

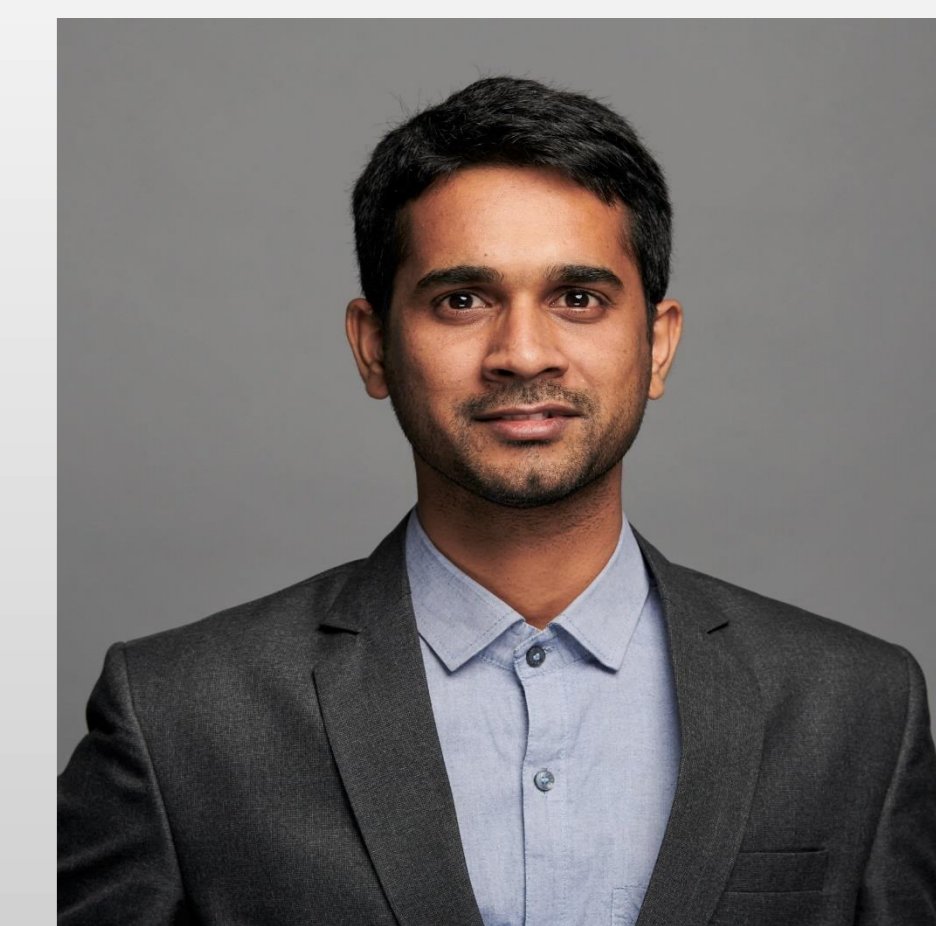
TAP TO GO
BACK TO
KIOSK MENU

SAS[®] GLOBAL FORUM 2020

MARCH 29 - APRIL 1
WASHINGTON, DC



USERS PROGRAM



Main Author

DATA
RELIABILITY

Abstract

- Archive
- Abort Process
- Change Datatype
- Replace Column Names
- E-mail Error Log
- Reliable Calculation
- Skip Report Generation
- Conditional Merge
- Auth Domain Use
- Global Macro Variable
- Conclusion

- This paper presents simple reliable techniques that can benefit SAS programmers to write reliable code. Before diving into the techniques, let me tell you why it is important. Reliable code can be dependent upon the delivery of the expected functionality and the performance in a consistent, repeatable manner for a specified duration.
- Reliability is application-specific and is largely dependent on the code written and very little on the language itself.
- Because unreliable code is considered worthless or not trusted upon, reliability is considered paramount to other coding requirements such as portability, efficiency, and performance.
- A reliable code handles failure very well. In this paper, we discuss a few possible scenarios of failure and how to handle them.

Summary of Scenarios:

1. Archiving datasets of the previous cycle before executing the current cycle.
2. Do not run the program if required input files are not available.
3. Handling change of column type while reading data from Excel file using PROC IMPORT.
4. Replacing special characters in column names with underscores.
5. When an error is encountered, aborting the program and sending an email to the user with the log file.
6. Replacing ‘.’ with ‘0’ while performing division.
7. Not generating a report if the data is not available and continuing for the next report.
8. Observation count when performing joining: if observations=0, aborting the program or implementing custom logic.
9. Using Auth Domain instead of UserID and passwords while connecting to ODBC.
10. Using the Global Macro variable with initialization.

