Paper 11778-2016

# Comparative study of Proc Export and ODS

Saurabh Nandy, SAS® OSU Data Mining and Certificate Program, Oklahoma State University;
Dr. Goutam Chakraborty, Oklahoma State University

## ABSTRACT

Imagine that you have a very large dataset and you have some specific values in one of the columns of the dataset and you want to classify the entire dataset into different csv sheets based on the values present in that specific column. Perhaps you think you will use codes using IF/THEN and ELSE statement conditions in SAS along with some OUTPUT statements. Considering the fact that you are thinking to divide that dataset into csv sheets, it kind of makes it more frustrating to do that using the conventional manual process of converting each of the separated datasets into csv files. This paper looks at a comparative study of using the Macro command in SAS with the help of proc Export statement and ODS command using proc tabulate. In these two processes, the whole tedious process is done automatically using the SAS code. .

## INTRODUCTION

Analyzing huge datasets often run into a problem related to memory issues of that system. To help users in handling huge datasets, the SAS software provides multiple ways to break a large dataset into separate datasets using the records of one of the columns present in that dataset.. Such as:

```
Data (new dataset being created);
Set (dataset from which the data is imported);
       If (argument checking the condition)
      Output (* dataset)
       Else (..)
       Output(* dataset);
Run;
```

The most common step to accomplish this task would be using the conventional IF/ELSE statements where the arguments of this statement determine the next possible step after checking the conditions and with particular OUTPUT statements as shown above.

## USING MACRO

Macros automate the general tiresome process of running long written codes. It generalizes the code for the entire dataset. Creating separate excel files using macros is relatively easy then using the data steps. One can also limit the number of records in the individual excel files created by using OBS and FIRSTOBS conditions. For example in the following code:

```
%let path = (destination path);
%put &path;

%macro export(data,file);
 proc export data=<dataset imported>(where=(<Column Name>=:"&data"))
    outfile="&path.\&file..csv"
    dbms=csv
    replace;
  run;
 %mend;
%export(<Column Name>,<File to be created>);
```

Here in the above code, the first macro i.e. %LET calls the variable “path” which stores the location of the destination folder and the following %PUT writes the destination path into the SAS log.

Next, the macro export consists of the dataset that is being classified into numerous small datasets belonging to the record in its one particular column. The PROC EXPORT command makes it easy to export the data into csv format with a where condition that generalizes for distinct column names present in that dataset. The OUTFILE statement consists of the generalized final destination path file format. The macro is closed with a %MEND macro statement. The next line is the final and the important line to call the macro defined above which exports the CSV files according to the names of the records present in that column.

## USING ODS

The ODS statement is a global statement that gives the instructions and commands to the output delivery system. It is mainly used to provide different destination, selecting templates for the generated output files or to include or exclude a particular output. ODS command also creates files which are Excel ready and using PROC TEMPLATE, it can be customized according to one’s demand. Using ODS gives the flexibility to convert the dataset into individual HTML with an XLS extension.

```
ods HTML file=<Dataset>;
proc print data= Destination data;
 run;
ods HTML close;
```

For converting to CSV file, the below code can be used.

```
ods CSV file=<Dataset>;
 proc print data= Destination data label;
 var <variables>;
 label ;
 run;
ods CSV close;
```

Using the PROC TABULATE statement with ODS, datasets can be created the customized excel sheets.

Sometimes, there is confusion with the nomenclature of the term ODS i.e., it is either an Output object or Output destination. ODS produces an output object no matter what file destination we provide. To simply produce the ODS Output, one has to write a statement i.e.

```
Ods output <ODS table name> = <designated table name>;
```

The ODS also gives us an option to give us information on the outputs generated. That command is known ODS Trace ON. It's written prior to Proc statement in the SAS command. Like:

```
ODS trace on;
Proc <statement>
Run;
```

The ODS SELECT command enables a user to select or de-select the tables in the output. Also, the ODS EXCLUDE statement does the same thing of excluding the tables from the output.

