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:

<table>
<thead>
<tr>
<th>ID</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>5/1/2002</td>
<td>9/30/2002</td>
</tr>
<tr>
<td>Mary</td>
<td>1/1/2002</td>
<td>4/17/2002</td>
</tr>
<tr>
<td>Mary</td>
<td>2/1/2002</td>
<td>3/31/2002</td>
</tr>
<tr>
<td>Ed</td>
<td>2/1/2002</td>
<td>6/30/2002</td>
</tr>
<tr>
<td>Ed</td>
<td>5/1/2002</td>
<td>8/15/2002</td>
</tr>
<tr>
<td>Ann</td>
<td>1/1/2002</td>
<td>6/30/2002</td>
</tr>
<tr>
<td>Ann</td>
<td>8/1/2002</td>
<td>8/31/2002</td>
</tr>
<tr>
<td>Bob</td>
<td>5/1/2002</td>
<td>7/31/2002</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>ID</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>2/1/2002</td>
<td>9/30/2002</td>
</tr>
</tbody>
</table>

Complete Overlap
This type of overlap occurs when one segment in 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:

<table>
<thead>
<tr>
<th>ID</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>1/1/2002</td>
<td>4/17/2002</td>
</tr>
</tbody>
</table>

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 in 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:

<table>
<thead>
<tr>
<th>ID</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ed</td>
<td>2/1/2002</td>
<td>8/15/2002</td>
</tr>
</tbody>
</table>

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;
```

```
ID Start_Date End_Date
-------------------------
John 2/1/2002 9/30/2002
```
else do;
    seg+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;

RESULTING DATASET
After running the above code the resulting dataset is as follows

<table>
<thead>
<tr>
<th>ID</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>2/1/2002</td>
<td>9/30/2002</td>
</tr>
<tr>
<td>Mary</td>
<td>1/1/2002</td>
<td>4/17/2002</td>
</tr>
<tr>
<td>Ed</td>
<td>2/1/2002</td>
<td>8/15/2002</td>
</tr>
<tr>
<td>Ann</td>
<td>1/1/2002</td>
<td>6/30/2002</td>
</tr>
<tr>
<td>Ann</td>
<td>8/1/2002</td>
<td>8/31/2002</td>
</tr>
<tr>
<td>Bob</td>
<td>1/1/2002</td>
<td>7/31/2002</td>
</tr>
</tbody>
</table>

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@mercer.com

Wade Bannister
Mercer Human Resource Consulting
3131 E. Camelback Rd. Suite 300
Phoenix, AZ 85016
Phone: (602) 522-8562
Fax: (602) 957-9573
Email: wade.bannister@mercer.com