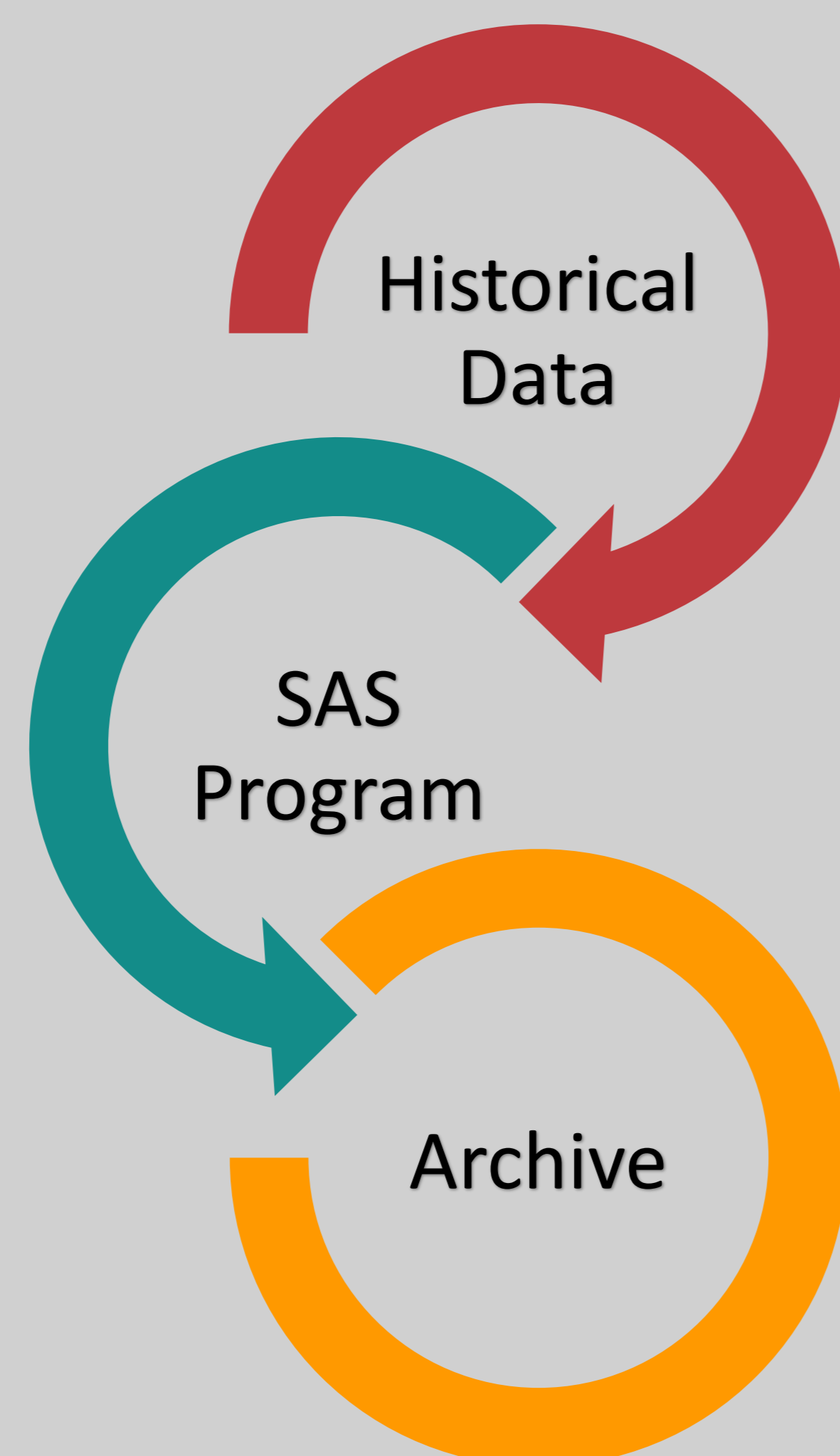
Archiving SAS Datasets

Why?

- Serves as a backup during data recovery.
- Reduces primary storage costs.
- Increases Security.

