```
%macro mu_check_data_and_var_exist(
 data_to_check =
,vars_to_check_in_all_data =
,vars_to_check_in_respective_data =
,abort_if_does_not_exist = No
,help =
,debug =
) / store source des='V1.0.0.1';
%*************************************************************************
FILENAME: MU_CHECK_DATA_EXIST.SAS
DEVELOPER: (b) (6)
PLATFORM: PC SAS 9.1.3, 9.2
DATE: 20120215
PURPOSE:  This macro will check to see if specified data sets and variables exist.
It will
                  also create global macro variables to indicate whether the
dataset / variable
                  exists or not.  See examples of the macro calls below for further
discussion
                  about these global macro variables.
NOTE:
MACROS USED: mu_wordscan

PARAMETERS:

DATA_TO_CHECK = the data set(s) to check if they exist.  In the form LIB.DATA,
        where lib may be ommitted if WORK.
        (REQUIRED)

VARS_TO_CHECK_IN_ALL_DATA = the variable(s) to check if they exist in all the data
        set(s) specified in parameter DATA_TO_CHECK. Space dellimited.
        (OPTIONAL)

VARS_TO_CHECK_IN_RESPECTIVE_DATA = the variable(s) to check if they exist in the
        respective data in parameter DATA_TO_CHECK. Delimit with the exclamation
point (!)
        if you want to check the next dataset in DATA_TO_CHECK for specified
variable(s).
        (OPTIONAL)

        The examples below illustrate how the 3 parameters above should be
populated in
        conjunction with each other.

        Example 1:

        Valid call: macro will check if datasets ADB.ADSL and BIGN exist. Global
macro variables
        DATA1_EXIST and DATA2_EXIST will be created. Note that if parameter
ABORT_IF_DOES_NOT_EXIST
```

```
        is Y(es), the macro may abort before it has a chance to create a macro
variable.  Thus, if
        dataset ADB.ADSL does not exist and ABORT_IF_DOES_NOT_EXIST is Y(es), the
macro variable
        DATA2_EXIST is not created, since the macro aborts after it recognizes that
ADB.ADSL
        does not exist.

        %mu_check_data_and_var_exist(
                data_to_check = adb.adsl bign
                ,vars_to_check_in_all_data =
                ,vars_to_check_in_respective_data =
        )

        Example 2:

        Valid call: macro will check if variables TRTVAR, USUBJID, and SAFETY exist
in dataset
        ADB.ADSL and if variables TRTVAR, COUNTRY, and N exist in dataset BIGN.
Global macro
        variables created will be DATA1_EXIST, DATA2_EXIST, VAR1_IN_DATA1_EXIST,
        VAR2_IN_DATA1_EXIST, VAR3_IN_DATA1_EXIST, VAR1_IN_DATA2_EXIST,
VAR2_IN_DATA2_EXIST,
        VAR3_IN_DATA2_EXIST.  Note that VAR1 always refers to TRTVAR, since the
macro always
        checks the variables in VARS_TO_CHECK_IN_ALL_DATA first. Note that if
parameter
        ABORT_IF_DOES_NOT_EXIST is Y(es), the macro may abort before it has a
chance to create a
        macro variable. See Example 1 above for further discussion.

        %mu_check_data_and_var_exist(
                data_to_check = adb.adsl bign
                ,vars_to_check_in_all_data = trtvar
                ,vars_to_check_in_respective_data = usubjid safety! country n
        )

        Example 3:

        Valid call: macro will check if variables SOC_TXT and PT_TXT exist in
dataset ADB.ADAE and
        if variables TRTVAR and USUBJID exist in dataset ADB.ADSL.

        %mu_check_data_and_var_exist(
                data_to_check = adb.adae adb.adsl
                ,vars_to_check_in_all_data =
                ,vars_to_check_in_respective_data = soc_txt pt_txt ! trtvar usubjid

        )
```

Example 4:

Valid call: macro will check if variables TRTVAR and USUBJID exist in all three datasets:
ADB.ADSL, WORK.BIGN, and ADB.ADAE

```
%mu_check_data_and_var_exist(
        data_to_check = adb.adsl work.bign adb.adae
        ,vars_to_check_in_all_data = trtvar usubjid
        ,vars_to_check_in_respective_data =
)
```

Example 5:

Valid call: macro will check if TRTVAR, USUBJID and SAFETY exist in ADB.ADSL,
if TRTVAR, COUNTRY, and N exist in WORK.BIGN, and if TRTVAR exists in ADB.ADAE.

```
%mu_check_data_and_var_exist(
        data_to_check = adb.adsl work.bign adb.adae
        ,vars_to_check_in_all_data = trtvar
        ,vars_to_check_in_respective_data = usubjid safety  ! country n
)
```

