THREE MODIFICATIONS TO THE AARDVARK STATISTICAL PACKAGE IMPLEMENTED AT KANSAS STATE UNIVERSITY

524
1226-5600

by

KRISTOPHER LEE ARHEART

B. S., Kansas State University, 1970

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1973

Approved by:

Tom L. Gallagher
Major Professor

LD
2668
R4
1973
A7
c.2
Doc.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

AARDVARK[1] is a modular statistical package developed at Iowa State University to provide analysis of variance and covariance for a broad scope of statistical problems. In particular, AARDVARK distinguishes two basic types of statistical models.

1. Analysis of variance and covariance for models of the balanced complete structures class.

2. General multiple linear regression analysis models.

Balanced complete structure models are those having an equal number of observations for every combination of levels for each factor. AARDVARK provides an analysis of variance for cases in which some factors are nested in the levels of other factors or in combinations of other factors. Therefore, AARDVARK can analyze factors which are completely crossed, completely nested, or a combination of crossed and nested. Analysis of covariance adds the capability of adjusting each factor for covariates either jointly or separately at the user's discretion. An analysis of variance on the covariates is also offered.

AARDVARK is designed with user convenience as a goal. It provides, in one package, a set of routines that performs analysis of variance and covariance on various statistical models. AARDVARK allows problems to be formulated in terms that are consistent with the natural statistical representation of the problem. The user is offered options to control

[1] AARDVARK is an acronym standing for analysis of variance system, algebraic model options, residual and means options, data format options, variate and covariate analysis, analysis on means option, requested pooled terms and key statistical transformations.

the printing of certain statistics, the type of input data, and certain model conditions. A set of standard mathematical data transformations is also provided as well as the ability to include user prescribed transformations.

Since its implementation at Kansas State University in 1968, AARDVARK has undergone changes to correct errors and to provide better service for its many users. This report will discuss three recent modifications. First, the correction of the routine that provides F statistics for the analysis of variance of mixed or random effects models will be discussed. The next topic will be the implementation of a set of routines that selects the best subset of predictor variables for a linear regression analysis. Providing the user with clear, intelligible error diagnostics instead of the usual error numbers will be discussed last.

CHAPTER 2

CORRECTION OF THE ERRONEOUS F TEST DENOMINATORS AND COMPONENTS OF VARIANCE FOR TERMS IN A MIXED OR RANDOM EFFECTS MODEL ANALYSIS OF VARIANCE

The major objectives of an analysis of variance for a random effects model are to estimate the variance components and to test hypotheses about their magnitude by using an F test. To compute estimates of variance components, an expected mean square for the denominator of the F statistic must be obtained. This is also true for testing hypotheses about variance components. Therefore, the denominator of the F statistic is of prime importance in an analysis of variance with random effects. In AARDVARK, random and mixed models are handled similarly, so only random effects models will be discussed.

In a random effects analysis of variance model with two factors, there are four mean squares: the A, the B, the AB interaction, and the error. Figure 1 illustrates a two factor random effects analysis of variance table. The expected mean square of the error is composed of only the error variance. The expected mean square for the AB interaction contains the error variance and a coefficient times the AB variance component. The coefficients for the AB interaction variance component, the A factor variance component, and the B factor variance component are the respective number of observations on each effect. The A and the B expected mean squares are composed of the error variance, the AB interaction variance times its coefficient, n, and the A or the B variance times their respective coefficients, bn, and an. For the coefficients above, a is the number of levels of factor A, b is the number of

FIGURE 1.

ANALYSIS OF VARIANCE TABLE FOR A TWO FACTOR RANDOM EFFECTS MODEL

Source	Degrees of Freedom	Expected Mean Square	Component of Variance Estimate
A	a-1	$\sigma_E^2+n\sigma_{AB}^2+bn\sigma_A^2$	$(MS_A-MS_{AB})/bn$
B	b-1	$\sigma_E^2+n\sigma_{AB}^2+an\sigma_B^2$	$(MS_B-MS_{AB})/an$
AB	(a-1) (b-1)	$\sigma_E^2+n\sigma_{AB}^2$	$(MS_{AB}-MS_E)/n$
ERROR	ab(n-1)	σ_E^2	

Source	F Statistic	Numerator Degrees of Freedom	Denominator Degrees of Freedom
A	MS_A/MS_{AB}	a-1	(a-1) (b-1)
B	MS_B/MS_{AB}	b-1	(a-1) (b-1)
AB	MS_{AB}/MS_E	(a-1) (b-1)	ab(n-1)

levels of factor B, and n is the number of observations of each AB combination.

