

A Literate Program for Converting Tables to LongForm Dataframes

Jimmy Oh

Department of Statistics
University of Auckland

Abstract

TableToLongForm automatically converts hierarchical Tables intended for a human reader into a simple LongForm Dataframe that is machine readable. It does this by recognising positional queues present in the hierarchical Table (which would normally be interpreted visually by the human brain) to decompose, then reconstruct the data into a LongForm Dataframe. This document provides a gallery of all recognised patterns and structures, with accompanying toy examples, before finally going into depth on the workings of the code itself.

Contents

1	Introduction	3
1.1	Motivation	3
1.2	The Plan of Attack	4
1.3	Recognised Patterns	4
1.3.1	By Empty Below	5
1.3.2	By Empty Below Transposed	5
1.3.3	By Empty Right 1	5
1.3.4	By Empty Right 2	6
1.3.5	By Empty Right 3	6
1.3.6	Multi-row Column Label	7
1.3.7	Misaligned Column Label	7
1.3.8	Find Single Table	7
1.3.9	Complete Example	8
2	Code Overview	8
2.1	Front End	9
2.2	Back End	10
3	Identification	11
3.1	Identification - Main Function	11
3.1.1	Ident By Most Common Boundary	12
3.1.2	Group Column Labels	13
3.2	Identification - Low Level Functions	14
3.2.1	IdentNonEmpty	15
3.2.2	IdentPattern	15
3.2.3	Ident Most Common Boundary	16

4	Discern Parentage	16
4.0.4	plist	17
4.1	Parentage - Front End Function	17
4.1.1	Pare Col	17
4.2	Parentage - Main Function	18
4.3	Parentage - Low Level Functions	20
4.3.1	Pare By Empty Right	20
4.3.2	Pare By Empty Below	22
5	Reconstruction	22
5.1	Reconstruction - Main Function	23
5.2	Reconstruction - Low Level Functions	25
5.2.1	Reconstruction - Row Labels	25
5.2.2	Reconstruction - Column Labels	26
6	Chunk Index	28
7	Identifier Index	28
	References	30
8	Appendix: TCRO	30
8.1	NZQAScholarships.xls	31
8.2	NZQASubjects.xls	96
8.3	StatsNZLabourForce.csv	104
8.4	ToyExByEmptyBelow.csv	124
8.5	ToyExByEmptyBelowT.csv	126
8.6	ToyExByEmptyRight1.csv	129
8.7	ToyExByEmptyRight2.csv	132
8.8	ToyExByEmptyRight3.csv	135
8.9	ToyExComplete.csv	139
8.10	ToyExFindSingleTable.csv	143
8.11	ToyExMisalignedColumnLabel.csv	145
8.12	ToyExMultiRowColumnLabel.csv	148

On Literate Programs

This software is presented as a *literate program* written in the *noweb* format (Ramsey 1994). It serves as both the documentation and container of the literate program. The **noweb** file can be used to produce both the *literate document* and the executable code.

The literate document is separated into *documentation chunks* and named *code chunks*. Each *code chunk* can contain code directly, or contain references to other *code chunks* which act as placeholders for the contents of the respective *code chunk*. The name of each *code chunk* should serve as a short description of the code it contains. Thus each *code chunk* provides an overview of its purpose by either directly containing code, or by containing the names of other *code chunks*. The reader is then free to delve deeper into the respective *code chunks* if desired.

1 Introduction

1.1 Motivation

In recent times there has been a movement toward *Open Data*, particularly for government data^{1,2}, yet there is still a prevalence of data releases being for direct human consumption, rather than for machine consumption. One symptom of this is the release of data in tabular form that relies on the human ability to identify patterns and discern structure, in order to decipher the data (henceforth referred to as a Table). Such tables are difficult to read and analyse with the computer, significantly limiting potential applications of this ‘open’ data.

LongForm is a simple alternative method of releasing the data that, due to its simplicity, is both easy to implement and is machine readable, greatly enhancing potential applications of the data. It is easy to go from a simple format such as a LongForm Dataframe to any number of other forms of presentation, including hierarchical Tables more suitable for direct human consumption. However the converse is rarely true. This is where TableToLongForm comes in, providing a way to automatically convert hierarchical Tables to a simple LongForm Dataframe, thus enabling much greater utilisation of the data.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Labour Force Status by Sex by Sing/Comb Ethnic Group (Qrtly–Mar/Jun/Sep/Dec)												
2		Male											
3		European Only								Maori Only			
4		Persons En	Persons Un	Not in Labo	Working Ag	Labour Forc	Unemploy	Employer	Total Labou	Persons En	Persons Un	Not in Labo	Working Ag
5	2007Q4	855.8	20.0	280.0	1155.8	75.8	2.3	74.0	875.8	71.1	6.1	28.1	105.3
6	2008Q1	863.0	25.4	283.5	1171.9	75.8	2.9	73.6	888.5	69.1	7.5	31.4	107.9
7	2008Q2	850.1	26.0	280.7	1156.8	75.7	3.0	73.5	876.1	67.2	5.7	27.4	100.2
8	2008Q3	839.6	29.8	285.9	1155.3	75.2	3.4	72.7	869.4	71.7	8.7	30.7	111.1
9	2008Q4	854.8	29.5	274.7	1158.9	76.3	3.3	73.8	884.2	76.1	8.5	28.5	113.1
10	2009Q1	845.0	35.4	279.4	1159.8	75.9	4.0	72.9	880.4	75.4	8.4	35.7	119.5
11	2009Q2	831.6	34.9	279.7	1146.2	75.6	4.0	72.6	866.5	74.2	9.9	33.1	117.3
12	2009Q3	813.3	42.5	290.4	1146.2	74.7	5.0	71.0	855.8	70.9	10.9	36.0	117.8
13	2009Q4	831.1	40.1	277.0	1148.2	75.9	4.6	72.4	871.2	71.7	13.6	33.2	118.5
14	2010Q1	822.5	36.4	283.2	1142.1	75.2	4.2	72.0	858.9	71.8	11.3	35.3	118.4
15	2010Q2	825.3	39.9	290.1	1155.3	74.9	4.6	71.4	865.2	71.9	13.7	33.7	119.2
16	2010Q3	836.9	31.0	287.1	1155.1	75.1	3.6	72.5	867.9	69.8	13.5	34.1	117.3
17	2010Q4	838.1	39.6	277.1	1154.8	76.0	4.5	72.6	877.7	70.7	14.4	36.4	121.5
18	2011Q1	829.6	36.8	281.2	1147.6	75.5	4.2	72.3	866.4	70.7	13.9	35.3	119.8
19	2011Q2	838.7	41.0	279.0	1158.7	75.9	4.7	72.4	879.6	67.1	10.5	37.1	114.7
20	2011Q3	830.5	34.6	280.2	1145.3	75.5	4.0	72.5	865.1	69.5	13.4	34.9	117.8
21	2011Q4	841.8	34.8	277.5	1154.1	76.0	4.0	72.9	876.6	69.2	12.7	36.1	118.0
22	2012Q1	843.1	43.3	282.7	1169.1	75.8	4.9	72.1	886.3	71.5	11.2	34.6	117.3
23	2012Q2	837.1	38.2	296.5	1171.8	74.7	4.4	71.4	875.3	66.2	10.6	33.0	109.7
24	2012Q3	833.0	38.0	298.4	1169.3	74.5	4.4	71.2	871.0	67.1	13.1	33.0	113.2
25	2012Q4	833.4	41.0	298.2	1172.6	74.6	4.7	71.1	874.4	63.0	12.3	35.4	110.7
26	2013Q1	832.0	35.8	294.7	1162.5	74.6	4.1	71.6	867.8	69.9	12.2	38.2	120.3
27	Table information:												
28	Units:												
29	Persons Employed in Labour Force: Number, Magnitude = Thousands												
30	Persons Unemployed in Labour Force: Number, Magnitude = Thousands												
31	Not in Labour Force: Number, Magnitude = Thousands												
32	Working Age Population: Number, Magnitude = Thousands												
33	Labour Force Participation Rate: Percent, Magnitude = Units												
34	Unemployment Rate: Percent, Magnitude = Units												
35	Employment Rate: Percent, Magnitude = Units												
36	Total Labour Force: Number, Magnitude = Thousands												
37	Footnotes:												
38													
39	Symbols:												
40	.. figure not available												
41	C: Confidential												
42	E: Early Estimate												
43	P: Provisional												

Figure 1: An example of a hierarchical Table. The data is Labour Force Status Survey data from Infoshare, Statistics New Zealand (2013) and in total spans 240 columns, making it suitable for neither man nor machine. However it is relatively tame in terms of how unsuitable for machines Tables can get and so could, with some manual labour, be read by a machine.

¹ “In many countries across the world, discussions, policies and developments are actively emerging around open access to government data.” Davies and Bawa (2012a)

² “Over 100 OGD [Open Government Data] initiatives are active across the globe, ranging from community-led OGD projects in urban India, to a World Bank sponsored OGD programme in Kenya, government-led developments in Brazil, civil-society initiated work in Russia, and a World Wide Web Foundation supported programme in Ghana.” Davies and Bawa (2012b)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1				Persons En	Persons Un	Not in Labo	Working Ag	Labour Forc	Unemployr	Employer	Total Labour Force		
2	Male	European C	2007Q4	855.8	20	280	1155.8	75.8	2.3	74	875.8		
3	Male	European C	2008Q1	863	25.4	283.5	1171.9	75.8	2.9	73.6	888.5		
4	Male	European C	2008Q2	850.1	26	280.7	1156.8	75.7	3	73.5	876.1		
5	Male	European C	2008Q3	839.6	29.8	285.9	1155.3	75.2	3.4	72.7	869.4		
6	Male	European C	2008Q4	854.8	29.5	274.7	1158.9	76.3	3.3	73.8	884.2		
7	Male	European C	2009Q1	845	35.4	279.4	1159.8	75.9	4	72.9	880.4		
8	Male	European C	2009Q2	831.6	34.9	279.7	1146.2	75.6	4	72.6	866.5		
9	Male	European C	2009Q3	813.3	42.5	290.4	1146.2	74.7	5	71	855.8		
10	Male	European C	2009Q4	831.1	40.1	277	1148.2	75.9	4.6	72.4	871.2		
11	Male	European C	2010Q1	822.5	36.4	283.2	1142.1	75.2	4.2	72	858.9		
12	Male	European C	2010Q2	825.3	39.9	290.1	1155.3	74.9	4.6	71.4	865.2		
13	Male	European C	2010Q3	836.9	31	287.1	1155.1	75.1	3.6	72.5	867.9		
14	Male	European C	2010Q4	838.1	39.6	277.1	1154.8	76	4.5	72.6	877.7		
15	Male	European C	2011Q1	829.6	36.8	281.2	1147.6	75.5	4.2	72.3	866.4		
16	Male	European C	2011Q2	838.7	41	279	1158.7	75.9	4.7	72.4	879.6		
17	Male	European C	2011Q3	830.5	34.6	280.2	1145.3	75.5	4	72.5	865.1		
18	Male	European C	2011Q4	841.8	34.8	277.5	1154.1	76	4	72.9	876.6		
19	Male	European C	2012Q1	843.1	43.3	282.7	1169.1	75.8	4.9	72.1	886.3		
20	Male	European C	2012Q2	837.1	38.2	296.5	1171.8	74.7	4.4	71.4	875.3		
21	Male	European C	2012Q3	833	38	298.4	1169.3	74.5	4.4	71.2	871		
22	Male	European C	2012Q4	833.4	41	298.2	1172.6	74.6	4.7	71.1	874.4		
23	Male	European C	2013Q1	832	35.8	294.7	1162.5	74.6	4.1	71.6	867.8		
24	Male	Maori Only	2007Q4	71.1	6.1	28.1	105.3	73.4	7.9	67.6	77.2		
25	Male	Maori Only	2008Q1	69.1	7.5	31.4	107.9	71	9.7	64.1	76.6		
26	Male	Maori Only	2008Q2	67.2	5.7	27.4	100.2	72.7	7.8	67	72.8		
27	Male	Maori Only	2008Q3	71.7	8.7	30.7	111.1	72.3	10.8	64.5	80.3		
28	Male	Maori Only	2008Q4	76.1	8.5	28.5	113.1	74.8	10	67.3	84.5		
29	Male	Maori Only	2009Q1	75.4	8.4	35.7	119.5	70.1	10.1	63.1	83.8		
30	Male	Maori Only	2009Q2	74.2	9.9	33.1	117.3	71.8	11.8	63.3	84.2		
31	Male	Maori Only	2009Q3	70.9	10.9	36	117.8	69.5	13.4	60.2	81.8		
32	Male	Maori Only	2009Q4	71.7	13.6	33.2	118.5	71.9	15.9	60.5	85.3		
33	Male	Maori Only	2010Q1	71.8	11.3	35.3	118.4	70.2	13.6	60.6	83.1		
34	Male	Maori Only	2010Q2	71.9	13.7	33.7	119.2	71.8	16	60.3	85.6		
35	Male	Maori Only	2010Q3	69.8	13.5	34.1	117.3	71	16.2	59.5	83.3		
36	Male	Maori Only	2010Q4	70.7	14.4	36.4	121.5	70	16.9	58.2	85.1		
37	Male	Maori Only	2011Q1	70.7	13.9	35.3	119.8	70.6	16.4	59	84.6		
38	Male	Maori Only	2011Q2	67.1	10.5	37.1	114.7	67.6	13.5	58.5	77.6		
39	Male	Maori Only	2011Q3	69.5	13.4	34.9	117.8	70.4	16.1	59	82.9		
40	Male	Maori Only	2011Q4	69.2	12.7	36.1	118	69.4	15.5	58.6	81.8		
41	Male	Maori Only	2012Q1	71.5	11.2	34.6	117.3	70.5	13.6	60.9	82.7		
42	Male	Maori Only	2012Q2	66.2	10.6	33	109.7	69.9	13.8	60.3	76.8		
43	Male	Maori Only	2012Q3	67.1	13.1	33	113.2	70.8	16.4	59.2	80.2		

Figure 2: An example of a LongForm Dataframe. This is the Labour Force Status Survey data after automatic conversion with TableToLongForm. Most, if not all statistical software (including spreadsheet software like Excel) can read, manipulate and run analyses on this without any problems.

1.2 The Plan of Attack

Unless the Table is horrible beyond mortal imagination, it should have some kind of pattern, such that a human will be able to discern the structure and hence understand the data it represents. This code attempts to algorithmically search for such patterns, discern the structure, then reconstruct the data into a LongForm Dataframe. Refer to Section 1.3 for a full gallery of currently recognised patterns.

The task can be seen to consist of three phases:

- Phase One is Identification (Section 3), which involves identifying the rows and columns where the labels and the data can be found.
- Phase Two is Discerning the Parentage (Section 4), which involves identifying the hierarchical structure of the data, based on the row and column labels.
- Phase Three is Reconstruction (Section 5), where we use what we've found in the first two phases to reconstruct the data into a LongForm Dataframe.

1.3 Recognised Patterns

Here we list, with toy examples, all the recognised patterns and structures. TableToLongForm should be able to process any combination of these patterns to automatically convert many different types of tables. The last example is a *complete* example that contains most of the recognised patterns in a single horrible table.

1.3.1 By Empty Below

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1	Row Child1	10	20	30	40
3		Row Child2	11	21	31	41
4	Row Parent2	Row Child1	12	22	32	42
5		Row Child2	13	23	33	43

The most simple type of parentage, here the *parent* and *children* are in different columns and we can see which of the children belong to which parent through the use of empty space below each parent.

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1	Row Child1	10	20	30	40
3	Row Parent1	Row Child2	11	21	31	41
4	Row Parent2	Row Child1	12	22	32	42
5	Row Parent2	Row Child2	13	23	33	43

1.3.2 By Empty Below Transposed

	1	2	3	4	5
1		Row Parent1		Row Parent2	
2		Row Child1	Row Child2	Row Child1	Row Child2
3	Column 1	10	11	12	13
4	Column 2	20	21	22	23
5	Column 3	30	31	32	33
6	Column 4	40	41	42	43

We note that parentage patterns recognised for row labels can often be applied to the transpose of column labels. This is how TableToLongForm deciphers most column labels, with some exceptions.

	1	2	3	4	5
1			Row Child1	Row Child2	
2	Row Parent1	Column 1	10	11	
3	Row Parent1	Column 2	20	21	
4	Row Parent1	Column 3	30	31	
5	Row Parent1	Column 4	40	41	
6	Row Parent2	Column 1	12	13	
7	Row Parent2	Column 2	22	23	
8	Row Parent2	Column 3	32	33	
9	Row Parent2	Column 4	42	43	

1.3.3 By Empty Right 1

	1	2	3	4	5	6	7
1				Column 1	Column 2	Column 3	Column 4
2	Row Parent1			10	20	30	40
3	Row Child1	Row Child-Child1		11	21	31	41
4	Row Child2	Row Child-Child2		12	22	32	42
5	Row Parent2			13	23	33	43
6	Row Child1	Row Child-Child1		14	24	34	44
7		Row Child-Child2		15	25	35	45

In this situation we have children in the same column as their parent. We can still recognise these as children if the children have children (*Child-Child*) in a different column, while the

parent does not (and hence is Empty Right).

Note the values pertaining to the Parent (if any) are discarded. This is because they are assumed to simply represent the sum of their children's values. It is planned for a sum-check to be implemented later to make this more robust.

	1	2	3	4	5	6	7
1				Column 1	Column 2	Column 3	Column 4
2	Row Parent1	Row Child1	Row Child-Ch	11	21	31	41
3	Row Parent1	Row Child2	Row Child-Ch	12	22	32	42
4	Row Parent2	Row Child1	Row Child-Ch	14	24	34	44
5	Row Parent2	Row Child1	Row Child-Ch	15	25	35	45

1.3.4 By Empty Right 2

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1		10	20	30	40
3		Row Child1	11	21	31	41
4		Row Child2	12	22	32	42
5	Row Parent2		13	23	33	43
6		Row Child1	14	24	34	44
7		Row Child2	15	25	35	45

Here we have both Empty Below and Empty Right. Either algorithm can handle this situation, but simply due to the ordering of the algorithms such situations are handled as Empty Right.

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1	Row Child1	11	21	31	41
3	Row Parent1	Row Child2	12	22	32	42
4	Row Parent2	Row Child1	14	24	34	44
5	Row Parent2	Row Child2	15	25	35	45

1.3.5 By Empty Right 3

	1	2	3	4	5	6	7	8
1				Column 1	Column 2	Column 3	Column 4	
2	Row Super-Parent1			10	20	30	40	
3	Row Parent1			11	21	31	41	
4	Row Child1	Row Child-Child1		12	22	32	42	
5	Row Parent2			13	23	33	43	
6	Row Child1	Row Child-Child1		14	24	34	44	
7	Row Super-Parent2			15	25	35	45	
8	Row Parent1			16	26	36	46	
9	Row Child1	Row Child-Child1		17	27	37	47	
10	Row Parent2			18	28	38	48	
11	Row Child1	Row Child-Child1		19	29	39	49	

The “parent-child in the same column” situation can be extended further. Here we have parents (*Super-Parent*) who have children (*Parent*), who each further have children (*Child*), all in the same column. Such situations can still be recognised if the lowest-level children in the column (*Child*) have children in a different column (*Child-Child*), while its direct parents (*Parent*) each have children in the same column (*Child*) but not in a different column (is Empty Right), and the top-most parents (*Super-Parents*) also have no children in a different column (is also Empty Right).

The algorithm cannot currently handle super-super-parents.

	1	2	3	4	5	6	7	8
1					Column 1	Column 2	Column 3	Column 4
2	Row Super-P	Row Parent1	Row Child1	Row Child-Ch	12	22	32	42
3	Row Super-P	Row Parent2	Row Child1	Row Child-Ch	14	24	34	44
4	Row Super-P	Row Parent1	Row Child1	Row Child-Ch	17	27	37	47
5	Row Super-P	Row Parent2	Row Child1	Row Child-Ch	19	29	39	49

1.3.6 Multi-row Column Label

	1	2	3	4	5
1		Column	Column	Column	Column
2		Child1	Child2	Child3	Child4
3	Row 1	10	20	30	40
4	Row 2	11	21	31	41
5	Row 3	12	22	32	42
6	Row 4	13	23	33	43

Often column labels are physically split over multiple rows rather than making use of line breaks in the same cell. In such occurrences, any row not identified as a parent are collapsed into a single row of labels. It is eventually planned for pattern recognition to be used here to make this collapsing smarter.

	1	2	3	4	5
1		Column Child1	Column Child2	Column Child3	Column Child4
2	Row 1	10	20	30	40
3	Row 2	11	21	31	41
4	Row 3	12	22	32	42
5	Row 4	13	23	33	43

1.3.7 Misaligned Column Label

	1	2	3	4	5	6	7	8	9
1			Column Parent1				Column Parent2		
2		Col Child1	Col Child2	Col Child3	Col Child4	Col Child1	Col Child2	Col Child3	Col Child4
3	Row 1	10	20	30	40	50	60	70	80
4	Row 2	11	21	31	41	51	61	71	81
5	Row 3	12	22	32	42	52	62	72	82
6	Row 4	13	23	33	43	53	63	73	83

Often column parents are physically centred over their children (N.B. where a spreadsheet's cell-merge feature is used to do the centering, the actual value is usually stored in the top-left cell and hence causes no problems). TableToLongForm makes use of pattern recognition to identify repeating patterns in the labels of the children, to help discern the correct parent for the children.

	1	2	3	4	5	6	7	8	9
1			Col Child1	Col Child2	Col Child3	Col Child4			
2	Column Paren	Row 1	10	20	30	40			
3	Column Paren	Row 2	11	21	31	41			
4	Column Paren	Row 3	12	22	32	42			
5	Column Paren	Row 4	13	23	33	43			
6	Column Paren	Row 1	50	60	70	80			
7	Column Paren	Row 2	51	61	71	81			
8	Column Paren	Row 3	52	62	72	82			
9	Column Paren	Row 4	53	63	73	83			

1.3.8 Find Single Table

	1	2	3	4	5
1	MISC INFORMATION				
2	MISC INFORMATION				
3		Column 1	Column 2	Column 3	Column 4
4	Row 1	10	20	30	40
5	Row 2	11	21	31	41
6	Row 3	12	22	32	42
7	Row 4	13	23	33	43
8	MISC INFORMATION		MISC INFORMATION		
9	MISC INFORMATION		MISC INFORMATION		

A table is often found amongst miscellaneous information we do not want. TableToLongForm is intended to have several algorithms to identify not only a single table, but multiple tables on the same ‘page’. Currently however, it can only identify a single table per ‘page’ by searching for a block (rectangular region) of numbers, which is assumed to be our table of data.

	1	2	3	4	5
1		Column 1	Column 2	Column 3	Column 4
2	Row 1	10	20	30	40
3	Row 2	11	21	31	41
4	Row 3	12	22	32	42
5	Row 4	13	23	33	43

1.3.9 Complete Example

	1	2	3	4	5	6	7	8	9	10	11
1	MISC INFORMATION										
2					Column Parent1				Column Parent2		
3				Column	Column	Column	Column	Column	Column	Column	Column
4				Child1	Child2	Child3	Child4	Child1	Child2	Child3	Child4
5	Row Super-Parent			10	20	30	40	50	60	70	80
6	Row Parent1			11	21	31	41	51	61	71	81
7	Row Child1	Row Child-Child1		12	22	32	42	52	62	72	82
8		Row Child-Child2		13	23	33	43	53	63	73	83
9	Row Child2	Row Child-Child1		14	24	34	44	54	64	74	84
10		Row Child-Child2		15	25	35	45	55	65	75	85
11	Row Parent2			16	26	36	46	56	66	76	86
12	Row Child1	Row Child-Child1		17	27	37	47	57	67	77	87
13		Row Child-Child2		18	28	38	48	58	68	78	88
14	Row Child2	Row Child-Child2		19	29	39	49	59	69	79	89
15	MISC INFORMATION										
16	MISC INFORMATION										

A complete example containing a combination of many of the patterns listed above.

	1	2	3	4	5	6	7	8	9	10	11
1						Column Child1	Column Child2	Column Child3	Column Child4		
2	Column Parent1	Row Super-Parent1	Row Parent1	Row Child1	Row Child-Child1	12	22	32	42		
3	Column Parent1	Row Super-Parent1	Row Parent1	Row Child1	Row Child-Child1	13	23	33	43		
4	Column Parent1	Row Super-Parent1	Row Parent1	Row Child2	Row Child-Child2	14	24	34	44		
5	Column Parent1	Row Super-Parent1	Row Parent1	Row Child2	Row Child-Child2	15	25	35	45		
6	Column Parent1	Row Super-Parent1	Row Parent2	Row Child1	Row Child-Child1	17	27	37	47		
7	Column Parent1	Row Super-Parent1	Row Parent2	Row Child1	Row Child-Child1	18	28	38	48		
8	Column Parent1	Row Super-Parent1	Row Parent2	Row Child2	Row Child-Child2	19	29	39	49		
9	Column Parent1	Row Super-Parent1	Row Parent1	Row Child1	Row Child-Child1	52	62	72	82		
10	Column Parent1	Row Super-Parent1	Row Parent1	Row Child1	Row Child-Child1	53	63	73	83		
11	Column Parent1	Row Super-Parent1	Row Parent1	Row Child2	Row Child-Child2	54	64	74	84		
12	Column Parent1	Row Super-Parent1	Row Parent1	Row Child2	Row Child-Child2	55	65	75	85		
13	Column Parent1	Row Super-Parent1	Row Parent2	Row Child1	Row Child-Child1	57	67	77	87		
14	Column Parent1	Row Super-Parent1	Row Parent2	Row Child1	Row Child-Child1	58	68	78	88		
15	Column Parent1	Row Super-Parent1	Row Parent2	Row Child2	Row Child-Child2	59	69	79	89		

2 Code Overview

TableToLongForm is structured as follows.

8 $\langle \text{TableToLongForm.R } 8 \rangle \equiv$
 $\langle \text{document header } 9a \rangle$
 $\langle \text{Front End } 9b \rangle$
 $\langle \text{Identification } 11a \rangle$
 $\langle \text{Discern Parentage } 16b \rangle$
 $\langle \text{Reconstruction } 22b \rangle$
 $\langle \text{Back End } 10 \rangle$

This code is written to file `TableToLongForm.R`.

