The Integration of SAS® Software in a Multicentric Tumor Registry

Harald Pitz, Frankfurt University Hospital
Hans-Peter Howaldt, Frankfurt University Hospital
Marcus Frenz, Frankfurt University Hospital

1. Introduction

In 1989 the German-Austrian-Swiss cooperative group on tumors of the maxillo-facial region (DOSAK) established a multicentric tumor registry for head and neck cancer financed by a grant of the Deutsche Krebshilfe - Dr. Mildred Scheel Stiftung.

All patients suffering from malignant disease and treated in the 78 departments of maxillo-facial surgery in Germany, Austria and Switzerland are supposed to be registered. After 21 months the registry's database contains more than 2,800 patients.

The documentation for each patient contains up to 8 forms. The average amount of data for each patient is about 100 data items or 500 bytes. Based on the increase of patients during the past 21 months we expect about 1200 - 1500 new patients yearly.

The main objectives of the registry are:
1. to report state of the art of therapy
2. evaluation of prognostic factors
3. proposal of a prognostically relevant tumor classification
4. facts concerning the value of different therapy modalities.

2. Requirements

The main problems a multicentric tumor registry has to deal with are data quantity (acceptance by the participating institutions) and data quality. Both can only be achieved by supporting the daily routine work of the involved physicians as well as by supporting scientific questions and areas physicians are interested in. Thus a software environment must have capabilities to support both individual processing of single patient's data and powerful processing of the whole database. The requirements in detail are:

1. data acquisition including special medical data types (e.g. time intervals ("2 to 4 months"), numeric fields with controlled, but variable limits ("size normally 100 to 220 cm, but 80 cm is accepted if the user agrees"), free-text controlled by a thesaurus)
2. analysis of free-text including a thesaurus for synonyms, different spelling etc.
3. generator for reports such as doctor's letters, patient summaries etc.
4. modules for standard descriptive analysis
5. modules for non-parametric survival analysis (Kaplan-Meier)
6. modules for semi-parametric survival analysis (COX's regression model)
7. modules for fully-parametric survival analysis
8. simple processing of subgroups, e.g. for a specific institution, for patients with several therapies
9. automatic generation of graphics
10. dialog and batch processing of all programs
11. distribution of data to participating institutions

For data entry and free-text report generation we are using the medical record system BAIK which supports special medical data types and thesaurus-based free-text search. For all further processing we use the SAS System. An intelligent interface has been developed to export data from BAIK to the SAS System. We are running BAIK and the SAS System (Release 6.04) on a PC (80386) under PC-DOS.

The most important BAIK-features, the BAIK-SAS interface and our SAS application will be described.

3. BAIK

BAIK ("Medical Record Keeping and Report Generating") has been developed at the Frankfurt University Hospital and includes special features necessary for documentation of medical data which are not provided by the SAS System. A detailed description can be found in [Giere86] and [Giere88]. A complete BAIK application for head and neck cancer has been developed since 1985 [Howaldt89]. BAIK is also the basis for a departmental information system [Pitz89, Pitz90].

Besides special medical data types and the report generator the BAIK free-text search is one of the most important features. A thesaurus contains various relations such as "preferred term", "narrow term", "broader term" etc. and is primarily used to detect words with different spelling (tumor or tumour) or same meaning (tumor or malignancy). Groups of words can be linked together to do an automated search without having regard for different spelling.

We perform frequency statistics of free-text to count specific kinds of surgery, therapy complications or previous diseases.

In addition, the free-text-search is used to make a preselection of patients for our BAIK-SAS interface which will be described later.

4. The export interface BAIK-SAS

The BAIK-SAS interface is used to export data entered in BAIK to the SAS System. An ASCII-file containing all necessary information to build up a SAS dataset is generated.

The interface allows the user to define

1. the patients whose data should be exported,
2. the data items of each patient to be exported,
3. the name of the ASCII-file (the same name will be used for the dataset),
4. the maximum line-length of the file and
5. the symbol used for missing values (e.g. dot ".")