The three F tests in a two factor random effects analysis of variance test the hypotheses that the variance components for the A, the B, and the AB interaction are equal to zero,.respectively. The F statistic numerator is the mean square of the term being considered, while the denominator is constructed so that its expected value includes all the variance components of the numerator except the variance component being tested under the null hypothesis. Thus, the F statistic for the AB interaction is the AB interaction mean square divided by the error mean square. For the A and the B factors, the F statistic is the respective mean square divided by the AB interaction mean square. The degrees of freedom for the respective F tests are the degrees of freedom associated with the respective mean squares in the F statistic.

The variance components in a two factor random effects analysis of variance model are computed by subtracting the respective F test denominators from the numerator mean squares and dividing this difference by the coefficient of the variance component in the expected mean square. The coefficient has the same definition as described earlier. For the AB interaction variance component, the computation is the AB mean square minus the error mean square divided by the number of observations per cell. The variance component for the A or the B factor is computed by subtracting the AB interaction mean square from the A or the B mean square and dividing this difference by the respective coefficients.

In the case of a two factor random effects analysis of variance, the F tests are exact because the F statistic denominator is the mean square of some other source of variance in the model. In the case of a three to n factor random effects analysis of variance, however, an exact F test cannot be performed for certain effects because the appropriate F statistic denominator does not exist. However, approximate F tests are possible by using Satterthwaite's procedure, which forms an approximate F statistic denominator from a linear combination of appropriate mean squares.[1] A three factor random effects analysis of variance model will be described now to illustrate Satterthwaite's procedure.

Figure 2 shows an analysis of variance table for a three factor random effects model. The expected mean square for each term is composed of variances as in the two factor model. The F statistic denominator for the interaction terms are the mean squares of some other term, so exact F tests are possible. However, this is not the case for the three main effects. No single mean square contains the appropriate variances for an F statistic denominator of the main effects. Satterthwaite's procedure approximates an F statistic denominator by linearly combining appropriate mean squares. For example, the approximate F statistic denominator for the A factor is the AB interaction mean square plus the AC interaction mean square minus the ABC interaction mean square. The numerator and its corresponding degrees of freedom for the approximate F statistic are identical to those in a regular F statistic. Approximate denominator degrees of freedom are computed by squaring the denominator and dividing this square by the sum of the squared terms in the denominator, divided by their corresponding degrees of freedom. In the case of the A factor mentioned above, the approximate F test denominator degrees of freedom would be the denominator squared and divided by the AB interaction mean

[1] William C. Guenther, Analysis of Variance, (Englewood Cliffs, New Jersey: John Wiley and Sons, Incorporated, 1966), 130-135.

FIGURE 2.

ANALYSIS OF VARIANCE TABLE FOR A THREE FACTOR RANDOM EFFECTS MODEL

Source	Degrees of Freedom	Expected Mean Square	Component of Variance Estimate
A	a-1	$\sigma_E^2+n\sigma_{ABC}^2+cn\sigma_{AB}^2+bn\sigma_{AC}^2+bcn\sigma_A^2$	$[MS_A-(MS_{AB}+MS_{AC}-MS_{ABC})]/bcn$
B	b-1	$\sigma_E^2+n\sigma_{ABC}^2+cn\sigma_{AB}^2+an\sigma_{BC}^2+acn\sigma_B^2$	$[MS_B-(MS_{AB}+MS_{BC}-MS_{ABC})]/acn$
C	c-1	$\sigma_E^2+n\sigma_{ABC}^2+bn\sigma_{AC}^2+an\sigma_{BC}^2+abn\sigma_C^2$	$[MS_C-(MS_{AC}+MS_{AC}-MS_{ABC})]/abn$
AB	(a-1) (b-1)	$\sigma_E^2+n\sigma_{ABC}^2+cn\sigma_{AB}^2$	$(MS_{AB}-MS_{ABC})/cn$
AC	(a-1) (c-1)	$\sigma_E^2+n\sigma_{ABC}^2+bn\sigma_{AC}^2$	$(MS_{AC}-MS_{ABC})/bn$
BC	(b-1) (c-1)	$\sigma_E^2+n\sigma_{ABC}^2+an\sigma_{BC}^2$	$(MS_{BC}-MS_{ABC})an$
ABC	(a-1) (b-1) (c-1)	$\sigma_E^2+n\sigma_{ABC}^2$	$(MS_{ABC}-MS_E)/n$
ERROR	abc(n-1)	σ_E^2	