We place a document header at the top of the extracted code to encourage people to read the literate description rather than attempting to study the code alone.

```
9a  <document header 9a>≡
    ##-----
    ## The code in this .R file is machine generated from the literate
    ## program, TableToLongForm.Rnw
    ## Documentation can be found in the literate description for this
    ## program, TableToLongForm.pdf
    ##-----
```

2.1 Front End

The main function `TableToLongForm` is defined here. For most users this is the only function they will call. However, the majority of the supporting functions are not hidden and therefore can easily be viewed and/or modified by users.

```
9b  <Front End 9b>≡
    TableToLongForm =
      function(datamat, IdentResult = NULL,
               fulloutput = FALSE, diag = FALSE, diagname = NULL){
        if(diag){
          if(is.null(diagname)) diagname = deparse(substitute(datamat))
          assign("TCRunout", file(paste0(diagname, ".TCRunout"),
                                   "w"), envir = .GlobalEnv)

          on.exit({
            close(TCRunout)
            rm("TCRunout", envir = .GlobalEnv)
          })
        }

        fullout = ReconsMain(datamat, IdentResult)
        if(fulloutput) fullout else fullout$datafr
      }
```

2.2 Back End

Various code, mainly to help produce diagnostic output, can be ignored by most users.

print.plist A print method for class **plist**, which are nested lists with a numeric vector at the lowest level; **print.default** is rather inefficient in displaying such nested lists.

TCRsink Sinks the output to TCR_{runout} for diagnostic output. Requires the existence of TCR_{runout} which is created by the main function **TableToLongForm** when **diag = TRUE**.

Spaces may be introduced by **match.call**, thus any spaces in the args of *variables to sink* (that is, the arguments supplied via ...) are removed without warning.

```
10 <Back End 10>≡
    print.plist = function(plist){
      plistC = function(plist){
        pLoc = attr(plist, "Loc")
        if(is.list(plist)){
          namevec = names(plist)
          if(!is.null(pLoc))
            namevec = paste0(names(plist),
              " (", pLoc[, "rows"], ", ", pLoc[, "cols"], ")")
          namelist = as.list(namevec)
          for(i in 1:length(namelist))
            namelist[[i]] =
              c(paste("+", namelist[[i]]),
                paste("-", plistC(plist[[i]])))
          do.call(c, namelist)
        } else{
          if(!is.null(names(plist))){
            namevec = names(plist)
            if(!is.null(pLoc))
              namevec = paste0(names(plist),
                " (", plist, ", ", pLoc[, "cols"], ")")
            paste("+", namevec)
          } else paste(plist, collapse = " ")
        }
      }
      cat(plistC(plist), sep = "\n")
    }

    attrLoc =
      function(plist, rows = NULL, cols = NULL){
        attr(plist, "Loc") = cbind(rows, cols)
        class(plist) = "plist"
        plist
      }

    TCRsink =
      function(ID, ...){
        if(exists("TCRunout", envir = .GlobalEnv)){
          varlist = list(...)
          names(varlist) = gsub(" ", "", as.character(match.call()[-(1:2)]))
          sink(TCRunout)
          for(i in 1:length(varlist)){
            cat("###TCR", ID, names(varlist)[i], "\n")
            print(varlist[[i]])
          }
          sink()
        }
      }
  }
```

}

Defines:

attrLoc, used in chunks 18–22.

TCRsink, used in chunks 11–14 and 18–24.

3 Identification

We separate the Identification functions into two groups.

Ident Main contains the main function that is called by the *Front End* function.

Ident Low Level contains supporting functions called by the *Ident Main* function.

11a $\langle \text{Identification } 11a \rangle \equiv$
 $\langle \text{Ident Main } 11b \rangle$
 $\langle \text{Ident Low Level } 14d \rangle$

3.1 Identification - Main Function

The purpose of the **IdentMain** function is to identify where in the file the data is found and where the accompanying labels are, while ignoring any extraneous information we do not want. It should also identify the presence of multiple tables in the same file.

It is intended for this procedure to involve a number of Identification algorithms that are used for a high degree of reliability and flexibility, but at this stage there is only a single algorithm.

The algorithms used are:

- Ident by Most Common Boundary.

Algorithms planned for the near future are:

- Ident by Runs.

The output of **IdentMain** will be a list containing two elements, **rows** and **cols**, each of which is a list containing these two elements:

label - a vector of the rows or columns where the labels are found.

data - a vector of the rows or columns where the data are found.

11b $\langle \text{Ident Main } 11b \rangle \equiv$
IdentMain =
 function(datamat){
 $\langle \text{Ident by Most Common Boundary } 12a \rangle$
 $\langle \text{Group Column Labels } 13b \rangle$
 TCRsink("IM", rowslist, colslist)
 list(rows = rowslist, cols = colslist)
 }
}

Defines:

IdentMain, used in chunk 23b.

Uses **TCRsink** 10.

Example values for **ToyExComplete.csv**

```
> rowslst
$label
[1] 1 2 3 4

$data
[1] 5 6 7 8 9 10 11 12 13 14

> colslst
$label
[1] 1 2

$data
$data[1]
[1] 4 5 6 7

$data[2]
[1] 8 9 10 11
```

3.1.1 Ident By Most Common Boundary

The `IdentMostCommonBoundary` Low Level function is used to find the most common start and end rows and columns (the boundary) to search for a block (rectangular region) of numbers, which is assumed to be our table of data.

- 12a *⟨Ident by Most Common Boundary 12a⟩*≡
 ⟨Get Non empty rows and cols 12b⟩
 ⟨Call Ident MostCommonBoundary 12c⟩
 ⟨Construct rowslst and colslst 13a⟩
- 12b *⟨Get Non empty rows and cols 12b⟩*≡
 rowNonempty = (1:nrow(datamat))[IdentNonEmpty(datamat, 1)]
 colNonempty = (1:ncol(datamat))[IdentNonEmpty(datamat, 2)]
Uses `IdentNonEmpty` 15a.
- 12c *⟨Call Ident MostCommonBoundary 12c⟩*≡
 rowData = `IdentMostCommonBoundary`(datamat, 2)
 colData = `IdentMostCommonBoundary`(datamat, 1)
 ## Temporary fix for first col being all numbers (e.g. years)
 if(colData[1] == 1) colData[1] = 2
 TCRsink("CIMCB", rowData, colData)
Uses `IdentMostCommonBoundary` 16a and `TCRsink` 10.

Example values for **ToyExComplete.csv**

```
> rowData
[1] 5 14

> colData
[1] 4 11
```

We construct the interim **rowslist** taking every non-empty row before the most common start of the numbers block (**rowData[1]**) and assigning these to the **label** region. The numbers block (which is bounded by **rowData[1]** and **rowData[2]**) is assigned to the **data** region. The interim **colslist** is constructed in the same manner.

```
13a <Construct rowslist and colslist 13a>≡
      rowslist = list(label = rowNonempty[rowNonempty < rowData[1]],
                      data = rowNonempty[(rowNonempty >= rowData[1]) &
                                         (rowNonempty <= rowData[2])])
      colslist = list(label = colNonempty[colNonempty < colData[1]],
                      data = colNonempty[(colNonempty >= colData[1]) &
                                         (colNonempty <= colData[2])])

      TCRsink("CRAC", rowslist, colslist)
Uses TCRsink 10.
```

Example values for **ToyExComplete.csv**

```
> rowslist
$label
[1] 1 2 3 4

$data
[1] 5 6 7 8 9 10 11 12 13 14

> colslist
$label
[1] 1 2

$data
[1] 4 5 6 7 8 9 10 11
```

3.1.2 Group Column Labels

We look for a repeating pattern in the column labels to handle cases of Misaligned Column Label (see Section 1.3.7).

```
13b <Group Column Labels 13b>≡
      <Generate Pattern vector 14a>
      <Take Largest Pattern 14b>
      <Group by Pattern 14c>
```

We loop through each row of the labels region and check for a pattern in either the contents of the cells (if they are all non-empty), or a pattern in which cells are empty (if any cells are empty), and store the results in `Patvec` (Pattern Vector).

```
14a <Generate Pattern vector 14a>≡
    curcol = colslist$data
    Patvec = NULL
    for(currow in rowslist$label){
        curlabel = datamat[currow, curcol]
        if(any(is.na(curlabel))){
            if(!all(is.na(curlabel))){
                Patvec = c(Patvec, IdentPattern(is.na(curlabel)))
            } else Patvec = c(Patvec, IdentPattern(curlabel))
        }
    }
    TCRsink("GPV", Patvec)
```

Uses `IdentPattern` 15b and `TCRsink` 10.

Example values for **ToyExComplete.csv**

```
> Patvec
[1] 4 1 4
```

Where multiple patterns are found, we assume the shorter patterns are patterns of families that are children to the parents of the largest pattern. Thus we always take the largest pattern found.

Some problems with this and next chunk, refer to BUG-ID 2.

```
14b <Take Largest Pattern 14b>≡
    Patvec = max(Patvec)

    If a pattern is found (NA = No patterns found), we group the columns into separate
    elements in a list and update the colslist. For easy handling later, irrespective of whether
    a pattern is found, colslist$data is a list.

14c <Group by Pattern 14c>≡
    if(!is.na(Patvec)){
        megacolnum = length(curcol)/Patvec
        megacollist = list()
        for(i in 1:megacolnum)
            megacollist =
                c(megacollist, list(curcol[1:Patvec + Patvec * (i - 1)]))
        colslist$data = megacollist
    } else colslist$data = list(curcol)
```

3.2 Identification - Low Level Functions

Here we discuss the low level functions that are called by the main Identification function. Each chunk corresponds to a separate low level function.

```
14d <Ident Low Level 14d>≡
    <Ident Non Empty 15a>
    <Ident Pattern 15b>
    <Ident Most Common Boundary 16a>
```

3.2.1 IdentNonEmpty

Given a matrix (`datamat`) and a margin (1 for rows, 2 for columns), return a vector giving the indices of non-empty rows or columns. Can specify a different empty identifying function (default `is.na`). Procedure:

1. Compute `isnonempty`, a logical vector about whether the rows or cols are not empty.
2. Use `which` on `isnonempty` to get indices.

```
15a  <Ident Non Empty 15a>≡
      IdentNonEmpty =
        function(datamat, margin, emptyident = is.na){
          isnonempty = apply(datamat, margin, function(x) !all(emptyident(x)))
          which(isnonempty)
        }
```

Defines:

`IdentNonEmpty`, used in chunks 12b and 23c.

3.2.2 IdentPattern

Attempt to discern a repeating pattern in `vec`, which can be a vector of any type (which is coerced to `character`). The returned value, `res` is either `NA` if no pattern is found. Or it is the grouping number for the repeating pattern, e.g.

- `vec = 1 1 1 1`, then `res = 1`
- `vec = 3 4 3 4`, then `res = 2`
- `vec = 1 2 3 1`, then `res = NA`

`IdentPattern` does this fairly efficiently by use of regular expressions. It combines the first `i` elements of `vec` and collapses this into a single string. A `grep` is then called on the entire `vec` that has been collapsed into a single string, checking to see if the entire string can be matched to some repeat of the aforementioned collapsed string of the first `i` elements.

For the moment it is possible for this to fail (and can even be intentionally gamed by providing something like `vec = c(12, 1, 2)`, which will return a pattern of 1, when it should return `NA`), so it should be changed to be more reliable (though less efficient).

```
15b  <Ident Pattern 15b>≡
      IdentPattern =
        function(vec){
          len = length(vec)
          res = NA
          for(i in 1:floor(len/2)){
            curseg = paste("^(", paste(vec[1:i], collapse = ""),
                          ")+$", sep = "")
            if(nchar(curseg) > 2559){
              warning("Label lengths too long for regular expressions to ",
                    "work. IdentPattern has been aborted. A pattern may ",
                    "exist but it cannot be found with the current ",
                    "algorithm.")
              break
            } else if(length(grep(curseg, paste(vec, collapse = ""))) > 0){
              res = i
              break
            }
          }
          res
        }
```

Defines:

`IdentPattern`, used in chunk 14a.

3.2.3 Ident Most Common Boundary

Search for the most common first and last rows/cols to identify a block (rectangular region) of numbers. Procedure:

1. Suppose `margin = 2`, then loop through each column and search for cells containing numbers.
2. Compute the first row with a number for each column (`nstarts`), and do the same for the last row (`nends`).
3. Return the most common first and last rows.

16a $\langle \text{Ident Most Common Boundary 16a} \rangle \equiv$

```
IdentMostCommonBoundary =
function(datamat, margin){
  isnumber = suppressWarnings(apply(datamat, margin,
    function(x) which(!is.na(as.numeric(x)))))
  nstarts = table(sapply(isnumber,
    function(x) if(length(x) > 0) min(x) else NA))
  nends = table(sapply(isnumber,
    function(x) if(length(x) > 0) max(x) else NA))
  as.numeric(names(c(which.max(nstarts), which.max(rev(nends))))))
}
```

Defines:

`IdentMostCommonBoundary`, used in chunk 12c.

4 Discern Parentage

We separate the Parentage functions into three groups.

Pare Front is a simple ‘front-end’ function that makes the appropriate first call to **PareMain**, and is the function called by the *Front End* function.

Pare Col A specialised front-end to **PareFront** that handles various fringe cases for Discerning Parentage for Column Labels, before eventually calling **PareFront**.

Pare Main contains the main function that recursively call itself until the all parentage is discerned.

Pare Low Level contains supporting functions called by the *Pare Main* function.

See the section on the main function (Section 4.2) for details on the purpose of the *Discern Parentage* stage.

16b $\langle \text{Discern Parentage 16b} \rangle \equiv$

```
 $\langle \text{Pare Front 17a} \rangle$ 
 $\langle \text{Pare Col 17b} \rangle$ 
 $\langle \text{Pare Main 18c} \rangle$ 
 $\langle \text{Pare Low Level 20b} \rangle$ 
```


4.0.4 plist

explanation.

4.1 Parentage - Front End Function

This front end function takes the `datamat` and constructs an initialising `plist` (Parentage List), which is used to make the first call to the main function.

```

17a  <Pare Front 17a>≡
      PareFront =
        function(datamat)
          PareMain(datamat = datamat, plist =
            list(rows = 1:nrow(datamat), cols = 1:ncol(datamat)))

```

Defines:

`PareFront`, used in chunks 18b and 23c.
Uses `PareMain` 18c.

4.1.1 Pare Col

The Parentage functions were initially designed to work with Row Labels only, however we can also use them to discern the parentage of Col Labels once we handle a few differences. We define a front-end to the front-end function called `PareCol` to do this.

```

17b  <Pare Col 17b>≡
      PareCol =
        function(datamat, datacols, labelrows){
          <Case Misaligned Col Parent 17c>
          datacols = unlist(datacols)
          <Collapse Fullrow Labels 18a>
          <Call Pare Front 18b>
        }

```

Defines:

`PareCol`, used in chunk 24.
Uses `datacols` 23b and `labelrows` 23b.

Unlike with Row Labels where the parents are reliably in the top-left corner of their family, Col Label parents are sometimes ‘misaligned’. In some cases this arises as Col Label parents might be centred over their family. Other times, it happens for no apparent logical reason.

Regardless of the cause, we need to correct for this. During the Identification phase, we Grouped the Column Labels based on repeating patterns. We use these groupings to identify the families, and if the parent is not found where it should be, we simply shift it over to the right place.

```

17c  <Case Misaligned Col Parent 17c>≡
      for(j in 1:length(datacols)){
        curfamily = datamat[labelrows, datacols[[j]], drop = FALSE]
        firstcolempty = is.na(curfamily[,1])
        if(any(firstcolempty))
          for(i in which(firstcolempty)){
            notempty = !is.na(curfamily[i,])
            if(sum(notempty) == 1){
              curfamily[i, 1] = curfamily[i, notempty]
              curfamily[i, notempty] = NA
            }
          }
        datamat[labelrows, datacols[[j]]] = curfamily
      }

```

Uses `datacols` 23b and `labelrows` 23b.

It is also quite common for Col Labels that are too wide to be physically split over multiple rows to manage the width of the labels. For now, we simply assume that any rows that are not full (and hence not parents) should all really be a single row of children, and collapse these.

```
18a  <Collapse Fullrow Labels 18a>≡
      notfullrows = apply(datamat[labelrows, datacols, drop = FALSE], 1,
        function(x) any(is.na(x)))
      if(any(diff(notfullrows) > 1))
        warning("full rows followed by not full rows!")
      pastestring = ""
      pasterows = which(!notfullrows)
      for(i in 1:length(pasterows))
        pastestring[i] = paste("datamat[labelrows[, pasterows[i],
          ], datacols]", sep = "")
      collapsedlabels = eval(parse(text = paste("paste(",
        paste(pastestring, collapse = ", "),
        ")", sep = "")))
```

Uses `datacols` 23b and `labelrows` 23b.

Once the above is handled, we can simply transpose our Col Labels and call `PareFront` on it.

```
18b  <Call Pare Front 18b>≡
      labeldatamat = rbind(datamat[labelrows[notfullrows], datacols],
        collapsedlabels)
      PareFront(t(labeldatamat))
```

Uses `datacols` 23b, `labelrows` 23b, and `PareFront` 17a.

4.2 Parentage - Main Function

The purpose of the `PareMain` function is to identify (or *Discern*, to better differentiate this stage from the *Identification* stage) hierarchical relationships (the *Parentage*) in the data.

It first makes various checks for fringe cases, then calls various detection algorithms (`Pare Low Levels`) to discern the parentage.

```
18c  <Pare Main 18c>≡
      PareMain =
        function(datamat, plist){
          <If only one column 18d>
          <If first column empty 19a>
          <If only one row 19b>
          <If first cell empty 19c>
          <Otherwise call Pare Low Levels 20a>
          class(res) = "plist"
          res
        }
      }
```

Defines:

`PareMain`, used in chunks 17a, 19a, and 20a.

If only one column is found then this means we are in the right-most column (or there was only one column to begin with), and hence the currently examined cells cannot be parents. We return the rows of these children as a vector, with names that correspond to their labels.

```
18d  <If only one column 18d>≡
      if(length(plist$cols) == 1){
        res = structure(plist$rows, .Names = datamat[plist$rows, plist$cols])
        res = attrLoc(res, cols = plist$col)
        TCRsink("IOOC", plist, res)
      }
```

Uses `attrLoc` 10 and `TCRsink` 10.

Example values for **ToyExComplete.csv**

```
> plist
$rows
[1] 3 4

$cols
[1] 2

> res
+ Row Child-Child1 (3, 2)
+ Row Child-Child2 (4, 2)
```

If the first column is found to be empty, then we will shift to the next column (which we know exists because we passed the check for only one column).

```
19a <If first column empty 19a>≡
      else if(all(is.na(datamat[plist$rows, plist$cols[1]]))){
        plist$cols = plist$cols[-1]
        res = PareMain(datamat, plist)
      }
```

Uses PareMain 18c.

If only one row is found then our row is a parent to itself (we know there are children in the row as we passed the check for only one column). We return the row as a numeric vector, nested in a list using correct parentage and names of the parentage within the row.

```
19b <If only one row 19b>≡
      else if(length(plist$rows) == 1){
        res = structure(plist$rows,
          .Names = datamat[plist$rows, plist$cols[length(plist$cols)]]
        res = attrLoc(res, cols = plist$cols[length(plist$cols)])
        for(i in (length(plist$cols) - 1):1){
          res = list(res)
          names(res) = datamat[plist$rows, plist$cols[i]]
          res = attrLoc(res, rows = plist$rows, cols = plist$cols[i])
        }
        TCRsink("IOOR", plist, res)
      }
```

Uses attrLoc 10 and TCRsink 10.

Example values for **ToyExComplete.csv**

```
> res
Never occurs
```

If the first cell is empty, after all previous checks, then this is an unrecognised format and we return a warning message.

```
19c <If first cell empty 19c>≡
      else if(is.na(datamat[plist$rows[1], plist$cols[1]])){
        warning("cell[1, 1] is empty")
        print(plist)
        res = NA
      }
```

If we have passed all the checks, we can then call the Low Level **Pare** functions. We first call **ByEmptyRight** to check for *empty right* situations. If none are found, it returns NA, in which case we try **ByEmptyBelow** instead.

We then loop through each element of the returned list and call the main function, as per the recursive nature of the function.

20a $\langle \text{Otherwise call Pare Low Levels 20a} \rangle \equiv$

```

else{
  res = PareByEmptyRight(datamat, plist)
  if(any(is.na(res)))
    res = PareByEmptyBelow(datamat, plist)
  for(i in 1:length(res))
    res[[i]] = PareMain(datamat, res[[i]])
  res
}

```

Uses **PareByEmptyBelow** 22a, **PareByEmptyRight** 20c, and **PareMain** 18c.

4.3 Parentage - Low Level Functions

The Low Level Parentage functions are called by the Main Parentage function. In particular, **ByEmptyRight** is always called first. Then **ByEmptyBelow** is called on the results of the above.

20b $\langle \text{Pare Low Level 20b} \rangle \equiv$

```

 $\langle \text{Pare By Empty Right 20c} \rangle$ 
 $\langle \text{Pare By Empty Below 22a} \rangle$ 

```

4.3.1 Pare By Empty Right

We check to see if we have an *empty right* situation. If we do not, we return NA.

20c $\langle \text{Pare By Empty Right 20c} \rangle \equiv$

```

PareByEmptyRight =
  function(datamat, plist)
  with(plist,
    if(all(is.na(datamat[rows[1], cols[-1]]))){
       $\langle \text{Check for Other Empty Rights 20d} \rangle$ 
       $\langle \text{Case Single Empty Right 20e} \rangle$ 
       $\langle \text{Case Multiple Empty Rights 21} \rangle$ 
      res
    } else NA)

```

Defines:

PareByEmptyRight, used in chunk 20a.

20d $\langle \text{Check for Other Empty Rights 20d} \rangle \equiv$

```

emptyrights = apply(datamat[rows, cols[-1], drop = FALSE], 1,
  function(x) all(is.na(x)))
rowemptyright = rows[emptyrights]

```

In the case of only a single empty right, we know there is only a single parent, which is the first line. Thus we take everything except the first line (which will be the rows of the children of this parent) and pass this through with correct naming.

20e $\langle \text{Case Single Empty Right 20e} \rangle \equiv$

```

if(length(rowemptyright) == 1){
  res = list(list(rows = rows[-1], cols = cols))
  names(res) = datamat[rows[1], cols[1]]
  res = attrLoc(res, rows = rows[1], cols = cols[1])
  TCRsink("CSER", res)
}

```

Uses **attrLoc** 10 and **TCRsink** 10.

1	New Zealand	
2	Auckland	
3	Accounting	Male
4		Female
5	Economics	Male
6		Female
7	Statistics	Male
8		Female
9	Wellington	
10	Economics	Male
11		Female
12	Statistics	Male
13		Female
14	Australia	
15	Sydney	
16	Accounting	Male
17		Female
18	Economics	Male
19		Female

Consider the toy example on the left.

In this case we do not have a simple `ByEmptyRight` structure. We have *super-parents* in the form of countries (New Zealand and Australia), and also *parents* in the form of cities (Auckland, Wellington and Sydney). To handle situations such as this, we must **Check for Other Empty Rights**.

If only a **Single Empty Right** is found, the situation is simple and we simply pass on the children of the single parent for the next iteration of `PareMain`.

However, if **Multiple Empty Rights** are found, we must identify the super-parents, and pass on the *children* of these super-parents (which would, in turn, contain parents and their children) as a list, to be handled in the next iteration of `PareMain`. In this example, we would have a list of length 2. The first element of the list would contain the `plist` with `rows` 2 to 13 (corresponding to the children of the New Zealand super-parent). The second element would have `rows` 15 to 19.

Example values for **ToyExComplete.csv**

```
> res
Never occurs
```

In the case of multiple empty rights, we first call `diff` to compute the gap in rows between the empty rights. If the value of `rowdiff[i]` is 1, this means there is no gap between the i^{th} `rowemptyright` and the $(i + 1)$ `rowemptyright`. This happens with *super-parents* as described in the example above. In this case, we gather these super-parents and ignore all other `rowemptyright` (the parents inside the super-parents will be handled at the next iteration of `PareMain`). Note, we assume there are never any super-super-parents (i.e. we can only handle a maximum of 2-levels of parentage in the same column).

Whether or not super-parents were identified, we compute the rows for the children of each parent (or super-parent) identified by `rowemptyright` and pass this through as a list, with correct naming.

```
21 <Case Multiple Empty Rights 21>≡
    else{
      rowdiff = diff(rowemptyright)
      if(any(rowdiff == 1))
        rowemptyright = rowemptyright[c(rowdiff == 1, FALSE)]

      rowstart = pmin(rowemptyright + 1, max(rows))
      rowend = c(pmax(rowemptyright[-1] - 1, min(rows)), max(rows))

      res = list()
      for(i in 1:length(rowstart))
        res[i] = list(list(rows = rowstart[i]:rowend[i], cols = cols))
        names(res) = datamat[rowemptyright, cols[1]]
        res = attrLoc(res, rows = rowemptyright, cols = cols[1])
        TCRsink("CMER", res)
    }
```

Uses `attrLoc` 10 and `TCRsink` 10.

Example values for **ToyExComplete.csv**

```
> res
+ Row Super-Parent (1, 1)
- + rows
- - 2 3 4 5 6 7 8 9 10
- + cols
- - 1 2
```

4.3.2 Pare By Empty Below

We check which cells are empty below (there should be at least 1 based on previous checks). Based on this, we compute the rows for the children of each parent and pass this through as a list, with correct naming.

22a $\langle \text{Pare By Empty Below 22a} \rangle \equiv$

```
PareByEmptyBelow =
  function(datamat, plist)
  with(plist, {
    emptybelow = is.na(datamat[rows, cols[1]])
    rowstart = rows[!emptybelow]
    rowend = c(rowstart[-1] - 1, max(rows))
    res = list()
    for(i in 1:length(rowstart))
      res[i] = list(list(rows = rowstart[i]:rowend[i], cols = cols[-1]))
    names(res) = datamat[rowstart, cols[1]]
    res = attrLoc(res, rows = rowstart, cols = cols[1])
    TCRsink("PBEB", res)
  })
```

Defines:

PareByEmptyBelow, used in chunk 20a.

Uses attrLoc 10 and TCRsink 10.

Example values for **ToyExComplete.csv**

```
> res
+ Row Child1 (3, 1)
- + rows
- - 3 4
- + cols
- - 2
+ Row Child2 (5, 1)
- + rows
- - 5 6
- + cols
- - 2
```

5 Reconstruction

We separate the Reconstruction functions into two groups.

Recons Main contains the main function that is called by the *Front End* function.

Recons Low Level contains supporting functions called by the *Recons Main* function.

