This file consists of Chapter 1 of **MacAnova User's Guide** by Gary W. Oehlert and Christopher Bingham, issued as Technical Report Number 617, School of Statistics, University of Minnesota, revised August 1998, describing Version 4.07 of MacAnova.

Fonts used in this manual are Palatino, Courier, and Symbol.

For information concerning MacAnova, write University of Minnesota, Department of Applied Statistics, 352 Classroom Office Building, 1994 Buford Avenue, St. Paul, MN 55108-6042.

1. Introduction

1.1 **What is MacAnova?** MacAnova is an interactive, programmable, multi-platform computer program for matrix manipulation and statistical analysis. It features powerful commands for the analysis of linear and generalized linear models, for multivariate analysis, and for the frequency analysis of time series. It is available for Apple Macintosh™ and IBM™ compatible computers and for computers running the Unix operating system. There are both DOS™ and Windows™ versions for IBM compatibles and a Motif version for Unix™.

This Users' Guide applies to all versions. See Appendices B, C, D, E and F for information specific to particular computer systems. This edition of the Users' Guide supersedes all previous editions. It describes many new features that were not available at the time the MacAnova 3.11 Users' Guide of March 1993 was prepared and some features not in the MacAnova 4.06 Users' Guide of April 1998.

MacAnova has:

- Commands for entering, modifying, combining, and selecting data
- Functions for standard statistical computations such as descriptive statistics, P-values, t-tests and power computations
- Commands for plotting data and results, including box plots and residuals plots
- A grammar for specifying linear models which includes the capability of transforming both the dependent and independent variables "on the fly"
- Functions for fitting linear and some generalized linear models to data
- Functions to compute fast Fourier transforms (FFT)
- Commands for clustering and factor rotation
- Matrix related operations, including inversion, linear equation solving, ordinary and relative eigenvalue/eigenvector computation, QR, Cholesky and singular value decompositions
- Conditional execution with syntax elements `if`, `else` and `elseif`
- Looping capability with syntax elements `for`, `while` and `break`
- A powerful macro capability that can be used to define new procedures.
- Ability to work conveniently with "ragged" (non-rectangular) data sets in the form of *structures*
- The ability to dynamically load and execute "user functions" – externally compiled C and Fortran routines
- On-line usage information and extensive help for all commands and most features
- Libraries of macros useful in frequency domain time series analysis and the design and analysis of experiments
- The ability to save a workspace containing current results and restore it later.

MacAnova is controlled primarily by typing commands and not by selecting items from menus. Its command syntax is modeled after S (Becker and Chambers 1984) but does not contain all the constructs in S. Sec. 11.4 summarizes some of the differences from S and its successor S-Plus™.

MacAnova was created in 1987 by Gary W. Oehlert of the Department of Applied

Statistics, University of Minnesota to provide for the design and analysis of experiments on Macintosh microcomputers in a classroom environment. It has been expanded and enhanced since 1991 by Christopher Bingham, with the goal of using it in connection with applied multivariate analysis and the frequency analysis of time series, as well as general statistical computing. Gary Oehlert programmed the interface for versions that run under Unix Motif, Microsoft Windows™ 3.1 with Win32S and Windows 95™.

1.2 **Comments on Version 4.07** MacAnova 3.0 was a major advance over all earlier versions, but MacAnova 4.07 represents only an evolutionary change from versions 3.0 and 3.1 and even less change from previous 4.0x versions. As compared to version 2.43, MacAnova 3.0 had dozens of new functions, enhancements to the grammar, including looping constructs, and a more powerful macro facility. Since then there have been many additional new functions and capabilities, but little has been obsoleted.

On the whole, MacAnova 4.07 is upwardly compatible with MacAnova 2.4x in the sense that most of what worked with MacAnova 2.4x also works with MacAnova 4.07. Incompatibilities include macro usage (arguments must not be quoted), pre-defined macro `readcols` (it has different arguments), command `print()` (arguments must not be quoted), and command `read()` (replaced by `vecread()`).

Similarly, with a few minor exceptions, whatever worked in Macanova 3.11 as described in March 1993 edition of this manual still works, but there are many new features. These include the extension of many operations to work with structures, the addition of new functions, the introduction of a syntax element `elseif`, and an increase in the number of variates and factors allowed in a linear model from 31 to 95. In addition, you can add coordinate labels to variables that are used in printing and modify the default labeling of coordinates. The enhanced capability of working with data in the form of a “structure” (See Sec. 2.8.16, 9.1) simplifies some analyses of “ragged” data sets. Some under-the-hood changes have resulted in much better memory management, resulting in up to a four fold increase in speed in macros and loops on some systems.

