Power Query (M): Adding multiple columns using Table.TransformRows

Suppose we have these data

	Group	Team	Won	Draw	GF-GA
1	Group A	Ecuador	1	1	4-3
2	Group A	Netherlands	2	1	5-1
3	Group A	Qatar	0	0	1-7
4	Group A	Senegal	2	0	5-4
5	Group B	England	2	1	9-2
6	Group B	Iran	1	0	4-7
7	Group B	United States	1	2	2-1
8	Group B	Wales	0	1	1-6
9	Group C	Argentina	2	0	5-2
10	Group C	Mexico	1	1	2-3
11	Group C	Poland	1	1	2-2
12	Group C	Saudi Arabia	1	0	3-5
13	Group D	Australia	2	0	3-4
14	Group D	Denmark	0	1	1-3
15	Group D	France	2	0	6-3
16	Group D	Tunisia	1	1	1-1
17	Group E	Costa Rica	1	0	3-11
18	Group E	Germany	1	1	6-5
19	Group E	Japan	2	0	4-3

And we want to add three columns

Won	Draw	GF-GA
1	1	4-3
2	1	5-1
0	0	1-7
2	0	5-4
2	1	9-2
1	0	4-7
1	2	2-1
0	1	1-6
2	0	5-2
1	1	2-3
1	1	2-2
1	0	3-5
2	0	3-4
0	1	1-3
2	0	6-3
1	1	1-1
1	0	3-11
1	1	6-5
2	0	4-3

1. Points = Won * 3 + Draw
2. Goals For = the number before the hyphen in the GF-GA column
3. Goal Diff = in the GF-GA column, the number before the hyphen minus the number after the hyphen

We can use Table.AddColumn three times

```
let
    Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content][[Group],[Team],[Won],[Draw],[#"GF-GA"]],

    //Method1
    AddPoints = Table.AddColumn(Source,
                                "Points",
                                each [Won]*3 + [Draw]),

    AddGF =     Table.AddColumn(AddPoints,
                                "GF",
                                each Number.From(Text.BeforeDelimiter([#"GF-GA"],"-"))),

    Result =    Table.AddColumn(AddGF,
                                "GD",
                                each [GF] - Number.From(Text.AfterDelimiter([#"GF-GA"],"-")))

in
    Result
```

First parameter is the table we want to add the column to

Second parameter is the name for the new column

Third parameter is the function to calculate the new column

Anatomy of Table.AddColumn

```
8
9       AddGF =     Table.AddColumn(AddPoints,
10                                  "GF",
11                                  each Number.From(Text.BeforeDelimiter([#"GF-GA"],"-"))),
12
13      Result =    Table.AddColumn(AddGF,
14                                  "GD",
15                                  each [GF] - Number.From(Text.AfterDelimiter([#"GF-GA"],"-")))
16
```

each in Table.AddColumn refers to the current record in the table

Column names in square brackets refer to fields in the current record. This can also be written as _[GF]

Here we are using Text.BeforeDelimiter and Text.AfterDelimiter to split the [#"GF-GA"] column and take the numbers from either side of the hyphen.

To subtract one from the other, they must be converted using Number.From

We can also use Text.Split in a nested function

The outer function takes one parameter – the delimiter, and returns a function

The inner function takes two parameters – the text to be split and the position in the resulting list to convert to a number

```
4    //Method2
5    SplitAsNumber = (delim as text) as function =>
6                        (text as text, position as number) as number =>
7                            Number.From( Text.Split( text , delim ){position}),
8
9    HyphenSplit = SplitAsNumber("-"),
```

We call the outer function of SplitAsNumber to create the two-parameter function using a hyphen as delimiter, and call it HyphenSplit

The function body splits the text by the delimiter, creating a list. It then selects that item from the list at the requested "position". It then converts the result to a number

Now we can simplify by using HyphenSplit

```
9     HyphenSplit = SplitAsNumber("-"),
10
11    AddPoints =     Table.AddColumn(Source,
12                                    "Points",
13                                    each [Won]*3 + [Draw]),
14
15    AddGFMethod2 =  Table.AddColumn(AddPoints,
16                                    "GF",
17                                    each HyphenSplit([#"GF-GA"],0)),
18
19    Result =        Table.AddColumn(AddGFMethod2,
20                                    "GD",
21                                    each [GF] - HyphenSplit([#"GF-GA"],1))
```