For example: On running the following SAS written code:

```
DATA sau23;
  INPUT id female race ses sctype $ prog
              read write math science socst;
DATALINES;
```

```
 147 1 1 3 pub 1 47  62  53  53  61
 108 0 1 2 pub 2 34  33  41  36  36
  18 0 3 2 pub 3 50  33  49  44  36
 153 0 1 2 pub 3 39  31  40  39  51
  50 0 2 2 pub 2 50  59  42  53  61
  51 1 2 1 pub 2 42  36  42  31  39
 102 0 1 1 pub 1 52  41  51  53  56
  57 1 1 2 pub 1 71  65  72  66  56
 160 1 1 2 pub 1 55  65  55  50  61
 136 0 1 2 pub 1 65  59  70  63  51
  88 1 1 1 pub 1 68  60  64  69  66
 177 0 1 2 pri 1 55  59  62  58  51
  95 0 1 1 pub 1 73  60  71  61  71
;
RUN;
```

Where sau23 is the dataset created. The input variables are ID, gender types i.e., male and female, race of the student, school type, program types and including the subject variables like math, science etc. On this dataset of student scores, we do a t-test on writing score and math scores for the different program types. And we want to save the p-values and t-values to use in other datasets. Without using ODS, it'll be a difficult thing and including ODS statement which is only one line will make this task easier. For this, first we'll sort the data and then use ODS statement to create a dataset with the required values.

```
proc sort data=sau23;
 by prog;
proc ttest data=sau23;
  by prog;
paired write*math;
ods output Ttests=ttest_output;
run;
proc print data=ttest_output;
run;
```

It gives the following results:

The temporary dataset table can be seen in the figure below:

| | prog | Variable1 | Variable2 | Difference | tValue | DF | Probt |
|---|---|---|---|---|---|---|---|
| 1 | 1 | write | math | write - math | -1.12 | 7 | 0.2985 |
| 2 | 2 | write | math | write - math | 0.12 | 2 | 0.9122 |
| 3 | 3 | write | math | write - math | -3.57 | 1 | 0.1738 |

**Display 1. Temporary dataset after running the above code**

```
prog=1

The TTEST Procedure

Difference:  write - math

 N          Mean       Std Dev      Std Err      Minimum      Maximum

 8       -3.3750        8.5011       3.0056     -11.0000      10.0000

    Mean        95% CL Mean         Std Dev       95% CL Std Dev

 -3.3750     -10.4821    3.7321       8.5011       5.6207   17.3020

    DF     t Value     Pr > |t|

     7       -1.12       0.2985
```

**Display 2. The results snapshot of the code written above**

The above result snapshot is of the one of the programs from the created dataset. Now, since the output is not that clear in terms of information, we go forward to use the ODS TRACE statement to get the information of the output respectively i.e. :

```
ods trace on /listing;
proc reg data=sau23;
        model write = female math;
run;
quit;
ods trace off;
ods listing close;
```

On using the ODS trace along with listing statement , we get the different information of the t-test done between the above two variables with the corresponding output. It is necessary to close the ODS listing

statement after the code is run. The results is shown below iving the information of different outputs of the analysis done on the data i.e. ANOVA analysis.

The output of the above code using ODS TRACE and LISTING command is shown below:

```
The REG Procedure
Model: MODEL1
Dependent Variable: write

Output Added:
-------------
Name:       NObs
Label:      Number of Observations
Template:   Stat.Reg.NObs
Path:       Reg.MODEL1.Fit.write.NObs
-------------


Number of Observations Read          13
Number of Observations Used          13

Output Added:
-------------
Name:       ANOVA
Label:      Analysis of Variance
Template:   Stat.REG.ANOVA
Path:       Reg.MODEL1.Fit.write.ANOVA
-------------


                          Analysis of Variance

                                   Sum of           Mean
Source                     DF        Squares         Square    F Value    Pr > F

Model                       2     1333.57424      666.78712       7.36    0.0108
Error                      10      906.42576       90.64258
Corrected Total            12     2240.00000

Output Added:
-------------
Name:       FitStatistics
Label:      Fit Statistics
Template:   Stat.REG.FitStatistics
Path:       Reg.MODEL1.Fit.write.FitStatistics
-------------


Root MSE              9.52064    R-Square     0.5953
Dependent Mean       51.00000    Adj R-Sq     0.5144
Coeff Var            18.66792

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL1.Fit.write.ParameterEstimates
-------------

The REG Procedure
```