FIGURE 2.--CONTINUED

Source	F Statistic	Numerator Degrees of Freedom	Denominator Degrees of Freedom
A	$\frac{MS_A}{MS_{AB}+MS_{AC}-MS_{ABC}}$	a-1	$\frac{(MS_{AB}+MS_{AC}-MS_{ABC})^2}{\frac{MS_{AB}^2}{(a-1)(b-1)}+\frac{MS_{AC}^2}{(a-1)(c-1)}+\frac{MS_{ABC}^2}{(a-1)(b-1)(c-1)}}$
B	$\frac{MS_B}{MS_{AB}+MS_{BC}-MS_{ABC}}$	b-1	$\frac{(MS_{AB}+MS_{BC}-MS_{ABC})^2}{\frac{MS_{AB}^2}{(a-1)(b-1)}+\frac{MS_{BC}^2}{(b-1)(c-1)}+\frac{MS_{ABC}^2}{(a-1)(b-1)(c-1)}}$
C	$\frac{MS_C}{MS_{AC}+MS_{BC}-MS_{ABC}}$	c-1	$\frac{(MS_{AC}+MS_{BC}-MS_{ABC})^2}{\frac{MS_{AC}^2}{(a-1)(c-1)}+\frac{MS_{BC}^2}{(b-1)(c-1)}+\frac{MS_{ABC}^2}{(a-1)(b-1)(c-1)}}$
AB	MS_{AB}/MS_{ABC}	(a-1) (b-1)	(a-1) (b-1) (c-1)
AC	MS_{AC}/MS_{ABC}	(a-1) (c-1)	(a-1) (b-1) (c-1)
BC	MS_{BC}/MS_{ABC}	(b-1) (c-1)	(a-1) (b-1) (c-1)
ABC	MS_{ABC}/MS_E	(a-1) (b-1) (c-1)	abc(n-1)

square squared divided by its degrees of freedom, plus the AC interaction mean square squared divided by its degrees of freedom, plus the ABC interaction mean square squared divided by its degrees of freedom. The variance components are still computed as for the two factor model using the approximate F statistic denominators.

In AARDVARK the subroutine VCAFT consists of 258 FORTRAN IV source statements. It computes the F statistic denominators and the variance components and then prints out the analysis of variance table. VCAFT is based on an algorithm provided by J. E. Schlater and W. J. Hemmerle.[1] This algorithm implements Satterthwaite's procedure to obtain the F statistic denominators and variance components in all cases, even when exact tests exist.

When VCAFT is called, it is passed three vectors and a parameter. The first vector, QMNSQ, contains the mean square for each effect of the model. The elements of the second vector, LMEFT, denote the presence or absence of each factor in the model for each effect. Each factor is assigned a specific bit position in the one-word elements of LMEFT. The bit position for a factor is determined by the order in which it appears in the model. The first factor is represented by the left-most bit, the second factor by the next bit to the right, and so forth. The presence of a certain factor in an effect is indicated by turning its corresponding bit on; the absence is shown by leaving the bit turned off. The third vector, LMSUB, is constructed in the same manner as LMEFT for the subscripts in a term. The parameter, IFOR, denotes which factors are random.

[1] J. E. Schlater and W. J. Hemmerle, "Statistical Computations Based on Algebraically Specified Models," Communications of the ACM, 9, 12, (December, 1966), 130-135.

The bit position for each factor is assigned the same position as in LMEFT. An "on" bit indicates that its corresponding factor is random; and an "off" bit shows a fixed factor.

VCAFT first forms a numerical value for each effect which denotes its subscript composition. The LMSUB vector contains the subscript structure for each effect as mentioned earlier. If the left-most bit of LMSUB is on, the numerical value becomes one, denoting the presence of the first subscript in the model. A two is added if the bit in the next position to the right is on. This process continues adding correspondingly higher powers of two to the value for each "on" bit in the manner described above, a new vector, IBVAL, has been created. Each element of IBVAL is a numerical value representing the subscript composition of each effect in the model.