Example 6:

Valid call: macro will check if USUBJID and SAFETY exists in ADB.ADSL, if N and COUNTRY
exists in WORK.BIGN, and if dataset ADB.ADAE exists (will not check if any variables
exist in ADB.ADAE dataset, since VAR_TO_CHECK_IN_ALL_DATA is blank and there is only one
exclamation point in VAR_TO_CHECK_IN_RESPECTIVE_DATA, even though there are three datasets
specified in DATA_TO_CHECK).

```
%mu_check_data_and_var_exist(
        data_to_check = adb.adsl work.bign adb.adae
        ,vars_to_check_in_all_data =
        ,vars_to_check_in_respective_data = usubjid safety  ! country n
)
```

Example 7:

Invalid call: macro is unable to determine which dataset to check for variables SOC_TXT
and PT_TXT. The number of exclamation points in

```
VAR_TO_CHECK_IN_RESPECTIVE_DATA must
        alwasy be less then the number of datasets specified in DATA_TO_CHECK.
Macro will abort.

        %mu_check_data_and_var_exist(
                data_to_check = adb.adsl work.bign
                ,vars_to_check_in_all_data =
                ,vars_to_check_in_respective_data = usubjid safety  ! country n !
soc_txt pt_txt
        )

ABORT_IF_DOES_NOT_EXIST = Y(es) or N(o) - if a dataset or a variable does not
exist,
        should the macro abort ? (OPTIONAL) (default = No)

HELP = Y(es) if information about whether datasets and variables exist or not is
        desired (OPTIONAL) (default = Yes)


MODIFIED BY:            DATE:
*********************************************************************************;

%PUT ---------------------------------------------- ;
%PUT INFO: (&SYSMACRONAME) ;
%PUT INFO: Version 1.0 ;
%PUT -START----------------------------------------- ;

%global MU_CHECK_DATA_AND_VAR_EXIST_RC  VAR_RC;
%let MU_CHECK_DATA_AND_VAR_EXIST_RC = 0;
%let VAR_RC = 0;

%mu_help_debug;

%***** get current setting of mprint;
%let mprint_setting = %sysfunc(getoption(mprint));
%if &debug = Y %then %do;
        options mprint;
%end;

%if "&abort_if_does_not_exist" eq "" %then %do;
        %let abort_if_does_not_exist = N;
%end;

%* check if data set(s) exist;

%if "&data_to_check" eq "" %then %do;
        %let MU_CHECK_DATA_AND_VAR_EXIST_RC = 1;
        %put ALERT_P: &SYSMACRONAME - Parameter DATA_TO_CHECK is null. It must be
populated.;
        %put ALERT_P: &SYSMACRONAME - Program will ABORT.;
```

```
%end;

%else %do;

        %mu_wordscan(string=&data_to_check, root=data, numw=num_data, delim=%str(
));

        %* if number of datasets specified is less than the number of exclamation
points in
                parameter VAR_TO_CHECK_IN_RESPECTIVE_DATA then abort program;

        %if "&vars_to_check_in_respective_data" ne "" %then %do;
                data temp;
                        resp="&vars_to_check_in_respective_data";
                        resp_=tranwrd(resp,'!',' ! ');
                        num_respective_vars = count(resp,'!')+1;
                        call
symputx('num_respective_vars',put(num_respective_vars,4.));
                run;

                data temp;
                        set temp;
                        length respective_var $200;
                        do iter=1 to &num_respective_vars;
                                respective_var = scan(resp_, iter, '!');
                                output;
                        end;
                drop iter;
                run;

                data _null_;
                        set temp;
                        call symputx('respective_vars' || compress(put(_n_,best.)),
respective_var);
                run;

                %if &num_respective_vars gt &num_data %then %do;
                        %let MU_CHECK_DATA_AND_VAR_EXIST_RC = 2;
                        %put ALERT_P: &SYSMACRONAME - The number of exclamation
points in parameter ;
                        %put ALERT_P: &SYSMACRONAME -
VAR_TO_CHECK_IN_RESPECTIVE_DATA must always be less than;
                        %put ALERT_P: &SYSMACRONAME - the number of datasets
specified in parameter;
                        %put ALERT_P: &SYSMACRONAME - DATA_TO_CHECK. Program will
ABORT.;
                %end;
        %end;
```

```
        %if &MU_CHECK_DATA_AND_VAR_EXIST_RC = 0 %then %do;

                %do iter = 1 %to &num_data;
                        %if %sysfunc(exist(&&data&iter)) eq 0 %then %do;
                                %let MU_CHECK_DATA_AND_VAR_EXIST_RC = 3;
                                %global data&iter._exist;
                                %let data&iter._exist = 0;
                                %put ALERT_I: &SYSMACRONAME - Dataset &&data&iter
does NOT exist.;
                                %if %substr(%upcase(&help,1,1)) = Y %then %do;