22b $\langle \text{Reconstruction 22b} \rangle \equiv$

```
 $\langle \text{Recons Main 23a} \rangle$ 
 $\langle \text{Recons Low Level 25a} \rangle$ 
```

5.1 Reconstruction - Main Function

The `ReconsMain` function is, in a manner of speaking, the true `TableToLongForm` function, as it makes the calls to `IdentMain` and `PareFront`, in conjunction with its own `Recons Low Level` functions, to carry out the conversion.

23a *⟨Recons Main 23a⟩*≡
`ReconsMain =`
`function(datamat, IdentResult){`
`⟨Call Ident Main 23b⟩`
`⟨Reconstruct Row Labels 23c⟩`
`⟨Reconstruct Col Labels 24⟩`
`}`

Call `IdentMain` and assign them meaningful names for convenience, the `labelcols` being the columns where the labels can be found, etc. These should all be a `vector`, except `datacols` which is a `list`.

23b *⟨Call Ident Main 23b⟩*≡
`if(is.null(IdentResult))`
`IdentResult = IdentMain(datamat)`
`labelcols = IdentResult$cols$label`
`datacols = IdentResult$cols$data`
`labelrows = IdentResult$rows$label`
`datarows = IdentResult$rows$data`

Defines:

`labelcols`, used in chunk 23c.
`datacols`, used in chunks 17, 18, and 24.
`labelrows`, used in chunks 17, 18, and 24.
`datarows`, used in chunks 23c and 24.

Uses `IdentMain` 11b.

We create a subset of `datamat` that contains just the Row Labels. We also remove any columns that are completely empty (N.B. this may no longer be necessary, need to do some testing).

We call `PareFront` on our subset to discern the parentage of the Row Labels. We then use this to reconstruct the portion of the LongForm Dataframe relating to the Row Labels and assign this to `rowvecs`.

23c *⟨Reconstruct Row Labels 23c⟩*≡
`datamatRowLabels = datamat[datarows, labelcols, drop = FALSE]`
`datamatRowLabels = datamatRowLabels[,`
`IdentNonEmpty(datamatRowLabels, 2), drop = FALSE]`
`rowplist = PareFront(datamatRowLabels)`
`rowvecs = ReconsRowLabels(rowplist)`
`TCRsink("RRL", rowplist, rowvecs[1:4,])`

Defines:

`rowplist`, used in chunk 24.
`rowvecs`, used in chunks 24–27.

Uses `datarows` 23b, `IdentNonEmpty` 15a, `labelcols` 23b, `PareFront` 17a, `ReconsRowLabels` 25b, and `TCRsink` 10.

Example values for **ToyExComplete.csv**

```
> rowplist
+ Row Super-Parent (1, 1)
- + Row Parent1 (2, 1)
- - + Row Child1 (3, 1)
- - - + Row Child-Child1 (3, 2)
- - - + Row Child-Child2 (4, 2)
- - + Row Child2 (5, 1)
- - - + Row Child-Child1 (5, 2)
- - - + Row Child-Child2 (6, 2)
- + Row Parent2 (7, 1)
- - + Row Child1 (8, 1)
- - - + Row Child-Child1 (8, 2)
- - - + Row Child-Child2 (9, 2)
- - + Row Child2 (10, 1)
- - - + Row Child-Child2 (10, 2)

> rowvecs[1:4,]
[,1]      [,2]      [,3]      [,4]
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child2"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child2"
```

Due to the various fringe cases that exist with Col Labels, we cannot simply call **PareFront** and must instead call **PareCol**. We then create a subset of **datamat** that contains just the Col Labels and call **ReconsColLabels** which in truth reconstruct the entire LongForm Dataframe by making use of the **rowvecs** generated above.

```
24 <Reconstruct Col Labels 24>≡
    colplist = PareCol(datamat, datacols, labelrows)
    datamatColLabels = datamat[datarows[unlist(rowplist)], unlist(datacols)]
    res = ReconsColLabels(colplist, datamatColLabels, rowvecs)
    TCRsink("RCL", colplist, res[1:4,])
    list(datafr = res, datamat = datamat, IdentResult = IdentResult,
          rowplist = rowplist, colplist = colplist)
```

Uses **datacols** 23b, **datarows** 23b, **labelrows** 23b, **PareCol** 17b, **ReconsColLabels** 26a, **rowplist** 23c, **rowvecs** 23c, and **TCRSink** 10.

Example values for **ToyExComplete.csv**

```
> colplist
+ Column Parent1 (1, 2)
- + Column Child1 (1, 3)
- + Column Child2 (2, 3)
- + Column Child3 (3, 3)
- + Column Child4 (4, 3)
+ Column Parent2 (5, 2)
- + Column Child1 (5, 3)
- + Column Child2 (6, 3)
- + Column Child3 (7, 3)
- + Column Child4 (8, 3)

> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN
1 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child1
2 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child2
3 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child1
4 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child2
  Column Child1 Column Child2 Column Child3 Column Child4
1          12          22          32          42
2          13          23          33          43
3          14          24          34          44
4          15          25          35          45
```

5.2 Reconstruction - Low Level Functions

The Low Level Reconstruction functions are called by the Main Reconstruction function. In particular, **ReconsRowLabels** is always called first and its results are one of the arguments for **ReconsColLabels**, which finishes the reconstruction of the entire LongForm Dataframe.

25a $\langle \text{Recons Low Level 25a} \rangle \equiv$
 $\langle \text{Recons Row Labels 25b} \rangle$
 $\langle \text{Recons Column Labels 26a} \rangle$

5.2.1 Reconstruction - Row Labels

ReconsRowLabels iterates down the row parentage list (**plist**) recursively, extracting the names and using this to construct the columns of the finished LongForm Dataframe corresponding to the row labels. The final output is what was shown in the *Reconstruct Row Labels* chunk above as **rowvecs[1:4,]**.

25b $\langle \text{Recons Row Labels 25b} \rangle \equiv$

```
ReconsRowLabels =
function(plist)
if(is.list(plist)){
  rowvecs = as.list(names(plist))
  for(i in 1:length(rowvecs))
    rowvecs[[i]] = cbind(rowvecs[[i]], ReconsRowLabels(plist[[i]]))
  do.call(rbind, rowvecs)
} else as.matrix(names(plist))
```

Defines:

ReconsRowLabels, used in chunk 23c.
Uses rowvecs 23c.

Example values for **ToyExComplete.csv**

```
> rowvecs[1:4,]
[,1]      [,2]      [,3]      [,4]
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child2"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child2"
```

5.2.2 Reconstruction - Column Labels

As with the row labels, **ReconsColLabels** iterates down the column parentage list (**plist**) recursively. We also need to handle the parents differently from the lowest level child. The final output is what was shown in the *Reconstruct Col Labels* chunk above as **res[1:4,]**.

26a *<Recons Column Labels 26a>*≡

```
ReconsColLabels =
  function(plist, datamat, rowvecs){
    <Recons Col Parents 26b>
    <Recons Col Children 27>
    datfr
  }
```

Defines:

ReconsColLabels, used in chunks 24 and 26b.

Uses **rowvecs** 23c.

Example values for **ToyExComplete.csv**

```
> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN
1 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child1
2 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child2
3 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child1
4 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child2
  Column Child1 Column Child2 Column Child3 Column Child4
1          12          22          32          42
2          13          23          33          43
3          14          24          34          44
4          15          25          35          45
```

Any parents are used to construct additional columns of factors (the labels of the parents) for the LongForm Dataframe, which is attached to the portion previously constructed in **ReconsRowLabels**.

26b *<Recons Col Parents 26b>*≡

```
if(is.list(plist)){
  colvecs = as.list(names(plist))
  for(i in 1:length(colvecs)){
    colvecs[[i]] = cbind(colvecs[[i]],
      ReconsColLabels(plist[[i]], datamat, rowvecs))
    colnames(colvecs[[i]])[1] = "UNKNOWN"
  }
  datfr = do.call(rbind, colvecs)
}
```

Uses **ReconsColLabels** 26a and **rowvecs** 23c.

For the lowest level child, we extract the relevant ‘data bits’ from the original table and bind it to our Dataframe, using the lowest level child as the labels of these columns of data values.

```
27  <Recons Col Children 27>≡
      else{
        datbit = datamat[,plist]
        mode(datbit) = "numeric"
        ## Specify row.names to avoid annoying warnings
        datfr =
          cbind(as.data.frame(rowvecs, row.names = 1:nrow(rowvecs)), datbit)
        colnames(datfr) =
          c(rep("UNKNOWN", length = ncol(rowvecs)), names(plist))
      }
      Uses rowvecs 23c.
```

6 Chunk Index

<Back End 10)
<Call Ident Main 23b)
<Call Ident MostCommonBoundary 12c)
<Call Pare Front 18b)
<Case Misaligned Col Parent 17c)
<Case Multiple Empty Rights 21)
<Case Single Empty Right 20e)
<Check for Other Empty Rights 20d)
<Collapse Fullrow Labels 18a)
<Construct rowlist and colslst 13a)
<Discern Parentage 16b)
<document header 9a)
<Front End 9b)
<Generate Pattern vector 14a)
<Get Non empty rows and cols 12b)
<Group by Pattern 14c)
<Group Column Labels 13b)
<Ident by Most Common Boundary 12a)
<Ident Low Level 14d)
<Ident Main 11b)
<Ident Most Common Boundary 16a)
<Ident Non Empty 15a)
<Ident Pattern 15b)
<Identification 11a)
<If first cell empty 19c)
<If first column empty 19a)
<If only one column 18d)
<If only one row 19b)
<Otherwise call Pare Low Levels 20a)
<Pare By Empty Below 22a)
<Pare By Empty Right 20c)
<Pare Col 17b)
<Pare Front 17a)
<Pare Low Level 20b)
<Pare Main 18c)
<Recons Col Children 27)
<Recons Col Parents 26b)
<Recons Column Labels 26a)
<Recons Low Level 25a)
<Recons Main 23a)
<Recons Row Labels 25b)
<Reconstruct Col Labels 24)
<Reconstruct Row Labels 23c)
<Reconstruction 22b)
<TableToLongForm.R 8)
<Take Largest Pattern 14b)

7 Identifier Index

Numbers indicate the chunks in which the function appears. Underline indicates the chunk where the function is defined.

`attrLoc`: 10, 18d, 19b, 20e, 21, 22a
`datacols`: 17b, 17c, 18a, 18b, 23b, 24
`datarows`: 23b, 23c, 24

IdentMain: 11b, 23b
 IdentMostCommonBoundary: 12c, 16a
 IdentNonEmpty: 12b, 15a, 23c
 IdentPattern: 14a, 15b
 labelcols: 23b, 23c
 labelrows: 17b, 17c, 18a, 18b, 23b, 24
 PareByEmptyBelow: 20a, 22a
 PareByEmptyRight: 20a, 20c
 PareCol: 17b, 24
 PareFront: 17a, 18b, 23c
 PareMain: 17a, 18c, 19a, 20a
 ReconsCollLabels: 24, 26a, 26b
 ReconsRowLabels: 23c, 25b
 rowplist: 23c, 24
 rowvecs: 23c, 24, 25b, 26a, 26b, 27
 TCRsink: 10, 11b, 12c, 13a, 14a, 18d, 19b, 20e, 21, 22a, 23c, 24

References

- Davies, T., Bawa, Z., 2012a. The Journal of Community Informatics 8 (2).
URL <http://ci-journal.net/index.php/ciej/issue/view/41>
- Davies, T., Bawa, Z., 2012b. The Promises and Perils of Open Government Data (OGD). The Journal of Community Informatics 8 (2).
URL <http://ci-journal.net/index.php/ciej/article/view/929>
- New Zealand Qualifications Authority, 2012. Secondary school statistics.
URL <http://www.nzqa.govt.nz/studying-in-new-zealand/secondary-school-and-ncea/secondary-school>
- R Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
URL <http://www.R-project.org/>
- Ramsey, N., Sept 1994. Literate programming simplified. IEEE Software 11 (5), 97–105.
URL <http://www.cs.tufts.edu/~nr/noweb/>
- Statistics New Zealand, 2013. Infoshare.
URL <http://www.stats.govt.nz/infoshare/>

8 Appendix: TCRO

8.1 NZQAScholarships.xls

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Scholarship Entries and Results by Gender and Ethnicity (Broken down by Decile)												
2													
3				Decile 1-3									Decile 4-7
4			# of	#			#	# Not	#	#		# of	#
5	Results		Entries	Absent	SNA		Assessed	Achieved	Scholarship	Outstanding		Entries	Absent
6													
7	All Subjects		714.0	148.0	13.0		553.0	462.0	81.0	10.0		6482.0	1772.0
8													
9	Accounting		22.0	4.0	0.0		18.0	16.0	2.0	0.0		156.0	41.0
10	NZ Maori	Male	2.0	1.0	0.0		1.0	1.0	0.0	0.0		2.0	1.0
11		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		7.0	2.0
12	NZ European	Male	2.0	0.0	0.0		2.0	1.0	1.0	0.0		51.0	13.0
13		Female	3.0	0.0	0.0		3.0	2.0	1.0	0.0		44.0	12.0
14		Unknown	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
15	Pasifika Pacific	Male	2.0	0.0	0.0		2.0	2.0	0.0	0.0		3.0	0.0
16		Female	6.0	2.0	0.0		4.0	4.0	0.0	0.0		4.0	2.0
17	Asian	Male	5.0	0.0	0.0		5.0	5.0	0.0	0.0		29.0	4.0
18		Female	2.0	1.0	0.0		1.0	1.0	0.0	0.0		15.0	7.0
19	Other/Unspecified	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
20		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		1.0	0.0
21													
22	Agricultural & Horticulture		0.0	0.0	0.0		0.0	0.0	0.0	0.0		15.0	3.0
23	NZ Maori	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
24		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
25	NZ European	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		10.0	1.0
26		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		5.0	2.0
27		Unknown	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
28	Pasifika Pacific	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
29		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
30	Asian	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
31		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
32	Other/Unspecified	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
33		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
34													
35	Art History		6.0	1.0	0.0		5.0	5.0	0.0	0.0		87.0	23.0
36	NZ Maori	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		1.0	0.0
37		Female	2.0	0.0	0.0		2.0	2.0	0.0	0.0		4.0	1.0
38	NZ European	Male	1.0	1.0	0.0		0.0	0.0	0.0	0.0		7.0	2.0
39		Female	3.0	0.0	0.0		3.0	3.0	0.0	0.0		60.0	17.0
40		Unknown	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
41	Pasifika Pacific	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
42		Female	0.0	0.0	0.0		0.0	0.0	0.0	0.0		0.0	0.0
43	Asian	Male	0.0	0.0	0.0		0.0	0.0	0.0	0.0		3.0	0.0

```
> rowData
[1] 7 462
> colData
[1] 3 28
> rowslist
$label
[1] 1 3 4 5
```

```
$data
[1] 7 9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26
[19] 27 28 29 30 31 32 33 35 36 37 38 39 40 41 42 43 44 45
[37] 46 48 49 50 51 52 53 54 55 56 57 58 59 61 62 63 64 65
[55] 66 67 68 69 70 71 72 74 75 76 77 78 79 80 81 82 83 84
[73] 85 87 88 89 90 91 92 93 94 95 96 97 98 100 101 102 103 104
[91] 105 106 107 108 109 110 111 113 114 115 116 117 118 119 120 121 122 123
[109] 124 126 127 128 129 130 131 132 133 134 135 136 137 139 140 141 142 143
[127] 144 145 146 147 148 149 150 152 153 154 155 156 157 158 159 160 161 162
[145] 163 165 166 167 168 169 170 171 172 173 174 175 176 178 179 180 181 182
[163] 183 184 185 186 187 188 189 191 192 193 194 195 196 197 198 199 200 201
[181] 202 204 205 206 207 208 209 210 211 212 213 214 215 217 218 219 220 221
[199] 222 223 224 225 226 227 228 230 231 232 233 234 235 236 237 238 239 240
[217] 241 243 244 245 246 247 248 249 250 251 252 253 254 256 257 258 259 260
[235] 261 262 263 264 265 266 267 269 270 271 272 273 274 275 276 277 278 279
[253] 280 282 283 284 285 286 287 288 289 290 291 292 293 295 296 297 298 299
[271] 300 301 302 303 304 305 306 308 309 310 311 312 313 314 315 316 317 318
[289] 319 321 322 323 324 325 326 327 328 329 330 331 332 334 335 336 337 338
```

```
[307] 339 340 341 342 343 344 345 347 348 349 350 351 352 353 354 355 356 357
[325] 358 360 361 362 363 364 365 366 367 368 369 370 371 373 374 375 376 377
[343] 378 379 380 381 382 383 384 386 387 388 389 390 391 392 393 394 395 396
[361] 397 399 400 401 402 403 404 405 406 407 408 409 410 412 413 414 415 416
[379] 417 418 419 420 421 422 423 425 426 427 428 429 430 431 432 433 434 435
[397] 436 438 439 440 441 442 443 444 445 446 447 448 449 451 452 453 454 455
[415] 456 457 458 459 460 461 462
```

```
> colslst
$label
[1] 1 2
```

```
$data
[1] 3 4 5 7 8 9 10 12 13 14 16 17 18 19 21 22 23 25 26 27 28
```

```
> Patvec
[1] 7 7 7
> rowslist
$label
[1] 1 3 4 5
```

```
$data
[1] 7 9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26
[19] 27 28 29 30 31 32 33 35 36 37 38 39 40 41 42 43 44 45
[37] 46 48 49 50 51 52 53 54 55 56 57 58 59 61 62 63 64 65
[55] 66 67 68 69 70 71 72 74 75 76 77 78 79 80 81 82 83 84
[73] 85 87 88 89 90 91 92 93 94 95 96 97 98 100 101 102 103 104
[91] 105 106 107 108 109 110 111 113 114 115 116 117 118 119 120 121 122 123
[109] 124 126 127 128 129 130 131 132 133 134 135 136 137 139 140 141 142 143
[127] 144 145 146 147 148 149 150 152 153 154 155 156 157 158 159 160 161 162
[145] 163 165 166 167 168 169 170 171 172 173 174 175 176 178 179 180 181 182
[163] 183 184 185 186 187 188 189 191 192 193 194 195 196 197 198 199 200 201
[181] 202 204 205 206 207 208 209 210 211 212 213 214 215 217 218 219 220 221
[199] 222 223 224 225 226 227 228 230 231 232 233 234 235 236 237 238 239 240
[217] 241 243 244 245 246 247 248 249 250 251 252 253 254 256 257 258 259 260
[235] 261 262 263 264 265 266 267 269 270 271 272 273 274 275 276 277 278 279
[253] 280 282 283 284 285 286 287 288 289 290 291 292 293 295 296 297 298 299
[271] 300 301 302 303 304 305 306 308 309 310 311 312 313 314 315 316 317 318
[289] 319 321 322 323 324 325 326 327 328 329 330 331 332 334 335 336 337 338
[307] 339 340 341 342 343 344 345 347 348 349 350 351 352 353 354 355 356 357
[325] 358 360 361 362 363 364 365 366 367 368 369 370 371 373 374 375 376 377
[343] 378 379 380 381 382 383 384 386 387 388 389 390 391 392 393 394 395 396
[361] 397 399 400 401 402 403 404 405 406 407 408 409 410 412 413 414 415 416
[379] 417 418 419 420 421 422 423 425 426 427 428 429 430 431 432 433 434 435
[397] 436 438 439 440 441 442 443 444 445 446 447 448 449 451 452 453 454 455
[415] 456 457 458 459 460 461 462
```

```
> colslst
$label
[1] 1 2
```

```
$data
$data[1]
[1] 3 4 5 7 8 9 10
```

```
$data[2]
[1] 12 13 14 16 17 18 19
```



```
$data[3]
[1] 21 22 23 25 26 27 28
```

```
> res
+ All Subjects (1, 1)
- + rows
- - 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 3
- + cols
- - 1 2
> res
+ Accounting (2, 1)
- + rows
- - 3 4 5 6 7 8 9 10 11 12 13
- + cols
- - 1 2
+ Agricultural & Horticultural Science (14, 1)
- + rows
- - 15 16 17 18 19 20 21 22 23 24 25
- + cols
- - 1 2
+ Art History (26, 1)
- + rows
- - 27 28 29 30 31 32 33 34 35 36 37
- + cols
- - 1 2
+ Biology (38, 1)
- + rows
- - 39 40 41 42 43 44 45 46 47 48 49
- + cols
- - 1 2
+ Chemistry (50, 1)
- + rows
- - 51 52 53 54 55 56 57 58 59 60 61
- + cols
- - 1 2
+ Chinese (62, 1)
- + rows
- - 63 64 65 66 67 68 69 70 71 72 73
- + cols
- - 1 2
+ Classical Studies (74, 1)
- + rows
- - 75 76 77 78 79 80 81 82 83 84 85
- + cols
- - 1 2
+ Dance (86, 1)
- + rows
- - 87 88 89 90 91 92 93 94 95 96 97
- + cols
- - 1 2
+ Design and Visual Communication (98, 1)
- + rows
- - 99 100 101 102 103 104 105 106 107 108 109
- + cols
- - 1 2
```

```

+ Drama (110, 1)
- + rows
- - 111 112 113 114 115 116 117 118 119 120 121
- + cols
- - 1 2
+ Economics (122, 1)
- + rows
- - 123 124 125 126 127 128 129 130 131 132 133
- + cols
- - 1 2
+ English (134, 1)
- + rows
- - 135 136 137 138 139 140 141 142 143 144 145
- + cols
- - 1 2
+ French (146, 1)
- + rows
- - 147 148 149 150 151 152 153 154 155 156 157
- + cols
- - 1 2
+ Geography (158, 1)
- + rows
- - 159 160 161 162 163 164 165 166 167 168 169
- + cols
- - 1 2
+ German (170, 1)
- + rows
- - 171 172 173 174 175 176 177 178 179 180 181
- + cols
- - 1 2
+ Graphics (182, 1)
- + rows
- - 183 184 185 186 187 188 189 190 191 192 193
- + cols
- - 1 2
+ History (194, 1)
- + rows
- - 195 196 197 198 199 200 201 202 203 204 205
- + cols
- - 1 2
+ Japanese (206, 1)
- + rows
- - 207 208 209 210 211 212 213 214 215 216 217
- + cols
- - 1 2
+ Latin (218, 1)
- + rows
- - 219 220 221 222 223 224 225 226 227 228 229
- + cols
- - 1 2
+ Mathematics with Calculus (230, 1)
- + rows
- - 231 232 233 234 235 236 237 238 239 240 241
- + cols
- - 1 2
+ Media Studies (242, 1)
- + rows

```

```

- - 243 244 245 246 247 248 249 250 251 252 253
- + cols
- - 1 2
+ Music Studies (254, 1)
- + rows
- - 255 256 257 258 259 260 261 262 263 264 265
- + cols
- - 1 2
+ Painting (266, 1)
- + rows
- - 267 268 269 270 271 272 273 274 275 276 277
- + cols
- - 1 2
+ Photography (278, 1)
- + rows
- - 279 280 281 282 283 284 285 286 287 288 289
- + cols
- - 1 2
+ Physical Education (290, 1)
- + rows
- - 291 292 293 294 295 296 297 298 299 300 301
- + cols
- - 1 2
+ Physics (302, 1)
- + rows
- - 303 304 305 306 307 308 309 310 311 312 313
- + cols
- - 1 2
+ Printmaking (314, 1)
- + rows
- - 315 316 317 318 319 320 321 322 323 324 325
- + cols
- - 1 2
+ Samoan (326, 1)
- + rows
- - 327 328 329 330 331 332 333 334 335 336 337
- + cols
- - 1 2
+ Science (338, 1)
- + rows
- - 339 340 341 342 343 344 345 346 347 348 349
- + cols
- - 1 2
+ Sculpture (350, 1)
- + rows
- - 351 352 353 354 355 356 357 358 359 360 361
- + cols
- - 1 2
+ Spanish (362, 1)
- + rows
- - 363 364 365 366 367 368 369 370 371 372 373
- + cols
- - 1 2
+ Statistics and Modelling (374, 1)
- + rows
- - 375 376 377 378 379 380 381 382 383 384 385
- + cols

```

```

- - 1 2
+ Te Reo Maori (386, 1)
- + rows
- - 387 388 389 390 391 392 393 394 395 396 397
- + cols
- - 1 2
+ Te Reo Rangatira (398, 1)
- + rows
- - 399 400 401 402 403 404 405 406 407 408 409
- + cols
- - 1 2
+ Technology (410, 1)
- + rows
- - 411 412 413 414 415 416 417 418 419 420 421
- + cols
- - 1 2
> res
+ NZ Maori (3, 1)
- + rows
- - 3 4
- + cols
- - 2
+ NZ European (5, 1)
- + rows
- - 5 6 7
- + cols
- - 2
+ Pasifika Peoples (8, 1)
- + rows
- - 8 9
- + cols
- - 2
+ Asian (10, 1)
- + rows
- - 10 11
- + cols
- - 2
+ Other/Unspecified Ethnicity (12, 1)
- + rows
- - 12 13
- + cols
- - 2
> plist
$rows
[1] 3 4

$cols
[1] 2

> res
+ Male (3, 2)
+ Female (4, 2)
> plist
$rows
[1] 5 6 7

$cols

```

```

[1] 2

> res
+ Male (5, 2)
+ Female (6, 2)
+ Unknown (7, 2)
> plist
$rows
[1] 8 9

$cols
[1] 2

> res
+ Male (8, 2)
+ Female (9, 2)
> plist
$rows
[1] 10 11

$cols
[1] 2

> res
+ Male (10, 2)
+ Female (11, 2)
> plist
$rows
[1] 12 13

$cols
[1] 2

> res
+ Male (12, 2)
+ Female (13, 2)
> res
+ NZ Maori (15, 1)
- + rows
- - 15 16
- + cols
- - 2
+ NZ European (17, 1)
- + rows
- - 17 18 19
- + cols
- - 2
+ Pasifika Peoples (20, 1)
- + rows
- - 20 21
- + cols
- - 2
+ Asian (22, 1)
- + rows
- - 22 23
- + cols
- - 2

```

```

+ Other/Unspecified Ethnicity (24, 1)
- + rows
- - 24 25
- + cols
- - 2
> plist
$rows
[1] 15 16

$cols
[1] 2

> res
+ Male (15, 2)
+ Female (16, 2)
> plist
$rows
[1] 17 18 19

$cols
[1] 2

> res
+ Male (17, 2)
+ Female (18, 2)
+ Unknown (19, 2)
> plist
$rows
[1] 20 21

$cols
[1] 2

> res
+ Male (20, 2)
+ Female (21, 2)
> plist
$rows
[1] 22 23

$cols
[1] 2

> res
+ Male (22, 2)
+ Female (23, 2)
> plist
$rows
[1] 24 25

$cols
[1] 2

> res
+ Male (24, 2)
+ Female (25, 2)
> res