The set of patients can be restricted to

1. individual ID-numbers (enumeration or interval)
2. data entered in a special time-interval
3. data resulting from a special time-interval
4. preselected patients stored as a result of a free-text search (see Section 3)
5. preselected patients stored as a result of a SAS analysis

All combinations of these parameters can be used together.

The restrictions 2 and 3 (time intervals) could also be evaluated by SAS statements later, but the preselection by the interface reduces the amount of data and thus the size of the ASCII-file.

To determine the data items to be exported the interface allows the user to generate, store and modify so-called export-schemas. Each export-schema contains a short description of the purpose of these data items and a list of data items including the BAIK identification of the item, the maximum length (longer items are cut off), the name of the SAS variable and a comment for this item. For different purposes, different export-schemas can be stored and used when necessary.

The interface automatically produces a file with the extension .SAS containing all SAS statements in order to build up a SAS dataset. The data type for each item is stored in the BAIK data dictionary, thus the interface can automatically generate format, informat, label and length statements. Figure 1 shows an example of an export file containing 3 data items of different data types. The BAIK id-numbers (Variable PNR) are automatically added to the dataset.

Figure 1: Example of an export-file DEMO.SAS with 5 patients and 3 data items generated by the BAIK-SAS interface:

```
/* BAIK-SAS-Datenexport vom 27.1.91 */

libname in '.';
data in.DEMO;
input PNR 1-5 LNR 8-9
      DATDIAG ddmmyy10. LOKSCHL 20-24 OPART $ 25-70 ;
format DATDIAG ddmmyy8.;
informat DATDIAG ddmmyy8.;
label DATDIAG = 'Diagnosedatum';
label LOKSCHL = 'Lokalisation';
label OPART = 'Operation';
length LOKSCHL 3;

cards4;
    1 1  06.09.89 1440Zungenteilresektion
    2 1  02.11.89 1456Unterkiefer-Teilresektion
    3 1  07.09.8914312Enorale Tumorresektion
    4 1  12.04.90 1441Mundbodenresektion
    5 1  12.02.8819603
;;;;
run;
```

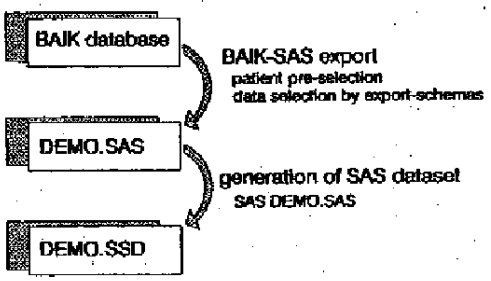
A comment file with extension .KOM with export date and time and description of the export-schema in easyly readable form is generated too.

The SAS dataset DEMO.SSD can now easily be generated by entering

SAS DEMO.SAS

at the DOS command-line.

In figure 2 the export process and the necessary steps are shown:



5. SAS application

5.1 General description

SAS software is used for statistical analysis and graphics. The programs are integrated in a menu environment actually based on the SAS macro statements %WINDOW and %DISPLAY. A user interface based on SAS/AF® software is under development (see section 5.5).

Most of our applications take some minutes or hours, especially if graphics are generated for a laser printer and/or the analysis is done separately for all participating institutions. For that purpose we developed a batch-like system where the user first enters all demands and

parameters which are then automatically processed by the system. The main parameters the user has to enter are:

1. what kind of analysis he wants to do (multiple choices are possible),
2. output device for graphics (screen, printer or graphic stream file) and
3. if the analysis are based on the total database, optional automatically seperated for each institution, or on a specific institution identified by its number.

The input of these parameters is actually implemented by using %WINDOW and %DISPLAY statements. As the users of the system are the registry's staff itself we did not consider it necessary to implement parameter checking, help functions etc. Figure 3 shows an example of a %WINDOW statement to select the institutions. We make use of the GROUP= option allowing a comprehensive view and better maintenance.

