Paper 3582-2019

SuperFreq: Using the REPORT Procedure to Create Multi-Level Frequency Tables

Jedediah J Teres, MDRC

ABSTRACT

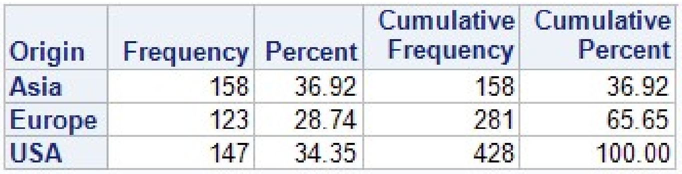
PROC FREQ is a powerful tool for summarizing categorical data. Cross-tabulating variables is a common application of PROC FREQ. Cumulative frequency and percents are calculated across the entire data PROC FREQ is summarizing. PROC FREQ supports BY group processing as well. Using BY group processing, cumulative frequencies and percents are calculated within each BY group. One limitation of PROC FREQ is that this is an either/or proposition-- cumulative frequencies and percents are calculated across the entire data set or within each BY group, but not both at once. Displaying multi-level cumulative frequencies in the same output table using PROC FREQ requires merging output data from multiple runs. An alternative approach that allows for displaying cumulative frequencies and percents from the full sample alongside cumulative frequencies and percents within each BY group uses PROC REPORT rather than PROC FREQ. This paper outlines an approach for creating such a summary table using PROC REPORT and highlights the advantages conferred through using this technique.

INTRODUCTION

Like many SAS programmers, I use FREQ procedure rather frequently[1] to check the distribution of categorical and discrete variables. However, while PROC FREQ is flexible and versatile, it does have limitations. For example, it does not show cumulative counts and percentages within subgroups—these measures are calculated only across the entire input dataset. In order to calculate cumulative counts and percentages for each subgroup within a sample, I needed to look elsewhere. Ultimately, I landed on the REPORT procedure. The techniques described in this paper require only familiarity with PROC REPORT rather than expertise. Readers will learn how to use PROC REPORT to create multi-level frequency tables that include cumulative counts and percentages within groups as well as for the entire sample. This approach can be applied in many situations across multiple industries. Truly, anyone who uses PROC FREQ can benefit. It's also a gentle introduction to PROC REPORT because of the familiar starting point of a standard frequency table. Experience using PROC FREQ is assumed as well.

THE DATA

Consider the SASHELP.CARS dataset. It contains pricing and performance data for a series of cars manufactured in the USA, Asia, and Europe. Using PROC FREQ, I can quickly get a count of the number of car models manufactured in each location as shown in Table 1.

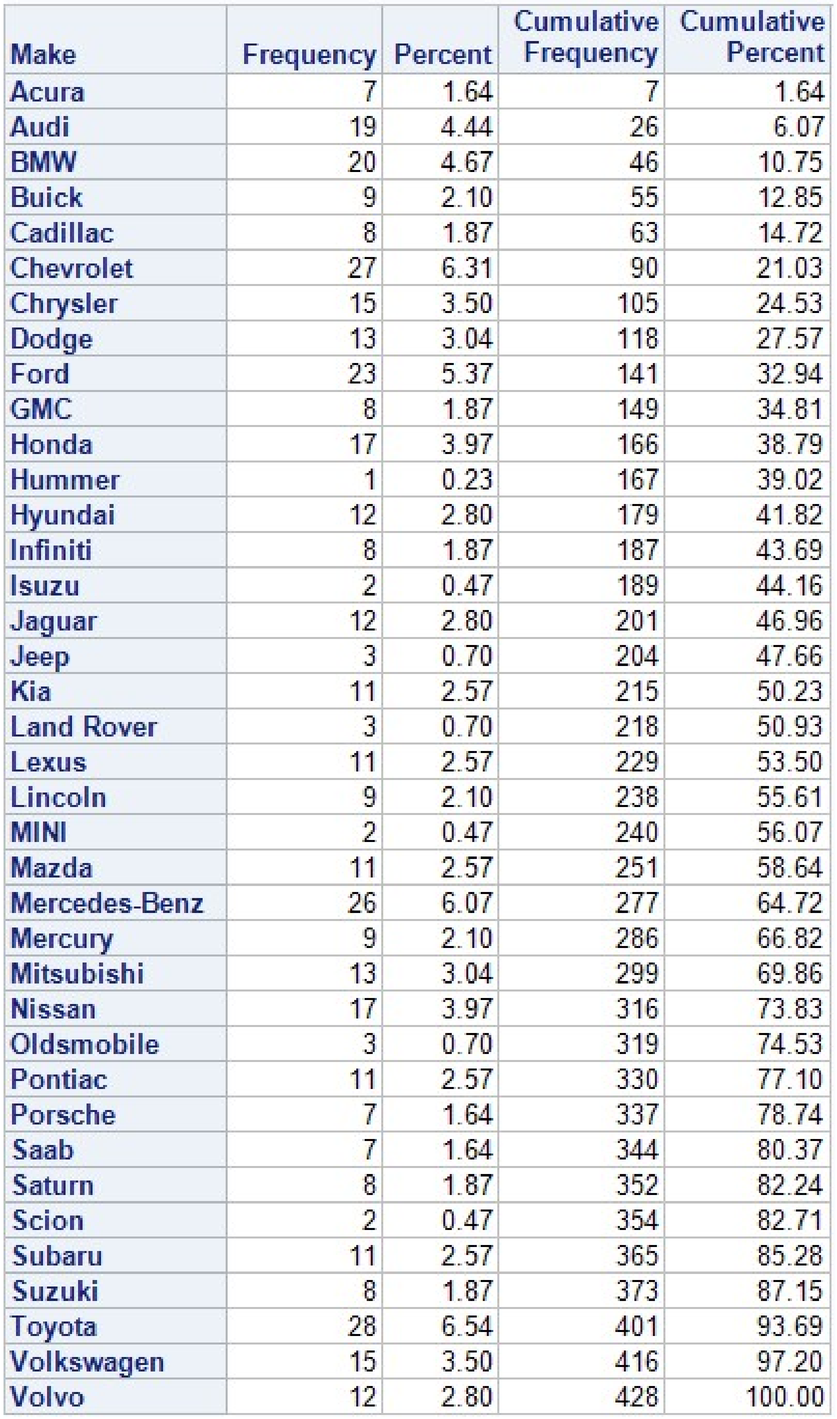


Origin	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Asia	158	36.92	158	36.92
Europe	123	28.74	281	65.65
USA	147	34.35	428	100.00

