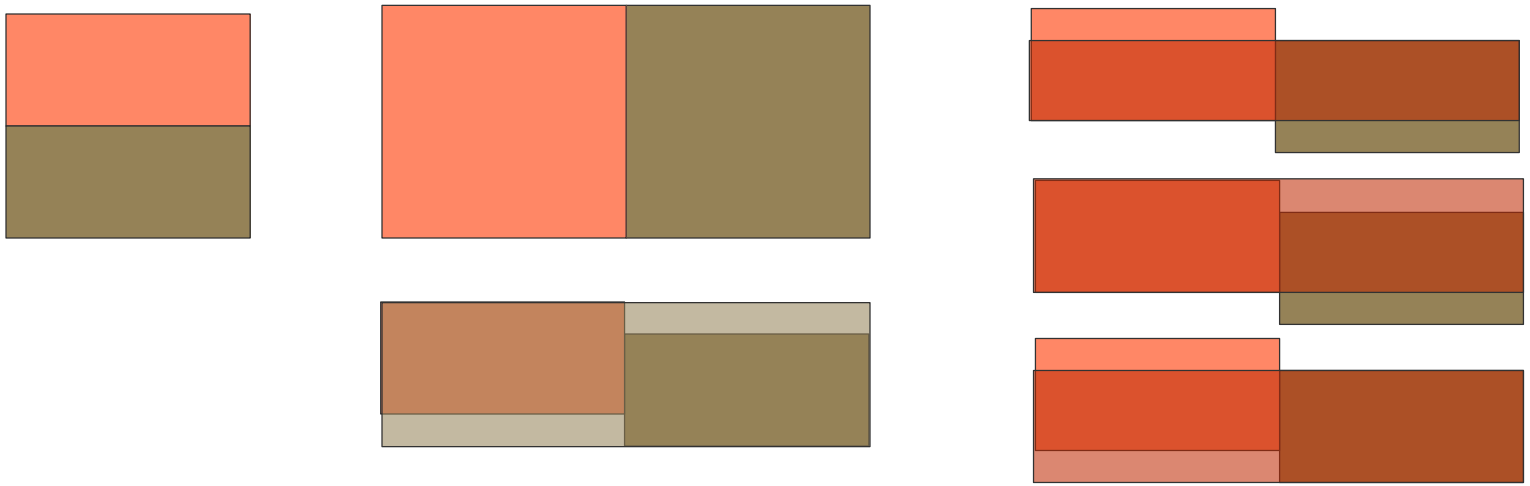
SAS Programming – TIP for TODAY

GHSUG- May 26, 2017

ArcelorMittal
Lesley Harschnitz
lesley.harschnitz@arcelormittal.com

Putting Data Sets Together

- Collecting, ordering, formatting data is often 80% of the work towards creating a report, dashboard, graph…..
- SAS provides numerous ways of reaching this objective
- Looking today at putting data sets together by various methods as follows:



We will be looking at both the DATA Step and PROC SQL as ways to join data.
Taking the “happy path”

Things to be aware of

- Pre-sorting: When using a BY statement you always need to pre-sort the data sets

- You may have unexpected results when:
  - You don't use a BY statement or a JOIN variable to control the record matching
  - Your join variables are not unique on one side of the join
  - Your join variables contain nulls (missing values) or errors

  - Hint: Read the log carefully

```
742    from test1 a, test2 b;
NOTE: The execution of this query involves performing one or more Cartesian product joins that can not be optimized.
743  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time               0.07 seconds
      cpu time                0.04 seconds


NOTE: MERGE statement has more than one data set with repeats of BY values.
NOTE: There were 4 observations read from the data set WORK.TEST1.
NOTE: There were 3 observations read from the data set WORK.TEST2.
NOTE: The data set WORK.TMERGE1 has 4 observations and 3 variables.
NOTE: DATA statement used (Total process time):
      real time               0.03 seconds
      cpu time                0.00 seconds
```