Figure 3: Example of a %WINDOW statement:

```
%window institut color = cyan
  group = start

  #5 @10  'DÖSAK tumor registry ' attr=highlight
          color=yellow
  #7 @10  'modul for analysis and graphics'

  #9 @10  'Analysis can be executed either'
  #10 @15 '(T0) for the total database       or'
  #11 @15 '(BY) seperated by institutions    or'
  #12 @15 '(##) for a single institution     or'
  #13 @15 '(0)  abort'

  group = input

  #16 @10 'Please enter your choice: ' color=blue kli 2
          required = yes

  group = nodata

  #18 @10 'This institution does not exist'
          ATTR=(BLINK,HIGHLIGHT) COLOR = RED
  #20 @10 'Please select another institution'
          ATTR=(BLINK,HIGHLIGHT) COLOR = RED

  group = end

  #18 @10 'You selected  "abort". Good bye.'
  #20 @10 'Continue with <RETURN>'
;
```

After entering all necessary parameters the system executes the required SAS programs. Status messages (e.g. institution processed) are produced using a %DISPLAY statement with the NOINPUT parameter to continue the execution without user acceptance.

The single application programs are all designed in the same way. They are implemented as macros with following standard parameters used in addition to program dependant parameters:

1. dataset to be used
2. title including name of institution (or 'total')

The application environment supposes that all necessary datasets have already been exported from the BAIK system and generated by the SAS System. The number of observations is automatically included in the title.

In the following sections the most frequently used procedures are described shortly. As example we present the plot of survival curves (see figures 5 and 6).

5.2 Descriptive analysis

The main procedures we use for descriptive analysis are PROC FREQ, PROC GCHART, PROC MEANS and PROC UNIVARIATE.

PROC UNIVARIATE provides detailed information about the distribution of a single variable and can be used for additional quality control. If we detect possibly incorrect data items, PROC FSEDIT and the FIND command are used to determine the patient and to correct the data manually. Corrections can be directly made in the dataset to continue or restart the analysis, but changes always have to be repeated in BAIK too.

5.3 Multivariate survival analysis

PROC LIFEREG and PROC LIFETEST are the main SAS procedures we use for survival analysis.

PROC LIFEREG is used for evaluation of prognostic factors which is one of the main objectives of the project. Normally we work with the log-normal distribution.

Figure 4 shows a sample program with PROC LIFEREG. The macro variables are already expanded for better readability.

```
proc lifereg data=in.demo;
 LogNor: model SurvTime*Death(0) = TestVar1 TestVar2
TestVar3 / Distribution = LNormal  MaxIt = 100;
 title "sample program using PROC LIFEREG (n=77)";
run;
```

PROC LIFETEST is used to compare different subgroups and to produce Kaplan-Meier survival curves. As PROC LIFETEST "only" produces an ASCII-plot, we developed

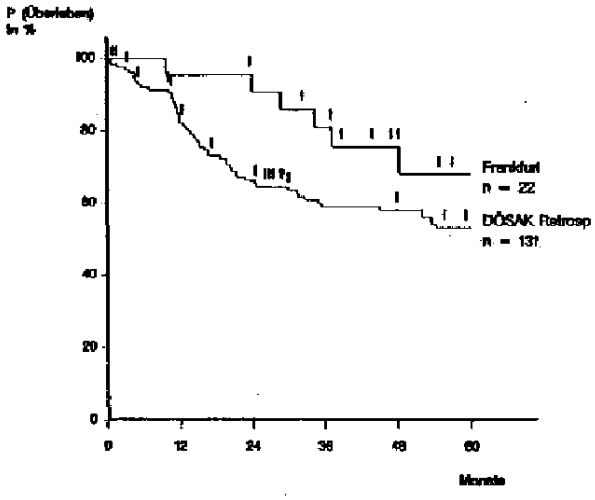
a module which transforms the output dataset of PROC LIFETEST into a form which can be used by PROC GPLOT. In addition this module allows to display the censored observations in the same graphic which is very important for the interpretation of the results.