The following features were new in versions 4.05 and 4.06:

- Enhancement of GLM model syntax: you can use `{expr}`, where `expr` is an arbitrary MacAnova expression, in place of a variable or factor name. This allows, for example, doing quadratic regression by `regress("y=x+{x^2}")` (Sec. 3.4.1)
- New model abbreviations such as `"y=P3(x)"` and `"y=C2(x)"` in place of `"y=({x}+{(x)^2}+{(x)^3})"` and `"y=({cos(x)}+{sin(x)}+{cos(2*(x))}+{sin(2*(x))})"`, respectively (Sec. 3.4.3).
- New pre-defined macro `regcoefs` for determining coefficients together with their standard errors and t-statistics for regression models (models with no factors) (Sec. 3.13.1).
- Enhancements to `modelvar()` allowing determination of the number of variates and factors in a model (Sec. 3.24.1).
- New functions `rpoi()` and `rbin()` to generate Poisson and binomial pseudo-random variates (Sec. 2.13).

- New functions `cumdunnett()` (Sec. 2.12.7) and `invdunnett()` (Sec. 2.12.8) for computing *P* values and critical values for Dunnett's test for comparing several treatments with a control.
- New function `evaluate()` to directly execute its argument (for example, `evaluate("sqrt(2*PI)")`) (Sec. 9.5.1).
- Enhanced use of `<<...>>`, for example, `<<"sqrt(2*PI)">>` (Sec. 9.5.2).
- Functions `loadUser()` and `User()` to load and execute user functions (Sec. 9.7).
- New function `asLong()` creating a variable of new type `LONG` for passing integer arguments to user functions (Sec. 9.7.4).
- Enhanced `changestr()` to allow `changestr(Struct,name:x)` instead of `changestr(Struct,name,x)` (Sec. 9.1.3).
- Enhanced keyboard shortcuts on windowed versions.
- Graphics limitations in DOS extended memory version have been eliminated.

Among the more important enhancements in version 4.07 are the following:

- New syntax element `next` to skip to the end of a loop without terminating it (Sec. 9.2.5)
- You can attach descriptive "notes" to all variables, including graphs and macros (Sec. 8.9).
- New functions and a macro `appendnotes()`, `getnotes()`, `attachnotes()` and `hasnotes` for working with such notes (Sec. 8.9).
- New keyword `notes` on `vector()`, `matrix()`, `array()`, `macro()` and all the plotting commands (Sec. 8.9).
- `matprint()` and `matwrite()` now write labels and notes and `macrowrite()` writes notes (Sec. 8.9).
- `keyvalue()` has been enhanced to allow checking of shape and value as well as type of keyword macro arguments (Sec. 9.4.4).
- New function `argvalue()` can be used to check type, shape and value of non-keyword macro arguments (Sec. 9.4.6).
- New function `getascii()` translates `CHARACTER` variables into ASCII codes; essentially an inverse to `putascii(x,keep:T)` (Sec. 7.3.2).
- New keyword phrase `mantexp:T` on function `det()` directs that the result is in base10 mantissa and exponent format (Sec. 2.10.6).
- Keyword phrase `exact:F` on `match()` allows use of wildcard character "`*`" in string being matched (Sec. 9.4.1).
- More ways wildcard character "`*`" can be used in `help()`, `list()` and `match()`. Patterns such as `"ab*cd"` and `"*ab*cd*ef*"` are now legal (Sec. 2.8.9, 2.9.1, 9.4.1).
- New keyword phrase `history:T` or `F` on `save()` on `asciisave()` controls the saving and restoring of recently entered command lines (Sec. 7.7).
- New option `savehistry` provides default value for `history` on `asciisave()` and `save()` (Sec. 7.7 and 8.1.3).
- Keyword phrase `go:c` on `vecread()` directs that reading should stop with the first line that does *not* start with character `c` (Sec. 2.11.1).
- Keyword phrase `bychars:T` on `vecread()` is like `bylines:T` except that each element of the result is a single character (Sec. 7.2).
- A new format for files written by `save()` and `asciisave()` allows saving tick information for `GRAPH` variables as well as notes attached to variables.
- New command line flags `-p prompt` and `-bp batchPrompt` allow setting

command line prompt and prompt echoed when reading a batch file (Sec. 7.6, C.6.3). Not on Macintosh.

- New command line flags `-e expression` and `-eq expression` set first expression to be executed (Sec. C.6). Not on Macintosh.

1.3 **Conventions in this manual** All references to MacAnova variables and functions will be in `Courier` font, as will all sample input and output. In examples, what is typed by the user will be in *`italics`*. Most annotations to output will be in the form of command line comments preceded by #. Other annotations will be in **`bold face Courier font`**.

Since the Macintosh, Windows and Motif versions have many features in common that are unavailable on versions which do not take advantage of a windowing environment, they are referred to as "windowed versions."