SAS Program

```
/*Zipping or Archiving SAS Datasets*/  
%let src=<<SOURCE_PATH>>;  
%let dest=<<DESTINATION_PATH>>;  
%let fname = <<ARCHIVE_FILE_NAME>>;  
/* Creating a ZIP file with ODS PACKAGE */  
ods package(newzip) open nopf;  
ods package(newzip)  
  add file="&src./<<SAS_DATASET>>.sas7bdat"  
  mimetype="application/x-compress";  
ods package(newzip)  
  add file="&src./<<SAS_DATASET>>.sas7bdat"  
  mimetype="application/x-compress";  
ods package(newzip) publish archive  
  properties (  
    archive_name="&fname..zip" archive_path="&dest."  
  );  
ods package(newzip) close clear;  
ods _all_ close;  
ods package close;
```



Info: A SAS Macro is written to make the above program work dynamically. For all the SAS datasets available in the folder, a Macro snippet will be available in the Abstract Submission.

Abstract

Archive

Abort Process

Change Datatype

Replace Column
Names

E-mail Error Log

Reliable

Calculation

Skip Report

Generation

Conditional Merge

Auth Domain Use

Global Macro

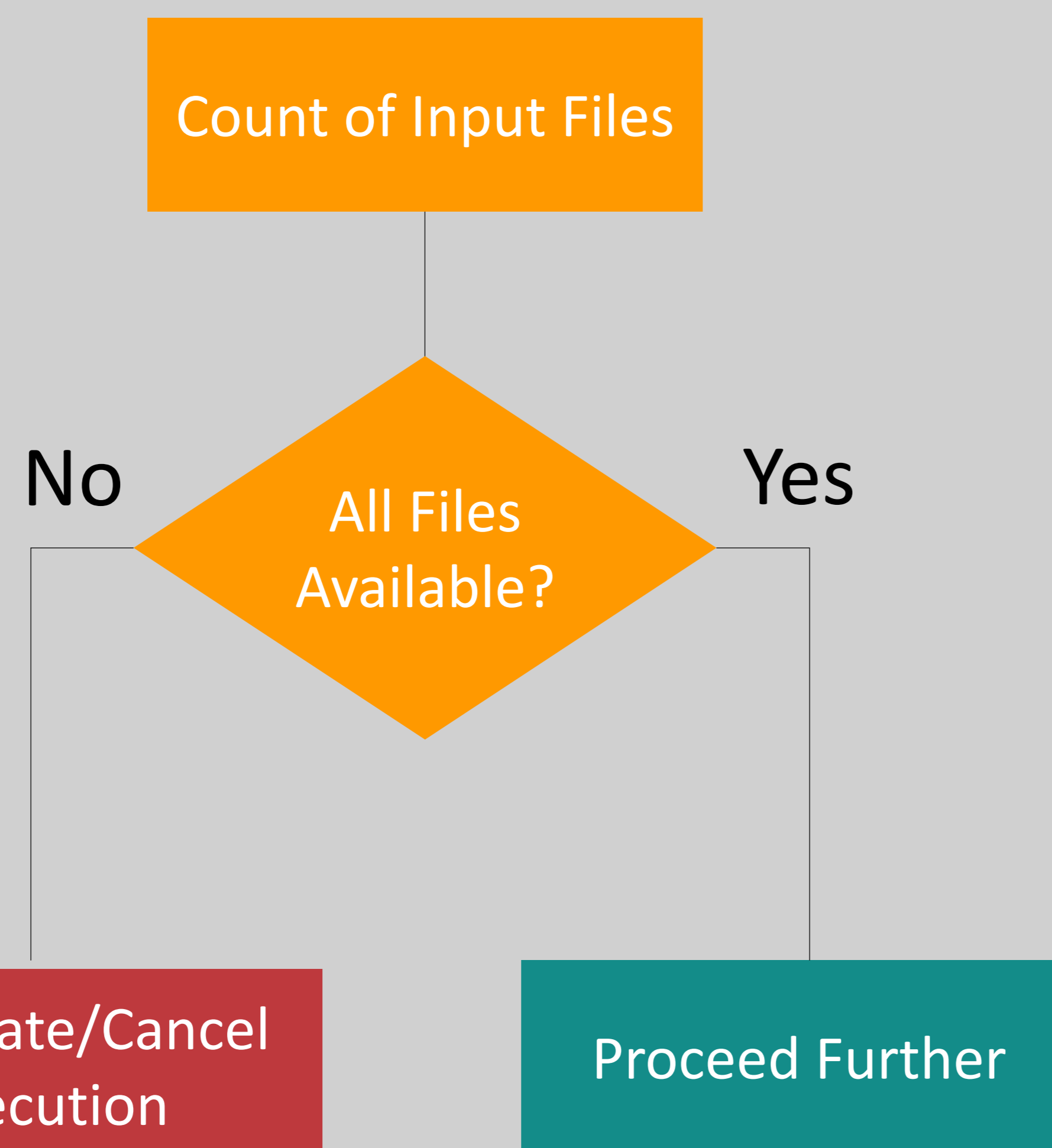
Variable

Conclusion

Abort Process if Required Input Files are Not Available

Why?

- Stops execution of DATA steps, if the required input files are not available.
- **ABORT CANCEL** statement aborts the program without terminating the SAS Session.
- Writes an error message to the SAS log.



SAS Program

```
/*Abort Program if Required input files are not available for data processing*/  
filename tmp pipe 'dir "&DIR_PATH\*.xlsx" ';  
  
data want;  
  infile tmp dlm="~";  
  length cmd_line $2000;  
  input cmd_line;  
  
run;  
  
data _null_;  
  set want nobs=n;  
  if n = 4 then  
    put 'All Files are available';  
  else  
  do;  
    put 'Some Files are Required';  
    abort cancel;  
  end;  
  
run;
```



- Abstract
- Archive
- Abort Process
- Change Datatype
- Replace Column Names
- E-mail Error Log
- Reliable Calculation
- Skip Report Generation
- Conditional Merge
- Auth Domain Use
- Global Macro Variable
- Conclusion

Data Type Change While Reading Excel File Using PROC IMPORT

Why?

