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

A	B	C
a	10	x1
b	20	y2
c	30	z3

```
myTbl : Table
myTbl =
    """A,B,C
a,10,x1
b,20,y2
c,30,z3"""
    |> fromCSV
```

```
myJson : String
myJson =
    """[
{"A":"a", "B":10, "C":"x1"},
{"A":"b", "B":20, "C":"y2"},
{"A":"c", "B":30, "C":"z3"}
    ]"""
```

```
myTbl : Table
myTbl =
    empty
        |> insertColumnFromJson "A" [] myJson
        |> insertColumnFromJson "B" [] myJson
        |> insertColumnFromJson "C" [] myJson
```

Adding rows and columns

A	B	C
a	10	x1
b	20	y2
c	30	z3
d	40	q4

```
myTbl
    |> insertRow
        [ ( "A", "d" )
        , ( "B", "40" )
        , ( "C", "q4" )
        ]
```

A	B	C	D
a	10	x1	🐼
b	20	y2	🐷
c	30	z3	🐠

```
myTbl
    |> insertColumn "D" [ "🐼", "🐷", "🐠" ]
```

Removing rows and columns

A	C
a	x1
b	y2
c	z3

```
myTbl
    |> removeColumn "B"
```

A	B	C
c	30	z3

```
myTbl
    |> filterRows "B" ((/=) "20")
    |> filterRows "C" (\cell -> cell > "x1")
```

Transforming columns

id	B	C
a	10	x1
b	20	y2
c	30	z3

```
myTbl
    |> renameColumn "A" "id"
```

A	B	C
a	10	1
b	20	2
c	30	3

```
myTbl
    |> mapColumn "C" (String.right 1)
```

A	B	D	E
a	10	x	1
b	20	y	2
c	30	z	3

```
myTbl
    |> bisect "C" headTail ( "D", "E" )
```

Exporting columns

A	B	C
a	10	x1
b	20	y2
c	30	z3

```
myTbl
    |> strColumn "A"
```
⇨ `["a","b","c"]`

```
myTbl
    |> 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".

A	LastYr	ThisYr
a	10	15
b	20	25
c	30	35

```
myMessyTbl
    |> gather "Year" "B"
        [ ( "LastYr", "2021" )
        , ( "ThisYr", "2022" )
        ]
```

⇨

A	Year	B
a	2021	10
a	2022	15
b	2021	20
b	2022	25
c	2021	30
c	2022	25

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

A	Year	B
a	2021	10
a	2022	15
b	2021	20
b	2022	25
c	2021	30
c	2022	25

```
myTidyTable
    |> spread "Year" "B"
```

⇨

A	2021	2022
a	10	15
b	20	25
c	30	35

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

A	Bx	By	Cx	Cy
a	10	15	40	45
b	20	25	50	55
c	30	35	60	65

```
myMessyTbl
    |> gather "BorC"
        "value"
        [ ( "Bx", "BX" )
        , ( "By", "BY" )
        , ( "Cx", "CX" )
        , ( "Cy", "CY" )
        ]
    |> bisect "BorC" headTail ( "BC", "XorY" )
    |> spread "XorY" "value"
```

⇨

A	BC	X	Y
a	B	10	15
a	C	40	45
b	B	20	25
b	C	50	55
c	B	30	35
c	C	60	65

Joining Tables

K1	A	B
a	1	10
b	2	20
c	3	30
d	4	40

`myTbl1`

K2	C	D
b	5	50
d	6	60
e	7	70
f	8	80

`myTbl2`

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	A	B	K2	C	D
a	1	10			
b	2	20	b	5	50
c	3	30			
d	4	40	d	6	60

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

K2	C	D	K1	A	B
b	5	50	b	2	20
d	6	60	d	4	40
e	7	70			
f	8	80			

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

Inner and outer joins:

K1	A	B	K2	C	D
b	2	20	b	5	50
d	4	40	d	6	60

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

K1	A	B	C	D
a	1	10		
b	2	20	5	50
c	3	30		
d	4	40	6	60
e			7	70
f			8	80

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

Table Differences

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

K1	A	B
a	1	10
c	3	30

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

K2	C	D
e	7	70
f	8	80

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