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 set 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 percents 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.

<table>
<thead>
<tr>
<th>Origin</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>158</td>
<td>36.92</td>
<td>158</td>
<td>36.92</td>
</tr>
<tr>
<td>Europe</td>
<td>123</td>
<td>28.74</td>
<td>281</td>
<td>65.65</td>
</tr>
<tr>
<td>USA</td>
<td>147</td>
<td>34.35</td>
<td>428</td>
<td>100.00</td>
</tr>
</tbody>
</table>

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.

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

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.
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.

<table>
<thead>
<tr>
<th>Origin</th>
<th>Make</th>
<th>Frequency</th>
<th>Origin Percent</th>
<th>Origin Cumulative Frequency</th>
<th>Origin Cumulative Percent</th>
<th>Overall Percent</th>
<th>Overall Cumulative Frequency</th>
<th>Overall Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>Acura</td>
<td>7</td>
<td>4.43</td>
<td>7</td>
<td>4.43</td>
<td>1.64</td>
<td>7</td>
<td>1.64</td>
</tr>
<tr>
<td>Honda</td>
<td>17</td>
<td>10.76</td>
<td>24</td>
<td>15.19</td>
<td>3.57</td>
<td>24</td>
<td>5.61</td>
<td></td>
</tr>
<tr>
<td>Hyundai</td>
<td>12</td>
<td>7.59</td>
<td>36</td>
<td>22.78</td>
<td>2.80</td>
<td>36</td>
<td>8.41</td>
<td></td>
</tr>
<tr>
<td>Infiniti</td>
<td>8</td>
<td>5.06</td>
<td>44</td>
<td>27.85</td>
<td>1.87</td>
<td>44</td>
<td>10.28</td>
<td></td>
</tr>
<tr>
<td>Isuzu</td>
<td>2</td>
<td>1.27</td>
<td>46</td>
<td>29.11</td>
<td>0.47</td>
<td>46</td>
<td>10.75</td>
<td></td>
</tr>
<tr>
<td>Kia</td>
<td>11</td>
<td>6.96</td>
<td>57</td>
<td>36.08</td>
<td>2.57</td>
<td>57</td>
<td>13.32</td>
<td></td>
</tr>
<tr>
<td>Lexus</td>
<td>11</td>
<td>6.96</td>
<td>60</td>
<td>43.04</td>
<td>2.57</td>
<td>60</td>
<td>15.69</td>
<td></td>
</tr>
<tr>
<td>Mazda</td>
<td>11</td>
<td>6.96</td>
<td>79</td>
<td>50.00</td>
<td>2.57</td>
<td>79</td>
<td>18.46</td>
<td></td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>13</td>
<td>8.23</td>
<td>92</td>
<td>58.23</td>
<td>3.04</td>
<td>92</td>
<td>21.50</td>
<td></td>
</tr>
<tr>
<td>Nissan</td>
<td>17</td>
<td>10.76</td>
<td>109</td>
<td>68.99</td>
<td>3.97</td>
<td>109</td>
<td>25.47</td>
<td></td>
</tr>
<tr>
<td>Scion</td>
<td>2</td>
<td>1.27</td>
<td>111</td>
<td>70.25</td>
<td>0.47</td>
<td>111</td>
<td>25.93</td>
<td></td>
</tr>
<tr>
<td>Subaru</td>
<td>11</td>
<td>6.96</td>
<td>122</td>
<td>77.22</td>
<td>2.57</td>
<td>122</td>
<td>28.50</td>
<td></td>
</tr>
<tr>
<td>Suzuki</td>
<td>8</td>
<td>5.06</td>
<td>130</td>
<td>82.28</td>
<td>1.87</td>
<td>130</td>
<td>30.37</td>
<td></td>
</tr>
<tr>
<td>Toyota</td>
<td>28</td>
<td>17.72</td>
<td>158</td>
<td>100.00</td>
<td>6.54</td>
<td>158</td>
<td>36.92</td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td>Audi</td>
<td>19</td>
<td>15.45</td>
<td>15.45</td>
<td>4.44</td>
<td>177</td>
<td>41.36</td>
<td></td>
</tr>
<tr>
<td>BMW</td>
<td>20</td>
<td>16.26</td>