- If you are reading Excel file using PROC IMPORT and the data type of column is not consistent from source systems.
- When using such a dataset in Joins, the code fails if the type of KEY column changes.
- Created a Macro which checks for the Data type and then updates it accordingly.



SAS Program

```
/*If you are expecting variable of type character, but you received variable as numeric type,
use below code to convert to character type*/
%macro NtoC(var,inputtable,outputtable);
data &outputtable.;
  set &inputtable.;
  %let app=&var;
  %put &app;
  if vtype(&var.)="N" then do;
    v=strip(put(round(&var.),best.));
    drop &var.;
    rename v=&var.;
  end;
run;
%mend;
```

/*If you are expecting variable of type numeric, but you received variable as character type,
use below code to convert to numeric type*/

```
%macro CtoN(var,inputtable,outputtable);
data &outputtable.;
  set &inputtable.;
  if vtype(&var.)="C" then do;
    v=input(&var.,best.);
    drop &var.;
    rename v=&var.;
  end;
run;
%mend;
```



Column Names: Replace Special Characters with Underscores

Why?

- Often we come across raw data that has special characters in their column names.
- Time-consuming to type such column names while programming.
- Helps in maintaining standards in naming Variables.

Limitations

- The showcased snippets will only work when all the columns have special characters in them.

SAS Program

```
/*Replace column names with special characters with underscores*/  
/*Code to Import Data*/  
proc import  
    datafile="<<DIRECTORY_PATH>>\Book1.xlsx"  
    dbms=xlsx  
    out=birthwgt replace;  
run;  
/*Code to Rename Columns with special characters in a Dataset*/  
data _null_;  
    set birthwgt (obs=1);  
    length VAR_NAME $32;  
    call execute('proc datasets noprint; modify birthwgt; rename');  
    do while(1);  
        call vnext (VAR_NAME);  
        if VAR_NAME = 'VAR_NAME' then leave;  
        call execute(catt(quote(trim(VAR_NAME), "'"), 'n=',  
            translate(trim(VAR_NAME), '_____', '&#%*^@'".>, <;: ' '))  
        );  
    end;  
    call execute('; quit;');  
run;
```