Figure 5 shows a sample program with PROC LIFETEST. Again most of the macro variables are expanded for better readability.

```
title f=swiss 'Demonstration of Kaplan-Meier';

proc lifetest data=demo method=km plots=(s)
              NoTables
              maxtime=1800 outsurv=kmlife;
time UeberLeZ*Tod(0);
strata Gruppe;

/* Transformations for PROC GPLOT */

%macro Treppe (Ueb, Cens);
 do;
  if FIRST._Strtum_  then do;
    AltUeb = 100;
    ToBig  = 0;
    end;
  Zeit = UeberLeZ / 30;
  if (Zeit > 60) then do;
    if NOT ToBig then do;
      ToBig = 1;
      Zeit  = 60;
      &Ueb =  AltUeb;
      output;
      end;
    end;
  else
    if _Censor_= 0 then do; /* nicht zensiert */
      &Ueb = AltUeb;
      output;
      &Ueb = Survival * 100;
      if AltUeb ^= &Ueb then output;
      AltUeb = &Ueb;
      end;
    else do;
      &Cens = AltUeb;
      output;
      end;
  end;
%mend Treppe;

data plot;
 set kmlife;
 by _Strtum_;
 drop AltUeb ToBig UeberLeZ;
 retain AltUeb ToBig;

 if _Strtum_ = 1 then %Treppe(Ueb1, Cens1);
 if _Strtum_ = 2 then %Treppe(Ueb2, Cens2);

%macro Markiere(Cens,Zeit);
  do; function='label'; x=&Zeit; y=&Cens; position='2';
text='I';
      output;
  end;
%mend Markiere;

data Zensiert;
 set plot;
 length function $ 8 style $ 8;
 length text $ 15;
 length position $ 1 when $ 1 xsys $ 1 ysys $ 1;
 retain style 'SWISSL' when '8' xsys '2' ysys '2';
 if Cens1 ^= . THEN %Markiere(Cens1,Zeit);
 if Cens2 ^= . THEN %Markiere(Cens2,Zeit);

data UebLast;
 keep UebLast1 UebLast2;
 set plot End=EOD;
 retain UebLast1 100;
 retain UebLast2 100;
 UebLast1 = Min(Ueb1, UebLast1);
 UebLast2 = Min(Ueb2, UebLast2);
 if EOD THEN output;

data Notiz;
 set UebLast;
 length function $ 8 style $ 8;
 length text $ 30;
 length position $ 1 when $ 1 xsys $ 1 ysys $ 1 hsys $ 1;
 retain style 'SWISSL' xsys '2' ysys '2' when '8';
 retain function 'label' x 60 hsys '4' size 0.8;
  y=UebLast1+3; position='6'; text='  Frankfurt'; output;
  y=UebLast1; position='F'; text='  n = 22';
output;
  y=UebLast2+3; position='6'; text='  DÖSAK Retrosp';
output;
  y=UebLast2; position='F'; text='  n = 131';
output;

data Notizen;
 set Zensiert Notiz;

/* Graphical output */

symbol1 i=join v=point l=1   c=red;
symbol2 i=join v=point l=1   c=green;
axis1 length=10 CM
      offset= (0,1.5 cm)
      width = 4
      minor = none
      order = 0 to 60 by 12
      value = (f=swiss1)
      label = (f=swiss j=r 'Monate');

axis2 length=9 CM
      offset=(0,0.5 cm)
      width = 4
      minor = none
      order = 0 to 100 by 20
      value = (f=swiss1)
      label = (f=swiss j=l 'P (Überleben)' j=l 'in %');
```

```
proc gplot data=plot;
 plot (Ueb1 Ueb2)*Zeit / annotate=Notizen
   overlay haxis=axis1 vaxis=axis2;

run;
```

Two compared survival curves printed with the procedure listed in figure 5 are shown in figure 6:



5.5 User interface with SAS/AF software

The interface by means of %WINDOW and %DISPLAY statements has been used to combine and test the macros and to get experience with the complete application. Presently we develop a user interface with SAS/AF software to run procedures and to display results immediately as well as to collect statements first and to run them simultaneously. In our view this capability which is obtained by choosing either the SUBMIT or the SUBMIT IMMEDIATE SCL statement is very important.

