

AD-A278 678



A COMPARISON OF VARIABLE SELECTION
CRITERIA FOR MULTIPLE LINEAR REGRE-
SSION: A THIRD SIMULATION STUDY

THESIS

Ertem MUTLU, 1st Lieutenant, TUAF
AFIT/GOR/ENS/94M-08

DTIC
ELECTE
APR 22 1994

S B D

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

DTIC QUALITY INSPECTED 3

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/94M-08

A COMPARISON OF VARIABLE SELECTION
CRITERIA FOR MULTIPLE LINEAR REGRE-
SSION: A THIRD SIMULATION STUDY

THESIS

Ertem MUTLU, 1st Lieutenant, TUAF
AFIT/GOR/ENS/94M-08

Approved for public release; distribution unlimited

94-12285



94 4 21 066

AFIT/GOR/ENS/94M-08

A COMPARISON OF VARIABLE SELECTION CRITERIA FOR
MULTIPLE LINEAR REGRESSION: A THIRD SIMULATION STUDY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirement for the Degree of
Master of Science in Operations Research

Ertem MUTLU, B.S.
1st Lieutenant TUAF

March, 1994

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited

THESIS APPROVAL

STUDENT: 1st Lieutenant Ertem Mutlu

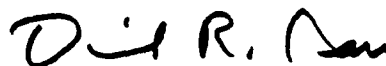
CLASS: GOR 94M

THESIS TITLE: A COMPARISON OF VARIABLE SELECTION CRITERIA
FOR MULTIPLE LINEAR REGRESSION: A THIRD SIMULATION STUDY

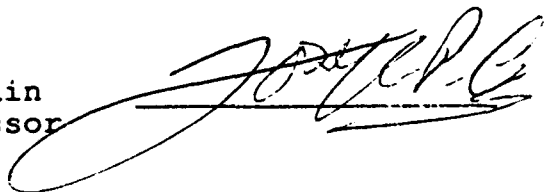
DEFENSE DATE: 04 MAR 94

COMMITTEE	NAME/DEPARTMENT	SIGNATURE
-----------	-----------------	-----------

Advisor	Dr. David R. Barr Associate Professor ENC
---------	---



ENS Rep.	Dr. Joseph P. Cain Associate Professor ENS
----------	--



Acknowledgement

I would like to express my appreciation to Dr. Barr for the support he has given me to accomplish this research project.

I would like to thank my friend John Camp for helping me in editing this document.

I also would like to thank my wife Nilgun and daughter Niler for their understanding and support.

Ertem Mutlu

Table of Contents

	Page
Acknowledgements	ii
List of Figures	vi
List of Tables	vii
Abstract	viii
I. Introduction	1
Background	1
Problem Statement	3
Assumptions	3
Scope	4
II. Concept Overview	5
Least Squares Regression	5
Assumptions	5
III. Literature Review	8
Regression	8
Subset Selection	9
All-Subsets Regression	11
Maximum Coefficient of Multiple Determination R^2	13
Maximum Adjusted R^2 or Minimum MSE	13
Minimum Mallows C_p	14
Minimum PRESS _p or Minimum Sp Criteria ...	15
Stepwise Procedures	16
Backward Elimination (BE)	16
Forward Selection Procedures (FS)	17
Stepwise Regression Method	17
Miller's Method	18
IV. Methodology and Model Development	20
Objective	20
Limitations	20
Overview	21
Overview of Data Generation Process	23
Factors	24
Data Generation	28
Data Sets	29
Performance Measurements	29
Performance Measurement for Real Predictors ..	30