The elements of IBVAL are next sorted into ascending order. At the same time, a vector, LORD, is formed. The elements of LORD are pointers that maintain a correspondence between the sorted values of IBVAL and the elements of QMNSQ, LMEFT, and LMSUB. The elements of LORD are used as subscripts when the QMNSQ, LMSUB, and LMEFT arrays are referenced, giving them the same order defined for the elements of IBVAL. The ordering defined by LORD insures that the mean squares needed to compute an F statistic denominator for a specific effect will be found only in effects ranked below it.

The F statistic denominator and the variance components are calculated for each effect according to the order of LORD. A control word, ICONT, is constructed by logically oring IFOR and the LMEFT element for the effect under consideration. A comparison is then made between the LMSUB element for the considered effect and the LMSUB element of the effect directly below it according to the order of LORD. If all the subscripts in the lower effect are present in the considered effect, another comparison is made between ICONT and the LMEFT element for the lower ranking effect. If there is a one to one correspondence between the bits of ICONT and LMEFT, the lower effect and its coefficient are a part of the F statistic denominator of the considered effect. When an effect is part of the F statistic denominator of the considered effect, its variance component, which has not yet been divided by its coefficient, is subtracted from the variance component of the considered effect and is added to the F statistic denominator. Note that by using the order defined in the LORD vector, the variance components of lower ranking effects are available for computing the F statistic denominator for a higher ranking effect.

The F statistic for each effect is computed by dividing the mean square of the effect by the F statistic denominator which was computed as described in the paragraph above. The variance component is computed for each effect by subtracting the denominator mean square from the numerator mean square and then dividing this difference by the variance component's coefficient. After these computations are made, the analysis of variance table is printed.

VCAFT optionally produces a table containing the F statistic denominators and their degrees of freedom for each effect. If the option is taken, the coefficients of the variance components are stored in a lower-triangular array, IEMS, as they are computed. This lower-triangular array represents a set of simultaneous equations which are solved by using a backward solution method.[1] The solution vector of the lower-triangular array is composed of the degrees of freedom for each effect. The solution vector is then printed.

To determine the reason why VCAFT produced incorrect F statistic denominators and variance components, it was necessary to learn how to compute these statistics for mixed and random effects models. Next, a small amount of data was traced through Schlater and Hemmerle's algorithm by hand to become familiar with its design. Then the intricate coding of VCAFT was studied to determine ways in which it paralleled the algorithm. The IBM DEBUG package was used next with the INIT option used to display the changes of value for pertinent variables. By comparing the values displayed during the debug procedure with those computed by hand, the error was discovered. The LORD order vector was not being used consistently when QMNSQ, LMSUB, and LMEFT were referenced. This resulted in using the wrong mean squares to compute the F statistic denominators and variance components. By revising 13 statements to use the LORD vector as intended, the error was remedied.

[1] N. R. Draper and H. Smith, Applied Regression Analysis, (New York: John Wiley and Sons, Incorporated, 1966), 167-169.

CHAPTER 3

IMPLEMENTING SELECT

In linear regression analyses, researchers often want to use the best subset of the independent variables to estimate the dependent variable. There are a number of criteria defining the "best" subset. The most common ones are based on functions of the residual mean squares resulting from a regression analysis. Smaller residual mean squares indicate a greater multiple correlation and a larger regression mean square. A larger multiple correlation coefficient denotes a better regression equation. In the following discussion, a "best" subset is one which yields the largest regression mean square.

The problem of choosing a subset of independent variables lies in the possible number of different subsets, which in many cases may be extremely large. In a regression analysis with ten independent variables, $2^{10}-1$ subsets are possible. Computing a regression for all 1,023 subsets is costly in terms of money and man-hours. Even if all subsets could be evaluated economically, most of them would not fit the criterion of a best subset.

L. R. La Motte and R. R. Hocking discuss a set of routines called SELECT that aid the researcher in choosing the best subset of independent variables for a linear regression analysis.[1] The algorithm upon which SELECT is based chooses the best subset and several nearly best subsets

[1] L. R. La Motte and R. R. Hocking, "Computational Efficiency in the Selection of Regression Variables," Technometrics, 12, (February, 1970), 83-93.

of a specified size or a number of different sizes.

