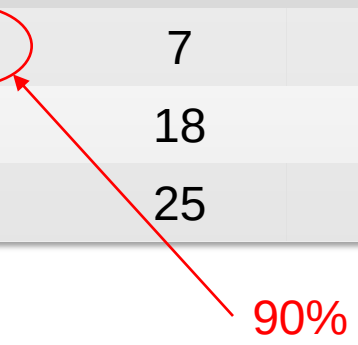
Redesign

Recoding spanning variables

Assign new categories to spanning variables, in order to ‘shuffle’ some respondents

Number of people with criminal record, by age classes

	<12	12-15	16-19	19+	Total
Yes	2	18	7	6	33
No	7	2	18	19	46
Total	9	20	25	25	79



Redesign

Recoding spanning variables

Assign new categories to spanning variables, in order to ‘shuffle’ some respondents

Number of people with criminal record, by age classes

	<13	13-16	17-20	20+	Total
Yes	3	17	7	6	33
No	7	4	16	19	46
Total	10	21	23	25	79

< 90%

Suppression

Primary suppressions:

suppress cells that are sensitive according to used measure

	Region				
	A	B	C	D	Total
Harps	58	**47**	36	89	230
Organs	71	124	24	31	250
Pianos	92	157	59	**28**	336
Other	800	934	651	742	3127
Total	1021	1262	770	890	3943

Suppression

Primary suppressions:

suppress cells that are sensitive according to used measure

	Region				
	A	B	C	D	Total
Harps	58	X	36	89	230
Organs	71	124	24	31	250
Pianos	92	157	59	X	336
Other	800	934	651	742	3127
Total	1021	1262	770	890	3943

Suppression

Secondary suppressions:

suppress additional cells to prevent recalculation

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column
not necessarily enough!

Can still be recalculated

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column
not necessarily enough!

Sum = 230 – 89 +
250 – (124 + 31) = 236

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column
not necessarily enough!

Sum = 1021 – (92 + 800) +
770 – (59 + 651) = 189

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column not necessarily enough!

X =

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column not necessarily enough!

X = 236

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

BEWARE:

two (or more) suppressions per row/column not necessarily enough!

X = 236 – 189 = 47

	Region				
	A	B	C	D	Total
Harps	X	X	X	89	230
Organs	X	124	X	31	250
Pianos	92	X	59	X	336
Other	800	X	651	X	3127
Total	1021	1262	770	890	3943

Suppression

Always:

using a-priori information (e.g., $T_C \geq 0$)
possible to obtain interval for each T_C

Feasibility intervals:

$[T_C - L_C, T_C + U_C]$ with $L_C , U_C \geq 0$

Sometimes $L_C = U_C = 0$

or bounds within p% of T_C

Feasibility intervals

How to obtain those intervals, given the suppressions?

Two LP-problems:

maximise T_C
under sum-restrictions of table and $T_C \geq 0$

minimise T_C
under sum-restrictions of table and $T_C \geq 0$

Feasibility intervals

E.g.: turnover (10^6 €)

	Region			
	A	B	C	Total
Books	20	50	10	80
Papers	x_1	19	x_2	49
Pens	x_3	32	x_4	61
Total	45	101	44	190

Feasibility intervals

E.g.: turnover (10^6 €)

	Region			
	A	B	C	Total
Books	20	50	10	80
Papers	x_1	19	x_2	49
Pens	x_3	32	x_4	61
Total	45	101	44	190

$$x_1 + x_2 = 49 - 19 = 30$$
$$x_3 + x_4 = 61 - 32 = 29$$
$$x_1 + x_3 = 45 - 20 = 25$$
$$x_2 + x_4 = 44 - 10 = 34$$

$$x_1, x_2, x_3, x_4 \geq 0$$

$$\min x_2: \quad x_2 \geq 5$$
$$\max x_2: \quad x_2 \leq 30$$

Feasibility intervals

example

	Region			
	A	B	C	Total
Books	20	50	10	80
Papers	[0,25]	19	**[5, 30]**	49
Pens	[0,25]	32	[4,29]	61
Total	45	101	44	190

Feasibility intervals

example

Other suppression pattern:

	Region			
	A	B	C	Total
Books	20	x_3	x_4	80
Papers	8	x_1	$\boldsymbol{x_2}$	49
Pens	17	32	12	61
Total	45	101	44	190

$$x_1 + x_2 = 49 - 8 = 41$$
$$x_3 + x_4 = 80 - 20 = 60$$
$$x_1 + x_3 = 101 - 32 = 69$$
$$x_2 + x_4 = 44 - 12 = 32$$

$$x_1, x_2, x_3, x_4 \geq 0$$

$$\min x_2: \; x_2 \geq 0$$
$$\max x_2: \; x_2 \leq 32$$

Feasibility intervals

example

Other suppression pattern:

	Region			
	A	B	C	Total
Books	20	[28,60]	[0,32]	80
Papers	8	[9,41]	**[0, 32]**	49
Pens	17	32	12	61
Total	45	101	44	190

Feasibility intervals

example

Simple 2-dimensional table

	A	B	Total
01	x_1	x_2	7
02	x_3	x_4	3
03	3	3	6
Total	9	7	16

subtract

Feasibility intervals

example

Simple 2-dimensional table

	A	B	Total
01	x_1	x_2	7
02	x_3	x_4	3
Total	6	4	10

Assuming every entry is non-negative

Second row: $x_4 \leq 3$

implies $x_2 \geq 1$

Feasibility intervals

example

Simple 2-dimensional table

	A	B	Total
01	(3,4,5,6)	(4,3,2,1)	7
02	(3,2,1,0)	(0,1,2,3)	3
Total	6	4	10

Safety ranges

Feasibility intervals for primary unsafe cells should be sufficiently wide

Possible rule:

Require minimum feasibility interval, i.e., require *safety ranges*, for each primary unsafe cell

Safety ranges

Possible choices for safety ranges:

- fixed percentage of cell value

  disadvantage: the “size” of sensitivity is not used

- related to sensitivity measure that is used

  e.g.: interval contains at least one (or *m*%) cell value(s) that satisfy the primary rules

  e.g.: value of *S*(*C*) determines the size of the safety-range

Safety ranges

Example (*p*% rule)

Feasibility interval of cell *C*:

$$[T_C - L_C, T_C + U_C]$$

Upper bound for largest contribution (x_2 attacks):

$$x_1^U = (T_C + U_C) - x_2$$

Requirement $x_1^U \geq (1 + p/100)x_1$

then yields

$$U_C \geq \frac{p}{100} x_1 - \sum_{i=3}^{N(C)} x_i = \frac{p}{100} x_1 - (T_C - x_1 - x_2)$$

Safety ranges

Example (*p*% rule)

Note:

- equivalently, L_C can be bounded using the lower estimate of x_1
- protection interval depends on used safety measure S_p

Safety ranges

Example (*p*% rule)

Problem:

two combined cells that provide the required interval do not necessarily satisfy the primary rule

Reason:

the structure of the ‘added’ cell is not taken into account, i.e., the second largest may change

Safety ranges

Example (*p*% rule)

51	**42**	55	148
9	**10**	31	50
28	43	29	100
88	95	115	298

$p = 25$

Cell C: **51** = 44 + 4 + 1 + 1 + 1

$S_p(C) = 0.25*44 - (1 + 1 + 1) = 8$ *Unsafe!*

Need $U_C \geq 0.25*44 - (1 + 1 + 1) = 8$

'Add' cell Y: **9** = 6 + 1 + 1 +1 $S_p(Y) = 0.25*6 - (1 + 1) = -0.5$ *Safe!*

$S_p(C + Y) = 0.25*44 - (4 + 1 + 1 + 1 + 1 + 1 + 1)$

$= 1$ *Unsafe!*

Safety ranges

Other reason for same problem: Holdings

Combination of cells may contain part of a holding

Other part knows suppressed value

May use that to get better estimate than allowed

Suppression

Given safety ranges, how to determine suppression pattern?

Remember:

two or more suppressions per row is necessary but not sufficient!

Moreover:

what about information loss?

Suppression

Goal:

Find a feasible suppression pattern

with a minimal loss of information

A suppression pattern is feasible, when the feasibility intervals of all primary suppressions provide enough protection

Finding a feasibility ***interval*** for each suppressed cell requires Linear Programming Methods

Finding a feasibility ***pattern with minimal information loss*** is a very complex Linear Programming Problem

Suppression

Minimum protection standard

- *Protection against exact disclosure*
  It is not possible to disclose the exact value of a sensitive cell if no additional knowledge is considered (like singletons) and if the analysis is carried out for each subtable separately

- *Protection against singleton disclosure*
  A suppression pattern with only two suppressed cells within a row (or column) of a table is not feasible if each one of the two corresponds to a single respondent

Suppression

Methods in τ-ARGUS:

- HyperCube: Applied to subtables
- Network: Network-flow method, only for up to 2-dimensional tables with at most 1 hierarchical spanning variable
- Optimal: Linear Optimization
- Modular: Linear Optimization, applied to subtables

Commercial LP solver?

