

TAP TO GO  
BACK TO  
KIOSK MENU

# SAS<sup>®</sup> GLOBAL FORUM 2020

MARCH 29 - APRIL 1  
WASHINGTON, DC



USERS PROGRAM





Andrea Barbo

Yale New Haven Health Services Corporation/Center for Outcomes Research and Evaluation (CORE)

## Abstract:

In the DATA step, merging data sets with common variables that are not included as BY variables can yield undesirable results. Specifically, the value of a common variable can be overwritten with an incorrect value. To prevent this from happening, you must ensure that the variable is read from only one "master" data set, by either dropping or renaming the variable in the other data sets. When working with data sets with just a few variables, you can quickly check which variables appear in more than one data set. However, as the number of data sets and variables increases, the chance of missing a common variable also increases. The SAS® macro CHECK\_VAR\_EXIST was written to identify variables that exist in more than one data set more efficiently and accurately. The macro prints all common variables, which data sets they appear in, and other pertinent information. You can then use the list to drop or rename variables where they are not relevant, thereby reducing the chance of unintentionally overwriting a large number of variables.



Andrea Barbo

### Abstract

Introduction  
SAS Macro  
Results  
Discussion

Please use the headings above to navigate through the different sections of the poster





## Introduction:

- ❑ SAS programmers are commonly taught that when you merge datasets in the DATA step, variables in the dataset listed later on the MERGE statement replace the values of variables that also exist in a previously listed dataset.
- ❑ This may be true for one-to-one merging, but not for one-to-many merging, because of how the Program Data Vector works.
- ❑ As such, you need to be careful when combining multiple datasets that have variables in common, and not all of them are included as BY variables.
- ❑ The best way to avoid seeing unexpected results is to drop or rename common variables so that they only show up in one dataset.
- ❑ Figuring out the common variables can be done easily if you're working with just a couple of datasets with few variables. However, it gets more cumbersome the more datasets and variables are involved.
- ❑ The SAS® macro **CHECK\_VAR\_EXIST**, which will be described in the next slides, provides an automated way of identifying common variables.

	Measure_Name	Measure_ID	Compa
1	Central Line Associated Bloodstream Infection (ICU + select Wards): Lower Confidence Limit	HAI_1_CILOWER	No Diff
2	Central Line Associated Bloodstream Infection (ICU + select Wards): Upper Confidence Limit	HAI_1_CIUPPER	No Diff
3	Central Line Associated Bloodstream Infection: Number of Device Days	HAI_1_DOPC	No Diff
4	Central Line Associated Bloodstream Infection (ICU + select Wards): Predicted Cases	HAI_1_ELIGCASES	No Diff

  

	Measure_Name	Measure_ID
	...e of complications for hip/knee placement patients	COMP_HIP_KNEE
	...th rate for heart attack patients	MORT_30_AMI
	...th rate for CABG surgery patients	MORT_30_CABG
	...th rate for COPD patients	MORT_30_COPD
	...th rate for heart failure patients	MORT_30_HF
	...th rate for pneumonia patients	MORT_30_PN
	...th rate for stroke patients	MORT_30_STK
	...toperative Acute Kidney Injury requiring Dialysis Rate	PSI_10_POST_KID
	...toperative Respiratory Failure Rate	PSI_11_POST_RE
	...ous blood clots after surgery	PSI_12_POSTOP_
	...lood stream infection after surgery	PSI_13_POST_SE
	...A wound that splits open after surgery on the abdomen or pelvis	PSI_14_POSTOP_
	...Accidental cuts and tears from medical treatment	PSI_15_ACC_LAC
	...Pressure sores	PSI_3_ULCER
	...Deaths among Patients with Serious	SI_4_SURG_COMP

Abstract

Introduction

SAS Macro

Results

Discussion

Please use the headings above to navigate through the different sections of the poster