Table 1. Frequency of country of origin

[1] Pun intended

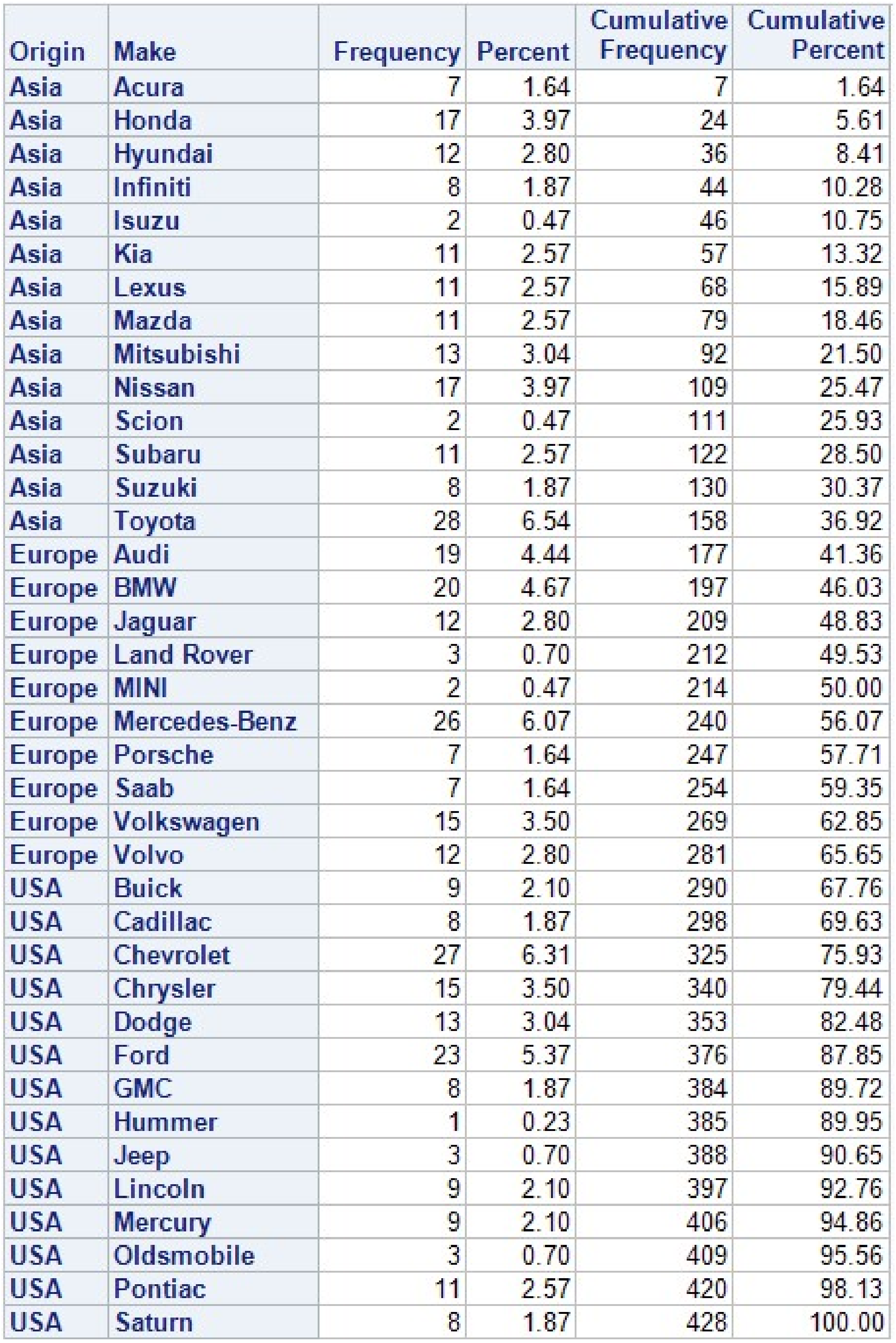
If the frequency of specific makes (i.e., brands) is of interest rather than the country of origin, you can create that table easily as shown in Table 2.



Make	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Acura	7	1.64	7	1.64
Audi	19	4.44	26	6.07
BMW	20	4.67	46	10.75
Buick	9	2.10	55	12.85
Cadillac	8	1.87	63	14.72
Chevrolet	27	6.31	90	21.03
Chrysler	15	3.50	105	24.53
Dodge	13	3.04	118	27.57
Ford	23	5.37	141	32.94
GMC	8	1.87	149	34.81
Honda	17	3.97	166	38.79
Hummer	1	0.23	167	39.02
Hyundai	12	2.80	179	41.82
Infiniti	8	1.87	187	43.69
Isuzu	2	0.47	189	44.16
Jaguar	12	2.80	201	46.96
Jeep	3	0.70	204	47.66
Kia	11	2.57	215	50.23
Land Rover	3	0.70	218	50.93
Lexus	11	2.57	229	53.50
Lincoln	9	2.10	238	55.61
MINI	2	0.47	240	56.07
Mazda	11	2.57	251	58.64
Mercedes-Benz	26	6.07	277	64.72
Mercury	9	2.10	286	66.82
Mitsubishi	13	3.04	299	69.86
Nissan	17	3.97	316	73.83
Oldsmobile	3	0.70	319	74.53
Pontiac	11	2.57	330	77.10
Porsche	7	1.64	337	78.74
Saab	7	1.64	344	80.37
Saturn	8	1.87	352	82.24
Scion	2	0.47	354	82.71
Subaru	11	2.57	365	85.28
Suzuki	8	1.87	373	87.15
Toyota	28	6.54	401	93.69
Volkswagen	15	3.50	416	97.20
Volvo	12	2.80	428	100.00

Table 2. Frequency of car makes on SASHELP.CARS

Using the LIST option, you can create a two-way table that essentially consists of the second table nested inside the first as show in Table 3.

