Data shaping with Tidy

gicentre/tidy: latest

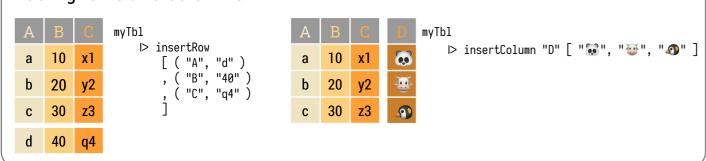
import Tidy exposing (..)

https://package.elm-lang.org/packages/gicentre/tidy/latest/Tidy

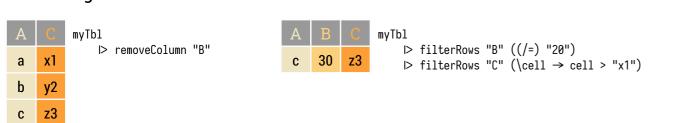
Creating a table from CSV or JSON

		_				
Α	В	C	myTbl : Table mvTbl =	myJson : String myJson =	myTbl : Table mvTbl =	
а	10	х1	"""A,B,C a,10,x1	/"""[{"A":"a", "B":10, "	empty 'C":"x1"},	ΙΛ" [] myleon
b	20	y2	b,20,y2	{"A":"b", "B":20, "	'C":"y2"},	'B" [] myJson
С	30	z3	c,30,z3""" I> fromCSV	{"A":"c", "B":30, "]"""	'C":"z3"}	C" [] myJson

Adding rows and columns



Removing rows and columns



Transforming columns

id	В	C		Α	В	С		Α	В	D	E
a	10	х1		а	10	1		а	10	Х	1
b	20	y2		b	20	2		b	20	у	2
С	30	z3		С	30	3		С	30	Z	3
myTbl I>		eColum	n "A" "id"	myTbl >		lumn "C	(String.right 1)	myTb:	l bised	ct "C	" hea

Exporting columns

Α	В	C	myTbl	_		
а	10	х1	I> strColumn "A"	["a","b","c"]		
b	20	y2	myTb1	 [10 00 00]		
С	30	z3	I> numColumn "B"	[10, 20, 30]		

To make use of table values, e.g. in Vega-Lite, export values column at a time, either as **strings** or **numbers**.

Gathering and Spreading tables

Tidy tables are where each **column** represents a distinct **variable** and each **row** an **observation**. If the same variable is in multiple columns, it can be **gathered** into a tidy table making it "longer and thinner".

Α	LastYr	ThisYr		Α	Year	В
а	10	15		а	2021	10
b	20	а	2022	15		
С	30	35		b	2021	20
-	ssyTbl	, , , , , , , , , , , , , , , , , , , ,		b	2022	25
	> gather "Y [("Las		С	2021	30	
	, ("Thi])	С	2022	25	

The reverse transformation is to **spread** a variable across more than one column, useful for compact table display.

Α	Year	В	myTidyTable	Α	2021	2022
a	2021	10	⊳ spread "Year" "B"	a	10	15
a	2022	15		b	20	25
b	2021	20		С	30	35
b	2022	25				
С	2021	30				
С	2022	25				

To create more than one column of gathered data, create temporary gathered columns (BorC and XorY) and then spread them:

В	X	Ву	Сх	Су	A	BC	X
10	-	15	40	45	а	В	10
20		25	50	55	a	С	40
30 3	3	5	60	65	b	В	20
ssyTbl					b	С	50
> gather "Bo "value"		30	orC"		С	В	30
[("Bx", , ("By",					С	С	60
, ("Cx", , ("Cy",]	("Cx", ("Cy",	,	"CX	")			
⊳ bisect ⊳ spread					XorY")		

Joining Tables

K1	Α	В	K2	С	D
а	1	10	b	5	50
b	2	20	d	6	60
С	3	30	е	7	70
d	4	40	f	8	80
myTbl1			myTb12	2	

Pairs of tables may be related using **keys** (columns containing unique row identifiers) common to both. Different joins prioritise first ('left') or second ('right') tables and determine what to join when keys do not match.

Left and right joins:

K1	Α	В	K2	С	D	
a	1	10				
b	2	20	b	5	50	
С	3	30				
d	4	40	d	6	60	
1 of + 1 o	in (muTh1	1 11/1	\ /	, muTl	٠1 <i>′</i>

b 5 50 b 2 20 d 6 60 d 4 40 e 7 70 f 8 80

leftJoin (myTbl1, "K1") (myTbl2, "K2")

rightJoin (myTbl1, "K1") (myTbl2, "K2")

Inner and outer joins:

K1	Α	В	K2	С	D
b	2	20	b	5	50
d	4	40	d	6	60

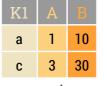
innerJoin (myTbl1, "K1") (myTbl2, "K2")

K1	Α	В	С	D
а	1	10		
b	2	20	5	50
С	3	30		
d	4	40	6	60
е			7	70
f			8	80
outer 1	oin (mvTh	11 "	K1" `

outerJoin (myTbl1, "K1") (myTbl2, "K2")

Table Differences

Rows in one table whose keys are not present in the other table:



leftDiff (myTbl1, "K1") (myTbl2, "K2")

E 7 70 f 8 80

rightDiff (myTbl1, "K1") (myTbl2, "K2")