WRITING RELIABLE SAS® PROGRAMS

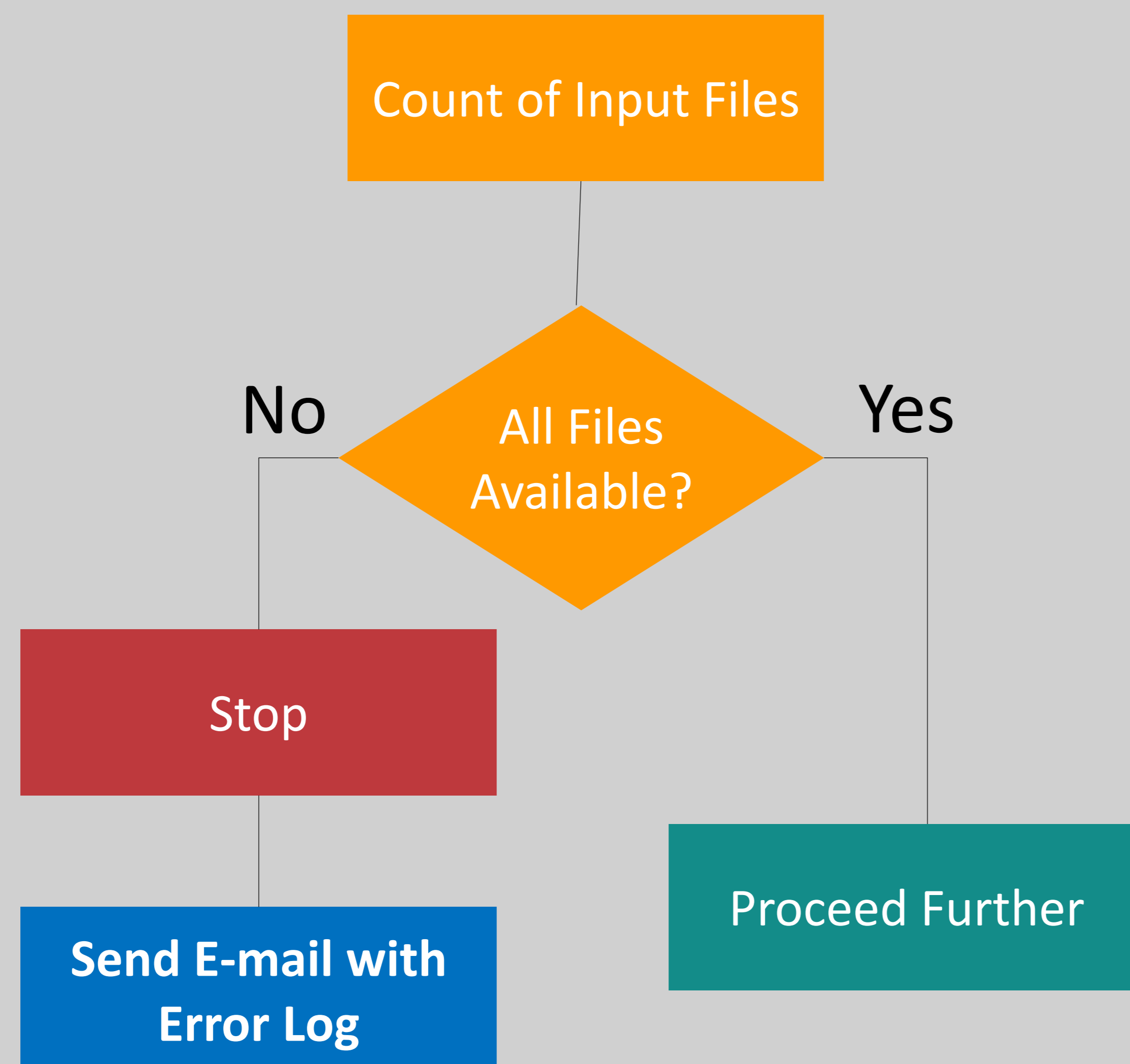
Balraj Pitlola & Venkata Karimiddela

Core Compete

E-mail - Error Log and Abort the Program

Why?

- While starting any data processing job, check whether the required number of files are available or not.
- If some files are not available, then stop the SAS program and send an E-MAIL to the recipient along with the error log.