```

```

+ NZ Maori (27, 1)
- + rows
- - 27 28
- + cols
- - 2
+ NZ European (29, 1)
- + rows
- - 29 30 31
- + cols
- - 2
+ Pasifika Peoples (32, 1)
- + rows
- - 32 33
- + cols
- - 2
+ Asian (34, 1)
- + rows
- - 34 35
- + cols
- - 2
+ Other/Unspecified Ethnicity (36, 1)
- + rows
- - 36 37
- + cols
- - 2
> plist
$rows
[1] 27 28

$cols
[1] 2

> res
+ Male (27, 2)
+ Female (28, 2)
> plist
$rows
[1] 29 30 31

$cols
[1] 2

> res
+ Male (29, 2)
+ Female (30, 2)
+ Unknown (31, 2)
> plist
$rows
[1] 32 33

$cols
[1] 2

> res
+ Male (32, 2)
+ Female (33, 2)
> plist

```

```

$rows
[1] 34 35

$cols
[1] 2

> res
+ Male (34, 2)
+ Female (35, 2)
> plist
$rows
[1] 36 37

$cols
[1] 2

> res
+ Male (36, 2)
+ Female (37, 2)
> res
+ NZ Maori (39, 1)
- + rows
- - 39 40
- + cols
- - 2
+ NZ European (41, 1)
- + rows
- - 41 42 43
- + cols
- - 2
+ Pasifika Peoples (44, 1)
- + rows
- - 44 45
- + cols
- - 2
+ Asian (46, 1)
- + rows
- - 46 47
- + cols
- - 2
+ Other/Unspecified Ethnicity (48, 1)
- + rows
- - 48 49
- + cols
- - 2
> plist
$rows
[1] 39 40

$cols
[1] 2

> res
+ Male (39, 2)
+ Female (40, 2)
> plist
$rows

```



```

[1] 41 42 43

$cols
[1] 2

> res
+ Male (41, 2)
+ Female (42, 2)
+ Unknown (43, 2)
> plist
$rows
[1] 44 45

$cols
[1] 2

> res
+ Male (44, 2)
+ Female (45, 2)
> plist
$rows
[1] 46 47

$cols
[1] 2

> res
+ Male (46, 2)
+ Female (47, 2)
> plist
$rows
[1] 48 49

$cols
[1] 2

> res
+ Male (48, 2)
+ Female (49, 2)
> res
+ NZ Maori (51, 1)
- + rows
- - 51 52
- + cols
- - 2
+ NZ European (53, 1)
- + rows
- - 53 54 55
- + cols
- - 2
+ Pasifika Peoples (56, 1)
- + rows
- - 56 57
- + cols
- - 2
+ Asian (58, 1)
- + rows

```

```

- - 58 59
- + cols
- - 2
+ Other/Unspecified Ethnicity (60, 1)
- + rows
- - 60 61
- + cols
- - 2
> plist
$rows
[1] 51 52

$cols
[1] 2

> res
+ Male (51, 2)
+ Female (52, 2)
> plist
$rows
[1] 53 54 55

$cols
[1] 2

> res
+ Male (53, 2)
+ Female (54, 2)
+ Unknown (55, 2)
> plist
$rows
[1] 56 57

$cols
[1] 2

> res
+ Male (56, 2)
+ Female (57, 2)
> plist
$rows
[1] 58 59

$cols
[1] 2

> res
+ Male (58, 2)
+ Female (59, 2)
> plist
$rows
[1] 60 61

$cols
[1] 2

> res

```

```

+ Male (60, 2)
+ Female (61, 2)
> res
+ NZ Maori (63, 1)
- + rows
- - 63 64
- + cols
- - 2
+ NZ European (65, 1)
- + rows
- - 65 66 67
- + cols
- - 2
+ Pasifika Peoples (68, 1)
- + rows
- - 68 69
- + cols
- - 2
+ Asian (70, 1)
- + rows
- - 70 71
- + cols
- - 2
+ Other/Unspecified Ethnicity (72, 1)
- + rows
- - 72 73
- + cols
- - 2
> plist
$rows
[1] 63 64

$cols
[1] 2

> res
+ Male (63, 2)
+ Female (64, 2)
> plist
$rows
[1] 65 66 67

$cols
[1] 2

> res
+ Male (65, 2)
+ Female (66, 2)
+ Unknown (67, 2)
> plist
$rows
[1] 68 69

$cols
[1] 2

> res

```

```

+ Male (68, 2)
+ Female (69, 2)
> plist
$rows
[1] 70 71

$cols
[1] 2

> res
+ Male (70, 2)
+ Female (71, 2)
> plist
$rows
[1] 72 73

$cols
[1] 2

> res
+ Male (72, 2)
+ Female (73, 2)
> res
+ NZ Maori (75, 1)
- + rows
- - 75 76
- + cols
- - 2
+ NZ European (77, 1)
- + rows
- - 77 78 79
- + cols
- - 2
+ Pasifika Peoples (80, 1)
- + rows
- - 80 81
- + cols
- - 2
+ Asian (82, 1)
- + rows
- - 82 83
- + cols
- - 2
+ Other/Unspecified Ethnicity (84, 1)
- + rows
- - 84 85
- + cols
- - 2
> plist
$rows
[1] 75 76

$cols
[1] 2

> res
+ Male (75, 2)

```

```

+ Female (76, 2)
> plist
$rows
[1] 77 78 79

$cols
[1] 2

> res
+ Male (77, 2)
+ Female (78, 2)
+ Unknown (79, 2)
> plist
$rows
[1] 80 81

$cols
[1] 2

> res
+ Male (80, 2)
+ Female (81, 2)
> plist
$rows
[1] 82 83

$cols
[1] 2

> res
+ Male (82, 2)
+ Female (83, 2)
> plist
$rows
[1] 84 85

$cols
[1] 2

> res
+ Male (84, 2)
+ Female (85, 2)
> res
+ NZ Maori (87, 1)
- + rows
- - 87 88
- + cols
- - 2
+ NZ European (89, 1)
- + rows
- - 89 90 91
- + cols
- - 2
+ Pasifika Peoples (92, 1)
- + rows
- - 92 93
- + cols

```

```

- - 2
+ Asian (94, 1)
- + rows
- - 94 95
- + cols
- - 2
+ Other/Unspecified Ethnicity (96, 1)
- + rows
- - 96 97
- + cols
- - 2
> plist
$rows
[1] 87 88

$cols
[1] 2

> res
+ Male (87, 2)
+ Female (88, 2)
> plist
$rows
[1] 89 90 91

$cols
[1] 2

> res
+ Male (89, 2)
+ Female (90, 2)
+ Unknown (91, 2)
> plist
$rows
[1] 92 93

$cols
[1] 2

> res
+ Male (92, 2)
+ Female (93, 2)
> plist
$rows
[1] 94 95

$cols
[1] 2

> res
+ Male (94, 2)
+ Female (95, 2)
> plist
$rows
[1] 96 97

$cols

```

```

[1] 2

> res
+ Male (96, 2)
+ Female (97, 2)
> res
+ NZ Maori (99, 1)
- + rows
- - 99 100
- + cols
- - 2
+ NZ European (101, 1)
- + rows
- - 101 102 103
- + cols
- - 2
+ Pasifika Peoples (104, 1)
- + rows
- - 104 105
- + cols
- - 2
+ Asian (106, 1)
- + rows
- - 106 107
- + cols
- - 2
+ Other/Unspecified Ethnicity (108, 1)
- + rows
- - 108 109
- + cols
- - 2
> plist
$rows
[1] 99 100

$cols
[1] 2

> res
+ Male (99, 2)
+ Female (100, 2)
> plist
$rows
[1] 101 102 103

$cols
[1] 2

> res
+ Male (101, 2)
+ Female (102, 2)
+ Unknown (103, 2)
> plist
$rows
[1] 104 105

$cols

```

```

[1] 2

> res
+ Male (104, 2)
+ Female (105, 2)
> plist
$rows
[1] 106 107

$cols
[1] 2

> res
+ Male (106, 2)
+ Female (107, 2)
> plist
$rows
[1] 108 109

$cols
[1] 2

> res
+ Male (108, 2)
+ Female (109, 2)
> res
+ NZ Maori (111, 1)
- + rows
- - 111 112
- + cols
- - 2
+ NZ European (113, 1)
- + rows
- - 113 114 115
- + cols
- - 2
+ Pasifika Peoples (116, 1)
- + rows
- - 116 117
- + cols
- - 2
+ Asian (118, 1)
- + rows
- - 118 119
- + cols
- - 2
+ Other/Unspecified Ethnicity (120, 1)
- + rows
- - 120 121
- + cols
- - 2
> plist
$rows
[1] 111 112

$cols
[1] 2

```



```

> res
+ Male (111, 2)
+ Female (112, 2)
> plist
$rows
[1] 113 114 115

$cols
[1] 2

> res
+ Male (113, 2)
+ Female (114, 2)
+ Unknown (115, 2)
> plist
$rows
[1] 116 117

$cols
[1] 2

> res
+ Male (116, 2)
+ Female (117, 2)
> plist
$rows
[1] 118 119

$cols
[1] 2

> res
+ Male (118, 2)
+ Female (119, 2)
> plist
$rows
[1] 120 121

$cols
[1] 2

> res
+ Male (120, 2)
+ Female (121, 2)
> res
+ NZ Maori (123, 1)
- + rows
- - 123 124
- + cols
- - 2
+ NZ European (125, 1)
- + rows
- - 125 126 127
- + cols
- - 2
+ Pasifika Peoples (128, 1)

```

```

- + rows
- - 128 129
- + cols
- - 2
+ Asian (130, 1)
- + rows
- - 130 131
- + cols
- - 2
+ Other/Unspecified Ethnicity (132, 1)
- + rows
- - 132 133
- + cols
- - 2
> plist
$rows
[1] 123 124

$cols
[1] 2

> res
+ Male (123, 2)
+ Female (124, 2)
> plist
$rows
[1] 125 126 127

$cols
[1] 2

> res
+ Male (125, 2)
+ Female (126, 2)
+ Unknown (127, 2)
> plist
$rows
[1] 128 129

$cols
[1] 2

> res
+ Male (128, 2)
+ Female (129, 2)
> plist
$rows
[1] 130 131

$cols
[1] 2

> res
+ Male (130, 2)
+ Female (131, 2)
> plist
$rows

```

```

[1] 132 133

$cols
[1] 2

> res
+ Male (132, 2)
+ Female (133, 2)
> res
+ NZ Maori (135, 1)
- + rows
- - 135 136
- + cols
- - 2
+ NZ European (137, 1)
- + rows
- - 137 138 139
- + cols
- - 2
+ Pasifika Peoples (140, 1)
- + rows
- - 140 141
- + cols
- - 2
+ Asian (142, 1)
- + rows
- - 142 143
- + cols
- - 2
+ Other/Unspecified Ethnicity (144, 1)
- + rows
- - 144 145
- + cols
- - 2
> plist
$rows
[1] 135 136

$cols
[1] 2

> res
+ Male (135, 2)
+ Female (136, 2)
> plist
$rows
[1] 137 138 139

$cols
[1] 2

> res
+ Male (137, 2)
+ Female (138, 2)
+ Unknown (139, 2)
> plist
$rows

```

```

[1] 140 141

$cols
[1] 2

> res
+ Male (140, 2)
+ Female (141, 2)
> plist
$rows
[1] 142 143

$cols
[1] 2

> res
+ Male (142, 2)
+ Female (143, 2)
> plist
$rows
[1] 144 145

$cols
[1] 2

> res
+ Male (144, 2)
+ Female (145, 2)
> res
+ NZ Maori (147, 1)
- + rows
- - 147 148
- + cols
- - 2
+ NZ European (149, 1)
- + rows
- - 149 150 151
- + cols
- - 2
+ Pasifika Peoples (152, 1)
- + rows
- - 152 153
- + cols
- - 2
+ Asian (154, 1)
- + rows
- - 154 155
- + cols
- - 2
+ Other/Unspecified Ethnicity (156, 1)
- + rows
- - 156 157
- + cols
- - 2
> plist
$rows
[1] 147 148

```

```

$cols
[1] 2

> res
+ Male (147, 2)
+ Female (148, 2)
> plist
$rows
[1] 149 150 151

$cols
[1] 2

> res
+ Male (149, 2)
+ Female (150, 2)
+ Unknown (151, 2)
> plist
$rows
[1] 152 153

$cols
[1] 2

> res
+ Male (152, 2)
+ Female (153, 2)
> plist
$rows
[1] 154 155

$cols
[1] 2

> res
+ Male (154, 2)
+ Female (155, 2)
> plist
$rows
[1] 156 157

$cols
[1] 2

> res
+ Male (156, 2)
+ Female (157, 2)
> res
+ NZ Maori (159, 1)
- + rows
- - 159 160
- + cols
- - 2
+ NZ European (161, 1)
- + rows
- - 161 162 163

```

```

- + cols
- - 2
+ Pasifika Peoples (164, 1)
- + rows
- - 164 165
- + cols
- - 2
+ Asian (166, 1)
- + rows
- - 166 167
- + cols
- - 2
+ Other/Unspecified Ethnicity (168, 1)
- + rows
- - 168 169
- + cols
- - 2
> plist
$rows
[1] 159 160

$cols
[1] 2

> res
+ Male (159, 2)
+ Female (160, 2)
> plist
$rows
[1] 161 162 163

$cols
[1] 2

> res
+ Male (161, 2)
+ Female (162, 2)
+ Unknown (163, 2)
> plist
$rows
[1] 164 165

$cols
[1] 2

> res
+ Male (164, 2)
+ Female (165, 2)
> plist
$rows
[1] 166 167

$cols
[1] 2

> res
+ Male (166, 2)

```

```

+ Female (167, 2)
> plist
$rows
[1] 168 169

$cols
[1] 2

> res
+ Male (168, 2)
+ Female (169, 2)
> res
+ NZ Maori (171, 1)
- + rows
- - 171 172
- + cols
- - 2
+ NZ European (173, 1)
- + rows
- - 173 174 175
- + cols
- - 2
+ Pasifika Peoples (176, 1)
- + rows
- - 176 177
- + cols
- - 2
+ Asian (178, 1)
- + rows
- - 178 179
- + cols
- - 2
+ Other/Unspecified Ethnicity (180, 1)
- + rows
- - 180 181
- + cols
- - 2
> plist
$rows
[1] 171 172

$cols
[1] 2

> res
+ Male (171, 2)
+ Female (172, 2)
> plist
$rows
[1] 173 174 175

$cols
[1] 2

> res
+ Male (173, 2)
+ Female (174, 2)

```

```

+ Unknown (175, 2)
> plist
$rows
[1] 176 177

$cols
[1] 2

> res
+ Male (176, 2)
+ Female (177, 2)
> plist
$rows
[1] 178 179

$cols
[1] 2

> res
+ Male (178, 2)
+ Female (179, 2)
> plist
$rows
[1] 180 181

$cols
[1] 2

> res
+ Male (180, 2)
+ Female (181, 2)
> res
+ NZ Maori (183, 1)
- + rows
- - 183 184
- + cols
- - 2
+ NZ European (185, 1)
- + rows
- - 185 186 187
- + cols
- - 2
+ Pasifika Peoples (188, 1)
- + rows
- - 188 189
- + cols
- - 2
+ Asian (190, 1)
- + rows
- - 190 191
- + cols
- - 2
+ Other/Unspecified Ethnicity (192, 1)
- + rows
- - 192 193
- + cols
- - 2

```



```

> plist
$rows
[1] 183 184

$cols
[1] 2

> res
+ Male (183, 2)
+ Female (184, 2)
> plist
$rows
[1] 185 186 187

$cols
[1] 2

> res
+ Male (185, 2)
+ Female (186, 2)
+ Unknown (187, 2)
> plist
$rows
[1] 188 189

$cols
[1] 2

> res
+ Male (188, 2)
+ Female (189, 2)
> plist
$rows
[1] 190 191

$cols
[1] 2

> res
+ Male (190, 2)
+ Female (191, 2)
> plist
$rows
[1] 192 193

$cols
[1] 2

> res
+ Male (192, 2)
+ Female (193, 2)
> res
+ NZ Maori (195, 1)
- + rows
- - 195 196
- + cols
- - 2

```

```

+ NZ European (197, 1)
- + rows
- - 197 198 199
- + cols
- - 2
+ Pasifika Peoples (200, 1)
- + rows
- - 200 201
- + cols
- - 2
+ Asian (202, 1)
- + rows
- - 202 203
- + cols
- - 2
+ Other/Unspecified Ethnicity (204, 1)
- + rows
- - 204 205
- + cols
- - 2
> plist
$rows
[1] 195 196

$cols
[1] 2

> res
+ Male (195, 2)
+ Female (196, 2)
> plist
$rows
[1] 197 198 199

$cols
[1] 2

> res
+ Male (197, 2)
+ Female (198, 2)
+ Unknown (199, 2)
> plist
$rows
[1] 200 201

$cols
[1] 2

> res
+ Male (200, 2)
+ Female (201, 2)
> plist
$rows
[1] 202 203

$cols
[1] 2

```

```

> res
+ Male (202, 2)
+ Female (203, 2)
> plist
$rows
[1] 204 205

$cols
[1] 2

> res
+ Male (204, 2)
+ Female (205, 2)
> res
+ NZ Maori (207, 1)
- + rows
- - 207 208
- + cols
- - 2
+ NZ European (209, 1)
- + rows
- - 209 210 211
- + cols
- - 2
+ Pasifika Peoples (212, 1)
- + rows
- - 212 213
- + cols
- - 2
+ Asian (214, 1)
- + rows
- - 214 215
- + cols
- - 2
+ Other/Unspecified Ethnicity (216, 1)
- + rows
- - 216 217
- + cols
- - 2
> plist
$rows
[1] 207 208

$cols
[1] 2

> res
+ Male (207, 2)
+ Female (208, 2)
> plist
$rows
[1] 209 210 211

$cols
[1] 2

```

```

> res
+ Male (209, 2)
+ Female (210, 2)
+ Unknown (211, 2)
> plist
$rows
[1] 212 213

```

```

$cols
[1] 2

```

```

> res
+ Male (212, 2)
+ Female (213, 2)
> plist
$rows
[1] 214 215

```

```

$cols
[1] 2

```

```

> res
+ Male (214, 2)
+ Female (215, 2)
> plist
$rows
[1] 216 217

```

```

$cols
[1] 2

```

```

> res
+ Male (216, 2)
+ Female (217, 2)
> res
+ NZ Maori (219, 1)
- + rows
- - 219 220
- + cols
- - 2
+ NZ European (221, 1)
- + rows
- - 221 222 223
- + cols
- - 2
+ Pasifika Peoples (224, 1)
- + rows
- - 224 225
- + cols
- - 2
+ Asian (226, 1)
- + rows
- - 226 227
- + cols
- - 2
+ Other/Unspecified Ethnicity (228, 1)
- + rows

```

```

- - 228 229
- + cols
- - 2
> plist
$rows
[1] 219 220

$cols
[1] 2

> res
+ Male (219, 2)
+ Female (220, 2)
> plist
$rows
[1] 221 222 223

$cols
[1] 2

> res
+ Male (221, 2)
+ Female (222, 2)
+ Unknown (223, 2)
> plist
$rows
[1] 224 225

$cols
[1] 2

> res
+ Male (224, 2)
+ Female (225, 2)
> plist
$rows
[1] 226 227

$cols
[1] 2

> res
+ Male (226, 2)
+ Female (227, 2)
> plist
$rows
[1] 228 229

$cols
[1] 2

> res
+ Male (228, 2)
+ Female (229, 2)
> res
+ NZ Maori (231, 1)
- + rows

```

```

- - 231 232
- + cols
- - 2
+ NZ European (233, 1)
- + rows
- - 233 234 235
- + cols
- - 2
+ Pasifika Peoples (236, 1)
- + rows
- - 236 237
- + cols
- - 2
+ Asian (238, 1)
- + rows
- - 238 239
- + cols
- - 2
+ Other/Unspecified Ethnicity (240, 1)
- + rows
- - 240 241
- + cols
- - 2
> plist
$rows
[1] 231 232

$cols
[1] 2

> res
+ Male (231, 2)
+ Female (232, 2)
> plist
$rows
[1] 233 234 235

$cols
[1] 2

> res
+ Male (233, 2)
+ Female (234, 2)
+ Unknown (235, 2)
> plist
$rows
[1] 236 237

$cols
[1] 2

> res
+ Male (236, 2)
+ Female (237, 2)
> plist
$rows
[1] 238 239

```

```

$cols
[1] 2

> res
+ Male (238, 2)
+ Female (239, 2)
> plist
$rows
[1] 240 241

$cols
[1] 2

> res
+ Male (240, 2)
+ Female (241, 2)
> res
+ NZ Maori (243, 1)
- + rows
- - 243 244
- + cols
- - 2
+ NZ European (245, 1)
- + rows
- - 245 246 247
- + cols
- - 2
+ Pasifika Peoples (248, 1)
- + rows
- - 248 249
- + cols
- - 2
+ Asian (250, 1)
- + rows
- - 250 251
- + cols
- - 2
+ Other/Unspecified Ethnicity (252, 1)
- + rows
- - 252 253
- + cols
- - 2
> plist
$rows
[1] 243 244

$cols
[1] 2

> res
+ Male (243, 2)
+ Female (244, 2)
> plist
$rows
[1] 245 246 247

```

```

$cols
[1] 2

> res
+ Male (245, 2)
+ Female (246, 2)
+ Unknown (247, 2)
> plist
$rows
[1] 248 249

$cols
[1] 2

> res
+ Male (248, 2)
+ Female (249, 2)
> plist
$rows
[1] 250 251

$cols
[1] 2

> res
+ Male (250, 2)
+ Female (251, 2)
> plist
$rows
[1] 252 253

$cols
[1] 2

> res
+ Male (252, 2)
+ Female (253, 2)
> res
+ NZ Maori (255, 1)
- + rows
- - 255 256
- + cols
- - 2
+ NZ European (257, 1)
- + rows
- - 257 258 259
- + cols
- - 2
+ Pasifika Peoples (260, 1)
- + rows
- - 260 261
- + cols
- - 2
+ Asian (262, 1)
- + rows
- - 262 263
- + cols

```



```

- - 2
+ Other/Unspecified Ethnicity (264, 1)
- + rows
- - 264 265
- + cols
- - 2
> plist
$rows
[1] 255 256

$cols
[1] 2

> res
+ Male (255, 2)
+ Female (256, 2)
> plist
$rows
[1] 257 258 259

$cols
[1] 2

> res
+ Male (257, 2)
+ Female (258, 2)
+ Unknown (259, 2)
> plist
$rows
[1] 260 261

$cols
[1] 2

> res
+ Male (260, 2)
+ Female (261, 2)
> plist
$rows
[1] 262 263

$cols
[1] 2

> res
+ Male (262, 2)
+ Female (263, 2)
> plist
$rows
[1] 264 265

$cols
[1] 2

> res
+ Male (264, 2)
+ Female (265, 2)

```

```

> res
+ NZ Maori (267, 1)
- + rows
- - 267 268
- + cols
- - 2
+ NZ European (269, 1)
- + rows
- - 269 270 271
- + cols
- - 2
+ Pasifika Peoples (272, 1)
- + rows
- - 272 273
- + cols
- - 2
+ Asian (274, 1)
- + rows
- - 274 275
- + cols
- - 2
+ Other/Unspecified Ethnicity (276, 1)
- + rows
- - 276 277
- + cols
- - 2
> plist
$rows
[1] 267 268

$cols
[1] 2

> res
+ Male (267, 2)
+ Female (268, 2)
> plist
$rows
[1] 269 270 271

$cols
[1] 2

> res
+ Male (269, 2)
+ Female (270, 2)
+ Unknown (271, 2)
> plist
$rows
[1] 272 273

$cols
[1] 2

> res
+ Male (272, 2)
+ Female (273, 2)

```

```

> plist
$rows
[1] 274 275

$cols
[1] 2

> res
+ Male (274, 2)
+ Female (275, 2)
> plist
$rows
[1] 276 277

$cols
[1] 2

> res
+ Male (276, 2)
+ Female (277, 2)
> res
+ NZ Maori (279, 1)
- + rows
- - 279 280
- + cols
- - 2
+ NZ European (281, 1)
- + rows
- - 281 282 283
- + cols
- - 2
+ Pasifika Peoples (284, 1)
- + rows
- - 284 285
- + cols
- - 2
+ Asian (286, 1)
- + rows
- - 286 287
- + cols
- - 2
+ Other/Unspecified Ethnicity (288, 1)
- + rows
- - 288 289
- + cols
- - 2
> plist
$rows
[1] 279 280

$cols
[1] 2

> res
+ Male (279, 2)
+ Female (280, 2)
> plist

```

```

$rows
[1] 281 282 283

$cols
[1] 2

> res
+ Male (281, 2)
+ Female (282, 2)
+ Unknown (283, 2)
> plist
$rows
[1] 284 285

$cols
[1] 2

> res
+ Male (284, 2)
+ Female (285, 2)
> plist
$rows
[1] 286 287

$cols
[1] 2

> res
+ Male (286, 2)
+ Female (287, 2)
> plist
$rows
[1] 288 289

$cols
[1] 2

> res
+ Male (288, 2)
+ Female (289, 2)
> res
+ NZ Maori (291, 1)
- + rows
- - 291 292
- + cols
- - 2
+ NZ European (293, 1)
- + rows
- - 293 294 295
- + cols
- - 2
+ Pasifika Peoples (296, 1)
- + rows
- - 296 297
- + cols
- - 2
+ Asian (298, 1)

```

```

- + rows
- - 298 299
- + cols
- - 2
+ Other/Unspecified Ethnicity (300, 1)
- + rows
- - 300 301
- + cols
- - 2
> plist
$rows
[1] 291 292

$cols
[1] 2

> res
+ Male (291, 2)
+ Female (292, 2)
> plist
$rows
[1] 293 294 295

$cols
[1] 2

> res
+ Male (293, 2)
+ Female (294, 2)
+ Unknown (295, 2)
> plist
$rows
[1] 296 297

$cols
[1] 2

> res
+ Male (296, 2)
+ Female (297, 2)
> plist
$rows
[1] 298 299

$cols
[1] 2

> res
+ Male (298, 2)
+ Female (299, 2)
> plist
$rows
[1] 300 301

$cols
[1] 2