                                        %put ALERT_I: &SYSMACRONAME - Global macro
variable DATA&iter._EXIST created - value is 0.;
                                %end;
                        %end;
                        %else %do;
                                %global data&iter._exist;
                                %let data&iter._exist = 1;
                                %if %substr(%upcase(&help,1,1)) = Y %then %do;
                                        %put ALERT_I: &SYSMACRONAME - Dataset
&&data&iter exists.;
                                        %put ALERT_I: &SYSMACRONAME - Global macro
variable DATA&iter._EXIST created - value is 1.;
                                %end;
                        %end;
                %end;


                %* check_var macro loops around and checks to see if specified
variables exist in specified
                        datasets.;


                %macro check_var;
                        %mu_wordscan(string=&allvars, root=var, numw=numvar,
delim=%str( ));
                        %do iter = 1 %to &numvar;
                                %if %sysfunc(varnum(&&dsid_in_data&loop,
&&var&iter)) eq 0 %then %do;
                                        %if &MU_CHECK_DATA_AND_VAR_EXIST_RC ne 3
%then %do;
                                                %let VAR_RC = 1;
                                        %end;
                                        %else %if &MU_CHECK_DATA_AND_VAR_EXIST_RC
eq 3 %then %do;
                                                %let VAR_RC = 2;
                                        %end;
                                        %global var&iter._in_data&loop._exist;
                                        %let var&iter._in_data&loop._exist = 0;
                                        %put ALERT_I: MU_CHECK_DATA_AND_VAR_EXIST -
The variable &&var&iter does NOT exist in dataset;
```

```
                                        %put ALERT_I: MU_CHECK_DATA_AND_VAR_EXIST -
&&data&loop...;
                                        %if %substr(%upcase(&help,1,1)) = Y %then
%do;
                                                %put ALERT_I:
MU_CHECK_DATA_AND_VAR_EXIST - Global macro variable VAR&iter._IN_DATA&loop._EXIST
created.;
                                                %put ALERT_I:
MU_CHECK_DATA_AND_VAR_EXIST - Value of VAR&iter._IN_DATA&loop._EXIST is 0.;
                                        %end;
                                %end;
                                %else %do;
                                        %global var&iter._in_data&loop._exist;
                                        %let var&iter._in_data&loop._exist = 1;
                                        %if %substr(%upcase(&help,1,1)) = Y %then
%do;
                                                %put ALERT_I:
MU_CHECK_DATA_AND_VAR_EXIST - The variable &&var&iter exists in dataset
&&data&loop...;
                                                %put ALERT_I:
MU_CHECK_DATA_AND_VAR_EXIST - Global macro variable VAR&iter._IN_DATA&loop._EXIST
created.;
                                                %put ALERT_I:
MU_CHECK_DATA_AND_VAR_EXIST - Value of VAR&iter._IN_DATA&loop._EXIST is 1.;
                                        %end;
                                %end;
                        %end;
                        %let close_dsid_in_data =
%sysfunc(close(&&dsid_in_data&loop));
                %mend check_var;


                %*check to see if variables exist;

                %if "&vars_to_check_in_all_data" eq "" and
"&vars_to_check_in_respective_data" ne ""
                         %then %do;

                        %do loop = 1 %to &num_respective_vars;
                                %if &&data&loop._exist = 1 %then %do;
                                        %let dsid_in_data&loop =
%sysfunc(open(&&data&loop));
                                        %let allvars = &&respective_vars&loop;
                                        %check_var;
                                %end;
                                %else %do;
                                        %let allvars = &&respective_vars&loop;
                                        %put ALERT_I: &SYSMACRONAME - Since dataset
&&data&loop does not exist,;
                                        %put ALERT_I: &SYSMACRONAME - macro will
```

```
not check for existence of variable(s) ;
                                        %put ALERT_I: &SYSMACRONAME - &allvars. in
dataset &&data&loop;
                                %end;
                        %end;

                %end;


                %else %if "&vars_to_check_in_all_data" ne "" and
"&vars_to_check_in_respective_data" ne ""
                         %then %do;