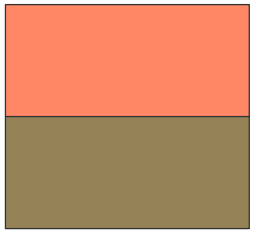
Sample Data

```
Data Sales;
  do Year='2015','2016','2017';
    do Month=1 to 12;
       do Region='North ','West','Export';
            tot_produced= 45000*rannor(0)+15000;
            tot_sales=1500*rannor(0)+25000;
            output;
       end;
      end;
  end;
run;

Data RawMatlV1;
  do Year='2015';
    do Month=1 to 12;
        do Region='North ','West','Export';
          tot_plastic= 3500*rannor(0)+35000;
          tot_paint=4500*rannor(0)+15000;
          if Year='2015' and Month=11 and Region='Export'
          then Region='Exprt';
          output;
        end;
      end;
  end;
run;
```

```
Data RawMatlV2;
  do Year='2016','2017';
    do Month=1 to 12;
      do Region='North ','West','Export';
          tot_plastic= 3500*rannor(0)+35000;
          tot_paint=4500*rannor(0)+15000;
          if Year='2016' and Month=3 and Region='Export'
          then Region='Exprt';
          output;
      end;
     end;
  end;
run;

Data Staff;
 length Name $10;
 do Year='2015','2016','2017';
   do Region='North ','West','Export';
      Namesel = INT((5*ranuni(0))+1);
      Name=scan("Patrick,Tom,John,Anna,Sophie", Namesel);
      output;
    end;
 end;
Run;
```

“These data sets are fictitious and were created for illustrative purposes only.
As such, this information should not be used for any other purpose.”

Stacking Data Sets



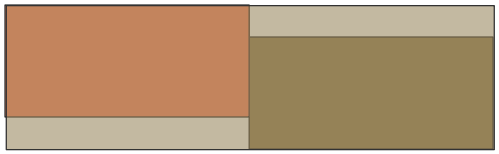
```
895
896  data RawMatl;
897    set RawMatlV1 RawMatlV2;
898  run;

NOTE: There were 36 observations read from the data set WORK.RAWMATLV1.
NOTE: There were 72 observations read from the data set WORK.RAWMATLV2.
NOTE: The data set WORK.RAWMATL has 108 observations and 5 variables.
NOTE: DATA statement used (Total process time):
      real time           0.07 seconds
      cpu time            0.00 seconds


899
900  proc sql;
901    select a.*
902      from RawMatlV1 a
903    UNION
904    select a.*
905      from RawMatlV2 a;
906  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.12 seconds
      cpu time            0.03 seconds
```

2015	9	Export	32868.69	10143.98
2015	9	North	37808.04	24809.32
2015	9	West	29163.32	10357.62
2015	10	Export	34028.21	12269.55
2015	10	North	43707.82	10833.15
2015	10	West	33955.78	16734.51
2015	11	Exprt	38124.07	26225.71
2015	11	North	28348.55	21432.83
2015	11	West	34734.07	13690.16
2015	12	Export	41278.9	10938.51
2015	12	North	42463.58	24781.03
2015	12	West	37084.96	12111.09
2016	1	Export	40102.1	8433.867
2016	1	North	32644.73	16709.3

Take care to ensure that the variables are common to both datasets.

"The information contained on this slide is fictitious and was created for illustrative purposes only. As such, this information should not be used for any other purpose."

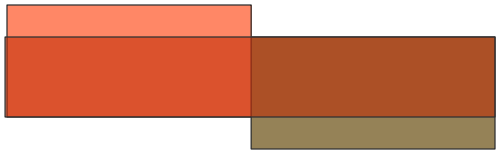
Full Outer Join



```
1067  proc sort data=Sales;
1068    by year month region;
1069  run;

NOTE: Input data set is already sorted, no sorting done.
NOTE: PROCEDURE SORT used (Total process time):
      real time             0.00 seconds
      cpu time              0.00 seconds


1070
1071  proc sort data=RawMatl;
1072    by year month region;
1073  run;

NOTE: Input data set is already sorted, no sorting done.
NOTE: PROCEDURE SORT used (Total process time):
      real time             0.00 seconds
      cpu time              0.00 seconds


1074
1075  data Compare;
1076   merge sales rawmatl;
1077   by year month region;
1078  run;

NOTE: There were 108 observations read from the data set WORK.SALES.
NOTE: There were 108 observations read from the data set WORK.RAWMATL.
NOTE: The data set WORK.COMPARE has 110 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time             0.03 seconds
      cpu time              0.01 seconds


1079
1080  proc sql;
1081    select s.*,
1082           r.tot_plastic,
1083           r.tot_paint
1084    from sales s full join rawmatl r
1085       on s.year=r.year and s.month=r.month and s.region=r.region;
1086  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time             0.15 seconds
      cpu time              0.04 seconds
```