The SELECT algorithm first computes a regression using all the independent variables. Measures of the reduction in the regression mean square resulting from deleting each independent variable from the equation are then computed. The reduction resulting from each independent variable alone is called the univariate reduction. The independent variables are then ranked in ascending order according to their respective univariate reductions. This ordered set of independent variables is divided into subsets whose sizes are the complement of the subset size specified by the user. Each subset of the complement size is searched to find the one set that yields the lowest reduction in the regression mean square. When this subset is found, its elements are deleted from the regression equation leaving the best subset of the specified size. The independent variables in the best subset contribute the most to the size of the regression mean square and make the residual mean square the smallest.

Nearly best subsets are computed in the same manner as the best subset. They are subsets of the same size as the best subset. The reduction of the regression mean square of the nearly best subsets is close to that of the best subsets within a tolerance margin.

SELECT was implemented in AARDVARK to provide the researcher with a method of selecting the best subset of independent variables for a linear regression analysis without having to analyze every subset.[1] The user specifies his regression analysis using the regular AARDVARK control cards. In addition, he includes another AARDVARK control card specifying the

[1] The original SELECT routines were provided to the Kansas State University Statistical Laboratory by R. Hocking of Texas A & M University and then modified as described by the author.

the minimum and maximum subset sizes. The control card is interpreted with the others in the SCAN1 routine in AARDVARK. SCAN1 places the subset sizes in a common block called ISEL.

After interpreting the control cards, control is passed to the REGMP routine where the regression analysis would be performed if SELECT were not called. A non-zero value for the maximum subset size indicates to REGMP that SELECT is to be used. REGMP forms a vector containing the sum of squares for each independent and dependent variable and another vector containing the correlations for all the variables, both dependent and independent. Then control is passed to SELDRV.

SELDRV is the driver routine for SELECT. SELDRV begins by reading one of the hypothesis cards stored on a work unit by SCAN1. The hypothesis card specifies the dependent and independent variables for a linear regression analysis. One regression equation will be computed for each dependent variable on the hypothesis card. A vector containing the sums of squares and a matrix containing the correlations for the independent variables specified on the hypothesis card are formed first. Then, for each dependent variable on the hypothesis card, a vector of correlations for the dependent variable and independent variables is formed. The sum of squares for the dependent variable is placed in the vector of sums of squares formed above, and then SELECT is called. The operations performed by SELDRV are repeated for each hypothesis card.

SELECT processes each subset of the specified sizes to find the best and nearly best subsets for the regression equation using the algorithm by La Motte and Hocking. The output from SELECT begins with the regression table for all the independent variables. This regression table includes the multiple correlation coefficient squared and the unrescaled and the rescaled regression coefficients. Next, regression tables are printed for each best and nearly best subset size specified. For each best subset, the reduction in the regression mean square resulting form eliminating the other variables is given.

Besides writing the fifty source statement FORTRAN IV routine SELDRV and the ten statement interface in REGMP, the greatest problem in implementing SELECT in AARDVARK was the huge amount of core space SELECT required. SELECT must be able to handle regression equations with 64 independent variables. This caused the original SELECT routines, consisting of 972 FORTRAN IV source statements in ten routines, to require approximately 250K bytes of core. By carefully studying the use of arguments and arrays by the ten different routines comprising SELECT, it was found that by using COMMON statements and by combining two routines the amount of core required was reduced by approximately 27K bytes. SELECT still requires so much core, that AARDVARK must be run in a 512K partition, even with an overlay structure. One possible solution to the problem of space is to segment the SELECT routines into a larger number of smaller routines that can be overlayed into a smaller amount of core.

CHAPTER 4

PROVIDING DETAILED EXECUTION ERROR MESSAGES FOR AARDVARK

When a FORTRAN IV program encounters an error condition, a call is made to the FORTRAN IV error monitor. The error monitor prints out an error message listing the error number, the name of the IBM FORTRAN IV routine in which the error occurred, and a brief statement as to the cause of the error. In addition to the error message, a traceback may also be given.

The traceback feature gives the user a table indicating the name of the user routines called in order to reach the routine containing the error. The contents of registers 0, 1, 14, and 15 are also given. At this time, the numbers indicating the statement that called the next routine in the table are incorrect because of an error in the IBM ERRTRA routine that prints the traceback. However, the list of user routine names does give the user the name of his routine where the error occurred.

