

# OVERLAPPING DATE SEGMENTS: HOW TO CLEAN UP THE MESS

Doug Shannon, Mercer Human Resource Consulting, Phoenix, AZ  
Wade Bannister, Arizona State University, Tempe, AZ

## ABSTRACT

Many times data are received that contain issues with the date variables. Overlapping segments is one of the more common problems that can occur, which causes difficulties with joins and analysis. These segments could be eligibility ranges, or they could be hospital stays that overlap. Either way, in most cases some type of data manipulation is required in order to correct the problem. This paper will go over these different situations and present ways to manipulate the data into a more desirable format. SAS code that corrects the issues will be reviewed and explained.

## INTRODUCTION

Overlapping date segments are one of the types of issues that can arise with dates. These can be eligibility ranges, hospitalization dates, or other types of date ranges which should be unique to a person or transaction. Regardless of the source of data, these overlaps can cause problems with joins and analysis. Some type of data manipulation usually needs to be done in order to correct the problem.

## TYPES OF OVERLAPS

Suppose you have a dataset that looks like this:

ID	Start Date	End Date
John	2/1/2002	4/30/2002
John	5/1/2002	9/30/2002
Mary	1/1/2002	4/17/2002
Mary	2/1/2002	3/31/2002
Ed	2/1/2002	6/30/2002
Ed	5/1/2002	8/15/2002
Ann	1/1/2002	6/30/2002
Ann	8/1/2002	8/31/2002
Bob	1/1/2002	3/31/2002
Bob	4/1/2002	6/30/2002
Bob	5/1/2002	7/31/2002

### Consecutive

Consecutive segments are not an actual overlap, but are within a day of each other. Depending on the situation, there are many times that these segments need to be collapsed. An example of one of the segments occurs for John. The two segments should actually be condensed to make one segment as follows.

ID	Start Date	End Date
John	2/1/2002	9/30/2002

### Complete Overlap

This type of overlap occurs when one segment is completely contained within another one. An example of this is found in Mary's eligibility; her second segment is entirely within the first. Thus, the second segment should be completely disregarded to leave the result as follows:

ID	Start Date	End Date
Mary	1/1/2002	4/17/2002

### Partial Overlap

This type of overlap occurs when portions of one of the segments are within portions of another segment. An example of this type of overlap is found in Ed's eligibility; part of his second segment is contained in the first. The true result should be a combination of the two segments for the following outcome:

ID	Start Date	End Date
Ed	2/1/2002	8/15/2002

### Gaps

On the other hand, we see that Ann's eligibility contains a gap for the month of July. While not technically an overlap, this somewhat complicates the issue of removing overlaps, since we cannot just take the first start date with the last end date as our date range.

### CODE

The code that fixes all of these overlap issues is listed below:

```
proc sort data=overlaps out=one;
  by id start_date end_date;
run;

data TWO(drop=id2 end_date
         rename=(start_date=begin_dos
                end2=end_dos));
  set one;
  retain end2;
  id2=lag1(id);
  if id2=id and start_date le (end2+1) then
  do;
    start_date=end2;
    end2=max(end_date, end2);
  end;
```

```

else do;
    seq+1;
    end2=end_date;
    end;
format end2 mmddyy10.;
run;

data THREE(drop=begin_dos end_dos seg);
retain start_date end_date;
set TWO;
by id seg;
format start_date end_date mmddyy10.;
if first.seg then do;
    start_date=begin_dos;
    end;
if last.seg then do;
    end_date = end_dos;
    output;
    end;
run;

```

### CODE DESCRIPTION

Now we will walk through the code.

#### Proc Sort

Though it is fairly straight forward, this is a very important step in the process. Sorting the data by ID and start date ensures that we are appropriately removing overlapping segments.

#### Data Two

The main purpose of this step is to link the date segments that should be collapsed. This is done by using the “LAG” function to “look back” to the previous record and see whether it is the same ID and whether the previous END\_DATE (actually END2) is within one day of the current START\_DATE. If they are, both records will be assigned the same SEG, and the END2 date will be assigned to the maximum of the previous END2 date (because of the RETAIN statement) and the current END\_DATE. Otherwise the SEG variable increases by 1 and thus differentiates itself from the previous record. Note that gaps larger than one day will not be considered overlaps, and thus Ann’s eligibility dates will not be altered.

#### Data Three

This step is for collapsing the overlapping segments together. It takes the first BEGIN\_DATE and last END\_DATE for each segment and then outputs the collapsed record.

## RESULTING DATASET

After running the above code the resulting dataset is as follows

ID	Start Date	End Date
John	2/1/2002	9/30/2002
Marv	1/1/2002	4/17/2002
Ed	2/1/2002	8/15/2002
Ann	1/1/2002	6/30/2002
Ann	8/1/2002	8/31/2002
Bob	1/1/2002	7/31/2002

## CONCLUSION

Many times we received data that contains date segments that are overlapping. By using the code provided in this paper, the issues that can occur will be corrected.

## CONTACT INFORMATION

Doug Shannon  
 Mercer Human Resource Consulting  
 3131 E. Camelback Rd. Suite 300  
 Phoenix, AZ 85016

Phone: (602) 522-8577  
 Fax: (602) 957-9573  
 Email: [doug.shannon@mercerc.com](mailto:doug.shannon@mercerc.com)

Wade Bannister  
 Arizona State University  
 Health Administration and Policy  
 W.P. Carey School of Business  
 Tempe, AZ 85287

Phone: 480.965.1623  
 Email: [wade.bannister@asu.edu](mailto:wade.bannister@asu.edu)