2015	8	West	-37150.1	24313.2	32476.95	14012.07
2015	9	Export	-69160.6	23981.52	32868.69	10143.98
2015	9	North	-43547.2	26650.69	37808.04	24809.32
2015	9	West	30659.86	25465.66	29163.32	10357.62
2015	10	Export	-17968.1	24589.23	34028.21	12269.55
2015	10	North	36176.82	25443.52	43707.82	10833.15
2015	10	West	68733.05	23345.35	33955.78	16734.51
2015	11	Export	-45736.7	26085.73	.	.
	.		.	.	38124.07	26225.71
2015	11	North	54318.98	25998.27	28348.55	21432.83
2015	11	West	60152.93	22719.81	34734.07	13690.16
2015	12	Export	16581.45	25928.96	41278.9	10938.51
2015	12	North	-753.296	25599.99	42463.58	24781.03
2015	12	West	85636.44	23405.33	37084.96	12111.09
2016	1	Export	-14077.4	26381.49	40102.1	8433.867
2016	1	North	-19988.6	22018.28	32644.73	16709.3
2016	1	West	12290.19	27594.58	36394.07	22753.87

Inner Join

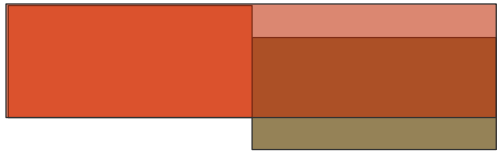


```
1087  data Compare;
1088   merge sales(in=a) rawmatl(in=b);
1089   by year month region;
1090   if a and b;
1091  run;

NOTE: There were 108 observations read from the data set WORK.SALES.
NOTE: There were 108 observations read from the data set WORK.RAWMATL.
NOTE: The data set WORK.COMPARE has 106 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.06 seconds
      cpu time            0.00 seconds


1092
1093  proc sql;
1094    select s.*,
1095           r.tot_plastic,
1096           r.tot_paint
1097    from sales s inner join rawmatl r
1098       on s.year=r.year and s.month=r.month and s.region=r.region;
1099  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.17 seconds
      cpu time            0.07 seconds
```

2015	9	North	-43547.2	26650.69	37808.04	24809.32
2015	9	West	30659.86	25465.66	29163.32	10357.62
2015	10	Export	-17968.1	24589.23	34028.21	12269.55
2015	10	North	36176.82	25443.52	43707.82	10833.15
2015	10	West	68733.05	23345.35	33955.78	16734.51
2015	11	North	54318.98	25998.27	28348.55	21432.83
2015	11	West	60152.93	22719.81	34734.07	13690.16
2015	12	Export	16581.45	25928.96	41278.9	10938.51
2015	12	North	-753.296	25599.99	42463.58	24781.03
2015	12	West	85636.44	23405.33	37084.96	12111.09
2016	1	Export	-14077.4	26381.49	40102.1	8433.867
2016	1	North	-19988.6	22018.28	32644.73	16709.3
2016	1	West	12290.19	27594.58	36394.07	22753.87
2016	2	Export	44123.05	25258.67	33708.95	15825.84
2016	2	North	-75227.9	24280.49	37870.27	10137.28

"The information contained on this slide is fictitious and was created for illustrative purposes only. As such, this information should not be used for any other purpose."