Performance Measurement for the Full Model ...	30
Results of previous Simulation Studies	34
Performance Measure (PM) for the Percentage of Correct Variables	34
Performance Measure of Model Accuracy	37
New Subset Selection Methods Studied in This Research	39
Two Stage Subset Selection Process, Where Miller's Method is Used for Screening	40
Two Stage Subset Selection Process, Where Miller's Method is Used for Final Model Selection	40
Modified-Miller's Method (MM Method)	41
Number of Known Extraneous Variables (augmentation coefficient)	43
Run Numbers	49
Finding the Best Model	52
Null Model	54
V. Experimental Design and the Results	58
Experiment	58
PM Equations	60
Results for PM Measurement	63
FM Equations	64
Results of FM Measurement	68
Analysis Based on Mean and Standard Deviation of PM and FM Values	69
IV. Conclusion and Recommendation for Further Research .	77
Conclusion	77
Objective	77
Techniques Studied	78
Methodology	78
Recommendations for Further Research	82
Appendix A: Flow Chart for Two Stage Subset Selection, Where Miller's Method is Used for Screening (1 extraneous)	85
Appendix B: Flow Chart for Two Stage Subset Selection, Where Miller's Method is Used for Screening (3 extraneous)	86
Appendix C: Flow Chart for Two Stage Subset Selection, Where Minimum MSE is Used for Screening and Miller's Method is Used for Final Model Selection	

Appendix D:	Flow Chart for Two Stage Subset Selection, Where Minimum S_p is Used for Screening and Miller's Method is Used for Final Model Selection	88
Appendix E:	Flow Chart for Two Stage Subset Selection, Where Minimum C_p is Used for Screening and Miller's Method is Used for Final Model Selection	89
Appendix F:	Flow Chart for MM Method for Sample Size 10 and 1 extraneous	90
Appendix G:	Flow Chart for MM Method for Sample Size 10 and 3 extraneous	91
Appendix H:	Flow Chart for MM Method for Sample Size 20 and 1 extraneous	92
Appendix I:	Flow Chart for MM Method for Sample Size 20 and 3 extraneous	93
Appendix J:	Combining Different Files and Finding PM and FM values for All Design Points	94
Appendix K:	Flow Chart for Miller's Method with augmentation coefficient 0.5	90
Appendix L:	FORTTRAN Programs	96
Appendix M:	SAS Programs	159
Bibliography		172
Vita		174

List of Figures

Figure	Page
2-1. Two-dimensional Representation of Linear Least Squares Regression	7
4-1. The PM Values of Table 4-4 Column 5	46
4-2. The FM Values of Table 4_5 Column 5	48
4-3. Comparison of Empirical Null Value and the Theoretical Null Value.....	56
5-1. Analysis of One Stage Subset Selection Methods According to Their PM and FM Mean Values and Standard Deviations	72
5-2. Analysis of Two Stage Subset Selection Methods where, Miller's Method is Used for Screening	73
5-3. Analysis of Two Stage Subset Selection Methods where, Miller's Method is Used for Final Model Selection	74
5-4 Search for the Best of the Best Subset Selection Methods	75

List of Tables

Table	Page
4-1. Summery Table for the Factors Used in Data Generation	26
4-2. Order of Factors According to the Design Point ...	27
4-3. Main Factor Coefficients of Effects by Method for PM measurement	36
4-4. The PM Values of Models Obtained by Using Different Augmentation Coefficients	45
4-5. The FM Values for Models Obtained for Different Augmentation Coefficients	47
4-6. Models for Design Point 14 and Data Set 31	53
5-1. Variable Coding for Response Surface Methodology	59
5-2. Summery Table of the Effects of Main Factors for PM Measurement	62
5-3. Summery Table of FM Values of the Effects of the Main Factors	67
5-4. The Mean and Standard Deviation of PM and FM value of the Employed Subset Selection Technique	70
6-1. Comparison of Models Obtained by 60 and by 6 runs	81

Abstract

The goal of this thesis research is to introduce and study a modification of Miller's Method which we call the Modified-Miller's method (MM method), study two step subset selection procedures, these of which apply Miller's Method in the first step and employs another method (Minimum MSE or Minimum Sp or Minimum Cp) in the second step and these of which employ another method (Minimum MSE or Minimum Sp or Minimum Cp) in the first step and applies Miller's Method in the second step. The results of all techniques will be compared including the results of the previous simulation studies done by Hanson in 1988 and by Woollard in 1993. In the researches mentioned above, Minimum MSE, Minimum Sp, Minimum Cp and Miller's method were studied.