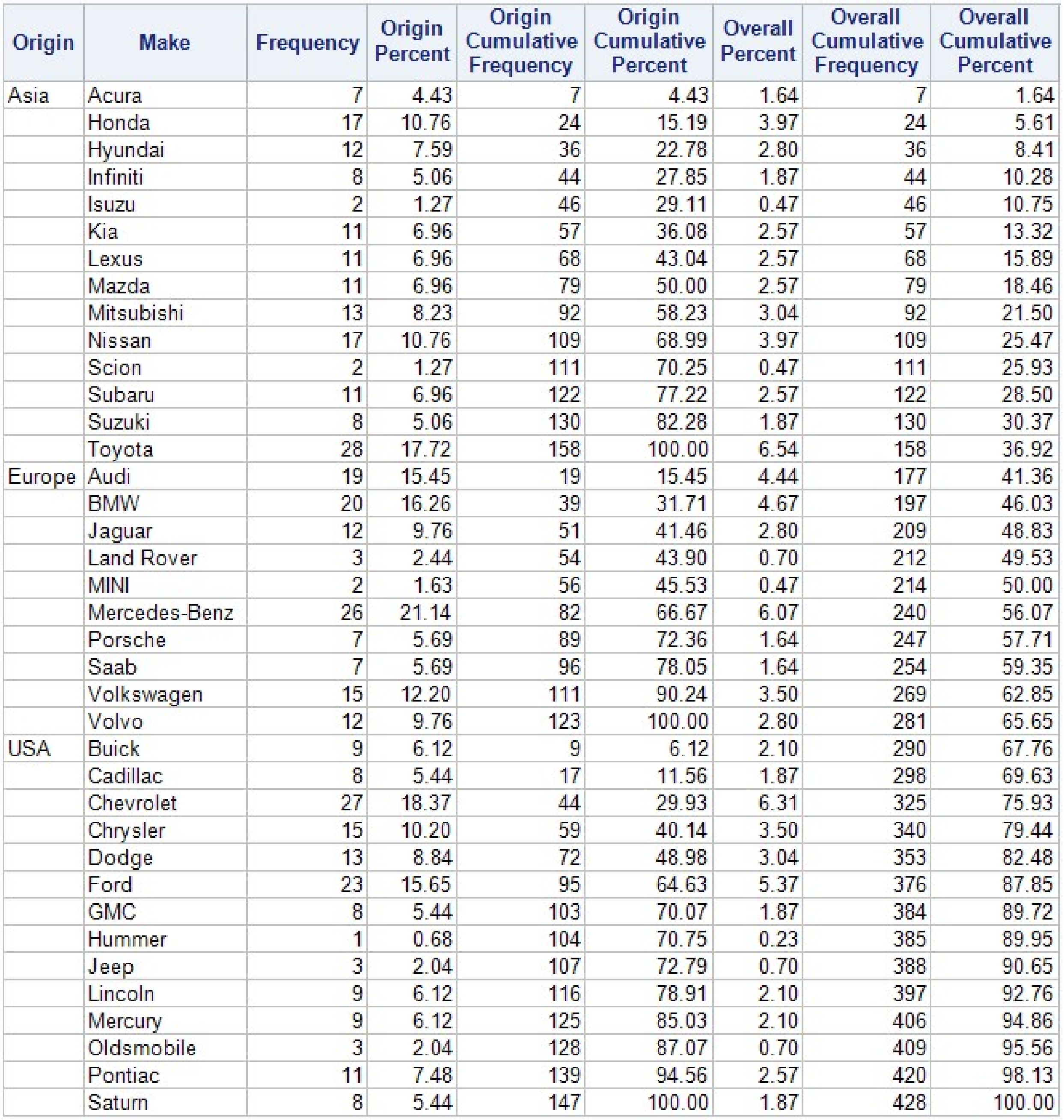


Origin	Make	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Asia	Acura	7	1.64	7	1.64
Asia	Honda	17	3.97	24	5.61
Asia	Hyundai	12	2.80	36	8.41
Asia	Infiniti	8	1.87	44	10.28
Asia	Isuzu	2	0.47	46	10.75
Asia	Kia	11	2.57	57	13.32
Asia	Lexus	11	2.57	68	15.89
Asia	Mazda	11	2.57	79	18.46
Asia	Mitsubishi	13	3.04	92	21.50
Asia	Nissan	17	3.97	109	25.47
Asia	Scion	2	0.47	111	25.93
Asia	Subaru	11	2.57	122	28.50
Asia	Suzuki	8	1.87	130	30.37
Asia	Toyota	28	6.54	158	36.92
Europe	Audi	19	4.44	177	41.36
Europe	BMW	20	4.67	197	46.03
Europe	Jaguar	12	2.80	209	48.83
Europe	Land Rover	3	0.70	212	49.53
Europe	MINI	2	0.47	214	50.00
Europe	Mercedes-Benz	26	6.07	240	56.07
Europe	Porsche	7	1.64	247	57.71
Europe	Saab	7	1.64	254	59.35
Europe	Volkswagen	15	3.50	269	62.85
Europe	Volvo	12	2.80	281	65.65
USA	Buick	9	2.10	290	67.76
USA	Cadillac	8	1.87	298	69.63
USA	Chevrolet	27	6.31	325	75.93
USA	Chrysler	15	3.50	340	79.44
USA	Dodge	13	3.04	353	82.48
USA	Ford	23	5.37	376	87.85
USA	GMC	8	1.87	384	89.72
USA	Hummer	1	0.23	385	89.95
USA	Jeep	3	0.70	388	90.65
USA	Lincoln	9	2.10	397	92.76
USA	Mercury	9	2.10	406	94.86
USA	Oldsmobile	3	0.70	409	95.56
USA	Pontiac	11	2.57	420	98.13
USA	Saturn	8	1.87	428	100.00

Table 3. Frequency of make by country of origin.

The counts and percentages associated with each specific make haven't changed—only the cumulative measures have changed because the makes are now shown within origin, so the order is different.

I have found myself in situations where it would helpful to know the percent within a group as well as across the entire sample. For example, Hondas represent 3.97 percent of all car makes, but what percent of Asian cars does that represent? According to the table of my dreams (Table 4), it’s 10.76 percent.



Origin	Make	Frequency	Origin Percent	Origin Cumulative Frequency	Origin Cumulative Percent	Overall Percent	Overall Cumulative Frequency	Overall Cumulative Percent
Asia	Acura	7	4.43	7	4.43	1.64	7	1.64
	Honda	17	10.76	24	15.19	3.97	24	5.61
	Hyundai	12	7.59	36	22.78	2.80	36	8.41
	Infiniti	8	5.06	44	27.85	1.87	44	10.28
	Isuzu	2	1.27	46	29.11	0.47	46	10.75
	Kia	11	6.96	57	36.08	2.57	57	13.32
	Lexus	11	6.96	68	43.04	2.57	68	15.89
	Mazda	11	6.96	79	50.00	2.57	79	18.46
	Mitsubishi	13	8.23	92	58.23	3.04	92	21.50
	Nissan	17	10.76	109	68.99	3.97	109	25.47
	Scion	2	1.27	111	70.25	0.47	111	25.93
	Subaru	11	6.96	122	77.22	2.57	122	28.50
	Suzuki	8	5.06	130	82.28	1.87	130	30.37
	Toyota	28	17.72	158	100.00	6.54	158	36.92
Europe	Audi	19	15.45	19	15.45	4.44	177	41.36
	BMW	20	16.26	39	31.71	4.67	197	46.03
	Jaguar	12	9.76	51	41.46	2.80	209	48.83
	Land Rover	3	2.44	54	43.90	0.70	212	49.53
	MINI	2	1.63	56	45.53	0.47	214	50.00
	Mercedes-Benz	26	21.14	82	66.67	6.07	240	56.07
	Porsche	7	5.69	89	72.36	1.64	247	57.71
	Saab	7	5.69	96	78.05	1.64	254	59.35
	Volkswagen	15	12.20	111	90.24	3.50	269	62.85
	Volvo	12	9.76	123	100.00	2.80	281	65.65
USA	Buick	9	6.12	9	6.12	2.10	290	67.76
	Cadillac	8	5.44	17	11.56	1.87	298	69.63
	Chevrolet	27	18.37	44	29.93	6.31	325	75.93
	Chrysler	15	10.20	59	40.14	3.50	340	79.44
	Dodge	13	8.84	72	48.98	3.04	353	82.48
	Ford	23	15.65	95	64.63	5.37	376	87.85
	GMC	8	5.44	103	70.07	1.87	384	89.72
	Hummer	1	0.68	104	70.75	0.23	385	89.95
	Jeep	3	2.04	107	72.79	0.70	388	90.65
	Lincoln	9	6.12	116	78.91	2.10	397	92.76
	Mercury	9	6.12	125	85.03	2.10	406	94.86
	Oldsmobile	3	2.04	128	87.07	0.70	409	95.56
	Pontiac	11	7.48	139	94.56	2.57	420	98.13
	Saturn	8	5.44	147	100.00	1.87	428	100.00

Table 4. Frequency Table Augmented with Subgroup Counts and Percentages

Clearly this table was not created using PROC FREQ. Rather, it’s output from PROC REPORT. The approach I took builds on techniques described by David D. Chapman in his paper “Using PROC REPORT to Produce Tables with Cumulative Totals and Row Differences.”