## SAS® Macro

### CHECK\_VAR\_EXIST:

- ❑ Identifies variables that exist in more than one dataset.
- ❑ Ideal to use before merging 2+ datasets as a check to prevent incorrect variables from overwriting correct ones with the same name.
- ❑ Input parameters: **DTA** is a list of datasets to check (preceded by a libref if stored as a permanent dataset), **LINK\_VAR** is a list of variables that should be excluded from the checking (usually the ones used as BY variables in the MERGE statement).
- ❑ Output: list of variables that appear in more than one dataset, with additional info like length & type, in the Results Window.

```
%macro check_var_exist(dta=,link_var=);
data _null_;
  /*remove excess blank characters from list of datasets*/
  _var="&dta";
  dta_list=tranwrd(compbl(strip(_var))," ",".");
  call symputx("dta_list",dta_list);

  /*count how many datasets to check for overlapping variables*/
  cnt_dta=count(strip(dta_list),"")+1;
  call symputx("cnt_dta",cnt_dta);

  /*list of variables to exclude from checking*/
  list_var=lowcase("'"||tranwrd(compbl(strip("&link_var")),",","'|'")||"'");
  call symputx("list_var",list_var);
run;
%put &dta_list &cnt_dta &list_var;

/*output variables that exist in more than 1 dataset*/
proc sql;
  select *
  from (select distinct upcase(name) as name label="Column Name",type,length,libname,memname
        from sashelp.vcolumn
        %if %sysfunc(find(%scan(%sysfunc(lowcase(&dta_list)),1,' '),.))>0 %then %do;
          where ( (lowcase(libname)="%scan(%scan(%sysfunc(lowcase(&dta_list)),1,' '),1,'.')" and
lowcase(memname)="%scan(%scan(%sysfunc(lowcase(&dta_list)),1,' '),2,'.)'"
        %end;
        %else %do;
          where ( (lowcase(libname)="work" and lowcase(memname)="%scan(%sysfunc(lowcase(&dta_list)),1,' ')'"
        %end;
        %do i=2 %to &cnt_dta;
          %if %sysfunc(find(%scan(%sysfunc(lowcase(&dta_list)),&i,' '),.))>0 %then %do;
            or (lowcase(libname)="%scan(%scan(%sysfunc(lowcase(&dta_list)),&i,' '),1,'.)'" and
lowcase(memname)="%scan(%scan(%sysfunc(lowcase(&dta_list)),&i,' '),2,'.)'"
          %end;
          %else %do;
            or (lowcase(libname)="work" and lowcase(memname)="%scan(%sysfunc(lowcase(&dta_list)),&i,' ')'"
          %end;
        ) and lowcase(name) not in (&list_var)
      )
  group by name
  having count(*)>1
  order by name,libname,memname
  ;
quit;
%mend check_var_exist;
```

Abstract  
Introduction  
SAS Macro  
Results  
Discussion

Please use the headings above to navigate through the different sections of the poster





## Results:

- ❑ To illustrate how the macro can be used, we downloaded a few CSV files from Data.Medicare.gov and imported into SAS.
- ❑ **Data.Medicare.gov** is a website where consumers can freely download official healthcare-related data produced by the Centers for Medicare & Medicaid Services (CMS).
- ❑ We checked 5 datasets, 3 of which are temporary and 2 are permanent datasets, for common variables. As we're interested in merging all 5 datasets by the variable, Provider ID, we exclude this from the check.
- ❑ `%check_var_exist(dta = Hospital_general_information  
Fy_2019_ipps_fr_impact_file  
sasgf.Complications_and_deaths___hospi  
Healthcare_associated_infections  
sasgf.Patient_survey__hcahps___hospit  
, link_var = Provider_ID)`

