



`%ProcFreq_Compare`

By Charles Patridge

ISO Innovative Analytics

Charles.S.Patridge@iso.com



The ISO Family of Companies



■ AIR Worldwide

Analytical tools and software systems to help customers estimate and manage the risks associated with natural and man-made catastrophes, weather, and climate



■ Atmospheric and Environmental Research (AER)

Research and consulting services to further understanding of the global environment and to enable better decision making in response to weather and climate risk



■ Domus Systems

Data collection and reporting for affordable housing



■ HealthCare Insight (HCI)

Clinically validated fraud prevention and cost containment for healthcare claims payors

The ISO Family of Companies



■ Insurance Information Exchange (iiX)

Motor-vehicle reports, undisclosed driver reports, credit-score reports, and other underwriting information



■ IntelliCorp

Background checks, employment screening, and drug-testing products for potential employees and volunteers



■ Interthinx

Tools to detect and prevent mortgage fraud



■ NIA Consulting

Fraud-detection and forensic audit services for the home mortgage and mortgage insurance industries

The ISO Family of Companies



■ National Equipment Register (NER)

Information services to help owners of heavy construction and agricultural equipment — and their insurers — recover stolen equipment



■ Predicted Solutions

Software and consulting services to detect and recover losses caused by healthcare fraud, waste, and abuse



■ Quality Planning (QPC)

Tools and services that help auto insurers identify rating errors, recover lost premium, and minimize future losses

The ISO Family of Companies



■ Regs Data

Information on mortgage broker and lender licensing



■ Verisk Health

Healthcare analytics and predictive modeling



■ Xactware

Software for estimating the cost of building and repair

%ProcFreq_Compare

1. %macro program_document;
2. This macro will take 2 SAS datasets, and run a ONE-WAY Frequency on a list of vars which are common to both datasets, and then compare the differences in the frequencies of these variables, and produce a proc print of the distributions with their differences.
3. Sample Call:
4.

```
***%procfreq_compare(base=old, compare=new  
,_vars_= var1 var2 var3 var4, format=7.1, cleanup=0);
```

1. Parameters:
2. **BASE** [REQUIRED] - name of BASE sas dataset with the lib reference.
3. **COMPARE** [REQUIRED] - name of COMPARE sas dataset with the lib reference.
4. **_VARS_** [REQUIRED] - a list of SAS variables which is common to both BASE and COMPARE sasdatasets, separated by blanks.
5. **XWHERE** [OPTIONAL] - a valid WHERE clause to subset both BASE and COMPARE datasets.
6. **WEIGHT** [OPTIONAL] - a sas variable common to both BASE and COMPARE datasets to weight the ONE-WAY Frequencies.
7. **FORMAT** [OPTIONAL] - a valid format that can be applied to all the variables listed in **_VARS_**. Default is best.2 for numeric variables.
8. **CLEANUP** [OPTIONAL] - 1 will cleanup all the work files not needed after comparison is completed. Any other value will keep work files around.

1. Notes:
2. 1. **The SAS Lib References need to be issued before calling this macro.**
3. 2. **Do not mix character and numeric variables in the same list of `_VARS_`.**
4. 3. **Need to choose a format that is appropriate for all variables listed in `_VARS_`.**
5. `%mend program_document;`

1. `%macro get_Var_Type(inputds=, var=);`
 1. `%local dsid varnum vartype rc;`
 2. `%let dsid = %sysfunc(open(&inputds));`
 3. `%let varnum = %sysfunc(varnum(&dsid,&var));`
 4. `%let vartype = %left(%sysfunc(vartype(&dsid,&varnum)));&vartype`
 5. `%let rc = %sysfunc(close(&dsid));`
2. `%mend get_Var_Type;`

1. `%macro procfreq_compare (base=, compare=, _vars_=, xwhere=, weight=, format=best.2, cleanup=1);`

2. `%if %length(&_vars_) gt 0 %then %do;`
 1. `%let _vars_ = %cmpres(&_vars_);`
 2. `%global howmany_vars;`
 3. `data _null_;`
 4. `howmany = 1 + (length(compbl("&_vars_")) - length(compress("&_vars_")));`
 5. `call symputx('howmany_vars', put(howmany,3.));`
 6. `run;`