```

```

> res
+ Male (300, 2)
+ Female (301, 2)
> res
+ NZ Maori (303, 1)
- + rows
- - 303 304
- + cols
- - 2
+ NZ European (305, 1)
- + rows
- - 305 306 307
- + cols
- - 2
+ Pasifika Peoples (308, 1)
- + rows
- - 308 309
- + cols
- - 2
+ Asian (310, 1)
- + rows
- - 310 311
- + cols
- - 2
+ Other/Unspecified Ethnicity (312, 1)
- + rows
- - 312 313
- + cols
- - 2
> plist
$rows
[1] 303 304

$cols
[1] 2

> res
+ Male (303, 2)
+ Female (304, 2)
> plist
$rows
[1] 305 306 307

$cols
[1] 2

> res
+ Male (305, 2)
+ Female (306, 2)
+ Unknown (307, 2)
> plist
$rows
[1] 308 309

$cols
[1] 2

```

```

> res
+ Male (308, 2)
+ Female (309, 2)
> plist
$rows
[1] 310 311

$cols
[1] 2

> res
+ Male (310, 2)
+ Female (311, 2)
> plist
$rows
[1] 312 313

$cols
[1] 2

> res
+ Male (312, 2)
+ Female (313, 2)
> res
+ NZ Maori (315, 1)
- + rows
- - 315 316
- + cols
- - 2
+ NZ European (317, 1)
- + rows
- - 317 318 319
- + cols
- - 2
+ Pasifika Peoples (320, 1)
- + rows
- - 320 321
- + cols
- - 2
+ Asian (322, 1)
- + rows
- - 322 323
- + cols
- - 2
+ Other/Unspecified Ethnicity (324, 1)
- + rows
- - 324 325
- + cols
- - 2
> plist
$rows
[1] 315 316

$cols
[1] 2

> res

```

```

+ Male (315, 2)
+ Female (316, 2)
> plist
$rows
[1] 317 318 319

$cols
[1] 2

> res
+ Male (317, 2)
+ Female (318, 2)
+ Unknown (319, 2)
> plist
$rows
[1] 320 321

$cols
[1] 2

> res
+ Male (320, 2)
+ Female (321, 2)
> plist
$rows
[1] 322 323

$cols
[1] 2

> res
+ Male (322, 2)
+ Female (323, 2)
> plist
$rows
[1] 324 325

$cols
[1] 2

> res
+ Male (324, 2)
+ Female (325, 2)
> res
+ NZ Maori (327, 1)
- + rows
- - 327 328
- + cols
- - 2
+ NZ European (329, 1)
- + rows
- - 329 330 331
- + cols
- - 2
+ Pasifika Peoples (332, 1)
- + rows
- - 332 333

```



```

- + cols
- - 2
+ Asian (334, 1)
- + rows
- - 334 335
- + cols
- - 2
+ Other/Unspecified Ethnicity (336, 1)
- + rows
- - 336 337
- + cols
- - 2
> plist
$rows
[1] 327 328

$cols
[1] 2

> res
+ Male (327, 2)
+ Female (328, 2)
> plist
$rows
[1] 329 330 331

$cols
[1] 2

> res
+ Male (329, 2)
+ Female (330, 2)
+ Unknown (331, 2)
> plist
$rows
[1] 332 333

$cols
[1] 2

> res
+ Male (332, 2)
+ Female (333, 2)
> plist
$rows
[1] 334 335

$cols
[1] 2

> res
+ Male (334, 2)
+ Female (335, 2)
> plist
$rows
[1] 336 337

```

```

$cols
[1] 2

> res
+ Male (336, 2)
+ Female (337, 2)
> res
+ NZ Maori (339, 1)
- + rows
- - 339 340
- + cols
- - 2
+ NZ European (341, 1)
- + rows
- - 341 342 343
- + cols
- - 2
+ Pasifika Peoples (344, 1)
- + rows
- - 344 345
- + cols
- - 2
+ Asian (346, 1)
- + rows
- - 346 347
- + cols
- - 2
+ Other/Unspecified Ethnicity (348, 1)
- + rows
- - 348 349
- + cols
- - 2
> plist
$rows
[1] 339 340

$cols
[1] 2

> res
+ Male (339, 2)
+ Female (340, 2)
> plist
$rows
[1] 341 342 343

$cols
[1] 2

> res
+ Male (341, 2)
+ Female (342, 2)
+ Unknown (343, 2)
> plist
$rows
[1] 344 345

```

```

$cols
[1] 2

> res
+ Male (344, 2)
+ Female (345, 2)
> plist
$rows
[1] 346 347

$cols
[1] 2

> res
+ Male (346, 2)
+ Female (347, 2)
> plist
$rows
[1] 348 349

$cols
[1] 2

> res
+ Male (348, 2)
+ Female (349, 2)
> res
+ NZ Maori (351, 1)
- + rows
- - 351 352
- + cols
- - 2
+ NZ European (353, 1)
- + rows
- - 353 354 355
- + cols
- - 2
+ Pasifika Peoples (356, 1)
- + rows
- - 356 357
- + cols
- - 2
+ Asian (358, 1)
- + rows
- - 358 359
- + cols
- - 2
+ Other/Unspecified Ethnicity (360, 1)
- + rows
- - 360 361
- + cols
- - 2
> plist
$rows
[1] 351 352

$cols

```

```

[1] 2

> res
+ Male (351, 2)
+ Female (352, 2)
> plist
$rows
[1] 353 354 355

$cols
[1] 2

> res
+ Male (353, 2)
+ Female (354, 2)
+ Unknown (355, 2)
> plist
$rows
[1] 356 357

$cols
[1] 2

> res
+ Male (356, 2)
+ Female (357, 2)
> plist
$rows
[1] 358 359

$cols
[1] 2

> res
+ Male (358, 2)
+ Female (359, 2)
> plist
$rows
[1] 360 361

$cols
[1] 2

> res
+ Male (360, 2)
+ Female (361, 2)
> res
+ NZ Maori (363, 1)
- + rows
- - 363 364
- + cols
- - 2
+ NZ European (365, 1)
- + rows
- - 365 366 367
- + cols
- - 2

```

```

+ Pasifika Peoples (368, 1)
- + rows
- - 368 369
- + cols
- - 2
+ Asian (370, 1)
- + rows
- - 370 371
- + cols
- - 2
+ Other/Unspecified Ethnicity (372, 1)
- + rows
- - 372 373
- + cols
- - 2
> plist
$rows
[1] 363 364

$cols
[1] 2

> res
+ Male (363, 2)
+ Female (364, 2)
> plist
$rows
[1] 365 366 367

$cols
[1] 2

> res
+ Male (365, 2)
+ Female (366, 2)
+ Unknown (367, 2)
> plist
$rows
[1] 368 369

$cols
[1] 2

> res
+ Male (368, 2)
+ Female (369, 2)
> plist
$rows
[1] 370 371

$cols
[1] 2

> res
+ Male (370, 2)
+ Female (371, 2)
> plist

```

```

$rows
[1] 372 373

$cols
[1] 2

> res
+ Male (372, 2)
+ Female (373, 2)
> res
+ NZ Maori (375, 1)
- + rows
- - 375 376
- + cols
- - 2
+ NZ European (377, 1)
- + rows
- - 377 378 379
- + cols
- - 2
+ Pasifika Peoples (380, 1)
- + rows
- - 380 381
- + cols
- - 2
+ Asian (382, 1)
- + rows
- - 382 383
- + cols
- - 2
+ Other/Unspecified Ethnicity (384, 1)
- + rows
- - 384 385
- + cols
- - 2
> plist
$rows
[1] 375 376

$cols
[1] 2

> res
+ Male (375, 2)
+ Female (376, 2)
> plist
$rows
[1] 377 378 379

$cols
[1] 2

> res
+ Male (377, 2)
+ Female (378, 2)
+ Unknown (379, 2)
> plist

```

```

$rows
[1] 380 381

$cols
[1] 2

> res
+ Male (380, 2)
+ Female (381, 2)
> plist
$rows
[1] 382 383

$cols
[1] 2

> res
+ Male (382, 2)
+ Female (383, 2)
> plist
$rows
[1] 384 385

$cols
[1] 2

> res
+ Male (384, 2)
+ Female (385, 2)
> res
+ NZ Maori (387, 1)
- + rows
- - 387 388
- + cols
- - 2
+ NZ European (389, 1)
- + rows
- - 389 390 391
- + cols
- - 2
+ Pasifika Peoples (392, 1)
- + rows
- - 392 393
- + cols
- - 2
+ Asian (394, 1)
- + rows
- - 394 395
- + cols
- - 2
+ Other/Unspecified Ethnicity (396, 1)
- + rows
- - 396 397
- + cols
- - 2
> plist
$rows

```

```

[1] 387 388

$cols
[1] 2

> res
+ Male (387, 2)
+ Female (388, 2)
> plist
$rows
[1] 389 390 391

$cols
[1] 2

> res
+ Male (389, 2)
+ Female (390, 2)
+ Unknown (391, 2)
> plist
$rows
[1] 392 393

$cols
[1] 2

> res
+ Male (392, 2)
+ Female (393, 2)
> plist
$rows
[1] 394 395

$cols
[1] 2

> res
+ Male (394, 2)
+ Female (395, 2)
> plist
$rows
[1] 396 397

$cols
[1] 2

> res
+ Male (396, 2)
+ Female (397, 2)
> res
+ NZ Maori (399, 1)
- + rows
- - 399 400
- + cols
- - 2
+ NZ European (401, 1)
- + rows

```



```

- - 401 402 403
- + cols
- - 2
+ Pasifika Peoples (404, 1)
- + rows
- - 404 405
- + cols
- - 2
+ Asian (406, 1)
- + rows
- - 406 407
- + cols
- - 2
+ Other/Unspecified Ethnicity (408, 1)
- + rows
- - 408 409
- + cols
- - 2
> plist
$rows
[1] 399 400

$cols
[1] 2

> res
+ Male (399, 2)
+ Female (400, 2)
> plist
$rows
[1] 401 402 403

$cols
[1] 2

> res
+ Male (401, 2)
+ Female (402, 2)
+ Unknown (403, 2)
> plist
$rows
[1] 404 405

$cols
[1] 2

> res
+ Male (404, 2)
+ Female (405, 2)
> plist
$rows
[1] 406 407

$cols
[1] 2

> res

```

```

+ Male (406, 2)
+ Female (407, 2)
> plist
$rows
[1] 408 409

$cols
[1] 2

> res
+ Male (408, 2)
+ Female (409, 2)
> res
+ NZ Maori (411, 1)
- + rows
- - 411 412
- + cols
- - 2
+ NZ European (413, 1)
- + rows
- - 413 414 415
- + cols
- - 2
+ Pasifika Peoples (416, 1)
- + rows
- - 416 417
- + cols
- - 2
+ Asian (418, 1)
- + rows
- - 418 419
- + cols
- - 2
+ Other/Unspecified Ethnicity (420, 1)
- + rows
- - 420 421
- + cols
- - 2
> plist
$rows
[1] 411 412

$cols
[1] 2

> res
+ Male (411, 2)
+ Female (412, 2)
> plist
$rows
[1] 413 414 415

$cols
[1] 2

> res
+ Male (413, 2)

```

```

+ Female (414, 2)
+ Unknown (415, 2)
> plist
$rows
[1] 416 417

$cols
[1] 2

> res
+ Male (416, 2)
+ Female (417, 2)
> plist
$rows
[1] 418 419

$cols
[1] 2

> res
+ Male (418, 2)
+ Female (419, 2)
> plist
$rows
[1] 420 421

$cols
[1] 2

> res
+ Male (420, 2)
+ Female (421, 2)
> rowplist
+ All Subjects (1, 1)
- + Accounting (2, 1)
- - + NZ Maori (3, 1)
- - - + Male (3, 2)
- - - + Female (4, 2)
- - + NZ European (5, 1)
- - - + Male (5, 2)
- - - + Female (6, 2)
- - - + Unknown (7, 2)
- - + Pasifika Peoples (8, 1)
- - - + Male (8, 2)
- - - + Female (9, 2)
- - + Asian (10, 1)
- - - + Male (10, 2)
- - - + Female (11, 2)
- - + Other/Unspecified Ethnicity (12, 1)
- - - + Male (12, 2)
- - - + Female (13, 2)
- + Agricultural & Horticultural Science (14, 1)
- - + NZ Maori (15, 1)
- - - + Male (15, 2)
- - - + Female (16, 2)
- - + NZ European (17, 1)
- - - + Male (17, 2)

```

- - - + Female (18, 2)
- - - + Unknown (19, 2)
- - + Pasifika Peoples (20, 1)
- - - + Male (20, 2)
- - - + Female (21, 2)
- - + Asian (22, 1)
- - - + Male (22, 2)
- - - + Female (23, 2)
- - + Other/Unspecified Ethnicity (24, 1)
- - - + Male (24, 2)
- - - + Female (25, 2)
- + Art History (26, 1)
- - + NZ Maori (27, 1)
- - - + Male (27, 2)
- - - + Female (28, 2)
- - + NZ European (29, 1)
- - - + Male (29, 2)
- - - + Female (30, 2)
- - - + Unknown (31, 2)
- - + Pasifika Peoples (32, 1)
- - - + Male (32, 2)
- - - + Female (33, 2)
- - + Asian (34, 1)
- - - + Male (34, 2)
- - - + Female (35, 2)
- - + Other/Unspecified Ethnicity (36, 1)
- - - + Male (36, 2)
- - - + Female (37, 2)
- + Biology (38, 1)
- - + NZ Maori (39, 1)
- - - + Male (39, 2)
- - - + Female (40, 2)
- - + NZ European (41, 1)
- - - + Male (41, 2)
- - - + Female (42, 2)
- - - + Unknown (43, 2)
- - + Pasifika Peoples (44, 1)
- - - + Male (44, 2)
- - - + Female (45, 2)
- - + Asian (46, 1)
- - - + Male (46, 2)
- - - + Female (47, 2)
- - + Other/Unspecified Ethnicity (48, 1)
- - - + Male (48, 2)
- - - + Female (49, 2)
- + Chemistry (50, 1)
- - + NZ Maori (51, 1)
- - - + Male (51, 2)
- - - + Female (52, 2)
- - + NZ European (53, 1)
- - - + Male (53, 2)
- - - + Female (54, 2)
- - - + Unknown (55, 2)
- - + Pasifika Peoples (56, 1)
- - - + Male (56, 2)
- - - + Female (57, 2)
- - + Asian (58, 1)

- - - + Male (58, 2)
- - - + Female (59, 2)
- - + Other/Unspecified Ethnicity (60, 1)
- - - + Male (60, 2)
- - - + Female (61, 2)
- + Chinese (62, 1)
- - + NZ Maori (63, 1)
- - - + Male (63, 2)
- - - + Female (64, 2)
- - + NZ European (65, 1)
- - - + Male (65, 2)
- - - + Female (66, 2)
- - - + Unknown (67, 2)
- - + Pasifika Peoples (68, 1)
- - - + Male (68, 2)
- - - + Female (69, 2)
- - + Asian (70, 1)
- - - + Male (70, 2)
- - - + Female (71, 2)
- - + Other/Unspecified Ethnicity (72, 1)
- - - + Male (72, 2)
- - - + Female (73, 2)
- + Classical Studies (74, 1)
- - + NZ Maori (75, 1)
- - - + Male (75, 2)
- - - + Female (76, 2)
- - + NZ European (77, 1)
- - - + Male (77, 2)
- - - + Female (78, 2)
- - - + Unknown (79, 2)
- - + Pasifika Peoples (80, 1)
- - - + Male (80, 2)
- - - + Female (81, 2)
- - + Asian (82, 1)
- - - + Male (82, 2)
- - - + Female (83, 2)
- - + Other/Unspecified Ethnicity (84, 1)
- - - + Male (84, 2)
- - - + Female (85, 2)
- + Dance (86, 1)
- - + NZ Maori (87, 1)
- - - + Male (87, 2)
- - - + Female (88, 2)
- - + NZ European (89, 1)
- - - + Male (89, 2)
- - - + Female (90, 2)
- - - + Unknown (91, 2)
- - + Pasifika Peoples (92, 1)
- - - + Male (92, 2)
- - - + Female (93, 2)
- - + Asian (94, 1)
- - - + Male (94, 2)
- - - + Female (95, 2)
- - + Other/Unspecified Ethnicity (96, 1)
- - - + Male (96, 2)
- - - + Female (97, 2)
- + Design and Visual Communication (98, 1)

- - + NZ Maori (99, 1)
- - - + Male (99, 2)
- - - + Female (100, 2)
- - + NZ European (101, 1)
- - - + Male (101, 2)
- - - + Female (102, 2)
- - - + Unknown (103, 2)
- - + Pasifika Peoples (104, 1)
- - - + Male (104, 2)
- - - + Female (105, 2)
- - + Asian (106, 1)
- - - + Male (106, 2)
- - - + Female (107, 2)
- - + Other/Unspecified Ethnicity (108, 1)
- - - + Male (108, 2)
- - - + Female (109, 2)
- + Drama (110, 1)
- - + NZ Maori (111, 1)
- - - + Male (111, 2)
- - - + Female (112, 2)
- - + NZ European (113, 1)
- - - + Male (113, 2)
- - - + Female (114, 2)
- - - + Unknown (115, 2)
- - + Pasifika Peoples (116, 1)
- - - + Male (116, 2)
- - - + Female (117, 2)
- - + Asian (118, 1)
- - - + Male (118, 2)
- - - + Female (119, 2)
- - + Other/Unspecified Ethnicity (120, 1)
- - - + Male (120, 2)
- - - + Female (121, 2)
- + Economics (122, 1)
- - + NZ Maori (123, 1)
- - - + Male (123, 2)
- - - + Female (124, 2)
- - + NZ European (125, 1)
- - - + Male (125, 2)
- - - + Female (126, 2)
- - - + Unknown (127, 2)
- - + Pasifika Peoples (128, 1)
- - - + Male (128, 2)
- - - + Female (129, 2)
- - + Asian (130, 1)
- - - + Male (130, 2)
- - - + Female (131, 2)
- - + Other/Unspecified Ethnicity (132, 1)
- - - + Male (132, 2)
- - - + Female (133, 2)
- + English (134, 1)
- - + NZ Maori (135, 1)
- - - + Male (135, 2)
- - - + Female (136, 2)
- - + NZ European (137, 1)
- - - + Male (137, 2)
- - - + Female (138, 2)

- - - + Unknown (139, 2)
- - + Pasifika Peoples (140, 1)
- - - + Male (140, 2)
- - - + Female (141, 2)
- - + Asian (142, 1)
- - - + Male (142, 2)
- - - + Female (143, 2)
- - + Other/Unspecified Ethnicity (144, 1)
- - - + Male (144, 2)
- - - + Female (145, 2)
- + French (146, 1)
- - + NZ Maori (147, 1)
- - - + Male (147, 2)
- - - + Female (148, 2)
- - + NZ European (149, 1)
- - - + Male (149, 2)
- - - + Female (150, 2)
- - - + Unknown (151, 2)
- - + Pasifika Peoples (152, 1)
- - - + Male (152, 2)
- - - + Female (153, 2)
- - + Asian (154, 1)
- - - + Male (154, 2)
- - - + Female (155, 2)
- - + Other/Unspecified Ethnicity (156, 1)
- - - + Male (156, 2)
- - - + Female (157, 2)
- + Geography (158, 1)
- - + NZ Maori (159, 1)
- - - + Male (159, 2)
- - - + Female (160, 2)
- - + NZ European (161, 1)
- - - + Male (161, 2)
- - - + Female (162, 2)
- - - + Unknown (163, 2)
- - + Pasifika Peoples (164, 1)
- - - + Male (164, 2)
- - - + Female (165, 2)
- - + Asian (166, 1)
- - - + Male (166, 2)
- - - + Female (167, 2)
- - + Other/Unspecified Ethnicity (168, 1)
- - - + Male (168, 2)
- - - + Female (169, 2)
- + German (170, 1)
- - + NZ Maori (171, 1)
- - - + Male (171, 2)
- - - + Female (172, 2)
- - + NZ European (173, 1)
- - - + Male (173, 2)
- - - + Female (174, 2)
- - - + Unknown (175, 2)
- - + Pasifika Peoples (176, 1)
- - - + Male (176, 2)
- - - + Female (177, 2)
- - + Asian (178, 1)
- - - + Male (178, 2)

- - - + Female (179, 2)
- - + Other/Unspecified Ethnicity (180, 1)
- - - + Male (180, 2)
- - - + Female (181, 2)
- + Graphics (182, 1)
- - + NZ Maori (183, 1)
- - - + Male (183, 2)
- - - + Female (184, 2)
- - + NZ European (185, 1)
- - - + Male (185, 2)
- - - + Female (186, 2)
- - - + Unknown (187, 2)
- - + Pasifika Peoples (188, 1)
- - - + Male (188, 2)
- - - + Female (189, 2)
- - + Asian (190, 1)
- - - + Male (190, 2)
- - - + Female (191, 2)
- - + Other/Unspecified Ethnicity (192, 1)
- - - + Male (192, 2)
- - - + Female (193, 2)
- + History (194, 1)
- - + NZ Maori (195, 1)
- - - + Male (195, 2)
- - - + Female (196, 2)
- - + NZ European (197, 1)
- - - + Male (197, 2)
- - - + Female (198, 2)
- - - + Unknown (199, 2)
- - + Pasifika Peoples (200, 1)
- - - + Male (200, 2)
- - - + Female (201, 2)
- - + Asian (202, 1)
- - - + Male (202, 2)
- - - + Female (203, 2)
- - + Other/Unspecified Ethnicity (204, 1)
- - - + Male (204, 2)
- - - + Female (205, 2)
- + Japanese (206, 1)
- - + NZ Maori (207, 1)
- - - + Male (207, 2)
- - - + Female (208, 2)
- - + NZ European (209, 1)
- - - + Male (209, 2)
- - - + Female (210, 2)
- - - + Unknown (211, 2)
- - + Pasifika Peoples (212, 1)
- - - + Male (212, 2)
- - - + Female (213, 2)
- - + Asian (214, 1)
- - - + Male (214, 2)
- - - + Female (215, 2)
- - + Other/Unspecified Ethnicity (216, 1)
- - - + Male (216, 2)
- - - + Female (217, 2)
- + Latin (218, 1)
- - + NZ Maori (219, 1)

- - - + Male (219, 2)
- - - + Female (220, 2)
- - + NZ European (221, 1)
- - - + Male (221, 2)
- - - + Female (222, 2)
- - - + Unknown (223, 2)
- - + Pasifika Peoples (224, 1)
- - - + Male (224, 2)
- - - + Female (225, 2)
- - + Asian (226, 1)
- - - + Male (226, 2)
- - - + Female (227, 2)
- - + Other/Unspecified Ethnicity (228, 1)
- - - + Male (228, 2)
- - - + Female (229, 2)
- + Mathematics with Calculus (230, 1)
- - + NZ Maori (231, 1)
- - - + Male (231, 2)
- - - + Female (232, 2)
- - + NZ European (233, 1)
- - - + Male (233, 2)
- - - + Female (234, 2)
- - - + Unknown (235, 2)
- - + Pasifika Peoples (236, 1)
- - - + Male (236, 2)
- - - + Female (237, 2)
- - + Asian (238, 1)
- - - + Male (238, 2)
- - - + Female (239, 2)
- - + Other/Unspecified Ethnicity (240, 1)
- - - + Male (240, 2)
- - - + Female (241, 2)
- + Media Studies (242, 1)
- - + NZ Maori (243, 1)
- - - + Male (243, 2)
- - - + Female (244, 2)
- - + NZ European (245, 1)
- - - + Male (245, 2)
- - - + Female (246, 2)
- - - + Unknown (247, 2)
- - + Pasifika Peoples (248, 1)
- - - + Male (248, 2)
- - - + Female (249, 2)
- - + Asian (250, 1)
- - - + Male (250, 2)
- - - + Female (251, 2)
- - + Other/Unspecified Ethnicity (252, 1)
- - - + Male (252, 2)
- - - + Female (253, 2)
- + Music Studies (254, 1)
- - + NZ Maori (255, 1)
- - - + Male (255, 2)
- - - + Female (256, 2)
- - + NZ European (257, 1)
- - - + Male (257, 2)
- - - + Female (258, 2)
- - - + Unknown (259, 2)

- - + Pasifika Peoples (260, 1)
- - - + Male (260, 2)
- - - + Female (261, 2)
- - + Asian (262, 1)
- - - + Male (262, 2)
- - - + Female (263, 2)
- - + Other/Unspecified Ethnicity (264, 1)
- - - + Male (264, 2)
- - - + Female (265, 2)
- + Painting (266, 1)
- - + NZ Maori (267, 1)
- - - + Male (267, 2)
- - - + Female (268, 2)
- - + NZ European (269, 1)
- - - + Male (269, 2)
- - - + Female (270, 2)
- - - + Unknown (271, 2)
- - + Pasifika Peoples (272, 1)
- - - + Male (272, 2)
- - - + Female (273, 2)
- - + Asian (274, 1)
- - - + Male (274, 2)
- - - + Female (275, 2)
- - + Other/Unspecified Ethnicity (276, 1)
- - - + Male (276, 2)
- - - + Female (277, 2)
- + Photography (278, 1)
- - + NZ Maori (279, 1)
- - - + Male (279, 2)
- - - + Female (280, 2)
- - + NZ European (281, 1)
- - - + Male (281, 2)
- - - + Female (282, 2)
- - - + Unknown (283, 2)
- - + Pasifika Peoples (284, 1)
- - - + Male (284, 2)
- - - + Female (285, 2)
- - + Asian (286, 1)
- - - + Male (286, 2)
- - - + Female (287, 2)
- - + Other/Unspecified Ethnicity (288, 1)
- - - + Male (288, 2)
- - - + Female (289, 2)
- + Physical Education (290, 1)
- - + NZ Maori (291, 1)
- - - + Male (291, 2)
- - - + Female (292, 2)
- - + NZ European (293, 1)
- - - + Male (293, 2)
- - - + Female (294, 2)
- - - + Unknown (295, 2)
- - + Pasifika Peoples (296, 1)
- - - + Male (296, 2)
- - - + Female (297, 2)
- - + Asian (298, 1)
- - - + Male (298, 2)
- - - + Female (299, 2)