Column Name	Column Type	Column Length	Library Name	Member Name
ADDRESS	char	51	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
ADDRESS	char	50	SASGF	PATIENT_SURVEY__HCAHPS___HOSPIT
ADDRESS	char	50	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
ADDRESS	char	50	WORK	HOSPITAL_GENERAL_INFORMATION
HOSPITAL_NAME	char	71	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
HOSPITAL_NAME	char	71	SASGF	PATIENT_SURVEY__HCAHPS___HOSPIT
HOSPITAL_NAME	char	50	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
HOSPITAL_NAME	char	50	WORK	HOSPITAL_GENERAL_INFORMATION
LOCATION	char	88	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
LOCATION	char	88	SASGF	PATIENT_SURVEY__HCAHPS___HOSPIT
LOCATION	char	86	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
LOCATION	char	89	WORK	HOSPITAL_GENERAL_INFORMATION
MEASURE_ID	char	25	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
MEASURE_ID	char	15	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
MEASURE_NAME	char	72	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
MEASURE_NAME	char	98	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
STATE	char	2	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
STATE	char	2	SASGF	PATIENT_SURVEY__HCAHPS___HOSPIT
STATE	char	2	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
STATE	char	2	WORK	HOSPITAL_GENERAL_INFORMATION
ZIP_CODE	num	8	SASGF	COMPLICATIONS_AND_DEATHS___HOSPI
ZIP_CODE	num	8	SASGF	PATIENT_SURVEY__HCAHPS___HOSPIT
ZIP_CODE	num	8	WORK	HEALTHCARE_ASSOCIATED_INFECTIONS
ZIP_CODE	num	8	WORK	HOSPITAL_GENERAL_INFORMATION

Abstract

Introduction

SAS Macro

Results

Discussion

Please use the headings above to navigate through the different sections of the poster





## Discussion:

- ❑ When variables exist in multiple datasets involved in a merge, and they're not listed as BY variables, you need to ensure they are read from a single "most correct" source, or there's a risk the incorrect value is saved.
- ❑ The SAS macro **CHECK\_VAR\_EXIST** was written to aid programmers in identifying more efficiently which variables could be wrongly overwritten even before the merging is done.
- ❑ The output of the macro is used to determine where to include a DROP or KEEP statement. It can also be used to determine the maximum length for each common variable, which could be handy when concatenating datasets using the SET statement, to prevent the truncation of the variable. Another use is to determine if any of the common variables have different types (character vs numeric).
- ❑ A simpler but less efficient way to check for common variables is by using **OPTIONS MSGLEVEL=I**. Setting MSGLEVEL to I will make the log display additional notes pertaining to the merge processing. However, this requires you to run the DATA step merging first and then check the log after.

```
124 data hospital_results;
125 merge Hospital_general_information Healthcare_associated_infections;
126 by Provider_ID;
127 run;

NOTE: There were 5334 observations read from the data set WORK.HOSPITAL_GENERAL_INFORMATION.
NOTE: There were 171288 observations read from the data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
NOTE: The data set WORK.HOSPITAL_RESULTS has 171864 observations and 36 variables.
NOTE: DATA statement used (Total process time):
      real time           0.42 seconds
      cpu time            0.46 seconds

128
129 options msglevel=i;
130 data hospital_results;
131 merge Hospital_general_information Healthcare_associated_infections;
132 by Provider_ID;
133 run;

INFO: The variable Hospital_Name on data set WORK.HOSPITAL_GENERAL_INFORMATION will be
overwritten by data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable Address on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten by
data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable City on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten by data
set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable State on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten by
data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable ZIP_Code on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten by
data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable County_Name on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten
by data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable Phone_Number on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten
by data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
INFO: The variable Location on data set WORK.HOSPITAL_GENERAL_INFORMATION will be overwritten by
data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
NOTE: There were 5334 observations read from the data set WORK.HOSPITAL_GENERAL_INFORMATION.
NOTE: There were 171288 observations read from the data set WORK.HEALTHCARE_ASSOCIATED_INFECTIONS.
NOTE: The data set WORK.HOSPITAL_RESULTS has 171864 observations and 36 variables.
NOTE: DATA statement used (Total process time):
      real time           0.60 seconds
      cpu time            0.61 seconds
```



The background of the banner features a scenic view of the Washington Monument at dusk, reflected in the water of the Tidal Basin. The sky is a mix of blue, purple, and orange. In the foreground, there are cherry blossom trees with pink and white flowers, and a stone walkway. A dark teal rectangular box is centered over the image, containing the event title in white and teal text.

# SAS<sup>®</sup> GLOBAL FORUM 2020

USERS PROGRAM

MARCH 29 - APRIL 1 | WASHINGTON, DC | #SASGF

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 trademarks of their respective companies.