1.4 **Disclaimer** Effort has been made to make MacAnova as error free as possible, but the program continues to evolve and and it is unlikely that no more bugs will be found. Neither the University of Minnesota, nor the Department of Applied Statistics, nor the authors of this program claim any responsibility for errors that may arise from this program. The University of Minnesota, the Department of Applied Statistics, and the authors make no warranties, either expressed or implied, regarding MacAnova or its fitness for any particular purpose.

1.5 **Acknowledgments** Development of MacAnova was supported in part by grants of Apple Macintosh™ computers by the Minnemac Project at the University of Minnesota.

Several colleagues, most notably Luke Tierney, Sanford Weisberg and Douglas Hawkins, have provided valuable suggestions and recommendations during the development of MacAnova.

The students in the Spring 1987 Statistics 5301 and 5163 classes and the Fall 1991 and several years of Statistics 5401 and 8511 classes survived (suffered through) developing versions of MacAnova and provided several useful suggestions.

We are indebted to Jan Kutylowski (jan.kutylowski@hibu.no) for many valued suggestions and tips where to find software tools and to Bernd Feige (charly@headmod.uni-muenster.de) for steering us toward the GNU readline software for editing command lines.

Special thanks are owed to Peter Fortini (Peter_Fortini@ST.cytec.com) who has been systematically searching for and finding bugs. Some recent changes in the handling of `NULL` variables and subscripts are the results of his suggestions.

Most Macintosh features are implemented using the TransSkel 3.12 environment, copyright © 1986 by Paul DuBois.

The Windows/Motif versions make use of the WxWin cross-platform windowing interface which is Copyright © 1995 Artificial Intelligence Applications Institute, The University of Edinburgh.

The syntax for specifying linear models is based in part on that used in GLIM (Aitken

et al. 1986).

Jay Andersen (supported by an EDP grant from the University of Minnesota) and Luke Tierney provided code for some of the cumulative and inverse distribution functions.

Sanford Weisberg provided a modification of the 1981 version of code for leaps and bounds regression supplied by George Furnival and based on Furnival and Wilson (1974).

The parser, written in `yacc` (a compiler compiler available in UNIX), evolved (a long way) from `hoc`, a desk calculator written also written in `yacc` (Kernighan and Pike 1984).

The Unix version and the extended memory DOS version (DJGPP) implement command line editing and recall of previous commands using the GNU Readline Library, Copyright © 1988, 1991 Free Software Foundation, Inc., distributed under the terms of the GNU public license.

The plotting routines use functions adapted from GNUplot, Copyright © 1986, 1987 Thomas Williams, Colin Kelley.

The Fast Fourier Transform algorithms were originally written in Fortran by Gordon Sande, then at the University of Chicago.

Code for stem and leaf displays was adapted from Fortran code in *ABCs of EDA* by David Hoaglin and Paul Velleman, 1981, Duxbury.

Code for hierarchical cluster analysis was adapted from Fortran program `hcl` of F. Murtagh, retrieved from Statlib. Code for k–means cluster analysis and varimax factor rotation was adapted from Fortran subroutines `trwcla` and `varmax` provided by Douglas Hawkins.

Code implementing a uniform pseudo-random number generator in P. L'Ecuyer (1988) was retrieved from netlib.

Code generating a pseudo-random Poisson variable was adapted from a Fortran program in Kemp and Kemp (1991).

Code generating a pseudo-random binomial variable was adapted from ACM Algorithm 678, Kachitvichyanukul and Schmeiser (1990).

Code to compute the cumulative distribution function and its inverse for the Studentized range is adapted from Algorithm AS 190 by R. E. Lund and J. R. Lund, *Appl. Statist.* **32** (1983, 1985), retrieved from statlib.

Code computing the cumulative distribution for Dunnett's *t* was adapted from Algorithm AS 251 by C. W. Dunnett (1989, 1993), and subroutine mvstud, also by Dunnett, that is part of the AS 251 distribution from statlib.

The eigenvalue and singular value decomposition functions are based on C adaptations of subroutines in the Eispack library (Smith *et al*, 1976).

Functions `cholesky()`, `qr()` and `solve()` use C adaptations of subroutines from the Linpack library (Dongarra et al, 1979).

1.6 **How to contact the authors** Our US mail and email addresses are

Gary W. Oehlert (gary@stat.umn.edu) and Christopher Bingham (kb@stat.umn.edu)
Department of Applied Statistics
University of Minnesota
352 Classroom Office Building
1994 Buford Avenue
St. Paul, MN 55108-6042

You can obtain the most recent release of any of Macintosh, DOS and Windows/Windows95 versions of MacAnova as well as the complete C and C++ source through the MacAnova home page with URL

http://www.stat.umn.edu/~gary/macanova/macanova.home.html.

A copy of this users' guide is also available through the MacAnova home page. See the Preface.