<td>39</td>
<td>31.71</td>
<td>4.67</td>
<td>197</td>
<td>46.03</td>
<td></td>
</tr>
<tr>
<td>Jaguar</td>
<td>12</td>
<td>9.76</td>
<td>51</td>
<td>41.46</td>
<td>2.80</td>
<td>209</td>
<td>48.83</td>
<td></td>
</tr>
<tr>
<td>Land Rover</td>
<td>3</td>
<td>2.44</td>
<td>54</td>
<td>43.90</td>
<td>0.70</td>
<td>212</td>
<td>49.53</td>
<td></td>
</tr>
<tr>
<td>MINI</td>
<td>2</td>
<td>1.63</td>
<td>56</td>
<td>45.53</td>
<td>0.47</td>
<td>214</td>
<td>50.00</td>
<td></td>
</tr>
<tr>
<td>Mercedes-Benz</td>
<td>26</td>
<td>21.14</td>
<td>82</td>
<td>66.67</td>
<td>6.07</td>
<td>240</td>
<td>56.07</td>
<td></td>
</tr>
<tr>
<td>Porsche</td>
<td>7</td>
<td>5.69</td>
<td>89</td>
<td>72.36</td>
<td>1.64</td>
<td>247</td>
<td>57.71</td>
<td></td>
</tr>
<tr>
<td>Saab</td>
<td>7</td>
<td>5.69</td>
<td>96</td>
<td>78.05</td>
<td>1.64</td>
<td>254</td>
<td>59.35</td>
<td></td>
</tr>
<tr>
<td>Volkswagen</td>
<td>15</td>
<td>12.20</td>
<td>111</td>
<td>90.24</td>
<td>3.50</td>
<td>259</td>
<td>62.85</td>
<td></td>
</tr>
<tr>
<td>Volvo</td>
<td>12</td>
<td>9.76</td>
<td>123</td>
<td>100.00</td>
<td>2.80</td>
<td>281</td>
<td>65.65</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>Buick</td>
<td>9</td>
<td>6.12</td>
<td>6.12</td>
<td>2.10</td>
<td>290</td>
<td>67.76</td>
<td></td>
</tr>
<tr>
<td>Cadillac</td>
<td>8</td>
<td>5.44</td>
<td>17</td>
<td>11.56</td>
<td>1.87</td>
<td>298</td>
<td>69.63</td>
<td></td>
</tr>
<tr>
<td>Chevrolet</td>
<td>27</td>
<td>18.37</td>
<td>44</td>
<td>29.93</td>
<td>6.31</td>
<td>325</td>
<td>75.93</td>
<td></td>
</tr>
<tr>
<td>Chrysler</td>
<td>15</td>
<td>10.20</td>
<td>59</td>
<td>40.14</td>
<td>3.50</td>
<td>340</td>
<td>79.44</td>
<td></td>
</tr>
<tr>
<td>Dodge</td>
<td>13</td>
<td>8.84</td>
<td>72</td>
<td>48.98</td>
<td>3.04</td>
<td>353</td>
<td>82.48</td>
<td></td>
</tr>
<tr>
<td>Ford</td>
<td>23</td>
<td>13.65</td>
<td>95</td>
<td>64.63</td>
<td>5.37</td>
<td>376</td>
<td>87.85</td>
<td></td>
</tr>
<tr>
<td>GMC</td>
<td>8</td>
<td>5.44</td>
<td>103</td>
<td>70.75</td>
<td>0.23</td>
<td>385</td>
<td>89.95</td>
<td></td>
</tr>
<tr>
<td>Hummer</td>
<td>1</td>
<td>0.68</td>
<td>104</td>
<td>70.75</td>
<td>0.23</td>
<td>385</td>
<td>89.95</td>
<td></td>
</tr>
<tr>
<td>Jeep</td>
<td>3</td>
<td>2.04</td>
<td>107</td>
<td>72.79</td>
<td>0.70</td>
<td>398</td>
<td>90.65</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>9</td>
<td>6.12</td>
<td>116</td>
<td>78.91</td>
<td>2.10</td>
<td>397</td>
<td>92.76</td>
<td></td>
</tr>
<tr>
<td>Mercury</td>
<td>9</td>
<td>6.12</td>
<td>125</td>
<td>85.03</td>
<td>2.10</td>
<td>406</td>
<td>94.86</td>
<td></td>
</tr>
<tr>
<td>Oldsmobile</td>
<td>3</td>
<td>2.04</td>
<td>128</td>
<td>87.07</td>
<td>0.70</td>
<td>409</td>
<td>95.56</td>
<td></td>
</tr>
<tr>
<td>Pontiac</td>
<td>11</td>
<td>7.48</td>
<td>139</td>
<td>94.56</td>
<td>2.57</td>
<td>420</td>
<td>98.13</td>
<td></td>
</tr>
<tr>
<td>Saturn</td>
<td>8</td>
<td>5.44</td>
<td>147</td>
<td>100.00</td>
<td>1.67</td>
<td>426</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