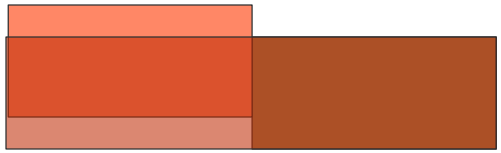
Left Join



```
1126  data Compare;
1127   merge sales(in=a) rawmatl(in=b);
1128   by year month region;
1129   if a;
1130  run;

NOTE: There were 108 observations read from the data set WORK.SALES.
NOTE: There were 108 observations read from the data set WORK.RAWMATL.
NOTE: The data set WORK.COMPARE has 108 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds


1131
1132  proc sql;
1133    select s.*,
1134           r.tot_plastic,
1135           r.tot_paint
1136    from sales s left join rawmatl r
1137       on s.year=r.year and s.month=r.month and s.region=r.region;
1138  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.07 seconds
      cpu time            0.01 seconds
```

2015	9	North	-43547.2	26650.69	37808.04	24809.32
2015	9	West	30659.86	25465.66	29163.32	10357.62
2015	10	Export	-17968.1	24589.23	34028.21	12269.55
2015	10	North	36176.82	25443.52	43707.82	10833.15
2015	10	West	68733.05	23345.35	33955.78	16734.51
2015	11	Export	-45736.7	26085.73	.	.
2015	11	North	54318.98	25998.27	28348.55	21432.83
2015	11	West	60152.93	22719.81	34734.07	13690.16
2015	12	Export	16581.45	25928.96	41278.9	10938.51
2015	12	North	-753.296	25599.99	42463.58	24781.03
2015	12	West	85636.44	23405.33	37084.96	12111.09
2016	1	Export	-14077.4	26381.49	40102.1	8433.867
2016	1	North	-19988.6	22018.28	32644.73	16709.3
2016	1	West	12290.19	27594.58	36394.07	22753.87
2016	2	Export	44123.05	25258.67	33708.95	15825.84
2016	2	North	75227.9	24280.49	37870.27	10137.28

Right Join



```
1113  data Compare;
1114   merge sales(in=a) rawmatl(in=b);
1115   by year month region;
1116   if b;
1117  run;

NOTE: There were 108 observations read from the data set WORK.SALES.
NOTE: There were 108 observations read from the data set WORK.RAWMATL.
NOTE: The data set WORK.COMPARE has 108 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.06 seconds
      cpu time            0.00 seconds


1118
1119  proc sql;
1120    select s.*,
1121           r.tot_plastic,
1122           r.tot_paint
1123    from sales s right join rawmatl r
1124      on s.year=r.year and s.month=r.month and s.region=r.region;
1125  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.12 seconds
      cpu time            0.04 seconds
```

2015	8	North	-17064.7	21632.26	37753.67	21598.09
2015	8	West	-37150.1	24313.2	32476.95	14012.07
2015	9	Export	-69160.6	23981.52	32868.69	10143.98
2015	9	North	-43547.2	26650.69	37808.04	24809.32
2015	9	West	30659.86	25465.66	29163.32	10357.62
2015	10	Export	-17968.1	24589.23	34028.21	12269.55
2015	10	North	36176.82	25443.52	43707.82	10833.15
2015	10	West	68733.05	23345.35	33955.78	16734.51
	.		.	.	38124.07	26225.71
2015	11	North	54318.98	25998.27	28348.55	21432.83
2015	11	West	60152.93	22719.81	34734.07	13690.16
2015	12	Export	16581.45	25928.96	41278.9	10938.51
2015	12	North	-753.296	25599.99	42463.58	24781.03
2015	12	West	85636.44	23405.33	37084.96	12111.09
2016	1	Export	-14077.4	26381.49	40102.1	8433.867
2016	1	North	-19988.6	22018.28	32644.73	16709.3
2016	1	West	12290.19	27594.58	36394.07	22753.87
2016	2	Export	44123.05	25258.67	33708.95	15825.84

Summary

- Can use either data step or SQL to do simple joins
- Power of the data step comes in when you want to do complex conditional processing
- Power of SQL comes in when you want to join tables with different column names and avoid pre-sorting
- Pay attention to the results:
  - Row counts
  - Missing values
  - Missing key column values