5.6 Export of data for participating institutions

Participating institutions can do analysis even if they don't use the SAS System. Using PROC DBF respectively PROC DIF one is able to generate DBASE files and LOTUS 1-2-3 worksheet files.

6. Discussion

The combination of BAIK and the SAS System proved to be very suitable for our project. Although the export of data from BAIK to the SAS System appears to be cumbersome it poses no problem for us, because analyses are not done daily but for specific events such as congresses, requests from institutions or the annual project report. On the contrary the separation from original data and analysis data guarantees a constant and consistent number of patients in all analyses being done at one time.

BAIK supports comprehensively the needs of an individual medical record including thesaurus-based processing of free-text.

The SAS System provides solutions for almost all our graphical and statistical tasks.

A very important aspect is the capability of automation and batch processing of all programs. This capability is primarily based on the powerful SAS macro language, which can be used everywhere in a SAS program. The consistent design of SAS procedures, especially use of the same name and meaning of keywords in different procedures, makes it easy to learn how to use new functions and to develop new applications. A developer can rely on special properties of SAS software everywhere, e.g. the use of the BY statement.

The mixture of data and procedure steps as a basic concept of the SAS System allows a set-oriented approach to a database which makes it easier to develop and documentate a program. Nevertheless the SAS language provides all necessary (procedural) commands and functions to manipulate datasets.

Restrictions and limitations

1. The customization of graphics including legends, titles, annotations etc. is often a quite troublesome work. Especially as a non-experienced user you have to perform a lot of tests to get the desired result. Under this aspect, SAS cannot be compared with PC-programs like Harvard Graphics or Freelance Plus. A post-editing of graphics produced with SAS/GRAPH® is possible by generating a Computer Graphics Metafile (CGM), but this process stands in contradiction to our need of automation.
2. The PC-Version of the SAS System often has problems with memory allocation due to the DOS limitation of 640 KBytes. The use of EMS will avoid some of these problems.
3. The Cox's regression model, which is an important and wide-spread method for survival analysis, is not yet available for PC's. The procedure COXREGR included in the SUGI Supplemental Library is only available under CMS and OS. In future we plan to implement the Cox's regression model in IML. Due to memory problems such an implementation will require Release 6.06 and the use of OS/2.

7. References

[Giere86] Giere, W.: BAIK, Befunddokumentation und Arztbriefschreibung im Krankenhaus. Media Verlag, Taunusstein 1986.

[Giere88] Giere, W.: Treatment, teaching and research - structural requirements for medical records, the information model of BAIK. In: Proc. of the 8th Joint Conference in Medical Informatics, Tokyo 1988.

[Howaldt88] Howaldt, H.-P.; Volke, M.; Pitz, H.; Neubert, J.; Oehlenschläger, W.: Computerdokumentation der Malignome des Mundes, der Kiefer und des Gesichts mit dem BAIK-System. Dtsch Z Mund Kiefer GesichtsChir 13, 1989.

[Pitz89] Pitz, H.; Howaldt, H.-P.; Giere, W.: A Heterogeneous MUMPS Network as Basis for a Departmental Clinical Information System. In: Proceedings of the 6th Conference on Medical Informatics, North Holland 1989.

[Pitz90] Pitz, H.; Howaldt, H.-P.: Experiences with a heterogeneous MUMPS Network. In: Newsletter of the MUMPS User's Group Europe, Vol. VII, No. 2/3 1990.

Author:

Dipl.-Inform. Harald Pitz

Department of Medical Informatics
J.W. Goethe University Hospital
Theodor-Stern-Kai 7
D-6000 Frankfurt / Main
Germany

Tel.: (49) 69 6301 6642
Fax.: (49) 69 6301 6301