3. `%end;`

1. `/***/` `FREQ` on `BASE` or `COMPARE` Table `*/**/`
2. `%macro run_freq(Base_Compare=BASE);`
3. `proc freq data=***Base_Compare;`
4. `%if %length(&xwhere) %then %do;`
 1. `where &xwhere;`
 2. `title2 "Where Clause = &xwhere";`
5. `%end;`

1. `%do _i_ = 1 %to &howmany_vars;`
 1. `table %Scan(&_vars_, &_i_,%STR()) / missing
outpct noprint out=&Base_Compare.&_i_;`
 2. `format %sysfunc(translate(%Scan(&_vars_,
&_i_,%STR()), ' ', '*')) &format ;`
2. `%end;`
3. `%if %length(&weight) gt 0 %then %do;`
4. `weight &weight;`
5. `%end;`
6. `%mend run_freq;`

1. `%orun_freq(BASE_COMPARE=BASE);`
2. `%orun_freq(BASE_COMPARE=COMPARE);`

1. `%macro Join_Base_Compare;`
2. `%do _i_ = 1 %to &howmany_vars;`
 1. `%let var_type = %get_Var_Type(inputds=base&_i_, var=%Scan(&_vars_, &_i_,%STR()));`
 2. `proc sort data=base&_i_ out=base&_i_ ; by %Scan(&_vars_, &_i_,%STR()); run;`
 3. `proc sort data=compare&_i_ out=compare&_i_; by %Scan(&_vars_, &_i_,%STR()); run;`

1. data base&_i_;
2. set base&_i_;
1. %if &var_type = N %then %do;
 - %Scan(&_vars_, &i_,%STR()) = input(put(%Scan(&_vars_, &i_,%STR()), &format), best.);
2. %end;
3. %else %do;
 - %Scan(&_vars_, &i_,%STR()) = input(put(%Scan(&_vars_, &i_,%STR()), &format), &format);
4. %end;
3. run;
4. data compare&_i_;
5. set compare&_i_;
1. %if &var_type = N %then %do;
 - %Scan(&_vars_, &i_,%STR()) = input(put(%Scan(&_vars_, &i_,%STR()), &format), best.);
2. %end;
3. %else %do;
 - %Scan(&_vars_, &i_,%STR()) = input(put(%Scan(&_vars_, &i_,%STR()), &format), &format);
4. %end;
6. run;

1. `data Diff&_i_;`
2. `length var_name $32.;`
3. `label var_values = "%Scan(&_vars_, &_i_,%STR())";`
4. `merge`
5. `base&_i_ (rename= (count=base_cnt percent=base_pct))`
6. `compare&_i_ (rename= (count=compare_cnt percent=compare_pct)) ;`
7. `by %Scan(&_vars_, &_i_,%STR());`
8. `var_name = "%Scan(&_vars_, &_i_,%STR())";`
9. `var_values = %Scan(&_vars_, &_i_,%STR());`

1. label

1. base_cnt = "Base Count"
2. base_pct = "Base Percent"
3. compare_cnt = "Compare Count"
4. compare_pct = "Compare Percent"
5. diff_cnt = "Difference Count"
6. diff_pct = "Difference Percent"

2. ;

```
1.  if base_cnt      = . then base_cnt      = 0;
2.  if compare_cnt = . then compare_cnt = 0;
3.  diff_cnt        = compare_cnt - base_cnt;
4.  if base_pct     = . then base_pct     = 0;
5.  if compare_pct = . then compare_pct = 0;
6.  diff_pct        = compare_pct - base_pct;
7.  drop %Scan(&_vars_, &_i_,%STR());
8.  run;
```

```
%end;
```

1. `/***/ now join all the difference files into a single file`
2. `delete _diff_ if it exists ***/`
3. `proc datasets library=work nolist;`
4. `delete _diff_;`
5. `quit;`

1. `%do _i_ = 1 %to &howmany_vars;`
 1. `proc append base=_diff_ data=diff&_i_; run;`
 2. `%if &cleanup = 1 %then %do;`
 3. `/** delete the individual base and compare files to clean up **/`
 1. `proc datasets library=work nolist; delete base&_i_ compare&_i_; quit;`
 4. `%end;`
2. `%end;`

