Dealing with Duplicates: Identify, Investigate and Purge
Ying Feng, CTB/McGraw-Hill, Hightstown, NJ

ABSTRACT

The best way to deal with duplicates is to not generate them in the first place. This is easier to say than do. Sometimes it is out of our control since the data coming to us is already “contaminated”. You certainly do not want a “Garbage in, Garbage out” scenario. Therefore, dealing with duplicates should be one of your first data-processing steps.

Records can be categorized as exact or partial duplicates. Dealing with exact duplicates is easy, but how to define and deal with partial duplicates requires discretion. This paper will detail how to identify, investigate and purge exact and partial duplicates under the SAS® environment.

With the help of PROC SORT, PROC SQL and the SAS Macro language, a simple and generalized method is developed to help deal with duplicates.

INTRODUCTION

Exact duplicates are the records where every variable has the same value. It should be purged without question. There are several ways to remove exact duplicates. You can either use the NODUP option of PROC SORT or the SELECT DISTINCT statement in PROC SQL to accomplish it.

Partial duplicates are the records where several variables have the same value. The determination of what constitutes duplicates is subject to exploration. For example, sometimes the records have the same values on all variables except for timestamps. In that case, the records should be treated as exact duplicates and be purged. While in another case, we need to determine a key to identify duplicates and whether we should use student id plus form or student id plus admin_date will depend on the specific situation.

During the exploration, we often ask such questions, “How many duplicates do we get if we use this combination of variables as a key?” “What if we switch to another combination of variables?” That is, we would like to get information on many different keys and on duplicates of various types. Therefore, we need to investigate the duplicates.

Since duplication is an issue we need to deal with on a daily basis, I developed a macro to handle identifying, investigating and purging all at once. In this paper, a detailed explanation of how the macro works will be illustrated step by step.

Step 1: Purge Exact Duplicates

First, we identify exact duplicates and purge them. The NODUP option in PROC SORT is used to remove the duplicates. The first occurrence of the duplicates is kept and the
other duplicates are removed. The DUPOUT option is used to store the records that are purged in case you need to report them.

```sas
%macro dupChk(in=, out=, varlst=, rmvPartInd=);
title;
*remove exact duplicates;
proc sort data=&in. out=&out. dupout=EDups nodup;
  by _all_
run;
```

**Step 2: Report Counts after Removing Exact Duplicates**

“Wait, we want to know how many cases are removed?”, “Can you give us a report on which cases are removed? We need to investigate”… We often hear people ask such questions. This step gives the records counts before and after removing the true duplicates, and the counts of how many duplicates are removed. The *EDups* duplicate dataset generated by step 1 can also be provided for more investigation.

PROC SQL provides a very convenient way to perform the reporting task. The SELECT INTO statement stores counts into macro variables, and PROC PRINT displays the values of the macro variables.

```sas
*display exact duplicates info;
proc sql noprint;
select count(*) into :cnt_orig
  from &in.;
select count(*) into :cnt_edup
  from Edups;
select count(*) into :cnt_noedup
  from &out.;

data info;
  count_orig=&cnt_orig;
count_exactDup=&cnt_edup;
count_afterRmvExactDup=&cnt_noedup;
run;

title "Counts of Removing Exact Dups";
proc print data=info noobs; run;
```

**Step 3: Investigate Partial Duplicates**

PROC SQL is also a powerful tool to investigate the partial duplicates. The GROUP BY and HAVING clauses allow PROC SQL to perform the summary function COUNT for
those records that have duplicated variables. Duplicated variable selection is defined in a macro variable `varlst`. You can define either one variable or multiple variables in `varlst`.

PROC PRINT displays all the records that have duplicated variables for further investigation.

```sql
*check other non-exact dup cases;
proc sql;
create table pDupChk as
select unique &varlst., count(*) as count
from &out
group by &varlst.
having count>1;

title 'After Removing Total Dups - Other Non-Exact Dups';
proc print data=pDupChk noobs;
run;
```

**Step 4: Purge Partial Duplicates**

After the investigation, we might want to purge the partial duplicates or leave them as is. The `%IF` statement in step 4 gives you the choice. If the macro variable `rmvPartInd` equals ‘Y’ then the purge will be performed and the information on the purge will be reported.

PROC PRINT displays the counts of the records after the partial duplicates are removed.

```sql
Step 4
%if &rmvPartInd=Y %then %do;
   proc sort data=&out. out=&out.2 dupout=PDups nodupkeys;
   by &varlst;
   run;

   *display counts after removing partial duplicates info;
   proc sql noprınt;
   select count(*) into :cnt_noPDup
   from &out.2;

data info;
   count_afterRmvPartialDup=&cnt_noPDup;
run;
title "Counts After Removing Dups By &varlst";
proc print data=info noobs; run;
%end;
%mend;
```
Example of Macro Call
The macro *dupChk* has been illustrated above. Here is an example of how to call the macro:

Example 1:
- purge the exact duplicates;
- use student name as a selected variable to check for partial duplicates;
- do not perform partial duplicate purge;
- print report

```%dupChk*(in=stu, out=studata_nodup, varlst=%str(name), rmvPartInd=);```

Example 2:
- purge the exact duplicates;
- use student name and adminDate as the selected variables to check for partial duplicates;
- perform partial duplicate purge;
- print report

```%dupChk*(in=stu, out=studata_nodup, varlst=%str(name adminDate), rmvPartInd=Y);```

CONCLUSION

Redundant records sometimes post challenges during our daily work. This paper has presented a simple, general method to identify, investigate and purge exact and partial duplicates. It’s a handy tool for me and I hope it will help you too.

AUTHOR CONTACT INFORMATION

Ying Feng
Statistical Analyst

CTB/McGraw-Hill
148 Princeton-Hightstown Road
East Windsor, NJ 08520

Email: ying_feng@ctb.com

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.