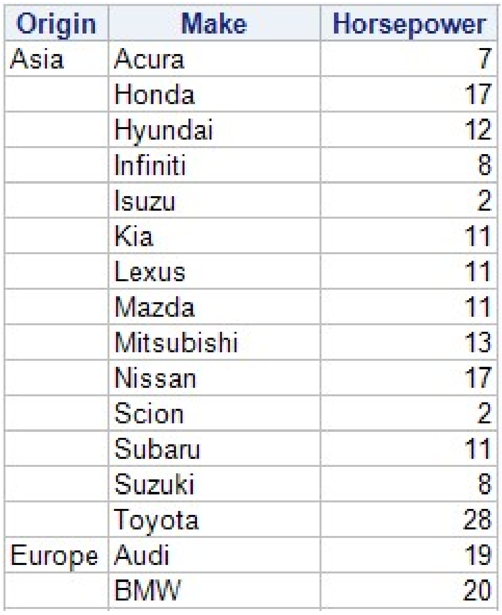
It is technically possible to get output resembling this from PROC FREQ, but it requires merging output data from different PROC FREQs. The SQL and TABULATE procedures can get close, but only PROC REPORT can calculate the cumulative measures I wanted to see.

Because I am not a PROC REPORT power-user, I wanted to first replicate the PROC FREQ output table using PROC REPORT to better understand the PROC REPORT syntax. Many papers have been written about PROC REPORT; this paper presupposes only basic familiarity with PROC REPORT syntax. Check the "recommended reading" list for some guides.

Some basic PROC REPORT code follows. PROC REPORT will generate the counts using a numeric ANALYSIS variable. In this case, I selected Horsepower, which I renamed "Frequency" in the output to avoid confusion:

```
proc report data = sashelp.cars nowd headline ;
     column     origin         /* crosstab var 1 */
                make           /* crosstab var 2 */
                Horsepower     /*   REPORT needs a
                                    numeric variable
                                    to use N */ ;

     ** define all the vars ;
     ** ORIGIN and MAKE are the variables we're grouping by ;
     define     origin         / group ;
     define     make           / group ;
     ** Horsepower is really just used for the N statistic. ;
     ** to do that, it must be declared an ANALYSIS column ;
     define     Horsepower     / analysis  n  'Frequency' ;
run ;
```

The result is familiar, though incomplete as shown in Table 5.



Origin	Make	Horsepower
Asia	Acura	7
	Honda	17
	Hyundai	12
	Infiniti	8
	Isuzu	2
	Kia	11
	Lexus	11
	Mazda	11
	Mitsubishi	13
	Nissan	17
	Scion	2
	Subaru	11
	Suzuki	8
	Toyota	28
Europe	Audi	19
	BMW	20

Table 5. Basic results from PROC REPORT (partial results only).

Notice that unlike PROC FREQ results, the values of Origin are not repeated across rows. PROC REPORT gave us the counts for "free," but in order to get the percentages, we will need to use COMPUTED variables. We want 3 additional variables in the output—percent, cumulative frequency, and cumulative percent. However, the percentage calculate requires a grand total variable (Total), which is not shown in the final output.

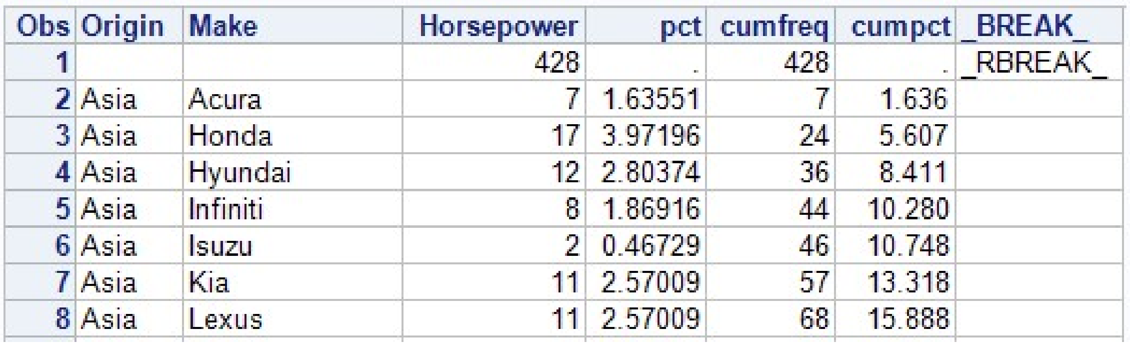
Each computed column has its own COMPUTE block. The grand total is calculated *before* any break points (i.e., new values of ORIGIN), which makes the sum available to each row of data processed by PROC REPORT. The cumulative total is initialized to 0 at the same time. Note that the cumulative total is incremented in the same compute block as the cumulative frequency.

The percentages shown in PROC FREQ are expressed as rational numbers ranging from 1 to 100 rather than 0 to 1. I have preserved that scale here.

Because cumtot is not included as a column in the REPORT output, it can't have its own COMPUTE block. The value of cumtot is used only to increment cumfreq, and so it made the most sense to include both calculations in the same block as shown below:

```
proc report data = sashelp.cars nowd headline out = cars_report_out ;
     column    origin         /* crosstab var 1 */
               make           /* crosstab var 2 */
               Horsepower     /* REPORT needs a
                                 numeric variable to use N */
               pct            /* percent */
               cumfreq        /* cumulative frequency */
               cumpct ;       /* cumulative percent */

     ** define all the var types ;
     ** ORIGIN and MAKE are the variables we're grouping by ;
     define    origin         / group ;
     define    make           / group ;
     ** Horsepower is really just used for count (N) ;
     ** to do that, it must be declared an ANALYSIS column ;
     define    Horsepower     /    analysis
                                   format = comma12. n 'Frequency'   ;

     ** the rest of the columns are computed ;
     ** this column is the percent across all
observations/combinations ;
     define    pct            /    computed
                                   format = 8.2
                                   'Percent'   ;
     ** this column is the cumulative frequency across all
combinations ;
     define    cumfreq        /    computed
                                   format = comma12.
                                   'Cumulative/Frequency'
                                   width = 10 ;
     ** this column is the cumulative percent across all combinations
;
     define    cumpct         /    computed
                                   format = 8.2
                                   'Cumulative/Percent'
                                   width = 10   ;

     ** PROC REPORT requires that the columns be listed in order on
the COLUMN statement ;
     ** it makes sense to keep them in order in the DEFINE block ;
     ** for these COMPUTE blocks, I'm ordering them in more of a
logical order for processing ;

     ** before anything happens, initialize total as the sum of
Horsepower.n and cumtot to 0 ;
     compute before ;
          total = Horsepower.n ;
          cumtot = 0 ;
     endcomp ;

     ** compute the percent of all combination (uses total) ;
     compute pct ;
          pct = 100 * (Horsepower.n / total) ;
     endcomp ;

     ** create global cumulative frequency ;
     compute cumfreq ;
          cumtot + Horsepower.n ;
          cumfreq = cumtot ;
     endcomp ;

     ** again, this is a relatively straightforward calculation ;
     compute cumpct ;
          cumpct = 100 * (cumfreq / total) ;
     endcomp ;
run ;
```

