NPAR under SAS75

Daniel M. Chilko
West Virginia University

Brief history and description of NPAR

NPAR is a computer program to perform nonparametric or "distribution-free" statistical tests. It was written originally by the Computer Institute for Social Science Research in 1966 for use on a CDC computer at Michigan State University Computer Laboratory, Michigan State University, East Lansing, Michigan.

The program was written to be of particular interest to research workers in the behavioral sciences, where the distribution of responses to a questionnaire, or measurements along a scale, may be unknown or known to be extremely erratic. The nonparametric tests in NPAR may be applied to these data, which fail to meet the requirements of the more familiar statistical tests of significance.

However, the program provides features that would recommend it to workers in every field of research. In particular, the program determines a precise probability or "level of significance" for the statistics it calculates. In addition, the program makes provisions for missing data items.

In the summer of 1974, the Computer Center at West Virginia University acquired NPAR-360 (a version of NPAR for use on IBM 360 or 370 computers) for inclusion in a library of statistical software already consisting of SAS, SPSS, BIOMED, and DATATEXT. The purchase of NPAR was made at the recommendation of the Statistical Software Advisory Group at WVU, who had surveyed our computer center users. The survey indicated an unanswered desire for a nonparametric statistical analysis computer program.

The initial response to NPAR was only slight use, in comparison to the initial response to SAS. The computer center at WVU had installed a version of SAS in early 1970 and SAS quickly became the most frequently used of our statistical computer programs. In fact, it is the most frequently used application program of any type.

At this time, the author was writing a local SAS procedure to allow users to punch SAS datasets or to create regular O.S. datasets. The procedure was written in PL/1 and was driven by PARMCARD information similar to the SAS INPUT statement. The experience of writing this procedure provided insight into the structure and organization of the SAS computer program.

The structure of the NPAR computer program and the SAS computer program are strikingly similar. This is reflected in similar control languages. Here, for example, are the control statements in NPAR to perform the Mann-Whitney U test.

```
OP,MWU
SIZE = 4,80
GROUP = 1
SCORE = 2,3,4
VARIABLE NAMES
SEX
ANXIETY
SELFCNPT
ATTITUDE
FORMAT(5X,F1.0,2X,3F1.0)
```

The OP statement serves a function similar to the SAS PROCEDURE statement and selects a particular statistical analysis to be performed.

The SCORE and GROUP statements serve the same functions as the SAS procedure information statements, VARIABLES and CLASSES. That is, they identify dependent (or score) variables and independent (or grouping) variables to a procedure.

The SIZE statement and the FORMAT statement are associated with data input. Also available is a DATA statement which can be used to indicate, for example, that data previously input is to be analyzed, or that the data is to be input from logical unit 9.

The VARIABLES NAMES statement is optional and provides for labelling variables in the printed output.

A MISSING DATA statement provides for identifing values of variables which are not to participate in the statistical analysis.

The NPAR program provides (1) a control statement processor which identifies and decodes control statement information, (2) an input processor to handle data input, and (3) an executive routine to call the appropriate subprogram to provide the requested analysis. The SAS program provides similar "system-type" functions.

The NPAR system has several disadvantages, in the author's opinion, both from the user's and installation's point of view.

1. organized as an overlay program which requires, at minimum, a 190K bytes core storage region.

2. a control language different from SAS for the user to learn.

3. unsophisticated input facilities with no internal facilities for data editing and recoding, limited error recovery.

4. fixed sized arrays for storing data and intermediate results. The program imposes a limit of 10,000 on the maximum number of data items (variables x observations). Since the data array is held in core storage, this is partly responsible for the large program size.

The NPAR system does have several advantages:

1. a high-level control language.

2. well-tested algorithms, especially for the calculation of probabilities.

3. good documentation.

4. missing value exclusion and optional variable labelling.

Interfacing SAS and NPAR

The author tried, then, to provide the best of NPAR to SAS users by including them as locally-added procedures in SAS.

The process consisted of:

1. discarding the NPAR "system-type" functions. These functions are performed superiorly by SAS.

2. writing an interface routine (actually three versions). The NPAR analysis procedures expect control information and data to be in common or named common storage when they are executing. In SAS, the same kind of information is obtained by the analysis procedures through a subroutine or function call. Thus the interface routine does little more than make the necessary subroutine calls and place the retrieved information into core storage accessible to the analysis procedures. Since the interface is able to determine how much core storage is necessary for the data and control information, it acquires, again via a SAS subroutine call, only the amount of core storage needed. This, generally, is smaller than the static size of 10,000 words in NPAR. There are three different versions of the interface routine depending on whether the analysis subprogram needs, does not need, or will accept, a CLASSES statement.

3. modest changes to the NPAR statistical analysis subprograms.
Changes to the NPAR subprograms were necessary only to meet some of our own design objectives. In fact, one of our design objectives was to keep the changes of the programs minimal, so as not to introduce program errors inadvertently.

The design objectives and the changes are:

1. dynamic core storage allocation.
Calling argument lists were expanded since we wanted to eliminate blank and named common.

2. double precision storage of "real" numbers.
We adopted the SAS convention that floating point numbers be stored in double precision, although, on reflection this seems unnecessary. This meant changing SQRT to DSQRT, etc.

3. use of character variables as CLASSES variables
To support this feature of SAS, some FORMAT statements were duplicated: One version of the format statement for numeric variables; another version for character variables.

NPAR, under SAS75, provides the following nonparametric tests:

- Cochran Q test
- Median test
- Kruskal-Wallis one-way analysis of variance
- Friedman two-way analysis of variance
- Wilson two-way analysis of variance
- Wilcoxon signed-ranks test
- Mann-Whitney U test
- Kolmogorov-Smirnov tests
- Runs test
- Moses test of extremes
- Sign test
- Binomial test
- Walsh test
- Randomization test
- Three-way analysis of variance
- Coombs nonmetric scaling
- Spearman rank correlation
- Kendall rank correlation
- Kendall partial rank correlation
- Kendall coefficient of concordance
- Pearson product moment correlation
- Printed plot of cumulative distributions
- Kolmogorov confidence limits of cumulative distributions

The author wishes to acknowledge the work of the authors of both NPAR and SAS. In particular, the author is in debt to the NPAR system authors for providing eminently modifiable source code and the SAS authors for an exceptionable system on which to build and a good document for writing local procedures.

In addition, a debt to the authors of WYLBUR, an interactive text-editing and remote job entry system, is acknowledged. Most of the clerical tasks associated with the interfacing and source modification were performed using WYLBUR.