SAS Program

```
/*Abort Program and send E-mail with Error Log as attachment*/
filename tmp pipe 'dir "<<DIRECTORY_PATH>>\*.xlsx" /b';
data want;
  infile tmp dlm="~"; length cmd_line $2000; input cmd_line;
run;
%let log_path = <<LOG_LOCATION>>;
%let toemail = "FIRST.LAST@XXX.com";
proc printto log= "&log_path.\test.log";run;
data _null_;
  set want nobs=n;
  if n = 4 then put 'All Files are available';
  else
  do;
    put 'Some Files are Required';
  end;
  stop;
run;
/*Setting Log to default*/
Proc printto;run;
/*Sending e-mail */
FILENAME Mailbox EMAIL TO=(&toemail)
  Subject="Monthly Job Execution Error"
  Attach("&log_path\test.log" );
data _null_;
  FILE Mailbox;
  PUT "Required Input files are not available";
  PUT "Re-run the job, once all the files are made available";
run;
```



Abstract
Archive
Abort Process
Change Datatype
Replace Column
Names

E-mail Error Log

Reliable
Calculation
Skip Report
Generation
Conditional Merge
Auth Domain Use
Global Macro
Variable
Conclusion

WRITING RELIABLE SAS® PROGRAMS

Balraj Pitlola & Venkata Karimiddela

Core Compete

Replace '.' With '0' While Performing Division

Why?

- While performing calculations on Numeric variables, if it contains '.' Or MISSING values then the calculation returns MISSING.
- Treat such variables before calculation to generate acceptable measures/values.

Output: Before

prod_id	yr_2018	yr_2019	yoy
B00005T3BD	1130	1190	5.31%
B00005T3C8	1120	.	.
B00006I506	5643	3243	-42.53%
B000092TTO	4323	.	.
B0002KVQBA	1000	1100	10.00%
B000FMNBXG	7632	5432	-28.83%

Output: After

prod_id	yr_2018	yr_2019	yoy
B00005T3BD	1130	1190	5.31%
B00005T3C8	1120	0	-100.0%
B00006I506	5643	3243	-42.53%
B000092TTO	4323	0	-100.0%
B0002KVQBA	1000	1100	10.00%
B000FMNBXG	7632	5432	-28.83%

SAS Program

```
/*BEFORE: Output*/
data yoy;
  set prod_data;
  format yoy percentn8.2;
  yoy = (yr_2019 - yr_2018)/yr_2018;
run;

/*Replacing Numeric variables missing values with 0's*/
data yoy1;
  set prod_data;
  array col _numeric_;
  do over col;
    if col=. then col=0;
  end;
run;

/*AFTER: Output*/
data yoy_final;
  set yoy1;
  format yoy percentn8.2;
  yoy = (yr_2019 - yr_2018)/yr_2018;
run;
```



- Abstract
- Archive
- Abort Process
- Change Datatype
- Replace Column Names
- E-mail Error Log
- Reliable Calculation**
- Skip Report Generation
- Conditional Merge
- Auth Domain Use
- Global Macro Variable
- Conclusion

WRITING RELIABLE SAS® PROGRAMS

Balraj Pitlola & Venkata Karimiddela

Core Compete

Not Generating a Report if the Data is Not Available and Continuing for Next Report

Why?

- To generate a report only when there is data available for a particular criteria/customer.
- If data is not available, then write a message to the Log.

Output: *When Data is available*

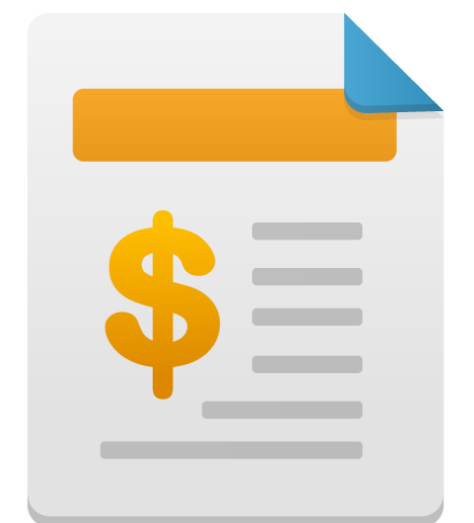
Name	Sex	Age	Height	Weight
Alfred	M	14	69.0	112.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83.0
Jeffrey	M	13	62.5	84.0
John	M	12	59.0	99.5
Philip	M	16	72.0	150.0

Output: *When Data is not available*

"NO DATA AVAILABLE FOR THE GIVEN SELECTIONS"