Model: MODEL1
Dependent Variable: write

Parameter Estimates

| Variable | DF | Parameter Estimate | Standard Error | t Value | Pr > \|t\| |
|---|---|---|---|---|---|
| Intercept | 1 | 6.04510 | 12.86771 | 0.47 | 0.6486 |
| female | 1 | 7.69630 | 5.50523 | 1.40 | 0.1923 |
| math | 1 | 0.76676 | 0.23323 | 3.29 | 0.0082 |

**Output 1. Information on the individual points from Display 2**

The ODS has some of the advantages such as it gives the ability to generate, format the SAS procedures and outputs. It can be used to create reports, presentations using the different destination styles and formats.

Taking a comparison on few of the statistics of the above commands i.e., Export, ODS (CSV, XML, and Excel) on a dataset of approximately equal to or less than 1 GB, we can see clear differences in their performances. We can see that Proc Export has some better results compared to ODS in terms of speed and accessibility. The table below summarizes the points..

## CONCLUSION

| Method | Performance | SAS Code | Pros | Cons |
|---|---|---|---|---|
| ODS CSV | Create time- 10.7 min<br>File size - 80 mb<br>Excel opening time- 3s | Ods _all_ close; Ods csv file='~/big.csv'; Proc print data=verybig; Run; Ods _all_ close; | SAS/Base Only. | No style. SAS formatted values may not be transformed correctly to Excel. |
| ODS Excel | Crashes after 10K observations | Ods Excel file = "~/big.xlsx"; Proc print data=big; Run; Ods _all_ close | SAS/Base Only. Integrated into ODS | Limited to very small data sets |
| ODS XML | Create time-35 mins<br>File size-1.04 GB<br>Excel open time- 4.5 mins | Ods _all_ close; Ods xml tagset=excelxp file='~/big.xml'; Proc print data=verybig; Run; Ods _all_ close; | SAS/Base Only. Supports complex styling and multi sheets. | Very slow. Inefficient file format for big tables. Complex control over XLS cell formats |
| Proc Export | Create time: 55 Sec<br>File Size: 44 MB<br>Excel Open Time: 25 Sec | Proc export dbms=xlsx data=verybig file='~/big.xlsx'; Run; | Very fast. Compact file format. Automatic translation of SAS formats to Excel cell formats | No style attributes. SAS 9.3 and above. |

**Table 1. Comparison chart**

## REFERENCES


Bruin, J. "Command to compute new test" http://www.ats.ucla.edu/stat/stata/ado/analysis/. 2006

Nathan Clausen,DC Edmond Cheng, 2012 "Integrating SAS and Excel: an Overview and Comparison of Three Methods for Using SAS® to Create and Access Data in Excel"

BS Everitt ,2008 "A handbook of statistical analyses using **SAS"**


## RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *Output Delivery System: Basics*
- *Using ODS and the Macro facility for SAS®*

## CONTACT INFORMATION

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

Saurabh Nandy
SAS and OSU Data Mining Certified
MIS in Oklahoma State University
Phone: 405-762-3232
Email: saurabh.nandy@okstate.edu

Dr. Goutam Chakraborty
Ralph A. and Peggy A. Brenneman Professor of Marketing
Director, Master of Science in Business Analytics
Director, Graduate Certificate Program in Business Data Mining
Director, Graduate Certificate Program in Marketing Analytics
Phone: 405 744 7644
Web site: http://analytics.okstate.edu/
Linked In: www.linkedin.com/in/goutamc