A Demonstration on the Use of R to Reconcile Similar but Not Identical Spreadsheets

Anthony J. Castellani

This is a demonstration of the use of R to solve a simple, common, yet often frustrating office situation: the presence of two similar but not identical spreadsheets that need to be reconciled. There are ways to do this in Microsoft Excel, but many users of R prefer to maximize the use of R, while minimizing the use of Excel. By using the `dplyr` package, this can be accomplished in very few lines of code.

Note that for the purposes of this demonstration, the package `pander` is also used. This package is only used here to generate more attractive tables, and plays no functional role in the manipulation of the data.

To begin, a sample data set will be created, using `set.seed` for repeatability.

```
set.seed(1)
spreadsheet0 <- dplyr::tbl_df(data.frame(matrix(
  data = c("a", "b", "c", "d", "e", "f", "g",
           round(runif(n = 28, min = 10001, max = 99999))),
  nrow = 7, ncol = 5, byrow = FALSE,
  dimnames = list(NULL, c("UniqueID", "Attribute1", "Attribute2",
                          "Attribute3", "Attribute4")))))
pander::pander(spreadsheet0)
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
a	43491	66620	54793	68650
b	61557	15562	74585	21301
c	91738	28538	99271	34050
d	28152	25891	44203	44750
e	90854	71832	79970	11206
f	95020	44570	94123	44415
g	69471	79285	29093	88271

From the source spreadsheet two similar but different spreadsheets will be generated.

Here is Spreadsheet "A":

```
spreadsheetA <- spreadsheet0[c(1,2,3,4,5),]
pander::pander(spreadsheetA)
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
a	43491	66620	54793	68650
b	61557	15562	74585	21301
c	91738	28538	99271	34050
d	28152	25891	44203	44750
e	90854	71832	79970	11206

And here is Spreadsheet "B":

```
spreadsheetB <- spreadsheet0[c(1,2,3,6,7),]
pander::pander(spreadsheetB)
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
a	43491	66620	54793	68650
b	61557	15562	74585	21301
c	91738	28538	99271	34050
f	95020	44570	94123	44415
g	69471	79285	29093	88271

The first exercise in reconciliation will be to use `dplyr` to recombine the two spreadsheets without any duplicates.

```
full <- dplyr::union(spreadsheetA, spreadsheetB)
full <- dplyr::arrange(full, UniqueID)
pander::pander(full)
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
a	43491	66620	54793	68650
b	61557	15562	74585	21301
c	91738	28538	99271	34050
d	28152	25891	44203	44750
e	90854	71832	79970	11206
f	95020	44570	94123	44415
g	69471	79285	29093	88271

The next exercise will return only those rows of data that are common to both spreadsheets.

```
pander::pander(dplyr::intersect(spreadsheetA, spreadsheetB))
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
a	43491	66620	54793	68650
b	61557	15562	74585	21301
c	91738	28538	99271	34050

After that we will highlight those rows of data that are unique to each spreadsheet.

First the unique rows in Spreadsheet "A":

```
pander::pander(dplyr::setdiff(spreadsheetA, spreadsheetB))
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
d	28152	25891	44203	44750
e	90854	71832	79970	11206

Then the unique rows in Spreadsheet "B":

```
pander::pander(dplyr::setdiff(spreadsheetB, spreadsheetA))
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
f	95020	44570	94123	44415
g	69471	79285	29093	88271

Finally, let's look at the unique rows from the two spreadsheets combined into one.

```
pander::pander(dplyr::bind_rows(dplyr::setdiff(spreadsheetA, spreadsheetB),
dplyr::setdiff(spreadsheetB, spreadsheetA)))
```

UniqueID	Attribute1	Attribute2	Attribute3	Attribute4
d	28152	25891	44203	44750
e	90854	71832	79970	11206
f	95020	44570	94123	44415
g	69471	79285	29093	88271