- - + Other/Unspecified Ethnicity (300, 1)
- - - + Male (300, 2)
- - - + Female (301, 2)
- + Physics (302, 1)
- - + NZ Maori (303, 1)
- - - + Male (303, 2)
- - - + Female (304, 2)
- - + NZ European (305, 1)
- - - + Male (305, 2)
- - - + Female (306, 2)
- - - + Unknown (307, 2)
- - + Pasifika Peoples (308, 1)
- - - + Male (308, 2)
- - - + Female (309, 2)
- - + Asian (310, 1)
- - - + Male (310, 2)
- - - + Female (311, 2)
- - + Other/Unspecified Ethnicity (312, 1)
- - - + Male (312, 2)
- - - + Female (313, 2)
- + Printmaking (314, 1)
- - + NZ Maori (315, 1)
- - - + Male (315, 2)
- - - + Female (316, 2)
- - + NZ European (317, 1)
- - - + Male (317, 2)
- - - + Female (318, 2)
- - - + Unknown (319, 2)
- - + Pasifika Peoples (320, 1)
- - - + Male (320, 2)
- - - + Female (321, 2)
- - + Asian (322, 1)
- - - + Male (322, 2)
- - - + Female (323, 2)
- - + Other/Unspecified Ethnicity (324, 1)
- - - + Male (324, 2)
- - - + Female (325, 2)
- + Samoan (326, 1)
- - + NZ Maori (327, 1)
- - - + Male (327, 2)
- - - + Female (328, 2)
- - + NZ European (329, 1)
- - - + Male (329, 2)
- - - + Female (330, 2)
- - - + Unknown (331, 2)
- - + Pasifika Peoples (332, 1)
- - - + Male (332, 2)
- - - + Female (333, 2)
- - + Asian (334, 1)
- - - + Male (334, 2)
- - - + Female (335, 2)
- - + Other/Unspecified Ethnicity (336, 1)
- - - + Male (336, 2)
- - - + Female (337, 2)
- + Science (338, 1)
- - + NZ Maori (339, 1)
- - - + Male (339, 2)

- - - + Female (340, 2)
- - + NZ European (341, 1)
- - - + Male (341, 2)
- - - + Female (342, 2)
- - - + Unknown (343, 2)
- - + Pasifika Peoples (344, 1)
- - - + Male (344, 2)
- - - + Female (345, 2)
- - + Asian (346, 1)
- - - + Male (346, 2)
- - - + Female (347, 2)
- - + Other/Unspecified Ethnicity (348, 1)
- - - + Male (348, 2)
- - - + Female (349, 2)
- + Sculpture (350, 1)
- - + NZ Maori (351, 1)
- - - + Male (351, 2)
- - - + Female (352, 2)
- - + NZ European (353, 1)
- - - + Male (353, 2)
- - - + Female (354, 2)
- - - + Unknown (355, 2)
- - + Pasifika Peoples (356, 1)
- - - + Male (356, 2)
- - - + Female (357, 2)
- - + Asian (358, 1)
- - - + Male (358, 2)
- - - + Female (359, 2)
- - + Other/Unspecified Ethnicity (360, 1)
- - - + Male (360, 2)
- - - + Female (361, 2)
- + Spanish (362, 1)
- - + NZ Maori (363, 1)
- - - + Male (363, 2)
- - - + Female (364, 2)
- - + NZ European (365, 1)
- - - + Male (365, 2)
- - - + Female (366, 2)
- - - + Unknown (367, 2)
- - + Pasifika Peoples (368, 1)
- - - + Male (368, 2)
- - - + Female (369, 2)
- - + Asian (370, 1)
- - - + Male (370, 2)
- - - + Female (371, 2)
- - + Other/Unspecified Ethnicity (372, 1)
- - - + Male (372, 2)
- - - + Female (373, 2)
- + Statistics and Modelling (374, 1)
- - + NZ Maori (375, 1)
- - - + Male (375, 2)
- - - + Female (376, 2)
- - + NZ European (377, 1)
- - - + Male (377, 2)
- - - + Female (378, 2)
- - - + Unknown (379, 2)
- - + Pasifika Peoples (380, 1)

- - - + Male (380, 2)
 - - - + Female (381, 2)
 - - + Asian (382, 1)
 - - - + Male (382, 2)
 - - - + Female (383, 2)
 - - + Other/Unspecified Ethnicity (384, 1)
 - - - + Male (384, 2)
 - - - + Female (385, 2)
 - + Te Reo Maori (386, 1)
 - - + NZ Maori (387, 1)
 - - - + Male (387, 2)
 - - - + Female (388, 2)
 - - + NZ European (389, 1)
 - - - + Male (389, 2)
 - - - + Female (390, 2)
 - - - + Unknown (391, 2)
 - - + Pasifika Peoples (392, 1)
 - - - + Male (392, 2)
 - - - + Female (393, 2)
 - - + Asian (394, 1)
 - - - + Male (394, 2)
 - - - + Female (395, 2)
 - - + Other/Unspecified Ethnicity (396, 1)
 - - - + Male (396, 2)
 - - - + Female (397, 2)
 - + Te Reo Rangatira (398, 1)
 - - + NZ Maori (399, 1)
 - - - + Male (399, 2)
 - - - + Female (400, 2)
 - - + NZ European (401, 1)
 - - - + Male (401, 2)
 - - - + Female (402, 2)
 - - - + Unknown (403, 2)
 - - + Pasifika Peoples (404, 1)
 - - - + Male (404, 2)
 - - - + Female (405, 2)
 - - + Asian (406, 1)
 - - - + Male (406, 2)
 - - - + Female (407, 2)
 - - + Other/Unspecified Ethnicity (408, 1)
 - - - + Male (408, 2)
 - - - + Female (409, 2)
 - + Technology (410, 1)
 - - + NZ Maori (411, 1)
 - - - + Male (411, 2)
 - - - + Female (412, 2)
 - - + NZ European (413, 1)
 - - - + Male (413, 2)
 - - - + Female (414, 2)
 - - - + Unknown (415, 2)
 - - + Pasifika Peoples (416, 1)
 - - - + Male (416, 2)
 - - - + Female (417, 2)
 - - + Asian (418, 1)
 - - - + Male (418, 2)
 - - - + Female (419, 2)
 - - + Other/Unspecified Ethnicity (420, 1)

```

- - - + Male (420, 2)
- - - + Female (421, 2)
> rowvecs[1:4,]
      [,1]      [,2]      [,3]      [,4]
[1,] "All Subjects" "Accounting" "NZ Maori" "Male"
[2,] "All Subjects" "Accounting" "NZ Maori" "Female"
[3,] "All Subjects" "Accounting" "NZ European" "Male"
[4,] "All Subjects" "Accounting" "NZ European" "Female"
> res
+ Decile 1-3 (1, 2)
- + rows
- - 1 2 3 4 5 6 7
- + cols
- - 3
+ Decile 4-7 (8, 2)
- + rows
- - 8 9 10 11 12 13 14
- + cols
- - 3
+ Decile 8-10 (15, 2)
- + rows
- - 15 16 17 18 19 20 21
- + cols
- - 3
> plist
$rows
[1] 1 2 3 4 5 6 7

$cols
[1] 3

> res
+ # of Entries (1, 3)
+ # Absent (2, 3)
+ # SNA (3, 3)
+ # Assessed (4, 3)
+ # Not Achieved (5, 3)
+ # Scholarship (6, 3)
+ # Outstanding (7, 3)
> plist
$rows
[1] 8 9 10 11 12 13 14

$cols
[1] 3

> res
+ # of Entries (8, 3)
+ # Absent (9, 3)
+ # SNA (10, 3)
+ # Assessed (11, 3)
+ # Not Achieved (12, 3)
+ # Scholarship (13, 3)
+ # Outstanding (14, 3)
> plist
$rows
[1] 15 16 17 18 19 20 21

```

```
$cols
[1] 3
```

```
> res
+ # of Entries (15, 3)
+ # Absent (16, 3)
+ # SNA (17, 3)
+ # Assessed (18, 3)
+ # Not Achieved (19, 3)
+ # Scholarship (20, 3)
+ # Outstanding (21, 3)
> colplist
+ Decile 1-3 (1, 2)
- + # of Entries (1, 3)
- + # Absent (2, 3)
- + # SNA (3, 3)
- + # Assessed (4, 3)
- + # Not Achieved (5, 3)
- + # Scholarship (6, 3)
- + # Outstanding (7, 3)
+ Decile 4-7 (8, 2)
- + # of Entries (8, 3)
- + # Absent (9, 3)
- + # SNA (10, 3)
- + # Assessed (11, 3)
- + # Not Achieved (12, 3)
- + # Scholarship (13, 3)
- + # Outstanding (14, 3)
+ Decile 8-10 (15, 2)
- + # of Entries (15, 3)
- + # Absent (16, 3)
- + # SNA (17, 3)
- + # Assessed (18, 3)
- + # Not Achieved (19, 3)
- + # Scholarship (20, 3)
- + # Outstanding (21, 3)
```

```
> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN UNKNOWN # of Entries
1 Decile 1-3 All Subjects Accounting NZ Maori      Male          2
2 Decile 1-3 All Subjects Accounting NZ Maori      Female         0
3 Decile 1-3 All Subjects Accounting NZ European    Male          2
4 Decile 1-3 All Subjects Accounting NZ European    Female         3
# Absent # SNA # Assessed # Not Achieved # Scholarship # Outstanding
1         1     0         1         1         0         0         0
2         0     0         0         0         0         0         0
3         0     0         2         1         1         1         0
4         0     0         3         2         1         1         0
```

8.2 NZQASubjects.xls

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Standard(s) and Subject(s) (broken down by Ethnicity and Gender)												
2													
3	Accounting	Externally Assessed Achievement S	National										
4	Standard L	Ethnicity	Gender		# of Results		# Not Achieved		# Achieved		# Merit		# Excellenc
5													
6	Level 1				17692.0		3859.0		5683.0		4978.0		3172.0
7		NZ Maori			1633.0		545.0		609.0		350.0		129.0
8			Male		847.0		260.0		327.0		193.0		67.0
9			Female		786.0		285.0		282.0		157.0		62.0
10		NZ European			9404.0		1610.0		3081.0		2873.0		1840.0
11			Male		5508.0		976.0		1935.0		1673.0		924.0
12			Female		3896.0		634.0		1146.0		1200.0		916.0
13		Pasifika Peoples			1490.0		694.0		494.0		227.0		75.0
14			Male		538.0		244.0		168.0		96.0		30.0
15			Female		952.0		450.0		326.0		131.0		45.0
16		Asian			4749.0		893.0		1359.0		1425.0		1072.0
17			Male		2265.0		471.0		685.0		667.0		442.0
18			Female		2484.0		422.0		674.0		758.0		630.0
19		Other/Unspecified Ethnicity			416.0		117.0		140.0		103.0		56.0
20			Male		206.0		53.0		80.0		50.0		23.0
21			Female		210.0		64.0		60.0		53.0		33.0
22													
23	Level 2				13540.0		3617.0		4762.0		3355.0		1806.0
24		NZ Maori			901.0		390.0		323.0		138.0		50.0
25			Male		454.0		181.0		177.0		72.0		24.0
26			Female		447.0		209.0		146.0		66.0		26.0
27		NZ European			7642.0		1719.0		2760.0		2023.0		1140.0
28			Male		4247.0		991.0		1532.0		1101.0		623.0
29			Female		3395.0		728.0		1228.0		922.0		517.0
30		Pasifika Peoples			911.0		455.0		326.0		112.0		18.0
31			Male		328.0		157.0		126.0		41.0		4.0
32			Female		583.0		298.0		200.0		71.0		14.0
33		Asian			3796.0		973.0		1253.0		999.0		571.0
34			Male		1834.0		484.0		633.0		460.0		257.0
35			Female		1962.0		489.0		620.0		539.0		314.0
36		Other/Unspecified Ethnicity			290.0		80.0		100.0		83.0		27.0
37			Male		137.0		38.0		44.0		40.0		15.0
38			Female		153.0		42.0		56.0		43.0		12.0
39													
40	Level 3				9373.0		2802.0		3183.0		2277.0		1111.0
41		NZ Maori			504.0		240.0		162.0		73.0		29.0
42			Male		265.0		117.0		92.0		36.0		20.0
43			Female		239.0		123.0		70.0		37.0		9.0

```

> rowData
[1] 6 55
> colData
[1] 5 13
> rowslist
$label
[1] 1 3 4

$data
[1] 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31
[26] 32 33 34 35 36 37 38 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55

> colslist
$label
[1] 1 2 3

$data
[1] 5 7 9 11 13

> Patvec
[1] NA NA
> rowslist
$label
[1] 1 3 4

$data

```



```

[1] 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 23 24 25 26 27 28 29 30 31
[26] 32 33 34 35 36 37 38 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55

```

```
> colslst
```

```
$label
```

```
[1] 1 2 3
```

```
$data
```

```
$data[1]
```

```
[1] 5 7 9 11 13
```

```
> res
```

```
+ Level 1 (1, 1)
```

```
- + rows
```

```
- - 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

```
- + cols
```

```
- - 1 2 3
```

```
+ Level 2 (17, 1)
```

```
- + rows
```

```
- - 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
```

```
- + cols
```

```
- - 1 2 3
```

```
+ Level 3 (33, 1)
```

```
- + rows
```

```
- - 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
```

```
- + cols
```

```
- - 1 2 3
```

```
> res
```

```
+ NZ Maori (2, 2)
```

```
- + rows
```

```
- - 3 4
```

```
- + cols
```

```
- - 2 3
```

```
+ NZ European (5, 2)
```

```
- + rows
```

```
- - 6 7
```

```
- + cols
```

```
- - 2 3
```

```
+ Pasifika Peoples (8, 2)
```

```
- + rows
```

```
- - 9 10
```

```
- + cols
```

```
- - 2 3
```

```
+ Asian (11, 2)
```

```
- + rows
```

```
- - 12 13
```

```
- + cols
```

```
- - 2 3
```

```
+ Other/Unspecified Ethnicity (14, 2)
```

```
- + rows
```

```
- - 15 16
```

```
- + cols
```

```
- - 2 3
```

```
> plist
```

```
$rows
```

```
[1] 3 4
```

```

$cols
[1] 3

> res
+ Male (3, 3)
+ Female (4, 3)
> plist
$rows
[1] 6 7

$cols
[1] 3

> res
+ Male (6, 3)
+ Female (7, 3)
> plist
$rows
[1] 9 10

$cols
[1] 3

> res
+ Male (9, 3)
+ Female (10, 3)
> plist
$rows
[1] 12 13

$cols
[1] 3

> res
+ Male (12, 3)
+ Female (13, 3)
> plist
$rows
[1] 15 16

$cols
[1] 3

> res
+ Male (15, 3)
+ Female (16, 3)
> res
+ NZ Maori (18, 2)
- + rows
- - 19 20
- + cols
- - 2 3
+ NZ European (21, 2)
- + rows
- - 22 23
- + cols

```

```

- - 2 3
+ Pasifika Peoples (24, 2)
- + rows
- - 25 26
- + cols
- - 2 3
+ Asian (27, 2)
- + rows
- - 28 29
- + cols
- - 2 3
+ Other/Unspecified Ethnicity (30, 2)
- + rows
- - 31 32
- + cols
- - 2 3
> plist
$rows
[1] 19 20

$cols
[1] 3

> res
+ Male (19, 3)
+ Female (20, 3)
> plist
$rows
[1] 22 23

$cols
[1] 3

> res
+ Male (22, 3)
+ Female (23, 3)
> plist
$rows
[1] 25 26

$cols
[1] 3

> res
+ Male (25, 3)
+ Female (26, 3)
> plist
$rows
[1] 28 29

$cols
[1] 3

> res
+ Male (28, 3)
+ Female (29, 3)
> plist

```

```

$rows
[1] 31 32

$cols
[1] 3

> res
+ Male (31, 3)
+ Female (32, 3)
> res
+ NZ Maori (34, 2)
- + rows
- - 35 36
- + cols
- - 2 3
+ NZ European (37, 2)
- + rows
- - 38 39
- + cols
- - 2 3
+ Pasifika Peoples (40, 2)
- + rows
- - 41 42
- + cols
- - 2 3
+ Asian (43, 2)
- + rows
- - 44 45
- + cols
- - 2 3
+ Other/Unspecified Ethnicity (46, 2)
- + rows
- - 47 48
- + cols
- - 2 3
> plist
$rows
[1] 35 36

$cols
[1] 3

> res
+ Male (35, 3)
+ Female (36, 3)
> plist
$rows
[1] 38 39

$cols
[1] 3

> res
+ Male (38, 3)
+ Female (39, 3)
> plist
$rows

```

```
[1] 41 42
```

```
$cols
```

```
[1] 3
```

```
> res
```

```
+ Male (41, 3)
```

```
+ Female (42, 3)
```

```
> plist
```

```
$rows
```

```
[1] 44 45
```

```
$cols
```

```
[1] 3
```

```
> res
```

```
+ Male (44, 3)
```

```
+ Female (45, 3)
```

```
> plist
```

```
$rows
```

```
[1] 47 48
```

```
$cols
```

```
[1] 3
```

```
> res
```

```
+ Male (47, 3)
```

```
+ Female (48, 3)
```

```
> rowplist
```

```
+ Level 1 (1, 1)
```

```
- + NZ Maori (2, 2)
```

```
- - + Male (3, 3)
```

```
- - + Female (4, 3)
```

```
- + NZ European (5, 2)
```

```
- - + Male (6, 3)
```

```
- - + Female (7, 3)
```

```
- + Pasifika Peoples (8, 2)
```

```
- - + Male (9, 3)
```

```
- - + Female (10, 3)
```

```
- + Asian (11, 2)
```

```
- - + Male (12, 3)
```

```
- - + Female (13, 3)
```

```
- + Other/Unspecified Ethnicity (14, 2)
```

```
- - + Male (15, 3)
```

```
- - + Female (16, 3)
```

```
+ Level 2 (17, 1)
```

```
- + NZ Maori (18, 2)
```

```
- - + Male (19, 3)
```

```
- - + Female (20, 3)
```

```
- + NZ European (21, 2)
```

```
- - + Male (22, 3)
```

```
- - + Female (23, 3)
```

```
- + Pasifika Peoples (24, 2)
```

```
- - + Male (25, 3)
```

```
- - + Female (26, 3)
```

```
- + Asian (27, 2)
```

```
- - + Male (28, 3)
```

```

- - + Female (29, 3)
- + Other/Unspecified Ethnicity (30, 2)
- - + Male (31, 3)
- - + Female (32, 3)
+ Level 3 (33, 1)
- + NZ Maori (34, 2)
- - + Male (35, 3)
- - + Female (36, 3)
- + NZ European (37, 2)
- - + Male (38, 3)
- - + Female (39, 3)
- + Pasifika Peoples (40, 2)
- - + Male (41, 3)
- - + Female (42, 3)
- + Asian (43, 2)
- - + Male (44, 3)
- - + Female (45, 3)
- + Other/Unspecified Ethnicity (46, 2)
- - + Male (47, 3)
- - + Female (48, 3)
> rowvecs[1:4,]
      [,1]      [,2]      [,3]
[1,] "Level 1" "NZ Maori"  "Male"
[2,] "Level 1" "NZ Maori"  "Female"
[3,] "Level 1" "NZ European" "Male"
[4,] "Level 1" "NZ European" "Female"
> res
+ National (1, 2)
- + rows
- - 1 2 3 4 5
- + cols
- - 3
> plist
$rows
[1] 1 2 3 4 5

$cols
[1] 3

> res
+ # of Results (1, 3)
+ # Not Achieved (2, 3)
+ # Achieved (3, 3)
+ # Merit (4, 3)
+ # Excellence (5, 3)
> colplist
+ National (1, 2)
- + # of Results (1, 3)
- + # Not Achieved (2, 3)
- + # Achieved (3, 3)
- + # Merit (4, 3)
- + # Excellence (5, 3)
> res[1:4,]
      UNKNOWN UNKNOWN      UNKNOWN UNKNOWN # of Results # Not Achieved # Achieved
1 National Level 1   NZ Maori   Male           847           260           327
2 National Level 1   NZ Maori   Female          786           285           282
3 National Level 1 NZ European   Male          5508           976          1935

```

4	National Level 1 NZ European Female	3896	634	1146
	# Merit # Excellence			
1	193 67			
2	157 62			
3	1673 924			
4	1200 916			

8.3 StatsNZLabourForce.csv

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Labour Force Status by Sex by Sing/Comb Ethnic Group (Qrtly-Mar/Jun/Sep/Dec)												
2		Male											
3		European Only								Maori Only			
4		Persons En	Persons Un	Not in Labo	Working Ag	Labour Forc	Unemploy	Employer	Total Labou	Persons En	Persons Un	Not in Labo	Working Ag
5	2007Q4	855.8	20.0	280.0	1155.8	75.8	2.3	74.0	875.8	71.1	6.1	28.1	105.3
6	2008Q1	863.0	25.4	283.5	1171.9	75.8	2.9	73.6	888.5	69.1	7.5	31.4	107.9
7	2008Q2	850.1	26.0	280.7	1156.8	75.7	3.0	73.5	876.1	67.2	5.7	27.4	100.2
8	2008Q3	839.6	29.8	285.9	1155.3	75.2	3.4	72.7	869.4	71.7	8.7	30.7	111.1
9	2008Q4	854.8	29.5	274.7	1158.9	76.3	3.3	73.8	884.2	76.1	8.5	28.5	113.1
10	2009Q1	845.0	35.4	279.4	1159.8	75.9	4.0	72.9	880.4	75.4	8.4	35.7	119.5
11	2009Q2	831.6	34.9	279.7	1146.2	75.6	4.0	72.6	866.5	74.2	9.9	33.1	117.3
12	2009Q3	813.3	42.5	290.4	1146.2	74.7	5.0	71.0	855.8	70.9	10.9	36.0	117.8
13	2009Q4	831.1	40.1	277.0	1148.2	75.9	4.6	72.4	871.2	71.7	13.6	33.2	118.5
14	2010Q1	822.5	36.4	283.2	1142.1	75.2	4.2	72.0	858.9	71.8	11.3	35.3	118.4
15	2010Q2	825.3	39.9	290.1	1155.3	74.9	4.6	71.4	865.2	71.9	13.7	33.7	119.2
16	2010Q3	836.9	31.0	287.1	1155.1	75.1	3.6	72.5	867.9	69.8	13.5	34.1	117.3
17	2010Q4	838.1	39.6	277.1	1154.8	76.0	4.5	72.6	877.7	70.7	14.4	36.4	121.5
18	2011Q1	829.6	36.8	281.2	1147.6	75.5	4.2	72.3	866.4	70.7	13.9	35.3	119.8
19	2011Q2	838.7	41.0	279.0	1158.7	75.9	4.7	72.4	879.6	67.1	10.5	37.1	114.7
20	2011Q3	830.5	34.6	280.2	1145.3	75.5	4.0	72.5	865.1	69.5	13.4	34.9	117.8
21	2011Q4	841.8	34.8	277.5	1154.1	76.0	4.0	72.9	876.6	69.2	12.7	36.1	118.0
22	2012Q1	843.1	43.3	282.7	1169.1	75.8	4.9	72.1	886.3	71.5	11.2	34.6	117.3
23	2012Q2	837.1	38.2	296.5	1171.8	74.7	4.4	71.4	875.3	66.2	10.6	33.0	109.7
24	2012Q3	833.0	38.0	298.4	1169.3	74.5	4.4	71.2	871.0	67.1	13.1	33.0	113.2
25	2012Q4	833.4	41.0	298.2	1172.6	74.6	4.7	71.1	874.4	63.0	12.3	35.4	110.7
26	2013Q1	832.0	35.8	294.7	1162.5	74.6	4.1	71.6	867.8	69.9	12.2	38.2	120.3
27	Table information:												
28	Units:												
29	Persons Employed in Labour Force: Number, Magnitude = Thousands												
30	Persons Unemployed in Labour Force: Number, Magnitude = Thousands												
31	Not in Labour Force: Number, Magnitude = Thousands												
32	Working Age Population: Number, Magnitude = Thousands												
33	Labour Force Participation Rate: Percent, Magnitude = Units												
34	Unemployment Rate: Percent, Magnitude = Units												
35	Employment Rate: Percent, Magnitude = Units												
36	Total Labour Force: Number, Magnitude = Thousands												
37	Footnotes:												
38													
39	Symbols:												
40	.. figure not available												
41	C: Confidential												
42	E: Early Estimate												
43	P: Provisional												

```
> rowData
[1] 5 26
> colData
[1] 2 241
> rowslist
$label
[1] 1 2 3 4
```

```
$data
[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
> colslist
$label
[1] 1
```

```
$data
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
[19] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
[37] 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
[55] 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
[73] 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
[91] 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109
[109] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
[127] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
[145] 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163
[163] 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
```



```

[181] 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
[199] 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217
[217] 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235
[235] 236 237 238 239 240 241

> Patvec
[1] 80 8 8
> rowslist
$label
[1] 1 2 3 4

$data
[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

> colslist
$label
[1] 1

$data
$data[1]
[1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[26] 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
[51] 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
[76] 77 78 79 80 81

$data[2]
[1] 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[20] 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
[39] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138
[58] 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
[77] 158 159 160 161

$data[3]
[1] 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
[20] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
[39] 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218
[58] 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237
[77] 238 239 240 241

> plist
$rows
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

$cols
[1] 1

> res
+ 2007Q4 (1, 1)
+ 2008Q1 (2, 1)
+ 2008Q2 (3, 1)
+ 2008Q3 (4, 1)
+ 2008Q4 (5, 1)
+ 2009Q1 (6, 1)
+ 2009Q2 (7, 1)
+ 2009Q3 (8, 1)
+ 2009Q4 (9, 1)

```

```

+ 2010Q1 (10, 1)
+ 2010Q2 (11, 1)
+ 2010Q3 (12, 1)
+ 2010Q4 (13, 1)
+ 2011Q1 (14, 1)
+ 2011Q2 (15, 1)
+ 2011Q3 (16, 1)
+ 2011Q4 (17, 1)
+ 2012Q1 (18, 1)
+ 2012Q2 (19, 1)
+ 2012Q3 (20, 1)
+ 2012Q4 (21, 1)
+ 2013Q1 (22, 1)
> rowplist
+ 2007Q4 (1, 1)
+ 2008Q1 (2, 1)
+ 2008Q2 (3, 1)
+ 2008Q3 (4, 1)
+ 2008Q4 (5, 1)
+ 2009Q1 (6, 1)
+ 2009Q2 (7, 1)
+ 2009Q3 (8, 1)
+ 2009Q4 (9, 1)
+ 2010Q1 (10, 1)
+ 2010Q2 (11, 1)
+ 2010Q3 (12, 1)
+ 2010Q4 (13, 1)
+ 2011Q1 (14, 1)
+ 2011Q2 (15, 1)
+ 2011Q3 (16, 1)
+ 2011Q4 (17, 1)
+ 2012Q1 (18, 1)
+ 2012Q2 (19, 1)
+ 2012Q3 (20, 1)
+ 2012Q4 (21, 1)
+ 2013Q1 (22, 1)
> rowvecs[1:4,]
[1] "2007Q4" "2008Q1" "2008Q2" "2008Q3"
> res
+ Male (1, 2)
- + rows
- - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
- + cols
- - 3 4
+ Female (81, 2)
- + rows
- - 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108
- + cols
- - 3 4
+ Total Both Sexes (161, 2)
- + rows
- - 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
- + cols
- - 3 4
> res
+ European Only (1, 3)
- + rows

```