The standard FORTRAN IV execution error messages are inadequate for most AARDVARK users who are researchers having had little or no interface with the computer. Therefore, when the user gets an error message, he must decipher the message himself by consulting the IBM System/360 Operating System FORTRAN IV (G and H) Programmer's Guide. For someone with little knowledge of the FORTRAN IV language, this often can be an involved, fruitless task. If the user cannot understand these manuals, he must then consult with someone who has had experience with

FORTRAN IV, thereby spending not only his time but also that of someone else.

To provide better service to the general user, new, more detailed error messages were implemented in AARDVARK for errors arising from illegal data in arithmetic computations, incorrectly keypunched data, improper data formatting, and incorrect observation counts. The new error messages contain the cause of the error, the results of the standard error correction provided by the error monitor, and the error number. A traceback is also provided after each error. A detailed list of these new error messages is given in Appendix A.

New error messages were implemented in AARDVARK by using the FORTRAN IV supplied subroutine ERRSET. ERRSET gives the programmer control of the error diagnostic routine provided by the extended error handling facilities generated in FORTRAN IV during systems generation. The controls initiated by calling ERRSET are effective for a specified error during the execution of a program containing the call to ERRSET.

The calling parameters to ERRSET are IERNO, INOAL, INOMES, ITRACE, IUSADR, and IRANGE. IERNO is an integer representing the number of the error which is to be modified by the call to ERRSET. INOAL is an integer specifying the number of errors that can occur before the execution of the program is terminated. If INOAL is zero or negative, the parameter does not change the system default value; if it is greater than 255, an unlimited number of errors is permitted. The number 255 is the upper limit since one byte is used to store INOAL, and the maximum value a one-byte number can contain is 255. The number of standard error messages to be printed is specified by the parameter INOMES. If INOMES is a negative value, no standard error messages will be printed. A value of zero for INOMES leaves the system default value unchanged. By specifying a positive integer, the number of system error messages is limited to the number set by the user. ITRACE controls the printing of a traceback for each occurrance of an error. If a value of zero is specified for ITRACE, the system default value is unchanged. A value of one for ITRACE suppresses the traceback. When ITRACE is given a value of two, a traceback will be printed for each occurrence of the error. The routine to be used in correcting an error is specified by the value of IUSADR. If IUSADR is deleted from the argument list of the call to ERRSET or set to zero, the standard system error correction routine will be used. When a user wants to provide his own routine to correct an error, the name of the user routine to be used is used for IUSADR. In addition, the name of the user routine must be listed in a FORTRAN IV EXTERNAL statement. The IRANGE parameter is used in conjunction with the parameter IERNO to specify a list of errors to be affected by the call to ERRSET. If IRANGE is not included in the argument list of the call to ERRSET, only the error denoted by the value of IERNO will by handled by the call to ERRSET. When IRANGE is included, the consecutive errors, beginning with IERNO and including IRANGE, will be controlled by the call to ERRSET.

In AARDVARK, new, more detailed error messages were implemented by using the ERRSET routine in conjunction with five new user routines to handle error corrections. There are five calls to ERRSET in the main routine of AARDVARK, each specifying one of the new error handling routines. Four of these new routines handle errors resulting from arithmetic operations performed in AARDVARK or user-supplied data transformation routines. The other routine handles data conversion errors arising from invalid input data characters.

For all errors except error 215, illegal decimal character, INOAL is set to 256 indicating an unlimited number of errors will be permitted. For error 215, 50 errors are allowed before the program is terminated. All calls to ERRSET have INOMES specified as -1 to suppress the printing of standard error messages since the new error messages will be printed. A traceback is requested for all errors except error 215. The first call to ERRSET specifies the range of error numbers to be 207 through 209, exponent overflow, exponent underflow, and division by zero respectively, and the user routine to be ER207. Error 215 is controlled by the second call to ERRSET. The user routine for error 215 is ER215. The range of errors from 251 to 254, negative square root argument, exponential argument greater than 174.673, natural or common logarithm argument less than or equal to zero, and sine or cosine argument greater than or equal to $2\pi^{18}$ respectively, are specified by the third call to ERRSET which also specifies a user routine named ER251. ER255 is the user routine name specified for error 255, zero arctangent argument, in the fourth call to ERRSET. The range of errors 256 to 259, hyperbolic sine or cosine argument greater than or equal to 175.366, arcsine or arccosine argument greater than one, tangent or cotangent argument greater than or equal to $2\pi^{18}$, and tangent or cotangent argument approaching singularity respectively, is controlled by the fifth call to ERRSET. The user routine ER 256 is assigned to handle the range of errors 256 through 259. All user routine names are specified in an EXTERNAL statement.