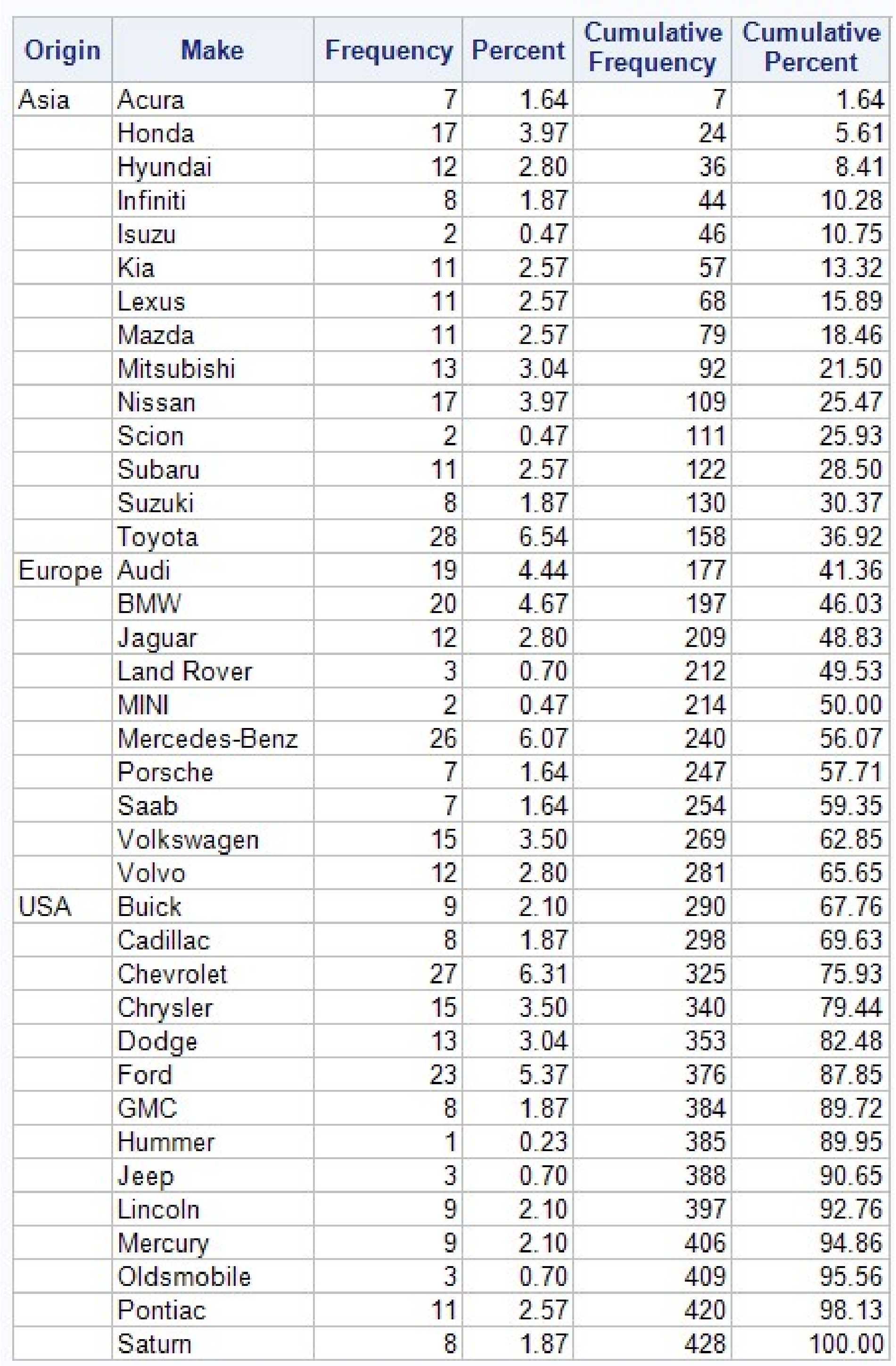
To better understand the implications of COMPUTE BEFORE, it is helpful to examine the dataset that PROC REPORT is building behind the scenes. Like many other PROCs, using the OUT = [dataset] option in the data will generate an output dataset. Note the value of Horsepower. When Origin and Make are missing, it is equal to 428. The value _RBREAK_ in the BREAK column lets us know that this is the result of the COMPUTE BEFORE calculation. The idea that the sum of a variable would be available -before- the data are processed is counterintuitive, but it's part of what makes PROC REPORT so powerful. Output 1 shows the dataset created by PROC REPORT.



Obs	Origin	Make	Horsepower	pct	cumfreq	cumpct	_BREAK_
1			428	.	428	.	_RBREAK_
2	Asia	Acura	7	1.63551	7	1.636	
3	Asia	Honda	17	3.97196	24	5.607	
4	Asia	Hyundai	12	2.80374	36	8.411	
5	Asia	Infiniti	8	1.86916	44	10.280	
6	Asia	Isuzu	2	0.46729	46	10.748	
7	Asia	Kia	11	2.57009	57	13.318	
8	Asia	Lexus	11	2.57009	68	15.888	

Output 1. PROC REPORT output dataset (partial).

At this point, I have essentially re-created the PROC FREQ output (with minor cosmetic differences) as shown in Table 6.



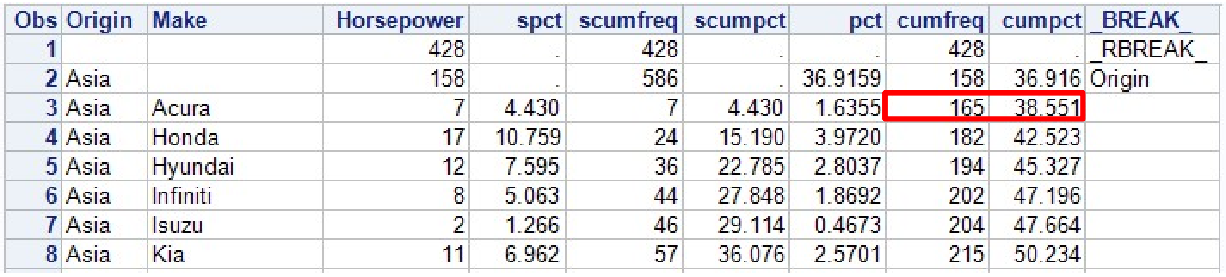
Origin	Make	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Asia	Acura	7	1.64	7	1.64
	Honda	17	3.97	24	5.61
	Hyundai	12	2.80	36	8.41
	Infiniti	8	1.87	44	10.28
	Isuzu	2	0.47	46	10.75
	Kia	11	2.57	57	13.32
	Lexus	11	2.57	68	15.89
	Mazda	11	2.57	79	18.46
	Mitsubishi	13	3.04	92	21.50
	Nissan	17	3.97	109	25.47
	Scion	2	0.47	111	25.93
	Subaru	11	2.57	122	28.50
	Suzuki	8	1.87	130	30.37
	Toyota	28	6.54	158	36.92
Europe	Audi	19	4.44	177	41.36
	BMW	20	4.67	197	46.03
	Jaguar	12	2.80	209	48.83
	Land Rover	3	0.70	212	49.53
	MINI	2	0.47	214	50.00
	Mercedes-Benz	26	6.07	240	56.07
	Porsche	7	1.64	247	57.71
	Saab	7	1.64	254	59.35
	Volkswagen	15	3.50	269	62.85
	Volvo	12	2.80	281	65.65
USA	Buick	9	2.10	290	67.76
	Cadillac	8	1.87	298	69.63
	Chevrolet	27	6.31	325	75.93
	Chrysler	15	3.50	340	79.44
	Dodge	13	3.04	353	82.48
	Ford	23	5.37	376	87.85
	GMC	8	1.87	384	89.72
	Hummer	1	0.23	385	89.95
	Jeep	3	0.70	388	90.65
	Lincoln	9	2.10	397	92.76
	Mercury	9	2.10	406	94.86
	Oldsmobile	3	0.70	409	95.56
	Pontiac	11	2.57	420	98.13
	Saturn	8	1.87	428	100.00