                        %do loop = 1 %to &num_data;
                                %if &&data&loop._exist = 1 %then %do;
                                        %let dsid_in_data&loop =
%sysfunc(open(&&data&loop));
                                        %if &loop gt &num_respective_vars %then
%do;
                                                %let allvars =
&vars_to_check_in_all_data;
                                        %end;
                                        %else %do;
                                                %let allvars =
&vars_to_check_in_all_data &&respective_vars&loop;
                                        %end;
                                        %check_var;
                                %end;
                                %else %do;
                                        %if &loop gt &num_respective_vars %then
%do;
                                                %let allvars =
&vars_to_check_in_all_data;
                                        %end;
                                        %else %do;
                                                %let allvars =
&vars_to_check_in_all_data &&respective_vars&loop;
                                        %end;
                                        %put ALERT_I: &SYSMACRONAME - Since dataset
&&data&loop does not exist,;
                                        %put ALERT_I: &SYSMACRONAME - macro will
not check for existence of variable(s) ;
                                        %put ALERT_I: &SYSMACRONAME - &allvars. in
dataset &&data&loop;
                                %end;
                        %end;

                %end;
```

```
                %else %if "&vars_to_check_in_all_data" ne "" and
"&vars_to_check_in_respective_data" eq ""
                         %then %do;

                        %do loop = 1 %to &num_data;
                                %if &&data&loop._exist = 1 %then %do;
                                        %let dsid_in_data&loop =
%sysfunc(open(&&data&loop));
                                        %let allvars = &vars_to_check_in_all_data;
                                        %check_var;
                                %end;
                                %else %do;
                                        %let allvars = &vars_to_check_in_all_data;
                                        %put ALERT_I: &SYSMACRONAME - Since dataset
&&data&loop does not exist,;
                                        %put ALERT_I: &SYSMACRONAME - macro will
not check for existence of variable(s) ;
                                        %put ALERT_I: &SYSMACRONAME - &allvars. in
dataset &&data&loop;
                                %end;
                        %end;

                %end;

        %end;

%end;


%if  &VAR_RC = 1 %then %let MU_CHECK_DATA_AND_VAR_EXIST_RC = 4;
%else %if &VAR_RC = 2 %then %let MU_CHECK_DATA_AND_VAR_EXIST_RC = 5;

%PUT ------------------------------------------------ ;
%PUT INFO: MU_CHECK_DATA_AND_VAR_EXIST_RC = &MU_CHECK_DATA_AND_VAR_EXIST_RC;
%PUT ------------------------------------------------ ;

%if &MU_CHECK_DATA_AND_VAR_EXIST_RC = 1 or &MU_CHECK_DATA_AND_VAR_EXIST_RC = 2
%then %do;
        data _null_;
                abort;
        run;
%end;


%else %if &MU_CHECK_DATA_AND_VAR_EXIST_RC > 2 and
%substr(%upcase(&abort_if_does_not_exist,1,1)) = Y %then %do;
        %if &MU_CHECK_DATA_AND_VAR_EXIST_RC = 3 %then %do;
                %put ALERT_P: &SYSMACRONAME - At least one of the datasets in
parameter DATA_TO_CHECK ;
                %put ALERT_P: &SYSMACRONAME - does not exist.  Program will ABORT.;
```

```
	%end;

	%else %if &MU_CHECK_DATA_AND_VAR_EXIST_RC = 4 %then %do;
		%put ALERT_P: &SYSMACRONAME - At least one of the variables in
parameters;
		%put ALERT_P: &SYSMACRONAME - VARS_TO_CHECK_IN_ALL_DATA or ;
		%put ALERT_P: &SYSMACRONAME - VARS_TO_CHECK_IN_ALL_RESPECTIVE_DATA
does not exist.;
		%put ALERT_P: &SYSMACRONAME - Program will ABORT.;
	%end;

	%else %if &MU_CHECK_DATA_AND_VAR_EXIST_RC = 5 %then %do;
		%put ALERT_P: &SYSMACRONAME - At least one of the datasets in
parameter DATA_TO_CHECK ;
		%put ALERT_P: &SYSMACRONAME - does not exist. Also, at least one of
the variables in;
		%put ALERT_P: &SYSMACRONAME - parameters VARS_TO_CHECK_IN_ALL_DATA
or;
		%put ALERT_P: &SYSMACRONAME - VARS_TO_CHECK_IN_ALL_RESPECTIVE_DATA
does not exist.;
		%put ALERT_P: &SYSMACRONAME - Program will ABORT.;
	%end;

	data _null_;
		abort;
	run;
%end;

options &mprint_setting ;

%PUT ------------------------------------------------ ;
%PUT INFO: (&SYSMACRONAME) ;
%PUT INFO: Version 1.0 ;
%PUT -END------------------------------------------- ;

%mend mu_check_data_and_var_exist;
```