Use HyphenSplit to get the number before the hyphen (position 0)

Use HyphenSplit to get the number after the hyphen (position 1)

But there's another way to add columns...

```
7       HyphenSplit = SplitAsNumber("-"),
8
9       //Method3
10      TransformedRows =
11          Table.TransformRows(
12              Source,
13              each _ & [
14                      Points = [Won]*3 + [Draw],
15                      GF = HyphenSplit([#"GF-GA"],0),
16                      GD = GF - HyphenSplit([#"GF-GA"],1)
17                  ]
18          ),
19
20      Result = Table.FromRecords( TransformedRows )
21  in
22      Result
```

Table.TransformRows applies a function to each record in a table and returns a list of records

each and its associated underscore in Table.TransformRows represent the current record

The ampersand operator unions two records

```
7       HyphenSplit = SplitAsNumber("-"),
8
9       //Method3
10      TransformedRows =
11          Table.TransformRows(
12              Source,
13              each _ & [
14                          Points = [Won]*3 + [Draw],
15                          GF = HyphenSplit([#"GF-GA"],0),
16                          GD = GF - HyphenSplit([#"GF-GA"],1)
17                      ]
18          ),
```

The underscore is the current record.

The ampersand will UNION the record on its right with the record on its left.

We define a record containing the new columns we want to add

New columns are defined as fields in a record

```
7       HyphenSplit = SplitAsNumber("-"),
8
9       //Method3
10      TransformedRows =
11          Table.TransformRows(
12              Source,
13              each _ & [
14                          Points = [Won]*3 + [Draw],
15                          GF = HyphenSplit([#"GF-GA"],0),
16                          GD = GF - HyphenSplit([#"GF-GA"],1)
17                      ]
18          ),
19
```

Each new column is a field in the record. A column definition begins with the new column name, then an =, then the calculation for the new column

Column names defined previously in the new record are not enclosed in square brackets

Column names from the already-existing record in the table are enclosed in square brackets

Table.FromRecords converts a list of records to a table

```
7       HyphenSplit = SplitAsNumber("-"),
8
9       //Method3
10      TransformedRows =
11          Table.TransformRows(
12              Source,
13              each _ & [
14                          Points = [Won]*3 + [Draw],
15                          GF = HyphenSplit([#"GF-GA"],0),
16                          GD = GF - HyphenSplit([#"GF-GA"],1)
17                      ]
18          ),
19
20      Result = Table.FromRecords( TransformedRows )
21  in
22      Result
```

Because Table.TransformRows returns a list of records, we use Table.FromRecords to convert it back to a Table

For adding multiple columns whose calculations depend on each other, consider Table.TransformRows

```
9     TransformedRows = Table.TransformRows( Source,each _ & [Points = [Won]*3 + [Draw],
10                                                            GF = HyphenSplit([#"GF-GA"],0),
11                                                            GD = GF - HyphenSplit([#"GF-GA"],1)]),
12
13    Result = Table.FromRecords( TransformedRows )
```

= Table.FromRecords(TransformedRows)

	Group	Team	Won	Draw	GF-GA	Points	GF	GD
1	Group A	Ecuador	1	1	4-3	4	4	1
2	Group A	Netherlands	2	1	5-1	7	5	4
3	Group A	Qatar	0	0	1-7	0	1	-6
4	Group A	Senegal	2	0	5-4	6	5	1
5	Group B	England	2	1	9-2	7	9	7
6	Group B	Iran	1	0	4-7	3	4	-3
7	Group B	United States	1	2	2-1	5	2	1
8	Group B	Wales	0	1	1-6	1	1	-5
9	Group C	Argentina	2	0	5-2	6	5	3
10	Group C	Mexico	1	1	2-3	4	2	-1
11	Group C	Poland	1	1	2-2	4	2	0
12	Group C	Saudi Arabia	1	0	3-5	3	3	-2
13	Group D	Australia	2	0	3-4	6	3	-1