Table 6. PROC FREQ-like output created with PROC REPORT.

Calculating the percent, cumulative frequency, and cumulative percent within each group requires some careful consideration. As before, each column must be prespecified—note that the columns spct, scumfreq, and scumpct have been added (the "s" prefix stands for "subgroup"). In addition to calculating a grand total before any rows of data have been processed, I am now calculating a subgroup total (stotal) before each new value of Origin. This value will be used to calculate "local" percentages within each subgroup. The code used to calculate percent and cumulative percent within subgroup parallels the code calculating the percent and cumulative percent columns for the full sample. One crucial difference concerns the cumulative frequency for the overall sample. It is necessary to increment the cumulative count only when there is a non-missing value for Make, for reasons explained below. First the revised code:

```
proc report    data = sashelp.cars nowd
               out  = car_report ;
     column    origin
               make
               Horsepower
               spct
               scumfreq
               scumpct
               pct
               cumfreq
               cumpct ;

     define    origin         /    group ;
     define    make           /    group ;

     define    Horsepower     /    'Frequency'
                                   analysis n
                                   format = comma12. ;

     define    spct           /    'Origin/Percent'
                                   computed
                                   format = 8.2 ;
     define    scumfreq       /    'Origin/Cumulative/Frequency'
                                   computed
                                   format = comma12.0 ;
     define    scumpct        /    'Origin/Cumulative/Percent'
                                   computed
                                   format =  8.2 ;

     define    pct            /    'Overall/Percent'
                                   computed
                                   format = 8.2 ;
     define    cumfreq        /    'Overall/Cumulative/Frequency'
                                   computed
                                   format = comma12. ;
     define    cumpct         /    'Overall/Cumulative/Percent'
                                   computed
                                   format = 8.2 ;

     compute before ;
          total = Horsepower.n ;
          cumtot = 0 ;
     endcomp ;

     compute before origin ;
          stotal = Horsepower.n ;
          scumtot = 0 ;
     endcomp ;

     compute spct ;
          spct = 100 * (Horsepower.n / stotal) ;
     endcomp ;

     compute scumfreq ;
          scumtot + Horsepower.n  ;
          scumfreq = scumtot ;
     endcomp ;

     compute pct ;
          pct = 100 * (Horsepower.n / total) ;
     endcomp ;

     compute scumpct ;
          scumpct = 100 * (scumfreq / stotal) ;
     endcomp ;

     compute cumfreq ;
          if not(missing(make)) then cumtot + Horsepower.n ;
          cumfreq = cumtot ;
     endcomp ;

     compute cumpct ;
          cumpct = 100 * (cumfreq / total) ;
     endcomp ;
run ;
```

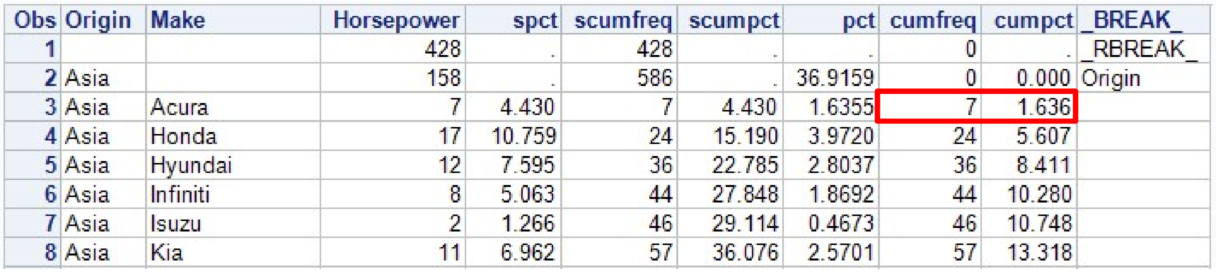
Again, it is helpful to examine the dataset PROC REPORT creates behind the scenes. When cumfreq is calculated at every step, the counts get inflated. Note the highlighted values—they are carrying over the values from the preceding row, thereby inflating the measures. Because this is the first instance of a Make within the first value of Origin, the cumulative values should equal the percent and frequency. That is clearly not the case, as shown in Output 2.



Obs	Origin	Make	Horsepower	spct	scumfreq	scumpct	pct	cumfreq	cumpct	_BREAK_
1			428	.	428	.	.	428	.	_RBREAK_
2	Asia		158	.	586	.	36.9159	158	36.916	Origin
3	Asia	Acura	7	4.430	7	4.430	1.6355	165	38.551	
4	Asia	Honda	17	10.759	24	15.190	3.9720	182	42.523	
5	Asia	Hyundai	12	7.595	36	22.785	2.8037	194	45.327	
6	Asia	Infiniti	8	5.063	44	27.848	1.8692	202	47.196	
7	Asia	Isuzu	2	1.266	46	29.114	0.4673	204	47.664	
8	Asia	Kia	11	6.962	57	36.076	2.5701	215	50.234	

Output 2. PROC REPORT dataset without IF Logic (partial).