SAS Program

```
/******  
***** PRINTING CUSTOMERS WHOSE GENDER MALE/FEMALE AND AGE BETWEEN XX AND YY *****  
*****  
%MACRO GENDER_AGE (GENDER=M, MINIMUM_AGE=21, MAXIMUM_AGE=75);  
%macro d;  
%mend d;  
DATA CLASS;  
    SET SASHELP.CLASS;  
    WHERE SEX="&GENDER." AND AGE BETWEEN &MINIMUM_AGE. AND &MAXIMUM_AGE.;  
RUN;  
  
PROC SQL NOPRINT;  
    SELECT COUNT(*) INTO: NUMBER_OF_RECORDS FROM CLASS;  
QUIT;  
  
%IF &NUMBER_OF_RECORDS. GT 0 %THEN %DO;  
    PROC PRINT DATA=CLASS; /* IDEAL CASE WE WILL HAVE MORE PROCESSING AT THIS PLACE */  
    RUN;  
%END;  
%ELSE %DO;  
    %PUT "NO DATA AVAILABLE FOR THE GIVEN SELECTIONS";  
%END;  
%MEND GENDER_AGE;  
%GENDER_AGE (GENDER=M, MINIMUM_AGE=10, MAXIMUM_AGE=20);
```



Abstract
Archive
Abort Process
Change Datatype
Replace Column
Names
E-mail Error Log
Reliable
Calculation
Skip Report
Generation
Conditional Merge
Auth Domain Use
Global Macro
Variable
Conclusion

WRITING RELIABLE SAS® PROGRAMS

Balraj Pitlola & Venkata Karimiddela

Core Compete

Joining Only When Datasets Have Observations

Why?

- To keep a check that the **MERGE** has happened as expected.
- There could be a case where there are no observations in the datasets while performing **MERGE** operation.
- The Column from the other dataset is used in the next steps and could lead to improper **MERGE**.

Output: Before

prod_id	yr_2018	yr_2019	prod_name
B00005T3BD	1130	1190	
B00005T3C8	.	1120	
B00006BMQT	2345	7865	
B00006I506	5643	3243	
B000092TTO	.	4323	

Output: After

prod_id	yr_2018	yr_2019	prod_name
---------	---------	---------	-----------

Output: Log

Merge failed due to insufficient observations

SAS Program

```
/*Data Step to read 0 Observations*/
data prod_name;
  set prod_name(obs=0);
run;

/*Data Step to create Macro variable for Observation Count*/
data _null_;
  call symput('obsCnt', 0);
  set work.prod_name nobs=n;
  call symput('obsCnt', n);
stop;
run;

/*Data Step, to merge based on observation count*/
data merged_data;
  if input(&obsCnt, 8.) > 0 then
    do;
      merge work.prod_data work.prod_name;
      by prod_id;
    end;
  else;
  do;
    put "Merge failed due to insufficient observations";
    /*Trigger E-mail to Notify the User*/
    stop;
  end;
end;
```



Abstract
Archive
Abort Process
Change Datatype
Replace Column
Names
E-mail Error Log
Reliable
Calculation
Skip Report
Generation

Conditional Merge

Auth Domain Use
Global Macro
Variable
Conclusion

Auth Domain

Why?

- The **AUTHDOMAIN=** option enables you to avoid explicitly specifying a database user ID and password in a SAS program. You define an authorization domain in metadata and define the associated credentials in a metadata repository. Then, you can simply reference the authorization domain in a **LIBNAME** statement or in the SQL procedure pass-through code.
- The **SAS/ACCESS** engine accepts the credentials as if the user-specified them in the **USER=** and **PASSWORD=** options.
- The authentication domain and the associated login definition must be stored in a metadata repository and the metadata server must be running in order to resolve the metadata object specification.

SAS Program

```
/* 10. Connecting to ODBC using Auth Domain*/  
proc sql;  
    connect to oracle(authdomain="oracle-authorization-domain" path=Oracle-database);  
    create table foo as  
        select * from connection to oracle  
            (select * from employees where jobcode=602);  
quit;  
  
libname oralib oracle authdomain="oracle-authorization-domain" path=alien;
```



Using Global Macro Variable With Initialization

Why?

- Programmers mistakenly assume that the %GLOBAL macro statement both creates and initializes a macro variable, resetting it to a blank value.
- If a global macro variable already exists and you specify that variable in a %GLOBAL statement, the existing value remains unchanged.
- To prevent failure, explicitly clear the value in global macro variables.