```

- - 1 2 3 4 5 6 7 8
- + cols
- - 4
+ Maori Only (9, 3)
- + rows
- - 9 10 11 12 13 14 15 16
- + cols
- - 4
+ Pacific Peoples Only (17, 3)
- + rows
- - 17 18 19 20 21 22 23 24
- + cols
- - 4
+ Asian Only (25, 3)
- + rows
- - 25 26 27 28 29 30 31 32
- + cols
- - 4
+ MELAA Only (33, 3)
- + rows
- - 33 34 35 36 37 38 39 40
- + cols
- - 4
+ Other Ethnicity Only (41, 3)
- + rows
- - 41 42 43 44 45 46 47 48
- + cols
- - 4
+ European/Maori (49, 3)
- + rows
- - 49 50 51 52 53 54 55 56
- + cols
- - 4
+ Two or More Groups Not Elsewhere Included (57, 3)
- + rows
- - 57 58 59 60 61 62 63 64
- + cols
- - 4
+ Residual Categories (65, 3)
- + rows
- - 65 66 67 68 69 70 71 72
- + cols
- - 4
+ Total All Ethnic Groups (73, 3)
- + rows
- - 73 74 75 76 77 78 79 80
- + cols
- - 4
> plist
$rows
[1] 1 2 3 4 5 6 7 8

$cols
[1] 4

> res
+ Persons Employed in Labour Force (1, 4)

```

```

+ Persons Unemployed in Labour Force (2, 4)
+ Not in Labour Force (3, 4)
+ Working Age Population (4, 4)
+ Labour Force Participation Rate (5, 4)
+ Unemployment Rate (6, 4)
+ Employment Rate (7, 4)
+ Total Labour Force (8, 4)
> plist
$rows
[1] 9 10 11 12 13 14 15 16

$cols
[1] 4

> res
+ Persons Employed in Labour Force (9, 4)
+ Persons Unemployed in Labour Force (10, 4)
+ Not in Labour Force (11, 4)
+ Working Age Population (12, 4)
+ Labour Force Participation Rate (13, 4)
+ Unemployment Rate (14, 4)
+ Employment Rate (15, 4)
+ Total Labour Force (16, 4)
> plist
$rows
[1] 17 18 19 20 21 22 23 24

$cols
[1] 4

> res
+ Persons Employed in Labour Force (17, 4)
+ Persons Unemployed in Labour Force (18, 4)
+ Not in Labour Force (19, 4)
+ Working Age Population (20, 4)
+ Labour Force Participation Rate (21, 4)
+ Unemployment Rate (22, 4)
+ Employment Rate (23, 4)
+ Total Labour Force (24, 4)
> plist
$rows
[1] 25 26 27 28 29 30 31 32

$cols
[1] 4

> res
+ Persons Employed in Labour Force (25, 4)
+ Persons Unemployed in Labour Force (26, 4)
+ Not in Labour Force (27, 4)
+ Working Age Population (28, 4)
+ Labour Force Participation Rate (29, 4)
+ Unemployment Rate (30, 4)
+ Employment Rate (31, 4)
+ Total Labour Force (32, 4)
> plist
$rows

```

```

[1] 33 34 35 36 37 38 39 40

$cols
[1] 4

> res
+ Persons Employed in Labour Force (33, 4)
+ Persons Unemployed in Labour Force (34, 4)
+ Not in Labour Force (35, 4)
+ Working Age Population (36, 4)
+ Labour Force Participation Rate (37, 4)
+ Unemployment Rate (38, 4)
+ Employment Rate (39, 4)
+ Total Labour Force (40, 4)
> plist
$rows
[1] 41 42 43 44 45 46 47 48

$cols
[1] 4

> res
+ Persons Employed in Labour Force (41, 4)
+ Persons Unemployed in Labour Force (42, 4)
+ Not in Labour Force (43, 4)
+ Working Age Population (44, 4)
+ Labour Force Participation Rate (45, 4)
+ Unemployment Rate (46, 4)
+ Employment Rate (47, 4)
+ Total Labour Force (48, 4)
> plist
$rows
[1] 49 50 51 52 53 54 55 56

$cols
[1] 4

> res
+ Persons Employed in Labour Force (49, 4)
+ Persons Unemployed in Labour Force (50, 4)
+ Not in Labour Force (51, 4)
+ Working Age Population (52, 4)
+ Labour Force Participation Rate (53, 4)
+ Unemployment Rate (54, 4)
+ Employment Rate (55, 4)
+ Total Labour Force (56, 4)
> plist
$rows
[1] 57 58 59 60 61 62 63 64

$cols
[1] 4

> res
+ Persons Employed in Labour Force (57, 4)
+ Persons Unemployed in Labour Force (58, 4)
+ Not in Labour Force (59, 4)

```

```

+ Working Age Population (60, 4)
+ Labour Force Participation Rate (61, 4)
+ Unemployment Rate (62, 4)
+ Employment Rate (63, 4)
+ Total Labour Force (64, 4)
> plist
$rows
[1] 65 66 67 68 69 70 71 72

$cols
[1] 4

> res
+ Persons Employed in Labour Force (65, 4)
+ Persons Unemployed in Labour Force (66, 4)
+ Not in Labour Force (67, 4)
+ Working Age Population (68, 4)
+ Labour Force Participation Rate (69, 4)
+ Unemployment Rate (70, 4)
+ Employment Rate (71, 4)
+ Total Labour Force (72, 4)
> plist
$rows
[1] 73 74 75 76 77 78 79 80

$cols
[1] 4

> res
+ Persons Employed in Labour Force (73, 4)
+ Persons Unemployed in Labour Force (74, 4)
+ Not in Labour Force (75, 4)
+ Working Age Population (76, 4)
+ Labour Force Participation Rate (77, 4)
+ Unemployment Rate (78, 4)
+ Employment Rate (79, 4)
+ Total Labour Force (80, 4)
> res
+ European Only (81, 3)
- + rows
- - 81 82 83 84 85 86 87 88
- + cols
- - 4
+ Maori Only (89, 3)
- + rows
- - 89 90 91 92 93 94 95 96
- + cols
- - 4
+ Pacific Peoples Only (97, 3)
- + rows
- - 97 98 99 100 101 102 103 104
- + cols
- - 4
+ Asian Only (105, 3)
- + rows
- - 105 106 107 108 109 110 111 112
- + cols

```

```

- - 4
+ MELAA Only (113, 3)
- + rows
- - 113 114 115 116 117 118 119 120
- + cols
- - 4
+ Other Ethnicity Only (121, 3)
- + rows
- - 121 122 123 124 125 126 127 128
- + cols
- - 4
+ European/Maori (129, 3)
- + rows
- - 129 130 131 132 133 134 135 136
- + cols
- - 4
+ Two or More Groups Not Elsewhere Included (137, 3)
- + rows
- - 137 138 139 140 141 142 143 144
- + cols
- - 4
+ Residual Categories (145, 3)
- + rows
- - 145 146 147 148 149 150 151 152
- + cols
- - 4
+ Total All Ethnic Groups (153, 3)
- + rows
- - 153 154 155 156 157 158 159 160
- + cols
- - 4
> plist
$rows
[1] 81 82 83 84 85 86 87 88

$cols
[1] 4

> res
+ Persons Employed in Labour Force (81, 4)
+ Persons Unemployed in Labour Force (82, 4)
+ Not in Labour Force (83, 4)
+ Working Age Population (84, 4)
+ Labour Force Participation Rate (85, 4)
+ Unemployment Rate (86, 4)
+ Employment Rate (87, 4)
+ Total Labour Force (88, 4)
> plist
$rows
[1] 89 90 91 92 93 94 95 96

$cols
[1] 4

> res
+ Persons Employed in Labour Force (89, 4)
+ Persons Unemployed in Labour Force (90, 4)

```

```

+ Not in Labour Force (91, 4)
+ Working Age Population (92, 4)
+ Labour Force Participation Rate (93, 4)
+ Unemployment Rate (94, 4)
+ Employment Rate (95, 4)
+ Total Labour Force (96, 4)
> plist
$rows
[1] 97 98 99 100 101 102 103 104

$cols
[1] 4

> res
+ Persons Employed in Labour Force (97, 4)
+ Persons Unemployed in Labour Force (98, 4)
+ Not in Labour Force (99, 4)
+ Working Age Population (100, 4)
+ Labour Force Participation Rate (101, 4)
+ Unemployment Rate (102, 4)
+ Employment Rate (103, 4)
+ Total Labour Force (104, 4)
> plist
$rows
[1] 105 106 107 108 109 110 111 112

$cols
[1] 4

> res
+ Persons Employed in Labour Force (105, 4)
+ Persons Unemployed in Labour Force (106, 4)
+ Not in Labour Force (107, 4)
+ Working Age Population (108, 4)
+ Labour Force Participation Rate (109, 4)
+ Unemployment Rate (110, 4)
+ Employment Rate (111, 4)
+ Total Labour Force (112, 4)
> plist
$rows
[1] 113 114 115 116 117 118 119 120

$cols
[1] 4

> res
+ Persons Employed in Labour Force (113, 4)
+ Persons Unemployed in Labour Force (114, 4)
+ Not in Labour Force (115, 4)
+ Working Age Population (116, 4)
+ Labour Force Participation Rate (117, 4)
+ Unemployment Rate (118, 4)
+ Employment Rate (119, 4)
+ Total Labour Force (120, 4)
> plist
$rows
[1] 121 122 123 124 125 126 127 128

```



```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (121, 4)
+ Persons Unemployed in Labour Force (122, 4)
+ Not in Labour Force (123, 4)
+ Working Age Population (124, 4)
+ Labour Force Participation Rate (125, 4)
+ Unemployment Rate (126, 4)
+ Employment Rate (127, 4)
+ Total Labour Force (128, 4)
> plist
$rows
[1] 129 130 131 132 133 134 135 136

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (129, 4)
+ Persons Unemployed in Labour Force (130, 4)
+ Not in Labour Force (131, 4)
+ Working Age Population (132, 4)
+ Labour Force Participation Rate (133, 4)
+ Unemployment Rate (134, 4)
+ Employment Rate (135, 4)
+ Total Labour Force (136, 4)
> plist
$rows
[1] 137 138 139 140 141 142 143 144

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (137, 4)
+ Persons Unemployed in Labour Force (138, 4)
+ Not in Labour Force (139, 4)
+ Working Age Population (140, 4)
+ Labour Force Participation Rate (141, 4)
+ Unemployment Rate (142, 4)
+ Employment Rate (143, 4)
+ Total Labour Force (144, 4)
> plist
$rows
[1] 145 146 147 148 149 150 151 152

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (145, 4)
+ Persons Unemployed in Labour Force (146, 4)
+ Not in Labour Force (147, 4)
+ Working Age Population (148, 4)

```

```

+ Labour Force Participation Rate (149, 4)
+ Unemployment Rate (150, 4)
+ Employment Rate (151, 4)
+ Total Labour Force (152, 4)
> plist
$rows
[1] 153 154 155 156 157 158 159 160

$cols
[1] 4

> res
+ Persons Employed in Labour Force (153, 4)
+ Persons Unemployed in Labour Force (154, 4)
+ Not in Labour Force (155, 4)
+ Working Age Population (156, 4)
+ Labour Force Participation Rate (157, 4)
+ Unemployment Rate (158, 4)
+ Employment Rate (159, 4)
+ Total Labour Force (160, 4)
> res
+ European Only (161, 3)
- + rows
- - 161 162 163 164 165 166 167 168
- + cols
- - 4
+ Maori Only (169, 3)
- + rows
- - 169 170 171 172 173 174 175 176
- + cols
- - 4
+ Pacific Peoples Only (177, 3)
- + rows
- - 177 178 179 180 181 182 183 184
- + cols
- - 4
+ Asian Only (185, 3)
- + rows
- - 185 186 187 188 189 190 191 192
- + cols
- - 4
+ MELAA Only (193, 3)
- + rows
- - 193 194 195 196 197 198 199 200
- + cols
- - 4
+ Other Ethnicity Only (201, 3)
- + rows
- - 201 202 203 204 205 206 207 208
- + cols
- - 4
+ European/Maori (209, 3)
- + rows
- - 209 210 211 212 213 214 215 216
- + cols
- - 4
+ Two or More Groups Not Elsewhere Included (217, 3)

```

```

- + rows
- - 217 218 219 220 221 222 223 224
- + cols
- - 4
+ Residual Categories (225, 3)
- + rows
- - 225 226 227 228 229 230 231 232
- + cols
- - 4
+ Total All Ethnic Groups (233, 3)
- + rows
- - 233 234 235 236 237 238 239 240
- + cols
- - 4
> plist
$rows
[1] 161 162 163 164 165 166 167 168

$cols
[1] 4

> res
+ Persons Employed in Labour Force (161, 4)
+ Persons Unemployed in Labour Force (162, 4)
+ Not in Labour Force (163, 4)
+ Working Age Population (164, 4)
+ Labour Force Participation Rate (165, 4)
+ Unemployment Rate (166, 4)
+ Employment Rate (167, 4)
+ Total Labour Force (168, 4)
> plist
$rows
[1] 169 170 171 172 173 174 175 176

$cols
[1] 4

> res
+ Persons Employed in Labour Force (169, 4)
+ Persons Unemployed in Labour Force (170, 4)
+ Not in Labour Force (171, 4)
+ Working Age Population (172, 4)
+ Labour Force Participation Rate (173, 4)
+ Unemployment Rate (174, 4)
+ Employment Rate (175, 4)
+ Total Labour Force (176, 4)
> plist
$rows
[1] 177 178 179 180 181 182 183 184

$cols
[1] 4

> res
+ Persons Employed in Labour Force (177, 4)
+ Persons Unemployed in Labour Force (178, 4)
+ Not in Labour Force (179, 4)

```

```

+ Working Age Population (180, 4)
+ Labour Force Participation Rate (181, 4)
+ Unemployment Rate (182, 4)
+ Employment Rate (183, 4)
+ Total Labour Force (184, 4)
> plist
$rows
[1] 185 186 187 188 189 190 191 192

$cols
[1] 4

> res
+ Persons Employed in Labour Force (185, 4)
+ Persons Unemployed in Labour Force (186, 4)
+ Not in Labour Force (187, 4)
+ Working Age Population (188, 4)
+ Labour Force Participation Rate (189, 4)
+ Unemployment Rate (190, 4)
+ Employment Rate (191, 4)
+ Total Labour Force (192, 4)
> plist
$rows
[1] 193 194 195 196 197 198 199 200

$cols
[1] 4

> res
+ Persons Employed in Labour Force (193, 4)
+ Persons Unemployed in Labour Force (194, 4)
+ Not in Labour Force (195, 4)
+ Working Age Population (196, 4)
+ Labour Force Participation Rate (197, 4)
+ Unemployment Rate (198, 4)
+ Employment Rate (199, 4)
+ Total Labour Force (200, 4)
> plist
$rows
[1] 201 202 203 204 205 206 207 208

$cols
[1] 4

> res
+ Persons Employed in Labour Force (201, 4)
+ Persons Unemployed in Labour Force (202, 4)
+ Not in Labour Force (203, 4)
+ Working Age Population (204, 4)
+ Labour Force Participation Rate (205, 4)
+ Unemployment Rate (206, 4)
+ Employment Rate (207, 4)
+ Total Labour Force (208, 4)
> plist
$rows
[1] 209 210 211 212 213 214 215 216

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (209, 4)
+ Persons Unemployed in Labour Force (210, 4)
+ Not in Labour Force (211, 4)
+ Working Age Population (212, 4)
+ Labour Force Participation Rate (213, 4)
+ Unemployment Rate (214, 4)
+ Employment Rate (215, 4)
+ Total Labour Force (216, 4)
> plist
$rows
[1] 217 218 219 220 221 222 223 224

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (217, 4)
+ Persons Unemployed in Labour Force (218, 4)
+ Not in Labour Force (219, 4)
+ Working Age Population (220, 4)
+ Labour Force Participation Rate (221, 4)
+ Unemployment Rate (222, 4)
+ Employment Rate (223, 4)
+ Total Labour Force (224, 4)
> plist
$rows
[1] 225 226 227 228 229 230 231 232

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (225, 4)
+ Persons Unemployed in Labour Force (226, 4)
+ Not in Labour Force (227, 4)
+ Working Age Population (228, 4)
+ Labour Force Participation Rate (229, 4)
+ Unemployment Rate (230, 4)
+ Employment Rate (231, 4)
+ Total Labour Force (232, 4)
> plist
$rows
[1] 233 234 235 236 237 238 239 240

```

```

$cols
[1] 4

> res
+ Persons Employed in Labour Force (233, 4)
+ Persons Unemployed in Labour Force (234, 4)
+ Not in Labour Force (235, 4)
+ Working Age Population (236, 4)
+ Labour Force Participation Rate (237, 4)

```

```

+ Unemployment Rate (238, 4)
+ Employment Rate (239, 4)
+ Total Labour Force (240, 4)
> colplist
+ Male (1, 2)
- + European Only (1, 3)
- - + Persons Employed in Labour Force (1, 4)
- - + Persons Unemployed in Labour Force (2, 4)
- - + Not in Labour Force (3, 4)
- - + Working Age Population (4, 4)
- - + Labour Force Participation Rate (5, 4)
- - + Unemployment Rate (6, 4)
- - + Employment Rate (7, 4)
- - + Total Labour Force (8, 4)
+ Maori Only (9, 3)
- - + Persons Employed in Labour Force (9, 4)
- - + Persons Unemployed in Labour Force (10, 4)
- - + Not in Labour Force (11, 4)
- - + Working Age Population (12, 4)
- - + Labour Force Participation Rate (13, 4)
- - + Unemployment Rate (14, 4)
- - + Employment Rate (15, 4)
- - + Total Labour Force (16, 4)
+ Pacific Peoples Only (17, 3)
- - + Persons Employed in Labour Force (17, 4)
- - + Persons Unemployed in Labour Force (18, 4)
- - + Not in Labour Force (19, 4)
- - + Working Age Population (20, 4)
- - + Labour Force Participation Rate (21, 4)
- - + Unemployment Rate (22, 4)
- - + Employment Rate (23, 4)
- - + Total Labour Force (24, 4)
+ Asian Only (25, 3)
- - + Persons Employed in Labour Force (25, 4)
- - + Persons Unemployed in Labour Force (26, 4)
- - + Not in Labour Force (27, 4)
- - + Working Age Population (28, 4)
- - + Labour Force Participation Rate (29, 4)
- - + Unemployment Rate (30, 4)
- - + Employment Rate (31, 4)
- - + Total Labour Force (32, 4)
+ MELAA Only (33, 3)
- - + Persons Employed in Labour Force (33, 4)
- - + Persons Unemployed in Labour Force (34, 4)
- - + Not in Labour Force (35, 4)
- - + Working Age Population (36, 4)
- - + Labour Force Participation Rate (37, 4)
- - + Unemployment Rate (38, 4)
- - + Employment Rate (39, 4)
- - + Total Labour Force (40, 4)
+ Other Ethnicity Only (41, 3)
- - + Persons Employed in Labour Force (41, 4)
- - + Persons Unemployed in Labour Force (42, 4)
- - + Not in Labour Force (43, 4)
- - + Working Age Population (44, 4)
- - + Labour Force Participation Rate (45, 4)
- - + Unemployment Rate (46, 4)

```

- - + Employment Rate (47, 4)
- - + Total Labour Force (48, 4)
- + European/Maori (49, 3)
- - + Persons Employed in Labour Force (49, 4)
- - + Persons Unemployed in Labour Force (50, 4)
- - + Not in Labour Force (51, 4)
- - + Working Age Population (52, 4)
- - + Labour Force Participation Rate (53, 4)
- - + Unemployment Rate (54, 4)
- - + Employment Rate (55, 4)
- - + Total Labour Force (56, 4)
- + Two or More Groups Not Elsewhere Included (57, 3)
- - + Persons Employed in Labour Force (57, 4)
- - + Persons Unemployed in Labour Force (58, 4)
- - + Not in Labour Force (59, 4)
- - + Working Age Population (60, 4)
- - + Labour Force Participation Rate (61, 4)
- - + Unemployment Rate (62, 4)
- - + Employment Rate (63, 4)
- - + Total Labour Force (64, 4)
- + Residual Categories (65, 3)
- - + Persons Employed in Labour Force (65, 4)
- - + Persons Unemployed in Labour Force (66, 4)
- - + Not in Labour Force (67, 4)
- - + Working Age Population (68, 4)
- - + Labour Force Participation Rate (69, 4)
- - + Unemployment Rate (70, 4)
- - + Employment Rate (71, 4)
- - + Total Labour Force (72, 4)
- + Total All Ethnic Groups (73, 3)
- - + Persons Employed in Labour Force (73, 4)
- - + Persons Unemployed in Labour Force (74, 4)
- - + Not in Labour Force (75, 4)
- - + Working Age Population (76, 4)
- - + Labour Force Participation Rate (77, 4)
- - + Unemployment Rate (78, 4)
- - + Employment Rate (79, 4)
- - + Total Labour Force (80, 4)
- + Female (81, 2)
- + European Only (81, 3)
- - + Persons Employed in Labour Force (81, 4)
- - + Persons Unemployed in Labour Force (82, 4)
- - + Not in Labour Force (83, 4)
- - + Working Age Population (84, 4)
- - + Labour Force Participation Rate (85, 4)
- - + Unemployment Rate (86, 4)
- - + Employment Rate (87, 4)
- - + Total Labour Force (88, 4)
- + Maori Only (89, 3)
- - + Persons Employed in Labour Force (89, 4)
- - + Persons Unemployed in Labour Force (90, 4)
- - + Not in Labour Force (91, 4)
- - + Working Age Population (92, 4)
- - + Labour Force Participation Rate (93, 4)
- - + Unemployment Rate (94, 4)
- - + Employment Rate (95, 4)
- - + Total Labour Force (96, 4)

- + Pacific Peoples Only (97, 3)
- - + Persons Employed in Labour Force (97, 4)
- - + Persons Unemployed in Labour Force (98, 4)
- - + Not in Labour Force (99, 4)
- - + Working Age Population (100, 4)
- - + Labour Force Participation Rate (101, 4)
- - + Unemployment Rate (102, 4)
- - + Employment Rate (103, 4)
- - + Total Labour Force (104, 4)
- + Asian Only (105, 3)
- - + Persons Employed in Labour Force (105, 4)
- - + Persons Unemployed in Labour Force (106, 4)
- - + Not in Labour Force (107, 4)
- - + Working Age Population (108, 4)
- - + Labour Force Participation Rate (109, 4)
- - + Unemployment Rate (110, 4)
- - + Employment Rate (111, 4)
- - + Total Labour Force (112, 4)
- + MELAA Only (113, 3)
- - + Persons Employed in Labour Force (113, 4)
- - + Persons Unemployed in Labour Force (114, 4)
- - + Not in Labour Force (115, 4)
- - + Working Age Population (116, 4)
- - + Labour Force Participation Rate (117, 4)
- - + Unemployment Rate (118, 4)
- - + Employment Rate (119, 4)
- - + Total Labour Force (120, 4)
- + Other Ethnicity Only (121, 3)
- - + Persons Employed in Labour Force (121, 4)
- - + Persons Unemployed in Labour Force (122, 4)
- - + Not in Labour Force (123, 4)
- - + Working Age Population (124, 4)
- - + Labour Force Participation Rate (125, 4)
- - + Unemployment Rate (126, 4)
- - + Employment Rate (127, 4)
- - + Total Labour Force (128, 4)
- + European/Maori (129, 3)
- - + Persons Employed in Labour Force (129, 4)
- - + Persons Unemployed in Labour Force (130, 4)
- - + Not in Labour Force (131, 4)
- - + Working Age Population (132, 4)
- - + Labour Force Participation Rate (133, 4)
- - + Unemployment Rate (134, 4)
- - + Employment Rate (135, 4)
- - + Total Labour Force (136, 4)
- + Two or More Groups Not Elsewhere Included (137, 3)
- - + Persons Employed in Labour Force (137, 4)
- - + Persons Unemployed in Labour Force (138, 4)
- - + Not in Labour Force (139, 4)
- - + Working Age Population (140, 4)
- - + Labour Force Participation Rate (141, 4)
- - + Unemployment Rate (142, 4)
- - + Employment Rate (143, 4)
- - + Total Labour Force (144, 4)
- + Residual Categories (145, 3)
- - + Persons Employed in Labour Force (145, 4)
- - + Persons Unemployed in Labour Force (146, 4)

- - + Not in Labour Force (147, 4)
- - + Working Age Population (148, 4)
- - + Labour Force Participation Rate (149, 4)
- - + Unemployment Rate (150, 4)
- - + Employment Rate (151, 4)
- - + Total Labour Force (152, 4)
- + Total All Ethnic Groups (153, 3)
- - + Persons Employed in Labour Force (153, 4)
- - + Persons Unemployed in Labour Force (154, 4)
- - + Not in Labour Force (155, 4)
- - + Working Age Population (156, 4)
- - + Labour Force Participation Rate (157, 4)
- - + Unemployment Rate (158, 4)
- - + Employment Rate (159, 4)
- - + Total Labour Force (160, 4)
- + Total Both Sexes (161, 2)
- + European Only (161, 3)
- - + Persons Employed in Labour Force (161, 4)
- - + Persons Unemployed in Labour Force (162, 4)
- - + Not in Labour Force (163, 4)
- - + Working Age Population (164, 4)
- - + Labour Force Participation Rate (165, 4)
- - + Unemployment Rate (166, 4)
- - + Employment Rate (167, 4)
- - + Total Labour Force (168, 4)
- + Maori Only (169, 3)
- - + Persons Employed in Labour Force (169, 4)
- - + Persons Unemployed in Labour Force (170, 4)
- - + Not in Labour Force (171, 4)
- - + Working Age Population (172, 4)
- - + Labour Force Participation Rate (173, 4)
- - + Unemployment Rate (174, 4)
- - + Employment Rate (175, 4)
- - + Total Labour Force (176, 4)
- + Pacific Peoples Only (177, 3)
- - + Persons Employed in Labour Force (177, 4)
- - + Persons Unemployed in Labour Force (178, 4)
- - + Not in Labour Force (179, 4)
- - + Working Age Population (180, 4)
- - + Labour Force Participation Rate (181, 4)
- - + Unemployment Rate (182, 4)
- - + Employment Rate (183, 4)
- - + Total Labour Force (184, 4)
- + Asian Only (185, 3)
- - + Persons Employed in Labour Force (185, 4)
- - + Persons Unemployed in Labour Force (186, 4)
- - + Not in Labour Force (187, 4)
- - + Working Age Population (188, 4)
- - + Labour Force Participation Rate (189, 4)
- - + Unemployment Rate (190, 4)
- - + Employment Rate (191, 4)
- - + Total Labour Force (192, 4)
- + MELAA Only (193, 3)
- - + Persons Employed in Labour Force (193, 4)
- - + Persons Unemployed in Labour Force (194, 4)
- - + Not in Labour Force (195, 4)
- - + Working Age Population (196, 4)

