

Paper PP14

Dynamic footnotes – be more dynamic than your statistician -

Wilfried Esken, HMS analytical software GmbH, Heidelberg, Germany

ABSTRACT

Researchers want to get most out of clinical data. Consequently, statistical models are getting more and more complex. The statistical programmer must cope with this fact when writing analysis programs and creating outputs. One aspect in this field is how to display the statistical outputs, especially the p values. In our experience it turns out that most often all kinds of p values are required to appear and to be explained in the footnotes. And each time the statistician applies a new model, the programmer must flexibly adapt the footnotes.

INTRODUCTION

In this poster a simple but effective approach is presented how to dynamically generate footnotes with p values, when the underlying model is under constant matter of change or unknown beforehand.

PROBLEM

In the above described area, the programmer must cope with the following subproblems:

1. Get the pvalues from the statistical procedure.
2. Assure that the pvalues are displayed together with the correct descriptions.
Any mismatches must be strictly avoided.
3. Properly format the footnotes.

Especially the third subproblem often turns into a tedious task, when the model is under constant matter of change or descriptions and formats must be adapted repeatedly.

The problems get worse, when you don't know the exact model beforehand. This might be the case when for example one must implement an analysis in two steps:

- a) In a first step one should implement the model with a certain number of specified covariates.
- b) In a second step repeat the analysis, but now use only those covariates from step a), which have significant pvalues in a).

So, when the model from step a) change, you subsequently must recheck all outputs generated in step b). Even more worse, when for example after an unblinding the significant covariates in step a) changes – without change of the model itself -, you might get into trouble: You again must recheck and correct all outputs produced in step b) - One more point on your to-do list after a database lock, and you even might not aware of it.

SOLUTION

In this poster, a dynamic generation of footnotes is presented as a solution, to overcome the problems described above. The subproblems are solved as following:

1. GET THE PVALUES FROM THE STATISTICAL PROCEDURE

Perform the statistical procedure and define the proper ODS output dataset which contains the pvalues:

```
ods output type3 = type3;
proc phreg data = ...;
  class trt01pn(ref="1");
  model aval * cnsr(1) = trt01pn age bmi /RL;
run;
```

2. ASSURE THAT CERTAIN PVALUES ARE DISPLAYED WITH THE CORRECT DESCRIPTIONS

The names of the variables, which are used as covariates in the model, are given in the ODS dataset in variable "EFFECT". This is exploited in the following piece of code:

```
data footpvals;
  set type3;
  call symput(effect, put(probchisq, pvalue6.4));
run;
```

When performing this data step, SAS® creates a set of macro variables. The names of the macro variables are identical to the names used in the model, whereas the contents of the macro variables are identical to their pvalues. So, using the generated macro variables in subsequent code guarantees the correct usage of the pvalues. For example:

```
proc format lib=work;
  value $footfmt;
    "TRT01PN" = "*** The time dependent...(p=&trt01pn) "
    "AGE"      = "baseline age (p=&age) "
    "BMI"      = "baseline BMI (p=&bmi) ";
run;
```

3. PROPERLY FORMAT THE FOOTNOTES

The generated formats are then used to create one big string containing all footnotes. Again, it is exploited that the covariates are given in ODS dataset variable "EFFECT".

```
data foots;
  length foottmp $ 2000;
  set footpvals end=eof;
  foottmp = put(effect, $footfmt.);
  if eof then foottmp = "and " || strip(foottmp);
run;
proc sql noprint;
  select foottmp into: _footlst separated by ", "
  from foots;
quit;
```

From this string the single footnotes are created. The length of the footnotes must be aligned with the available space. A length of 80 is assumed to be appropriate but can be adapted if necessary:

```
data _null_;
  length %do i = 1 %to 10; foot&i %end; $ 80;
  allfoot = "&_footlst";
  %do i = 1 %to 10;
    do while (length(compress(foot&i)) <= 80
      and allfoot ne "");
      _ind = index(allfoot, " ");
      foot&i = strip(foot&i) || " " || strip(substr(allfoot, 1, _ind));
      allfoot = substr(allfoot, _ind + 1);
    end;
    call symput("foot&i", strip(foot&i));
  %end;
run;
```

At the poster, examples of outputs are presented.

CONCLUSION

In our daily work we find it helpful to have a tool which generates dynamic footnotes. During development it turned out that the implementation was easier than expected, as no special program skills were necessary.

As a general conclusion we would recommend checking if repeated tasks can be automated. Repeatedly doing tedious tasks is rather time consuming and error-prone. This is especially true when changes of requirements are likely to come.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact: Wilfried Esken

wilfried.esken@analytical-software.de



Brand and product names are trademarks of their respective companies.