Each of the techniques, studied in this thesis effort was applied on generated data with known multicollinearities, random predictors, variances, sample sizes and error terms.

The performance of each method is evaluated by employing two performance measurements. The first one, denoted PM (Performance Measurement), calculates the percentage of correct variables in a model. The second one, denoted FM

(Full Model) calculates the ratio of perfect models, the set of which is {X1 X2 X3} for this research, to the number of all models.

The Response Surface Methodology technique was employed to set up a 2^6 full factorial experiment in order to study the effectiveness of each subset selection method with respect to the known data characteristics.

Including the results of Hansen's and Woollard's study, eleven different subset selection techniques were compared. Considering the PM and FM measurements together, the CpMLR (method which is a two stage subset selection method) performs the best, and with a very small difference the SpMLR method (which is another two stage subset selection procedure) is close behind the CpMLR. The Modified-Miller's Method, (which is a one step subset selection method) performs the third best.

A COMPARISON OF VARIABLE SELECTION CRITERIA FOR MULTIPLE LINEAR REGRESSION: A THIRD SIMULATION STUDY

I. Introduction

Background

Regression analysis is a statistical methodology that is used to relate variables. The variable of interest, which is called the dependent or response variable is related to one or more predictor or in other words independent variables. In the literature the dependent variable is mostly denoted by Y , and the independent variables are denoted by $X_1, X_2 \dots X_p$. The objective in regression analysis is to build a regression model (equation). This model is used to describe, predict and control the dependent variable on the basis of independent variables (Bowerman and O'Connell, 1990:3).

In real world applications, many times an analyst is involved in regression analysis, where the number of independent variables is over 100. The cost of maintaining

a model with large number of independent variables is high, and it is hard to understand and gain insight into the studied phenomena. Because of the large variance the predictive capability of such a model is also small. In order to minimize these problems, the analyst needs to find the correct number of independent variables to include in the regression model. The concern of subset selection in multiple regression is to find the best solution to this requirement. It is not an easy task. The topic of subset selection is one which is viewed by many statisticians as "unclean" or "distasteful" (Miller, 1990:ix).

Subset selection procedures mentioned in the literature can be classified as (1) considering all-possible subsets, and, (2) finding the sub-optimal models. All-possible subsets methods are called guaranteed methods, but because of the large computational requirement often they are hard to implement. For a set of 25 independent variables 33,554,431 possible models need to be examined. The guarantee of these methods is to find the best model for the pool of variables already available. But what if the pool doesn't include all significant variables? Also the final best set of variables obtained for a sampled data might be different from the best set of variables obtained for another sampled data. On the other hand, the sub-optimal methods (the most widely used ones are the stepwise

procedures) don't consider all-possible models, therefore they need less computational effort. They are based on finding and including in the model the most significant variable given that the previously selected variables are in the model. These methods do not even guarantee that the final model is the best one for the given pool of variables.

The need for better subset selection methods is one of the things that is addressed this thesis research.

Problem Statement

The objectives of this thesis effort are: (1) introduce and implement two stage subset selection procedures, (2) make an improvement in Miller's method, and, (3) make an extension of Ross Hansen's 1988 and David Woollard's 1993 thesis research by comparing the results of the methods they have examined (Minimum MSE, Minimum Sp, Minimum Cp and Miller's Method) with the results of the methods implemented in this research.

Assumptions

It is assumed that: the data generated and used through the research is a random sample from the

predetermined population, the error terms are identically and independently distributed (iid) and come from normal populations with mean zero and variance σ^2 .

Scope

This study is an extension of Ross Hansen's 1988 and David Woollard's 1993 research. In this thesis effort six different two stage subset selection procedures were examined. In the first three, Miller's Method was used for screening, and in the second three Miller's Method was used for final model selection. Improvements were done in Miller's Method and the obtained method was called Modified-Miller's Method. All the results of the employed methods in this thesis research are compared including the results of methods studied by Hansen and Woollard.