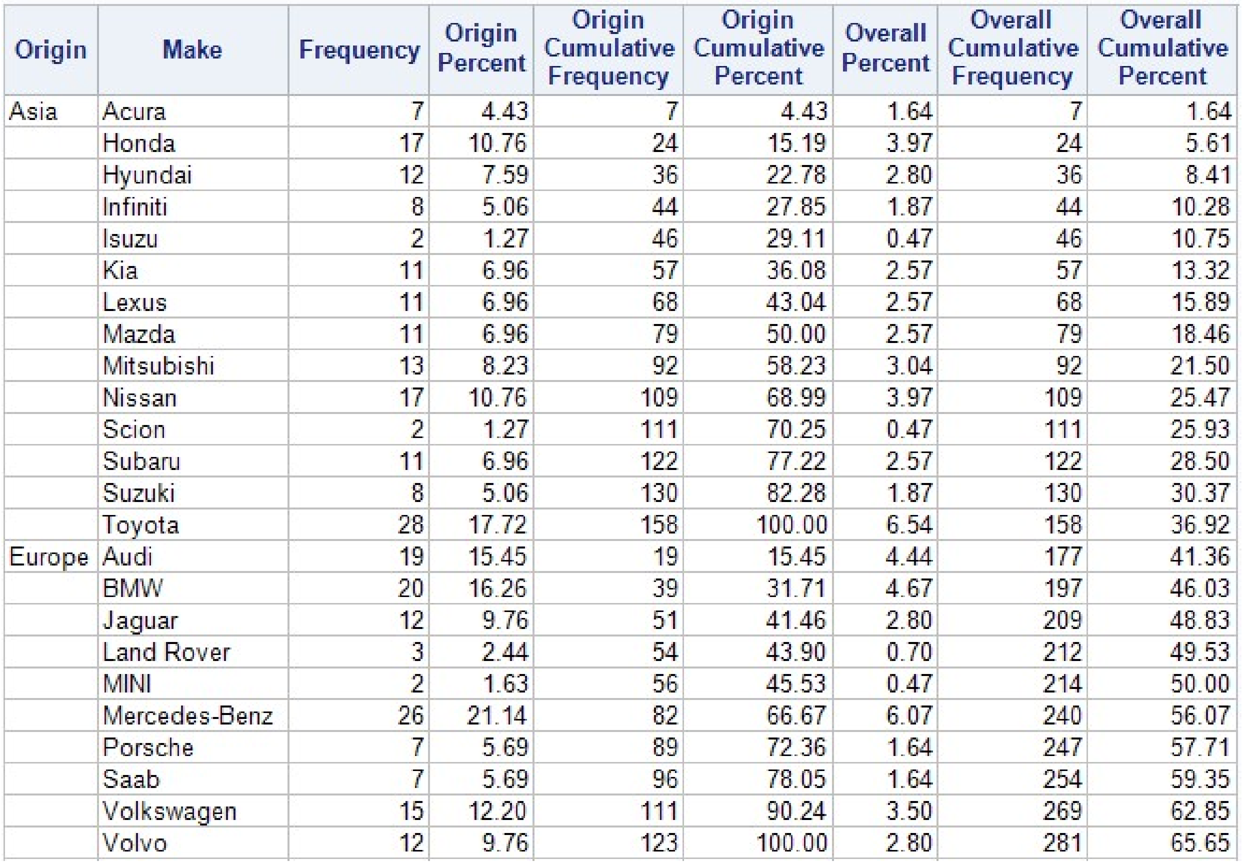
The issue is remedied by including the IF condition `if not(missing(make))` before calculating Cumulative Frequency (cumfreq). Since Cumulative Percent (cumpct) is derived from the Cumulative Frequency, its values are corrected without changing the syntax as shown in Output 3.



Obs	Origin	Make	Horsepower	spct	scumfreq	scumpct	pct	cumfreq	cumpct	_BREAK_
1			428	.	428	.	.	0	.	_RBREAK_
2	Asia		158	.	586	.	36.9159	0	0.000	Origin
3	Asia	Acura	7	4.430	7	4.430	1.6355	7	1.636	
4	Asia	Honda	17	10.759	24	15.190	3.9720	24	5.607	
5	Asia	Hyundai	12	7.595	36	22.785	2.8037	36	8.411	
6	Asia	Infiniti	8	5.063	44	27.848	1.8692	44	10.280	
7	Asia	Isuzu	2	1.266	46	29.114	0.4673	46	10.748	
8	Asia	Kia	11	6.962	57	36.076	2.5701	57	13.318	

Output 3. PROC REPORT dataset with IF Logic (partial).

At last, we have written the code required to create the result shown in Table 4, the table of my dreams. It is re-presented here in part as Table 7.



Origin	Make	Frequency	Origin Percent	Origin Cumulative Frequency	Origin Cumulative Percent	Overall Percent	Overall Cumulative Frequency	Overall Cumulative Percent
Asia	Acura	7	4.43	7	4.43	1.64	7	1.64
	Honda	17	10.76	24	15.19	3.97	24	5.61
	Hyundai	12	7.59	36	22.78	2.80	36	8.41
	Infiniti	8	5.06	44	27.85	1.87	44	10.28
	Isuzu	2	1.27	46	29.11	0.47	46	10.75
	Kia	11	6.96	57	36.08	2.57	57	13.32
	Lexus	11	6.96	68	43.04	2.57	68	15.89
	Mazda	11	6.96	79	50.00	2.57	79	18.46
	Mitsubishi	13	8.23	92	58.23	3.04	92	21.50
	Nissan	17	10.76	109	68.99	3.97	109	25.47
	Scion	2	1.27	111	70.25	0.47	111	25.93
	Subaru	11	6.96	122	77.22	2.57	122	28.50
	Suzuki	8	5.06	130	82.28	1.87	130	30.37
	Toyota	28	17.72	158	100.00	6.54	158	36.92
Europe	Audi	19	15.45	19	15.45	4.44	177	41.36
	BMW	20	16.26	39	31.71	4.67	197	46.03
	Jaguar	12	9.76	51	41.46	2.80	209	48.83
	Land Rover	3	2.44	54	43.90	0.70	212	49.53
	MINI	2	1.63	56	45.53	0.47	214	50.00
	Mercedes-Benz	26	21.14	82	66.67	6.07	240	56.07
	Porsche	7	5.69	89	72.36	1.64	247	57.71
	Saab	7	5.69	96	78.05	1.64	254	59.35
	Volkswagen	15	12.20	111	90.24	3.50	269	62.85
	Volvo	12	9.76	123	100.00	2.80	281	65.65

Table 7. The Final Result (partial).

CONCLUSION

Starting with a basic frequency table, it is straightforward to make the leap from using PROC FREQ to PROC REPORT. The PROC REPORT code to add cumulative counts and percentages within subgroups is a lighter lift if the basic structure of the frequency output table is in place. This approach can be extended to three (or more) grouping variables. With each additional grouping variable, the output gets denser and potentially harder to read and interpret. Appendix A contains code illustrating the use of three grouping variables and Appendix B contains sample macro code for a flexible implementation of creating frequency tables using PROC REPORT.

REFERENCES

Chapman, David D. 2002. "Using PROC REPORT to Produce Tables With Cumulative Totals and Row Differences." SUGI 27 Proceedings. Orlando, Florida: SAS Users Group International. Available at https://support.sas.com/resources/papers/proceedings/proceedings/sugi27/p120-27.pdf