All the user error routines, with the exception of ER215, are FORTRAN IV routine structured in the same manner. ER215 is a basic assembler routine. The entire set of routines consists of 262 FORTRAN IV statements and 237 BAL statements plus other interface statements in four of the AARDVARK routines. The calling parameters are IRETCD, IERNO, and a list of data values. If IRETCD is set to one, the standard error correction takes place. All the standard error corrections are noted in the error messages themselves, as shown in Appendix A. IERNO is the number of the error passed to the error handling routine to correct the error. The number and type of elements in the data list depend on the error being handled and determine the errors that are grouped together in the error handling routines written for AARDVARK. Since the standard error correction is used, these data values are not changed in any of the routines.

The error handling routines, with the exception of ER215, begin by transferring to the appropriate section of code to handle the particular error. If an invalid error number has been passed to the routine, the program execution is terminated after an error message is issued. Within each segment, the number of error messages allowed is checked. If the limit has been reached, no error message is printed. The number of messages allowed for each error is defined by an array called MSGNOS which is initialized in the BLOCK DATA subroutine. The current count of error messages printed is kept in array NMSNOS whose elements are set to zero in routine PREP each time a new set of data is processed. The number of tracebacks is checked after the error message is printed. Once the limit for the number of tracebacks has been reached, no more tracebacks are printed. The traceback limits for each error are kept in an array TRCNOS which is defined in BLOCK DATA. The current traceback count is kept in the array NTRNOS whose elements are set to zero by routine PREP each time a new set of data is read in. Control of the number of error messages printed is provided in order to simulate the standard error handling facility in the error monitor. The number of tracebacks permitted has been added for convenience. A logical flag called BYP is also turned on by the error handling routines. BYP is checked in the main routine after all the data has been input, but before the statistical analyses are performed. If an error has occurred, BYP is "on" and the data analysis is bypassed. After an error, if any other data sets are in the job step, they are subsequently input; otherwise, processing stops.

ER215 is an assembler program that handles data conversion errors.[1] When invalid data, such as character data or symbols other than a minus or a period are detected, ERRSET calls ER215. In ER215 the position of the erroneous data is ascertained and inspected. If the data is an "END OF DATA" card, a one is returned in the parameter IFLAG. The routine INPUT1, which does all the data reading for AARDVARK, handles the "END OF DATA" signal. If the data is invalid, a -1 is returned in IFLAG. Then control is returned to INPUT1 which calls routine BOMB. BOMB is a FORTRAN IV program that uses a pointer ICDCOL returned by ER215 to locate the offending character and the parameter IBUFNO to locate the buffer containing the record in error. BOMB prints the erroneous data record and places a "$" under the invalid character. On returning from BOMB, INPUT1 reads through the rest of the data set, and, if another data set is present, begins processing it. If no more data and parameters are present, the program is terminated. As long as

[1] ER215 and BOMB were supplied by Dr. Kenneth E. Kemp of the Kansas State University Statistical Laboratory.

IFLAG has the value zero, the data has no errors and is processed normally.

With the addition of the error handling routines, AARDVARK renders better service to the user. The errors caused by data in arithmetic routines are diagnosed by non-computer oriented users more easily because of the detailed error messages. Invalid data characters are handled better, too, because the invalid character is noted clearly.

CHAPTER 5

CONCLUSION

The three modifications to AARDVARK discussed in this paper have provided a more usable statistical package. The first modification, correcting the erroneous F statistic denominator, improved AARDVARK; not only are correct F statistics given now, but correct variance components and F test degrees of freedom are also available. AARDVARK modification number two was the addition of the SELECT routines. With SELECT, the user now has a tool to choose the best subset of independent variables for a linear regression analysis. By adding detailed error messages to AARDVARK in modification number three, the user is provided with more intelligible diagnostics in case he causes an error.

The major problem remaining with AARDVARK is its huge core space requirement. By adding the SELECT routines, the core requirement increased by approximately 23K. The inclusion of detailed error messages added approximately 8K more. AARDVARK now demands approximately 412K, even with an overlay structure. This is a problem which should be resolved in order to save core, decreasing the cost to the user and decreasing turn around time. Listings of the routine discussed in the paper are available at the Statistical Laboratory, Room 2, Calvin Hall, Kansas State University.