Output: Before

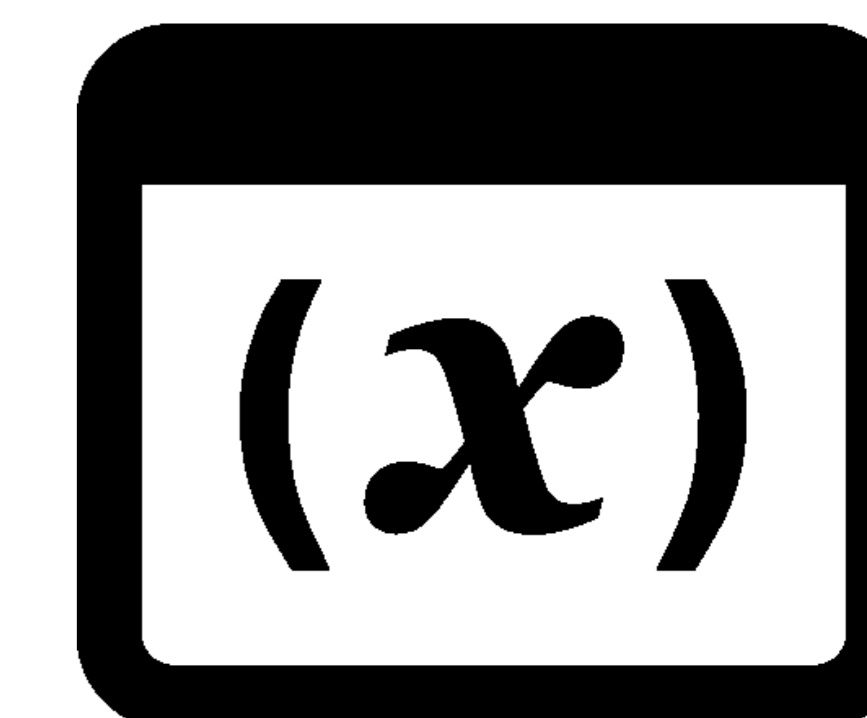
```
Running for 1st time
34      %put &=var;
VAR=1 2 3 4 5
35      %test;
36      %put Running for 2nd time;
Running for 2nd time
37      %put &=var;
VAR=1 2 3 4 5 1 2 3 4 5
```

Output: After

```
Running for 1st time
35      %put &=var;
VAR=1 2 3 4 5
36      %test;
37      %put Running for 2nd time;
Running for 2nd time
38      %put &=var;
VAR=1 2 3 4 5
```

SAS Program

```
/* 9. Using Global Macro Variable with Initialization*/
%macro test;
%global var;
%do i=1 %to 5;
    %let var = &var &i;
%end;
%mend;
%test;
%put Running for 1st time;
%put &=var;
%test;
%put Running for 2nd time;
%put &=var;
/*Global variable with Initialization*/
%macro test;
%global var;
%let var=; *Corrects the above defect;
%do i=1 %to 5;
    %let var = &var &i;
%end;
%mend;
%test;
%put Running for 1st time;
%put &=var;
%test;
%put Running for 2nd time;
%put &=var;
```



- Abstract
- Archive
- Abort Process
- Change Datatype
- Replace Column Names
- E-mail Error Log
- Reliable Calculation
- Skip Report Generation
- Conditional Merge
- Auth Domain Use
- Global Macro Variable**
- Conclusion

Thank You 😊

To conclude, Reliability is a measure of performance against failure. A reliable code is fault-tolerant and allows us to recover in case of failure. SAS Programmers can earn trust and respect of stakeholders by implementing **RELIABLE** programming techniques during development to enhance customer experience leading to Customer Delight.

Your comments and questions are valued and encouraged. Contact the author at:

balraj.pitlola@corecompete.com

Analyst, Data Science

CORE COMPETE

Hyderabad, India

References

- Book by **Troy Martin Hughes** titled – “**SAS Data Analytic Development**”
- SAS Communities: <https://communities.sas.com/>
- Auth Domain:
<https://documentation.sas.com/?docsetId=acrelldb&docsetTarget=n0aiq25zc8u8u6n1i81my0a24sd3.htm&docsetVersion=9.4&locale=en>

Abstract

Archive

Abort Process

Change Datatype

Replace Column
Names

E-mail Error Log

Reliable

Calculation

Skip Report

Generation

Conditional Merge

Auth Domain Use

Global Macro

Variable

Conclusion

The background of the banner is 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 pink. In the foreground, there are cherry blossom trees with pink and white flowers, and a stone walkway. A dark blue rectangular box is centered over the image, containing the event title in white and teal text.

SAS[®] 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.