II. CONCEPT OVERVIEW

Least Square Regression

In the subset selection criterias employed in this thesis research, Least Square technique is widely used. In this chapter properties and the concept of this technique are given. In general, the linear least squares regression equation is written:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + e \quad (1)$$

where:

- y is the observed value of the response
- β_0 is the constant term (intercept)
- β_i is the coefficient of the variable x_i
- p is the number of independent variables
- e is the error term

Assumptions

It is assumed that the collected data is a good representative of the population it is coming from. The

error term is assumed to be normally distributed with mean zero and unknown variance σ^2 .

The matrix notation of equation 1 is:

$$Y = \beta X + e \quad (2)$$

where:

Y is $n \times 1$ column vector of the observed response value

β is $p \times 1$ column vector of regression coefficients of the independent variables

X is $\{n \times (p+1)\}$ matrix of independent variables including the intercept

e is $n \times 1$ column vector of error terms

The least squares method is based on the criteria of minimizing the sum of the squared errors of prediction, that is, the sum of the squares of the observed values minus the corresponding prediction value.

$$\min \sum_1^n (e_i)^2 = \min \sum_1^n (y_i - \hat{y}_i)^2 \quad (3)$$

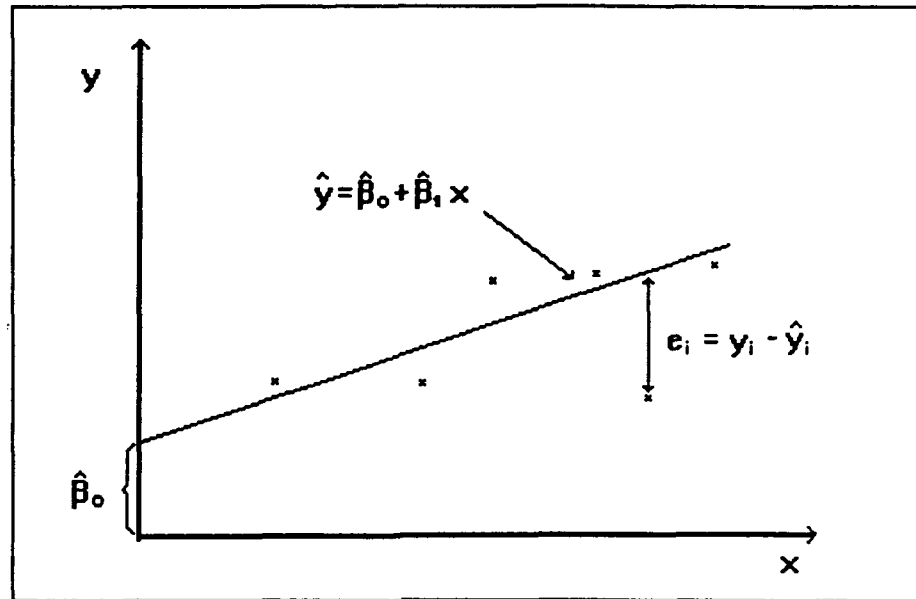
where:

e_i is the residual of the observation i

y_i is the observed response for the observation i

$\hat{y}_i$ is the predicted value for observation i

The graphical representation of the least squares method for the two dimensional case is given in Graph 2-1.



Graph 2-1.

Least Square estimates of the regression equation is the best among the unbiased estimators (Guttman, 1982:6-21).

III. LITERATURE REVIEW

Regression

Linear regression is a statistical model-building tool that utilizes data and the relationships between two or more quantitative variables to construct a mathematical expression in order to predict the value of one variable from the other, or others. This mathematical expression or in other words regression model can, with a certain degree of accuracy, predict the level of response of the associated phenomena, given a set of predictor values.