Suppression

Hypercube method

For each sensitive cell c_s, choose a hypercube of cells, such that

- c_s is one of the vertices of the hypercube
- suppression of all vertices yields the required safety range

Such a hypercube is called a

candidate suppression cube

Of all candidate suppression cubes, choose the one with minimum loss of information

Suppression

Optimal/Modular

Uses restrictions, based on table-structure, and costs that determine the information loss.

Costs can be specified

e.g.: each cell cost 1 (minimise number of suppressions)

each cell its cell-value as cost (minimise total sum of suppressed values)

each cell its number of contributions as cost (minimise total number of suppressed contributions)

costs depending on distance to “closest” primary unsafe cell

Suppression

	Modular	Optimal	Network	HyperCube
Exact or interval disclosure	Interval (subtable scenario)	Interval	Interval	Interval (subtable scenario)
Singleton disclosure	Yes (row/column scenario)	Yes (row/column scenario)	No	Yes

Suppression

- Network does not deal with singleton disclosure
- Modular and HyperCube both deal with interval disclosure (at subtable level)
- Modular outperforms Hypercube (smaller information loss)
- Modular can not be used in certain situations (like 5-dimensional tables)

Rounding

Often aesthetic.

But provides an interval to each cell as well

Basic idea:

- replace each cell by integer multiple of base *b*

Obviously: larger *b* implies more protection

Choose *b* according to used sensitivity measure

Rounding

Conventional

Each cell value replaced by nearest integer multiple of base *b*

	Region			
	A	B	C	Total
Books	20	50	10	80
Papers	8	19	22	49
Pens	17	32	12	61
Total	45	101	44	190

Suppose $b = 5$

Rounding

Controlled

Preserve additivity by replacing cell value by one of two nearest integer multiples of base *b*, i.e.,

$$\left\lfloor \frac{T_C}{b} \right\rfloor b \quad \text{or} \quad \left\lceil \frac{T_C}{b} \right\rceil b \qquad \text{(may be } T_C \text{ itself)}$$

Zero restricted:

preserve existing multiples of *b*

Unbiased:

integer multiple of *b* determined stochastically, such that expected rounding error equals zero

Rounding

Controlled

Methods:

- Method of Felligi (one dimensional tables)
- Method of Cox (two dimensional tables)

- MIP approach (Salazar, general tables)
  - specified safety ranges
  - *b* and safety ranges should match (or relax restrictions)

More generally linked tables

Often tables are linked:

- Spanning variables having common categories
  - Labour costs by NACE and Region
  - Labour costs by NACE and size class
  - Labour costs in NACE section G by Nace, Region and size class
- Should be protected consistently

More generally linked tables

Protection of complete multidimensional table?

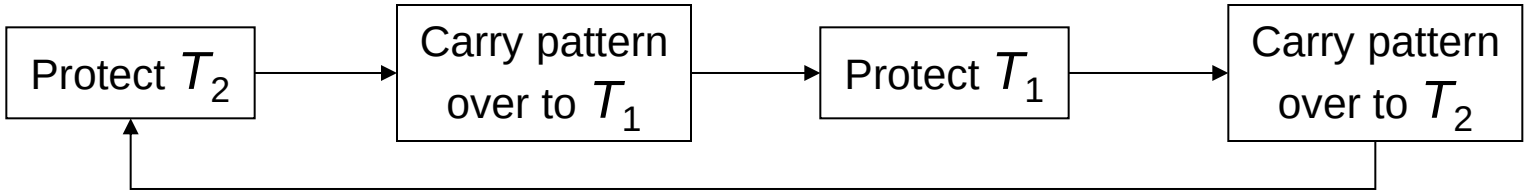
- Numerical problems
- Often heavy overprotection!

Iterative protection of individual tables

- “Backtracking”

Backtracking

Suppose two linked tables $\{T_1, T_2\}$



Carry pattern over: use *a priori file* to set secondary suppressions to primary suppressions

Repeat until no changes in patterns?

Three or more linked tables?

Modular Approach

Modular Approach for Linked Tables

- Use basics from Modular Approach
  - First construct ‘cover’-table
  - Apply Modular Approach on cover-table skipping subtables that do not appear in any table of the set of linked tables