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 origin         / group ;
   define make           / group ;
   define Horsepower     / analysis n 'Frequency' ;
run ;
```

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

<table>
<thead>
<tr>
<th>Origin</th>
<th>Make</th>
<th>Horsepower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>Acura</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Honda</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Hyundai</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Infiniti</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Isuzu</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Kia</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Lexus</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Mazda</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Mitsubishi</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Nissan</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Scion</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Subaru</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Suzuki</td>
<td>8</td>
</tr>
<tr>
<td>Europe</td>
<td>Audi</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>BMW</td>
<td>20</td>
</tr>
</tbody>
</table>

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:

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

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

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

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

** run:
```plaintext
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.

<table>
<thead>
<tr>
<th>Obs</th>
<th>Origin</th>
<th>Make</th>
<th>Horsepower</th>
<th>pct</th>
<th>cumfreq</th>
<th>cumpct</th>
<th>BREAK</th>
<th><em>RBREAK</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>428</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Asia</td>
<td>Acura</td>
<td>7</td>
<td>1.63551</td>
<td>7</td>
<td>1.636</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Asia</td>
<td>Honda</td>
<td>17</td>
<td>3.97196</td>
<td>24</td>
<td>5.607</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Asia</td>
<td>Hyundai</td>
<td>12</td>
<td>2.80374</td>
<td>36</td>
<td>8.411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Asia</td>
<td>Infiniti</td>
<td>8</td>
<td>1.86916</td>
<td>44</td>
<td>10.280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Asia</td>
<td>Isuzu</td>
<td>2</td>
<td>0.46729</td>
<td>46</td>
<td>10.748</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Asia</td>
<td>Kia</td>
<td>11</td>
<td>2.57009</td>
<td>57</td>
<td>13.318</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Asia</td>
<td>Lexus</td>
<td>11</td>
<td>2.57009</td>
<td>68</td>
<td>15.888</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
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

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

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

```sas
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.

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.
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.

### 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

RECOMMENDED READING
- A Step-by-Step Introduction to PROC REPORT
- So You’re Still Not Using PROC REPORT. Why Not?
- Advanced PROC REPORT:Doing More in the Compute Block

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

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

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.
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 = comma12.0 '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**

```sas
%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 */
    / 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 glpct / computed format = percent9.2 &GroupVar1/Percent ;
%this is the cumulative frequency column ;
define glcumfreq / computed format = comma12.0 &GroupVar1/Cumulative/Frequency width = 10 ;
%this is the cumulative percent column ;
define glcumpct / 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' ;
```

15
** 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 ;
  gltotal = &NumVar..n ;
  glcumtot = 0 ;
endcomp ;

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

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

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

** create running cumulative percent ;
compute glcumpct ;
  glcumpct = divide(glcumfreq, gltotal) ;
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 ;