BIBLIOGRAPHY

AARDVARK Reference Manual. Revised by A. G. Mexas. Ames, Iowa: Statistical Laboratory, Iowa State University, 1968.

Draper, N. R. and Smith, H. *Applied Regression Analysis*. New York: John Wiley and Sons, Incorporated, 1966.

Guenther, William C. *Analysis of Variance*. Englewood Cliffs, New Jersey: Prentice-Hall, Incorporated, 1964.

International Business Machines Corporation. *IBM System/360 Operating System: Messages and Codes*. New York: International Business Machines Corporation, 1970.

International Business Machines Corporation. *IBM System/360 Operating System FORTRAN IV (G and H) Programmer's Guide*. New York: International Business Machines Corporation, 1970.

La Motte, L. R. and Hocking, R. R. "Computational Efficiency in the Selection of Regression Variables." *Technometrics*, 12, (February, 1970), 83-93.

Scheffe, Henry. *The Analysis of Variance*. New York: John Wiley and Sons, Incorporated, 1958.

Schlater, J. E. and Hemmerle, W. J. "Statistical Computations Based on Algebraically Specified Models." *Communications of the ACM*, 9, 12, (December, 1966), 865-869.

Statistical Laboratory, Kansas State University. "Suggested Work to Be Done on AARDVARK." Request for programming services to be provided by the Kansas State University Computing Center to the Statistical Laboratory. Manhattan, Kansas, 1972.

APPENDIX A

DETAILED EXECUTION ERROR MESSAGES

Resulting number too large for machine to store. Resulting number set to approximately .7E76. (IBM error IHC207I.)

Resulting number too small for machine to store. Resulting number set to zero. (IBM error IHC208I.)

Division by zero attempted. Resulting number is set to dividend. (IBM error IHC209I)

Square root of a negative number attempted. Resulting number is square root of absolute value of number. (IBM error IHC251I.)

Exponentiation of a number larger than 174.673 attempted. Resulting number is approximately .7E76. (IBM error IHC252I.)

Natural or common logarithm of a zero or a negative number attempted. Resulting number is set to approximately -.7E76 for a zero or to the corresponding logarithm of the absolute value of the number. (IBM error IHC253I.)

Sine or cosine of a number whose absolute value is greater than or equal 2E18 X PI attempted. Resulting number is set to approximately .707. (IBM error IHC254I.)

Arctangent of a number equal to zero attempted. Resulting number is set to zero. (IBM error IHC255I.)

Hyperbolic sine or cosine of a number whose absolute value is greater than 175.352 attempted. Resulting number is set to approximately .7E76. (IBM error IHC256I.)

Arcsine or arccosine of a number whose absolute value is greater than one. Resulting number is set to zero. (IBM error IHC257I.)

Tangent or cotangent of a number whose absolute value is greater than 2E18 X PI attempted. Resulting number is set to one. (IBM error IHC258I.)

Tangent of a number too close to an odd multiple of PI / 2 attempted, or cotangent of a number that is too close to a multiple of PI attempted. Resulting number is set to approximately .7E76. (IBM error IHC259I.)

ACKNOWLEDGMENTS

I wish to thank the three members of my advisory committee: Dr. Tom L. Gallagher, chairman; Dr. Kenneth Conrow; and Dr. Kenneth E. Kemp. A special note of appreciation is given to Dr. Kemp for providing the basis of this report and for giving freely of his time. I would also like to thank my wife, Linda, and my family for their encouragement and help.

THREE MODIFICATIONS TO THE AARDVARK STATISTICAL PACKAGE IMPLEMENTED AT KANSAS STATE UNIVERSITY

by

KRISTOPHER LEE ARHEART

B. S., Kansas State University, 1970

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1973

ABSTRACT

This paper presents three modifications to the AARDVARK statistical package implemented at Kansas State University. The first change discussed is the correction of the VCAFT routine so that it would produce correct F statistic denominators and estimates of the variance components. The addition of the SELECT routines that aid the researcher in choosing the best subset of independent variables in a linear regression analysis is the second modification. The third and last alteration of AARDVARK is the incorporation of more detailed execution error messages.