RECOMMENDED READING

- *A Step-by-Step Introduction to PROC REPORT*
  https://support.sas.com/resources/papers/proceedings/pdfs/sgf2008/079-2008.pdf
- *So You're Still Not Using PROC REPORT. Why Not?*

  https://support.sas.com/resources/papers/proceedings/proceedings/sugi23/Handson/p122.pdf
- *Advanced PROC REPORT:Doing More in the Compute Block*

  https://www.lexjansen.com/pharmasug/2007/tu/TU02.pdf

CONTACT INFORMATION

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

Jedediah Teres
MDRC
(212) 340-4553
jed.teres@mdrc.org

APPENDIX A

```
proc report data = sashelp.cars nowd out = car_report headline ;
     column    origin
               drivetrain
               make
               Horsepower

               g2pct
               g2cumfreq
               g2cumpct

               g1pct
               g1cumfreq
               g1cumpct

               pct
               cumfreq
               cumpct ;

     define    origin         /    group ;
     define    drivetrain     /    group ;
     define    make           /    group ;
     define    Horsepower     /    analysis  format = comma12.   n
                                   'Frequency'  ;
     define    g2pct          /    computed  format = percent9.2
                                   'Drive Train/Percent' ;
     define    g2cumfreq      /    computed  format = comma12.0
                                   'Drive Train/Cumulative/Frequency'
;
     define    g2cumpct       /    computed  format =  percent9.2
                                   'Drive Train/Cumulative/Percent' ;
     define    g1pct          /    computed  format = percent9.2
                                   'Origin/Percent' ;
     define    g1cumfreq      /    computed  format = comma12.0
                                   'Origin/Cumulative/Frequency' ;
     define    g1cumpct       /    computed  format =  percent9.2
                                   'Origin/Cumulative/Percent' ;
     define    pct            /    computed  format = percent9.2
                                   'Overall/Percent'  ;
     define    cumfreq        /    computed  format = comma12.
                                   'Overall/Cumulative/Frequency' ;
     define    cumpct         /    computed  format = percent9.2
                                   'Overall/Cumulative/Percent' ;

     compute before ;
          total = Horsepower.n ;
          cumtot = 0 ;
     endcomp ;

     compute before origin ;
          g1total = Horsepower.n ;
          g1cumtot = 0 ;
     endcomp ;

     compute before drivetrain ;
          g2total = Horsepower.n ;
          g2cumtot = 0 ;
     endcomp ;

     compute g1pct ;
          g1pct = divide(Horsepower.n, g1total) ;
     endcomp ;

     compute g1cumfreq ;
          if not(missing(make)) then g1cumtot + Horsepower.n  ;
          g1cumfreq = g1cumtot ;
     endcomp ;

     compute g1cumpct ;
          g1cumpct = divide(g1cumfreq, g1total) ;
     endcomp ;

     compute g2pct ;
          g2pct = divide(Horsepower.n, g2total) ;
     endcomp ;

     compute g2cumfreq ;
          g2cumtot + Horsepower.n  ;
          g2cumfreq = g2cumtot ;
     endcomp ;

     compute g2cumpct ;
          g2cumpct = divide(g2cumfreq, g2total) ;
     endcomp ;

     compute pct ;
          pct = divide(Horsepower.n,  total) ;
     endcomp ;

     compute cumfreq ;
          if not(missing(make)) then cumtot + Horsepower.n ;
          cumfreq = cumtot ;
     endcomp ;

     compute cumpct ;
          cumpct = divide(cumfreq, total) ;
     endcomp ;
run ;
```

APPENDIX B

```
%macro SuperFreq    (inputDS        = /* Input Data Set */
                    ,GroupVar1      = /* Outer Category */
                    ,GroupVar2      = /* Inner Category */
                    ,NumVar         = /* Numeric Variable required for
report */
                    ) ;

     proc report data = &inputDS out = checkout nowd headline ;
          column    &GroupVar1     /* crosstab var 1 */
                    &GroupVar2     /* crosstab var 2 */
                    &NumVar        /* REPORT needs a numeric variable
to use N-- this was sort of arbitrary */
                                   /* it's essentially count within
group */
                    g1pct          /* percent within group */
                    g1cumfreq      /* cumulative count within group */
                    g1cumpct       /* cumulative percent within group
*/
                    pct            /* global percent */
                    cumfreq        /* global cumulative frequency */
                    cumpct ;       /* global cumulative percent */

          ** define all the vars ;
          ** thanks, PROC REPORT ;
          ** &GroupVar1 and &GroupVar2 are the variables we are
grouping by ;
          define    &GroupVar1  / group ;
          define    &GroupVar2  / group ;
          ** &NumVar is really just used for the N statistic. to do
that, it must be declared an ANALYSIS column ;
          define    &NumVar     / analysis     format = comma12. n
'Frequency'  ;
          ** the rest of the columns are computed ;

          ** to be calculated within the groups ;
          ** this is the percent column ;
          define    g1pct          / computed     format = percent9.2
"&GroupVar1/Percent"  ;
          ** this is the cumulative frequency column ;
          define    g1cumfreq      / computed     format = comma12.0
"&GroupVar1/Cumulative/Frequency" width = 10 ;
          ** this is the cumulative percent column ;
          define    g1cumpct       / computed     format =  percent9.2
"&GroupVar1/Cumulative/Percent" width = 10  ;

          ** these are for global/grand totals ;
          ** this column is the percent across all
observations/combinations ;
          define    pct            / computed     format = percent9.2
'Overall/Percent'  ;
```

```
          ** this column is the cumulative frequency across all
combinations ;
          define    cumfreq        / computed     format = comma12.
'Overall/Cumulative/Frequency' width = 10 ;
          ** this column is the cumulative percent across all
combinations ;
          define    cumpct         / computed     format = percent9.2
'Overall/Cumulative/Percent' width = 10  ;

          ** PROC REPORT requires that the columns be listed in order
on the COLUMN statement ;
          ** it makes sense to keep them in order in the DEFINE block
;
          ** for these COMPUTE blocks, they are ordered them in more
of a logical order for processing ;

          ** before anything happens, initialize gtotal as the sum of
&NumVar.n and gcumtot to 0 ;
          compute before ;
               total = &NumVar..n ;
               cumtot = 0 ;
          endcomp ;

          ** before each new &GroupVar1 value, re-initialize total and
cumtot ;
          compute before &GroupVar1 ;
               g1total = &NumVar..n ;
               g1cumtot = 0 ;
          endcomp ;

          ** pct is the count (&NumVar.n) over the total, very
straight forward ;
          compute g1pct ;
               g1pct = divide(&NumVar..n, g1total) ;
          endcomp ;

          ** the cumulative frequency is the running total ;
          ** reset it at the breaks ;
          compute g1cumfreq ;
               g1cumtot + &NumVar..n  ;
               g1cumfreq = g1cumtot ;
          endcomp ;


          ** compute the percent of all combination (uses gtotal) ;
          compute pct ;
               pct = divide(&NumVar..n, total) ;
          endcomp ;

          ** create running cumulative percent ;
          compute g1cumpct ;
               g1cumpct = divide(g1cumfreq, g1total) ;
          endcomp ;

          /*   create global cumulative frequency
               it's similar to the "local" counts but...
               in order to get this to work, i added the IF condition
               it makes more sense if you look at the output report
data, but some totals are computed BEFORE
               and obviously I didn't want that to happen because it
was inflating my counts */
          compute cumfreq ;
               if not(missing(&GroupVar2)) then cumtot + &NumVar..n ;
               cumfreq = cumtot ;
          endcomp ;

          ** again, this is a relatively straightforward calculation ;
          compute cumpct ;
               cumpct = divide(cumfreq, total) ;
          endcomp ;

          /*   this is required
               you have to choose SOMETHING for proc report to do here
               you can opt to have it SUPPRESS any subgroup summaries
               i kept the underline (UL) because it only shows up in
the listing output and makes it a little more readable
               one drawback is the goes across the entire report
field, which can probably be fixed but that's icing */
          break after &GroupVar1 / ul ;
     run ;
%mend ;
```