Paper 10821-2016

Four Lines of Code: Using Merge and Colons to Construct Historical Data from Status Tables

Jingyu She, Danica Pension

ABSTRACT

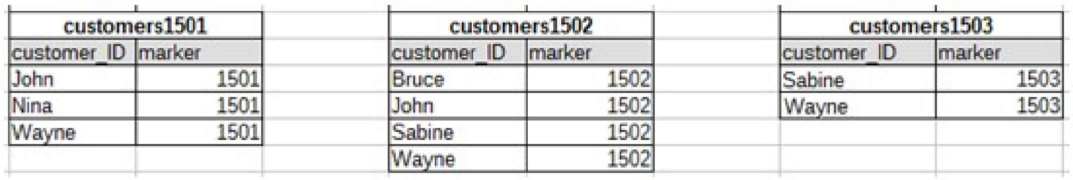
Collection of customer data is often done in status tables or snapshots, where, for example, for each month, the values for a handful of variables are recorded in a new status table whose name is marked with the value of the month. In this QuickTip, we present how to construct a table of last occurrence times for customers using a DATA step merge of such status tables and the colon (":") wildcard. If the status tables are sorted, this can be accomplished in four lines of code (where RUN; is the fourth). Also, we look at how to construct delta tables (for example, from one time period to another, or which customers have arrived or left) using a similar method of merge and colons.

INTRODUCTION

In this paper we explore how to use the inner workings of MERGE and DATA step wildcards (such as the dash and colon) to list last and first occurrences of records in a group of datasets. The main goal of the paper is not so much to demonstrate the technique as it is to encourage experimentation and creativity when it comes to using the DATA step.

LIST LAST OCCURENCES

Suppose we have the following three status tables listing customers in the months January, February, March of 2015, where all three are sorted on customer_ID:



customers1501	
customer_ID	marker
John	1501
Nina	1501
Wayne	1501

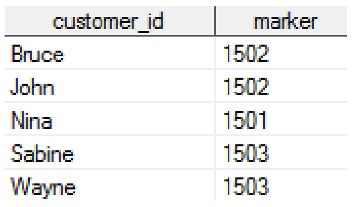
customers1502	
customer_ID	marker
Bruce	1502
John	1502
Sabine	1502
Wayne	1502

customers1503	
customer_ID	marker
Sabine	1503
Wayne	1503

If we want a table of all customers and the last time each of these customers were observed, we can choose to SET all datasets in one table, sort, and use another DATA step to keep the LAST occurrence of each customer. There is however another way, especially if the status tables are already sorted by *customer_ID*:

```
data lastoccurrences;
   merge customers: ;
   by customer_ID;
run;
```

This returns the following dataset of last occurrences:



customer_id	marker
Bruce	1502
John	1502
Nina	1501
Sabine	1503
Wayne	1503

WHY DOES IT WORK?

The colon wildcard tells the DATA step to merge all datasets in the WORK library starting with "customers", which would be *customers1501*, *customers1502*, and *customers1503*. When using the colon wildcard, the names of these data sets are listed in lexicographic order. Thus they are merged on by lexicographical order, in the sense that the value of the overlapping "marker" from *customers1501* is overwritten by the "marker" value from *customers1502*, which is then overwritten by the "marker" value from *customers1503*. For example, NINA is only a customer in 1501, hence there is no "marker" value from *customers1502* and *customers1503,* whereby her last occurrence is marked as 1501.

MAKING A DELTA TABLE

Let us use a similar trick to construct a delta data set, i.e. find out which customers arrive / leave and in which months this happens.

```
data deltas;
   merge customers1503-customers1501 (rename = (marker = first_seen));
   by customer_id;
   merge customers: (rename = (marker = last_seen));
   by customer_id;
run;
```

Here we need a slight modification of our code, since in order to get the list of first_seen customers, we must merge the datasets in reverse lexicographic order. We do this by listing the datasets in reverse using the dash "-". Note that here in particular, the naming of the datasets plays a big role. Also note that the rename option does in fact rename the marker variable on all data sets. By using this approach, each datasets is run through twice. First, the data sets are run through in reverse order so that the first occurrence of a customer is the last value of "first_seen" that overwrites the rest. Then, we repeat the exercise of finding the last occurrence, as we have done earlier.

Copy and run the code in the attachment to try out both this trick and the first trick on an empty WORK library.

CONCLUSION

The naming of the data sets is crucial for the colon and dash tricks to work. For the delta table, every dataset is read twice, which should be remembered when considering performance. The main goal of this paper is not to demonstrate this tip as such, but more to inspire and encourage creativity when using the DATA step, introducing wildcards, merge of datasets with the same variables, and the usage of multiple merge statements, all of which are not as dangerous actions as one might think.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jingyu She
Danica Pension
+45 45131261
jish@danicapension.dk
http://www.danicapension.dk/en-dk/