1. `proc sort data=_diff_ out=_diff_; by var_name; run;`
 2. `title1 "Base File = &base";`
 3. `title2 "Compare File = &compare";`
 4. `proc print data=_diff_;`
 5. `by var_name;`
 6. `id var_name;`
 7. `pageby var_name;`
 8. `run;`
1. `%mend Join_Base_Compare;`
 2. `%Join_Base_Compare; run;`
1. `%mend procfreq_compare;`

1. Example 01:
2. libname mylib "C:\DAD\sas_stuff\Proc_Freq_Compare";
3. title3 "Compare Year to Year";
4. %procfreq_compare(
5. base=mylib.sasalary_2000 ,
6. compare=mylib.sasalary_2001 ,
7. _vars_ = Gender,
8. format=\$6.,
9. cleanup=0
10.);

1. Example 01:

Base File = mylib.sasalary_2000
Compare File = mylib.sasalary_2001

var_name	var_ values	base_cnt	base_pct	compare_ cnt	compare_ pct	diff_cnt	diff_pct
Gender	Female	75	34.0909	62	38.9937	-13	4.90280
	Male	145	65.9091	97	61.0063	-48	-4.90280



1. Example 02:

```
proc format;  
  value birthyr  
    low- 1949 = '1950'  1950-1959 = '1960'  1960-1969 = '1970'  1970-1979 = '1980'  
    1980-1989 = '1990'  1990-1999 = '2000'  2000-2009 = '2010'  
  ;
```

```
%procfreq_compare(  
    base=mylib.sasalary_2000 ,  
  compare=mylib.sasalary_2001 ,  
    _vars_ = birthyr,  
    format=birthyr5.,  
  cleanup=0);
```

1. Example 02:

Base File = mylib.sasalary_2000

Compare File = mylib.sasalary_2001

var_name	var_ values	base_cnt	base_pct	compare_ cnt	compare_ pct	diff_cnt	diff_pct
birthyr	.	9	4.0909	8	5.0314	-1	0.94054
	1950	19	8.6364	13	8.1761	-6	-0.46026
	1960	55	25.0000	31	19.4969	-24	-5.50314
	1970	85	38.6364	54	33.9623	-31	-4.67410
	1980	52	23.6364	53	33.3333	1	9.69697

1. Example 03:

```
%procfreq_compare(  
    base=mylib.sasalary_2000 ,  
compare=mylib.sasalary_2001 ,  
    _vars_ = educate gender industry ,  
format= $15. ,  
cleanup=0);
```

1. Example 03:

Base File = mylib.sasalary_2000

Compare File = mylib.sasalary_2001

var_name	var_values	base_cnt	base_pct	compare_ cnt	compare_ pct	diff_cnt	diff_pct
educate	BA/BS	91	41.3636	67	42.1384	-24	0.7747
	High School Onl	11	5.0000	8	5.0314	-3	0.0314
	MA/MS	105	47.7273	70	44.0252	-35	-3.7021
	PhD	13	5.9091	14	8.8050	1	2.8959

var_name	var_values	base_cnt	base_pct	compare_ cnt	compare_ pct	diff_cnt	diff_pct
gender	Female	75	34.0909	62	38.9937	-13	4.9028
	Male	145	65.9091	97	61.0063	-48	-4.9028

var_name	var_values	base_cnt	base_pct	compare_ cnt	compare_ pct	diff_cnt	diff_pct
industry	Banking	16	7.2727	10	6.2893	-6	-0.9834
	Consulting	8	3.6364	18	11.3208	10	7.6844
	Direct Mail Mar	7	3.1818	9	5.6604	2	2.4786
	Educational Ins	18	8.1818	9	5.6604	-9	-2.5214
	Financial	8	3.6364	12	7.5472	4	3.9108
	Govt - Federal	15	6.8182	10	6.2893	-5	-0.5289
	Govt - State	5	2.2727	9	5.6604	4	3.3877
	Health Care	13	5.9091	13	8.1761	0	2.2670
	Insurance - Hea	13	5.9091	5	3.1447	-8	-2.7644
	Insurance - Pro	5	2.2727	3	1.8868	-2	-0.3859





`%ProcFreq_Compare`

By Charles Patridge

ISO Innovative Analytics

Charles.S.Patridge@iso.com



%ProcFreq_Compare

can be copied from

<http://www.sconsig.com/sastips/tip00417.htm>