

PROC REPORT: Tips and Customizations for Quickly Creating Customized Reports

Jonathan Duggins, NC State University
James Blum, UNCW

SESUG Paper 084-2021

Outline

- Begin with comparison to PRINT
- Start with fundamental REPORT syntax & effects
- Explore each of the six possible usages
- End with peek at more fully-fledged report

Simple PROC PRINT

Program 1

```
proc print data = SasHelp.Cars(obs = 5);  
  var Type MPG_City MPG_Highway;  
run;
```

Obs	Type	MPG_City	MPG_Highway
1	SUV	17	23
2	Sedan	24	31
3	Sedan	22	29
4	Sedan	20	28
5	Sedan	18	24

Modifying our PROC PRINT

Program 2

```
proc print data = SasHelp.Cars(obs = 5);  
  var ① MPG_City MPG_Highway;  
run;
```

Obs	MPG_City	MPG_Highway
1	17	23
2	24	31
3	22	29
4	20	28
5	18	24

Reproducing Print #1 with REPORT

Program 3

```
proc report data = SasHelp.Cars(obs = 5);1  
  column Type MPG_City MPG_Highway;2  
run;
```

Type	MPG (City)	MPG (Highway)
SUV	17	23
Sedan	24	31
Sedan	22	29
Sedan	20	28
Sedan	18	24

Reproducing Print #2 with REPORT

Program 4

```
proc report data = SasHelp.Cars(obs = 5);  
  columns ① MPG_City MPG_Highway;  
run;
```

MPG (City)	MPG (Highway)
101	135

Default Usages

Properties 1

DISPLAY:

- Each observation printed in its own row
- Default for character variables from data set

ANALYSIS:

- Calculates statistic for group of observations from data set.
 - Only ANALYSIS variables \Rightarrow group \equiv records in the report.
 - Default statistic is SUM.
- Default for numeric variables from data set.

Adding Explicit Usages

Program 5

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway;  
  define Type ① / display; ②  
  define MPG_City / display; ②  
  define MPG_Highway / display; ②  
run;
```

Type	MPG (City)	MPG (Highway)
SUV	17	23
Sedan	24	31
Sedan	22	29
Sedan	20	28
Sedan	18	24

Usage GPP: Include DEFINE with Explicit Usage

Program 6

```
proc report data = SasHelp.Cars(obs = 5);  
  column MPG_City MPG_Highway; ①  
  define MPG_City / display; ①  
  define MPG_Highway / display; ①  
run;
```

MPG (City)	MPG (Highway)
17	23
24	31
22	29
20	28
18	24

Aside: Building a Format

Program

```
proc format;①
  value mpg low - 20 = "Terrible" ②
           20 <- 35 = "OK"         ②
           35 <- 45 = "Good"       ②
           45 <- high = "Great"    ②
;
run;
```

- ① Create *mpg* format for MPG_City and MPG_Highway variables
- ② Designed so formatted order is not the same as internal order!

Additional Options: ID, FORMAT, Labels, NOPRINT

Program 7

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway;  
  define Type / display id; ①  
  define MPG_City / display format = mpg. ②  
                        'City Mileage' ③;  
  define MPG_Highway / display noprint ④;  
run;
```

Type	City Mileage
SUV	Terrible
Sedan	OK
Sedan	OK
Sedan	Terrible
Sedan	Terrible

Using ORDER with DESCENDING and FORMAT=

Program 8

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway;  
  define Type / order descending ❶;  
  define MPG_City / order  
    order = internal ❷  
    format = mpg. ❸;  
  define MPG_Highway / display;  
run;
```

Type	MPG (City)	MPG (Highway)
Sedan	Terrible	28
		24
	OK	31
		29
SUV	Terrible	23

Using ORDER= Options

Program 9

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway;  
  define Type / order;  
  define MPG_City / order format = mpg.;  
  define MPG_Highway / order format = mpg.;  
run;
```

Type	MPG (City)	MPG (Highway)
SUV	Terrible	OK
Sedan	OK	OK
	Terrible	OK

Intro to ACROSS

Program 10

```
proc report data = SasHelp.Cars;  
  column Type;  
  define Type / across ① order = freq ② descending ③ ;  
run;
```

Type					
Sedan	SUV	Sports	Wagon	Truck	Hybrid
262	60	49	30	24	3

Intro to ANALYSIS

Program 11

```
proc report data = SasHelp.Cars(obs = 5);  
  column MPG_City MPG_Highway;  
  define MPG_City / analysis ①;  
  define MPG_Highway / analysis ①;  
run;
```

MPG (City)	MPG (Highway)
101	135

Other ANALYSIS Statistics

Program 12

```
proc report data = SasHelp.Cars(obs = 5);  
  column MPG_City MPG_Highway;  
  define MPG_City / analysis mean ①;  
  define MPG_Highway / median ②;  
run;
```