```

- - + Labour Force Participation Rate (197, 4)
- - + Unemployment Rate (198, 4)
- - + Employment Rate (199, 4)
- - + Total Labour Force (200, 4)
- + Other Ethnicity Only (201, 3)
- - + Persons Employed in Labour Force (201, 4)
- - + Persons Unemployed in Labour Force (202, 4)
- - + Not in Labour Force (203, 4)
- - + Working Age Population (204, 4)
- - + Labour Force Participation Rate (205, 4)
- - + Unemployment Rate (206, 4)
- - + Employment Rate (207, 4)
- - + Total Labour Force (208, 4)
- + European/Maori (209, 3)
- - + Persons Employed in Labour Force (209, 4)
- - + Persons Unemployed in Labour Force (210, 4)
- - + Not in Labour Force (211, 4)
- - + Working Age Population (212, 4)
- - + Labour Force Participation Rate (213, 4)
- - + Unemployment Rate (214, 4)
- - + Employment Rate (215, 4)
- - + Total Labour Force (216, 4)
- + Two or More Groups Not Elsewhere Included (217, 3)
- - + Persons Employed in Labour Force (217, 4)
- - + Persons Unemployed in Labour Force (218, 4)
- - + Not in Labour Force (219, 4)
- - + Working Age Population (220, 4)
- - + Labour Force Participation Rate (221, 4)
- - + Unemployment Rate (222, 4)
- - + Employment Rate (223, 4)
- - + Total Labour Force (224, 4)
- + Residual Categories (225, 3)
- - + Persons Employed in Labour Force (225, 4)
- - + Persons Unemployed in Labour Force (226, 4)
- - + Not in Labour Force (227, 4)
- - + Working Age Population (228, 4)
- - + Labour Force Participation Rate (229, 4)
- - + Unemployment Rate (230, 4)
- - + Employment Rate (231, 4)
- - + Total Labour Force (232, 4)
- + Total All Ethnic Groups (233, 3)
- - + Persons Employed in Labour Force (233, 4)
- - + Persons Unemployed in Labour Force (234, 4)
- - + Not in Labour Force (235, 4)
- - + Working Age Population (236, 4)
- - + Labour Force Participation Rate (237, 4)
- - + Unemployment Rate (238, 4)
- - + Employment Rate (239, 4)
- - + Total Labour Force (240, 4)
> res[1:4,]
UNKNOWN      UNKNOWN UNKNOWN Persons Employed in Labour Force
1   Male European Only  2007Q4                        855.8
2   Male European Only  2008Q1                        863.0
3   Male European Only  2008Q2                        850.1
4   Male European Only  2008Q3                        839.6
Persons Unemployed in Labour Force Not in Labour Force Working Age Population
1                                20.0                      280.0                    1155.8

```

2		25.4	283.5	1171.9
3		26.0	280.7	1156.8
4		29.8	285.9	1155.3
	Labour Force Participation Rate	Unemployment Rate	Employment Rate	
1		75.8	2.3	74.0
2		75.8	2.9	73.6
3		75.7	3.0	73.5
4		75.2	3.4	72.7
	Total Labour Force			
1	875.8			
2	888.5			
3	876.1			
4	869.4			

8.4 ToyExByEmptyBelow.csv

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1	Row Child1	10	20	30	40
3		Row Child2	11	21	31	41
4	Row Parent2	Row Child1	12	22	32	42
5		Row Child2	13	23	33	43

```

> rowData
[1] 2 5
> colData
[1] 3 6
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5

> colslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> Patvec
[1] NA
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5

> colslist
$label
[1] 1 2

$data
$data[1]
[1] 3 4 5 6

> res
+ Row Parent1 (1, 1)
- + rows
- - 1 2
- + cols
- - 2
+ Row Parent2 (3, 1)
- + rows
- - 3 4
- + cols
- - 2

```

```

> plist
$rows
[1] 1 2

$cols
[1] 2

> res
+ Row Child1 (1, 2)
+ Row Child2 (2, 2)
> plist
$rows
[1] 3 4

$cols
[1] 2

> res
+ Row Child1 (3, 2)
+ Row Child2 (4, 2)
> rowplist
+ Row Parent1 (1, 1)
- + Row Child1 (1, 2)
- + Row Child2 (2, 2)
+ Row Parent2 (3, 1)
- + Row Child1 (3, 2)
- + Row Child2 (4, 2)
> rowvecs[1:4,]
      [,1]      [,2]
[1,] "Row Parent1" "Row Child1"
[2,] "Row Parent1" "Row Child2"
[3,] "Row Parent2" "Row Child1"
[4,] "Row Parent2" "Row Child2"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> colplist
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> res[1:4,]
  UNKNOWN      UNKNOWN Column 1 Column 2 Column 3 Column 4
1 Row Parent1 Row Child1      10      20      30      40
2 Row Parent1 Row Child2      11      21      31      41
3 Row Parent2 Row Child1      12      22      32      42
4 Row Parent2 Row Child2      13      23      33      43

```

8.5 ToyExByEmptyBelowT.csv

	1	2	3	4	5
1		Row Parent1		Row Parent2	
2		Row Child1	Row Child2	Row Child1	Row Child2
3	Column 1	10	11	12	13
4	Column 2	20	21	22	23
5	Column 3	30	31	32	33
6	Column 4	40	41	42	43

```

> rowData
[1] 3 6
> colData
[1] 2 5
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
[1] 2 3 4 5

> Patvec
[1] 2 2
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
$data[1]
[1] 2 3

$data[2]
[1] 4 5

> plist
$rows
[1] 1 2 3 4

$cols

```

```

[1] 1

> res
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> rowplist
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> rowvecs[1:4,]
[1] "Column 1" "Column 2" "Column 3" "Column 4"
> res
+ Row Parent1 (1, 1)
- + rows
- - 1 2
- + cols
- - 2
+ Row Parent2 (3, 1)
- + rows
- - 3 4
- + cols
- - 2
> plist
$rows
[1] 1 2

$cols
[1] 2

> res
+ Row Child1 (1, 2)
+ Row Child2 (2, 2)
> plist
$rows
[1] 3 4

$cols
[1] 2

> res
+ Row Child1 (3, 2)
+ Row Child2 (4, 2)
> colplist
+ Row Parent1 (1, 1)
- + Row Child1 (1, 2)
- + Row Child2 (2, 2)
+ Row Parent2 (3, 1)
- + Row Child1 (3, 2)
- + Row Child2 (4, 2)
> res[1:4,]
  UNKNOWN UNKNOWN Row Child1 Row Child2
1 Row Parent1 Column 1      10      11
2 Row Parent1 Column 2      20      21
3 Row Parent1 Column 3      30      31

```

4 Row Parent1 Column 4

40

41

8.6 ToyExByEmptyRight1.csv

	1	2	3	4	5	6	7
1				Column 1	Column 2	Column 3	Column 4
2	Row Parent1			10	20	30	40
3	Row Child1	Row Child-Child1		11	21	31	41
4	Row Child2	Row Child-Child2		12	22	32	42
5	Row Parent2			13	23	33	43
6	Row Child1	Row Child-Child1		14	24	34	44
7		Row Child-Child2		15	25	35	45

```

> rowData
[1] 2 7
> colData
[1] 4 7
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7

> colslist
$label
[1] 1 2

$data
[1] 4 5 6 7

> Patvec
[1] NA
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7

> colslist
$label
[1] 1 2

$data
$data[1]
[1] 4 5 6 7

> res
+ Row Parent1 (1, 1)
- + rows
- - 2 3
- + cols
- - 1 2
+ Row Parent2 (4, 1)
- + rows
- - 5 6
- + cols

```

```

- - 1 2
> res
+ Row Child1 (2, 1)
- + rows
- - 2
- + cols
- - 2
+ Row Child2 (3, 1)
- + rows
- - 3
- + cols
- - 2
> plist
$rows
[1] 2

$cols
[1] 2

> res
+ Row Child-Child1 (2, 2)
> plist
$rows
[1] 3

$cols
[1] 2

> res
+ Row Child-Child2 (3, 2)
> res
+ Row Child1 (5, 1)
- + rows
- - 5 6
- + cols
- - 2
> plist
$rows
[1] 5 6

$cols
[1] 2

> res
+ Row Child-Child1 (5, 2)
+ Row Child-Child2 (6, 2)
> rowplist
+ Row Parent1 (1, 1)
- + Row Child1 (2, 1)
- - + Row Child-Child1 (2, 2)
- + Row Child2 (3, 1)
- - + Row Child-Child2 (3, 2)
+ Row Parent2 (4, 1)
- + Row Child1 (5, 1)
- - + Row Child-Child1 (5, 2)
- - + Row Child-Child2 (6, 2)
> rowvecs[1:4,]

```

```

      [,1]      [,2]      [,3]
V2 "Row Parent1" "Row Child1" "Row Child-Child1"
V2 "Row Parent1" "Row Child2" "Row Child-Child2"
   "Row Parent2" "Row Child1" "Row Child-Child1"
   "Row Parent2" "Row Child1" "Row Child-Child2"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> colplist
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN Column 1 Column 2 Column 3 Column 4
1 Row Parent1 Row Child1 Row Child-Child1      11      21      31      41
2 Row Parent1 Row Child2 Row Child-Child2      12      22      32      42
3 Row Parent2 Row Child1 Row Child-Child1      14      24      34      44
4 Row Parent2 Row Child1 Row Child-Child2      15      25      35      45

```

8.7 ToyExByEmptyRight2.csv

	1	2	3	4	5	6
1			Column 1	Column 2	Column 3	Column 4
2	Row Parent1		10	20	30	40
3		Row Child1	11	21	31	41
4		Row Child2	12	22	32	42
5	Row Parent2		13	23	33	43
6		Row Child1	14	24	34	44
7		Row Child2	15	25	35	45

```

> rowData
[1] 2 7
> colData
[1] 3 6
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7

> colslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> Patvec
[1] NA
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7

> colslist
$label
[1] 1 2

$data
$data[1]
[1] 3 4 5 6

> res
+ Row Parent1 (1, 1)
- + rows
- - 2 3
- + cols
- - 1 2
+ Row Parent2 (4, 1)
- + rows
- - 5 6

```

```

- + cols
- - 1 2
> plist
$rows
[1] 2 3

$cols
[1] 2

> res
+ Row Child1 (2, 2)
+ Row Child2 (3, 2)
> plist
$rows
[1] 5 6

$cols
[1] 2

> res
+ Row Child1 (5, 2)
+ Row Child2 (6, 2)
> rowplist
+ Row Parent1 (1, 1)
- + Row Child1 (2, 2)
- + Row Child2 (3, 2)
+ Row Parent2 (4, 1)
- + Row Child1 (5, 2)
- + Row Child2 (6, 2)
> rowvecs[1:4,]
      [,1]      [,2]
[1,] "Row Parent1" "Row Child1"
[2,] "Row Parent1" "Row Child2"
[3,] "Row Parent2" "Row Child1"
[4,] "Row Parent2" "Row Child2"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> colplist
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> res[1:4,]
      UNKNOWN      UNKNOWN Column 1 Column 2 Column 3 Column 4
1 Row Parent1 Row Child1      11      21      31      41
2 Row Parent1 Row Child2      12      22      32      42
3 Row Parent2 Row Child1      14      24      34      44

```

4	Row	Parent2	Row	Child2	15	25	35	45
---	-----	---------	-----	--------	----	----	----	----

8.8 ToyExByEmptyRight3.csv

	1	2	3	4	5	6	7	8
1				Column 1	Column 2	Column 3	Column 4	
2	Row Super-Parent1			10	20	30	40	
3	Row Parent1			11	21	31	41	
4	Row Child1	Row Child-Child1		12	22	32	42	
5	Row Parent2			13	23	33	43	
6	Row Child1	Row Child-Child1		14	24	34	44	
7	Row Super-Parent2			15	25	35	45	
8	Row Parent1			16	26	36	46	
9	Row Child1	Row Child-Child1		17	27	37	47	
10	Row Parent2			18	28	38	48	
11	Row Child1	Row Child-Child1		19	29	39	49	

```

> rowData
[1] 2 11
> colData
[1] 4 7
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7 8 9 10 11

> colslist
$label
[1] 1 2

$data
[1] 4 5 6 7

> Patvec
[1] NA
> rowslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7 8 9 10 11

> colslist
$label
[1] 1 2

$data
$data[1]
[1] 4 5 6 7

> res
+ Row Super-Parent1 (1, 1)
- + rows
- - 2 3 4 5
- + cols
- - 1 2
+ Row Super-Parent2 (6, 1)

```

```

- + rows
- - 7 8 9 10
- + cols
- - 1 2
> res
+ Row Parent1 (2, 1)
- + rows
- - 3
- + cols
- - 1 2
+ Row Parent2 (4, 1)
- + rows
- - 5
- + cols
- - 1 2
> plist
$rows
[1] 3

$cols
[1] 1 2

> res
+ Row Child1 (3, 1)
- + Row Child-Child1 (3, 2)
> plist
$rows
[1] 5

$cols
[1] 1 2

> res
+ Row Child1 (5, 1)
- + Row Child-Child1 (5, 2)
> res
+ Row Parent1 (7, 1)
- + rows
- - 8
- + cols
- - 1 2
+ Row Parent2 (9, 1)
- + rows
- - 10
- + cols
- - 1 2
> plist
$rows
[1] 8

$cols
[1] 1 2

> res
+ Row Child1 (8, 1)
- + Row Child-Child1 (8, 2)
> plist

```



```

$rows
[1] 10

$cols
[1] 1 2

> res
+ Row Child1 (10, 1)
- + Row Child-Child1 (10, 2)
> rowplist
+ Row Super-Parent1 (1, 1)
- + Row Parent1 (2, 1)
- - + Row Child1 (3, 1)
- - - + Row Child-Child1 (3, 2)
- + Row Parent2 (4, 1)
- - + Row Child1 (5, 1)
- - - + Row Child-Child1 (5, 2)
+ Row Super-Parent2 (6, 1)
- + Row Parent1 (7, 1)
- - + Row Child1 (8, 1)
- - - + Row Child-Child1 (8, 2)
- + Row Parent2 (9, 1)
- - + Row Child1 (10, 1)
- - - + Row Child-Child1 (10, 2)
> rowvecs[1:4,]
      [,1]      [,2]      [,3]      [,4]
V2 "Row Super-Parent1" "Row Parent1" "Row Child1" "Row Child-Child1"
V2 "Row Super-Parent1" "Row Parent2" "Row Child1" "Row Child-Child1"
V2 "Row Super-Parent2" "Row Parent1" "Row Child1" "Row Child-Child1"
V2 "Row Super-Parent2" "Row Parent2" "Row Child1" "Row Child-Child1"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> colplist
+ Column 1 (1, 1)
+ Column 2 (2, 1)
+ Column 3 (3, 1)
+ Column 4 (4, 1)
> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN Column 1 Column 2
1 Row Super-Parent1 Row Parent1 Row Child1 Row Child-Child1      12      22
2 Row Super-Parent1 Row Parent2 Row Child1 Row Child-Child1      14      24
3 Row Super-Parent2 Row Parent1 Row Child1 Row Child-Child1      17      27
4 Row Super-Parent2 Row Parent2 Row Child1 Row Child-Child1      19      29
      Column 3 Column 4
1          32      42
2          34      44
3          37      47

```

4

39

49

8.9 ToyExComplete.csv

	1	2	3	4	5	6	7	8	9	10	11
1	MISC INFORMATION										
2					Column Parent1				Column Parent2		
3				Column	Column	Column	Column	Column	Column	Column	Column
4				Child1	Child2	Child3	Child4	Child1	Child2	Child3	Child4
5	Row Super-Parent			10	20	30	40	50	60	70	80
6	Row Parent1			11	21	31	41	51	61	71	81
7	Row Child1			12	22	32	42	52	62	72	82
8	Row Child-Child1			13	23	33	43	53	63	73	83
9	Row Child-Child2			14	24	34	44	54	64	74	84
10	Row Child-Child1			15	25	35	45	55	65	75	85
11	Row Child-Child2			16	26	36	46	56	66	76	86
12	Row Parent2			17	27	37	47	57	67	77	87
13	Row Child1			18	28	38	48	58	68	78	88
14	Row Child-Child1			19	29	39	49	59	69	79	89
15	Row Child-Child2										
16	MISC INFORMATION										

```

> rowData
[1] 5 14
> colData
[1] 4 11
> rowslist
$label
[1] 1 2 3 4

$data
[1] 5 6 7 8 9 10 11 12 13 14

> colslist
$label
[1] 1 2

$data
[1] 4 5 6 7 8 9 10 11

> Patvec
[1] 4 1 4
> rowslist
$label
[1] 1 2 3 4

$data
[1] 5 6 7 8 9 10 11 12 13 14

> colslist
$label
[1] 1 2

$data
$data[1]
[1] 4 5 6 7

$data[2]
[1] 8 9 10 11

> res
+ Row Super-Parent (1, 1)
- + rows
- - 2 3 4 5 6 7 8 9 10

```

```

- + cols
- - 1 2
> res
+ Row Parent1 (2, 1)
- + rows
- - 3 4 5 6
- + cols
- - 1 2
+ Row Parent2 (7, 1)
- + rows
- - 8 9 10
- + cols
- - 1 2
> res
+ Row Child1 (3, 1)
- + rows
- - 3 4
- + cols
- - 2
+ Row Child2 (5, 1)
- + rows
- - 5 6
- + cols
- - 2
> plist
$rows
[1] 3 4

$cols
[1] 2

> res
+ Row Child-Child1 (3, 2)
+ Row Child-Child2 (4, 2)
> plist
$rows
[1] 5 6

$cols
[1] 2

> res
+ Row Child-Child1 (5, 2)
+ Row Child-Child2 (6, 2)
> res
+ Row Child1 (8, 1)
- + rows
- - 8 9
- + cols
- - 2
+ Row Child2 (10, 1)
- + rows
- - 10
- + cols
- - 2
> plist
$rows

```

```

[1] 8 9

$cols
[1] 2

> res
+ Row Child-Child1 (8, 2)
+ Row Child-Child2 (9, 2)
> plist
$rows
[1] 10

$cols
[1] 2

> res
+ Row Child-Child2 (10, 2)
> rowplist
+ Row Super-Parent (1, 1)
- + Row Parent1 (2, 1)
- - + Row Child1 (3, 1)
- - - + Row Child-Child1 (3, 2)
- - - + Row Child-Child2 (4, 2)
- - + Row Child2 (5, 1)
- - - + Row Child-Child1 (5, 2)
- - - + Row Child-Child2 (6, 2)
- + Row Parent2 (7, 1)
- - + Row Child1 (8, 1)
- - - + Row Child-Child1 (8, 2)
- - - + Row Child-Child2 (9, 2)
- - + Row Child2 (10, 1)
- - - + Row Child-Child2 (10, 2)
> rowvecs[1:4,]
      [,1]      [,2]      [,3]      [,4]
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child1" "Row Child-Child2"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child1"
"Row Super-Parent" "Row Parent1" "Row Child2" "Row Child-Child2"
> res
+ Column Parent1 (1, 2)
- + rows
- - 1 2 3 4
- + cols
- - 3
+ Column Parent2 (5, 2)
- + rows
- - 5 6 7 8
- + cols
- - 3
> plist
$rows
[1] 1 2 3 4

$cols
[1] 3

> res

```

```

+ Column Child1 (1, 3)
+ Column Child2 (2, 3)
+ Column Child3 (3, 3)
+ Column Child4 (4, 3)
> plist
$rows
[1] 5 6 7 8

$cols
[1] 3

> res
+ Column Child1 (5, 3)
+ Column Child2 (6, 3)
+ Column Child3 (7, 3)
+ Column Child4 (8, 3)
> colplist
+ Column Parent1 (1, 2)
- + Column Child1 (1, 3)
- + Column Child2 (2, 3)
- + Column Child3 (3, 3)
- + Column Child4 (4, 3)
+ Column Parent2 (5, 2)
- + Column Child1 (5, 3)
- + Column Child2 (6, 3)
- + Column Child3 (7, 3)
- + Column Child4 (8, 3)
> res[1:4,]
      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN      UNKNOWN
1 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child1
2 Column Parent1 Row Super-Parent Row Parent1 Row Child1 Row Child-Child2
3 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child1
4 Column Parent1 Row Super-Parent Row Parent1 Row Child2 Row Child-Child2
  Column Child1 Column Child2 Column Child3 Column Child4
1          12          22          32          42
2          13          23          33          43
3          14          24          34          44
4          15          25          35          45

```

8.10 ToyExFindSingleTable.csv

	1	2	3	4	5
1	MISC INFORMATION				
2	MISC INFORMATION				
3		Column 1	Column 2	Column 3	Column 4
4	Row 1	10	20	30	40
5	Row 2	11	21	31	41
6	Row 3	12	22	32	42
7	Row 4	13	23	33	43
8	MISC INFORMATION		MISC INFORMATION		
9	MISC INFORMATION		MISC INFORMATION		

```

> rowData
[1] 4 7
> colData
[1] 2 5
> rowslist
$label
[1] 1 2 3

$data
[1] 4 5 6 7

> colslist
$label
[1] 1

$data
[1] 2 3 4 5

> Patvec
[1] NA
> rowslist
$label
[1] 1 2 3

$data
[1] 4 5 6 7

> colslist
$label
[1] 1

$data
$data[1]
[1] 2 3 4 5

> plist
$rows
[1] 1 2 3 4

$cols

```

```

[1] 1

> res
+ Row 1 (1, 1)
+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowplist
+ Row 1 (1, 1)
+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowvecs[1:4,]
[1] "Row 1" "Row 2" "Row 3" "Row 4"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 3

> res
+ Column 1 (1, 3)
+ Column 2 (2, 3)
+ Column 3 (3, 3)
+ Column 4 (4, 3)
> colplist
+ Column 1 (1, 3)
+ Column 2 (2, 3)
+ Column 3 (3, 3)
+ Column 4 (4, 3)
> res[1:4,]
  UNKNOWN Column 1 Column 2 Column 3 Column 4
1   Row 1      10      20      30      40
2   Row 2      11      21      31      41
3   Row 3      12      22      32      42
4   Row 4      13      23      33      43

```


8.11 ToyExMisalignedColumnLabel.csv

	1	2	3	4	5	6	7	8	9
1			Column Parent1				Column Parent2		
2		Col Child1	Col Child2	Col Child3	Col Child4	Col Child1	Col Child2	Col Child3	Col Child4
3	Row 1	10	20	30	40	50	60	70	80
4	Row 2	11	21	31	41	51	61	71	81
5	Row 3	12	22	32	42	52	62	72	82
6	Row 4	13	23	33	43	53	63	73	83

```

> rowData
[1] 3 6
> colData
[1] 2 9
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
[1] 2 3 4 5 6 7 8 9

> Patvec
[1] 4 4
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
$data[1]
[1] 2 3 4 5

$data[2]
[1] 6 7 8 9

> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Row 1 (1, 1)

```

```

+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowplist
+ Row 1 (1, 1)
+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowvecs[1:4,]
[1] "Row 1" "Row 2" "Row 3" "Row 4"
> res
+ Column Parent1 (1, 1)
- + rows
- - 1 2 3 4
- + cols
- - 2
+ Column Parent2 (5, 1)
- + rows
- - 5 6 7 8
- + cols
- - 2
> plist
$rows
[1] 1 2 3 4

$cols
[1] 2

> res
+ Col Child1 (1, 2)
+ Col Child2 (2, 2)
+ Col Child3 (3, 2)
+ Col Child4 (4, 2)
> plist
$rows
[1] 5 6 7 8

$cols
[1] 2

> res
+ Col Child1 (5, 2)
+ Col Child2 (6, 2)
+ Col Child3 (7, 2)
+ Col Child4 (8, 2)
> colplist
+ Column Parent1 (1, 1)
- + Col Child1 (1, 2)
- + Col Child2 (2, 2)
- + Col Child3 (3, 2)
- + Col Child4 (4, 2)
+ Column Parent2 (5, 1)
- + Col Child1 (5, 2)
- + Col Child2 (6, 2)
- + Col Child3 (7, 2)
- + Col Child4 (8, 2)
> res[1:4,]

```

	UNKNOWN	UNKNOWN	Col	Child1	Col	Child2	Col	Child3	Col	Child4
1	Column	Parent1	Row 1	10		20		30		40
2	Column	Parent1	Row 2	11		21		31		41
3	Column	Parent1	Row 3	12		22		32		42
4	Column	Parent1	Row 4	13		23		33		43

8.12 ToyExMultiRowColumnLabel.csv

	1	2	3	4	5
1		Column	Column	Column	Column
2		Child1	Child2	Child3	Child4
3	Row 1	10	20	30	40
4	Row 2	11	21	31	41
5	Row 3	12	22	32	42
6	Row 4	13	23	33	43

```

> rowData
[1] 3 6
> colData
[1] 2 5
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
[1] 2 3 4 5

> Patvec
[1] 1 NA
> rowslist
$label
[1] 1 2

$data
[1] 3 4 5 6

> colslist
$label
[1] 1

$data
$data[1]
[1] 2 3 4 5

> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res

```

```

+ Row 1 (1, 1)
+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowplist
+ Row 1 (1, 1)
+ Row 2 (2, 1)
+ Row 3 (3, 1)
+ Row 4 (4, 1)
> rowvecs[1:4,]
[1] "Row 1" "Row 2" "Row 3" "Row 4"
> plist
$rows
[1] 1 2 3 4

$cols
[1] 1

> res
+ Column Child1 (1, 1)
+ Column Child2 (2, 1)
+ Column Child3 (3, 1)
+ Column Child4 (4, 1)
> colplist
+ Column Child1 (1, 1)
+ Column Child2 (2, 1)
+ Column Child3 (3, 1)
+ Column Child4 (4, 1)
> res[1:4,]
UNKNOWN Column Child1 Column Child2 Column Child3 Column Child4
1 Row 1 10 20 30 40
2 Row 2 11 21 31 41
3 Row 3 12 22 32 42
4 Row 4 13 23 33 43

```