MPG (City)	MPG (Highway)
20.2	28

Understanding Analysis Groups

Program 13

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway;  
  define Type / display ①;  
  define MPG_City / analysis ② mean;  
  define MPG_Highway / analysis ② var;  
run;
```

Type	MPG (City)	MPG (Highway)
SUV	17	.
Sedan	24	.
Sedan	22	.
Sedan	20	.
Sedan	18	.

Intro to GROUP Usage

Program 14

```
proc report data = SasHelp.Cars;  
  column Type MPG_City MPG_Highway;  
  define Type          / group ①  
                      descending ②  
                      order = freq ② ;  
  define MPG_City      / analysis mean  
                      format = 4.1  
                      'Mean City MPG';  
  define MPG_Highway  / n  
                      'Record Count';  
run;
```

Type	Mean City MPG	Record Count
Sedan	21.1	262
SUV	16.1	60
Sports	18.4	49
Wagon	21.1	30
Truck	16.5	24
Hybrid	55.0	3

Intro to COMPUTED Usage

Program 15

```
proc report data = SasHelp.Cars(obs = 5);  
  column Type MPG_City MPG_Highway MPG_Ratio; ①  
  define Type / display ;  
  define MPG_City / display;  
  define MPG_Highway / display;  
  define MPG_Ratio / computed ② 'City/Highway';  
  compute ③ MPG_Ratio ④;  
    MPG_Ratio = MPG_City/MPG_Highway; ⑤  
  endcomp; ⑥  
run;
```

Type	MPG (City)	MPG (Highway)	City Highway
SUV	17	23	0.7391304
Sedan	24	31	0.7741935
Sedan	22	29	0.7586207
Sedan	20	28	0.7142857
Sedan	18	24	0.75

COMPUTED Positioning

Program 16

```
proc report data = SasHelp.Cars(obs = 5)
    split = "*" ① ;
    column Type MPG_Ratio ② MPG_City MPG_Highway;
    define Type / display ;
    define MPG_City / display;
    define MPG_Highway / display;
    define MPG_Ratio / computed 'City/Highway';
    compute MPG_Ratio;
        MPG_Ratio = MPG_City/MPG_Highway;
    endcomp;
run;
```

Type	City/Highway	MPG (City)	MPG (Highway)
SUV	.	17	23
Sedan	.	24	31
Sedan	.	22	29
Sedan	.	20	28
Sedan	.	18	24

COMPUTED Positioning: Fixed

Program 17

```
proc report data = SasHelp.Cars(obs = 5)
    split = "*";
    column Type MPG_Ratio MPG_City MPG_Highway;
    define Type / display ;
    define MPG_City / display;
    define MPG_Highway / display;
    define MPG_Ratio / computed
        'City/Highway'
        format = 4.2;
    compute MPG_Highway; ❶
        MPG_Ratio = MPG_City/MPG_Highway;
    endcomp;
run;
```

Type	City/Highway	MPG (City)	MPG (Highway)
SUV	0.74	17	23
Sedan	0.77	24	31
Sedan	0.76	22	29
Sedan	0.71	20	28
Sedan	0.75	18	24

Summarizing Usages

Properties 2

ACROSS: New column for each formatted unique value of of the ACROSS variable.

ANALYSIS: Calculates a statistic for a group of observations from the data set.

COMPUTED: Column derived in a COMPUTE block in the PROC REPORT.

DISPLAY: Each observation of the variable is printed in its own row in the report.

GROUP: Collects observations with the same formatted value of the GROUP variable.

ORDER: Sequences the rows of the report based on sort criteria and direction.

Styling a Report on MSRP

Origin	DriveTrain	Count	%	Mean	Median
USA	All	22	15.0%	\$33,972	\$32,448
	Front	90	61.2%	\$25,095	\$23,115
	Rear	35	23.8%	\$33,301	\$30,835
		147	34.3%	\$28,377	\$25,520
Europe	All	36	29.3%	\$45,103	\$39,445
	Front	37	30.1%	\$34,980	\$34,845
	Rear	50	40.7%	\$60,581	\$52,243
		123	28.7%	\$48,350	\$40,590
Asia	All	34	21.5%	\$28,982	\$26,898
	Front	99	62.7%	\$20,687	\$19,560
	Rear	25	15.8%	\$35,028	\$31,045
		158	36.9%	\$24,741	\$23,033
		428		\$32,775	\$27,635

Mean and median differ by more than 15%

- 1 STYLE controls aesthetics like color, font, etc.
- 2 *Aliasing* allows for multiple statistics on MSRP
- 3 BREAK and RBREAK allow for summaries
- 4 CALL DEFINE allows for changes within a row/column
- 5 COMPUTE + CALL DEFINE allows for banded rows
- 6 But none of it works without correct usages!

Summary

- Usages at the heart of good REPORTing
- Many additional tools
- Additional code in paper, more advanced examples in tutorial
- Thank you and, time permitting, any questions?!