The term "regression" emerged from Galton's studies of inheritance in biology in the late 1800's. In his particular example Galton mentioned that tall fathers had tall sons, but not as tall on average as the fathers. The same way, short fathers had short sons, but not as short on average as the fathers. The tendency of the average characteristic of the selected group in the next generation toward the mean of the population, rather than reproducing the averages of their fathers, Galton called regression-

regression toward the mean. Thus the term regression was born (Mosteller and Tukey, 1977:262).

In the real world, the analyst is concerned with the relationship between more than two variables to construct a mathematical expression in order to predict the value of one variable (called the response variable) given the others (called the independent or predictor variables). Unfortunately, it is often difficult to determine the "best" set of predictor variables to include in a linear regression model. All methods in the literature have certain disadvantages that can interfere with the process or in some cases, make it hard to implement the selected subset selection criteria.

Subset Selection

Subset selection is a subarea of regression analysis concerned with choosing variables from the given pool of predictive (independent) variables to include in the regression model.

The problem of determining the "best" subset of a given pool of variables has long been of interest to applied statisticians and because of the availability of high-speed computers in recent years, this problem has received

considerable attention in the recent statistical literature.

The problem of selection a subset of independent variables is mainly influenced by the set assumptions. That is, it is assumed that a) the analyst has data on a large number of potential variables which include all significant variables and some other extraneous variables. b) the analyst has available "good" data on which to base the final conclusions. In practice lack of satisfaction of these assumptions may make the performed subsets selection analysis a meaningless exercise (Hocking, 1976:2).

Assuming the analyst has the complete set of the variables, in order to make the regression equation useful for predictive purposes he should want the model to include as many real predictors as possible so that reliable fitted values can be determined (Drapper and Smith, 1966:163). On the other hand, the analyst also wishes to reduce the number of independent variables to be used in the final model. The reasons for this are that a regression model with a large number of independent variables is hard to maintain, and that further regression models with a limited number of independent variables are easier to work with and understand. Having a large number of independent variables in the final model increases the risk of including extraneous variables which behave like the error term. Finally, the presence of many highly correlated independent variables may add little to the predictive ability of the

model while substantially increasing the sample variation of the regression coefficients and roundoff errors (Netter and others, 1990:436).

There is no unique statistical procedure for finding the "best" subset and algorithms for finding best-fitting subsets of variables to a set of data require search strategies and computational algorithms. Search strategies can be divided conveniently into those which guarantee to find the best fitting subset of some or of all sizes, and the "cheap" methods which sometimes find the best-fitting subset (Miller, 1984:391).

All-Subsets Regression

All-subsets is a very direct method, which involves estimating all possible subset regression equations. For a pool consisting of p independent variables p equations

(models) including only one independent variable, $\binom{p}{1}$

equations including 2 independent variables and so forth

until $\binom{p}{p} = 1$ equation including all variables were

examined. Total number of equations examined for p

independent variables is 2^p-1 (Hawkins and Weber, 1980:460). Since this method examines all-possible models it needs long computational time. Thus it requires high-speed computers. With current computer support it is not feasible to employ this method for $p>20$. For a variable set of 20 these are $2^{20}-1 = 1,048,575$ models that need to be examined, and for $p=25$ there are $2^{25}-1 = 33,554,432$ models that need to be examined.

Miller refers to all-subsets method as guaranteed to find the best-fitting model (Miller, 1984:391), but all-subsets selection is only guaranteed to find the best model if all significant predictors are considered; if the set under consideration doesn't include all significant variables, then the all-subsets approach can not find the "best" overall model but will provide the best model for variables considered (Woollard, 1993:14).

There are different subset selection criteria in the literature employed by all-subsets method. In this thesis effort the following are covered:

1. Maximum Coefficient of Multiple Determination (R^2)
2. Maximum Adjusted R^2 or Minimum MSE
3. Minimum Mallows' C_p
4. Minimum Prediction Sum of Squares (PRESSp) or Minimum Sp

1. Maximum Coefficient of Multiple Determination R^2

The coefficient of multiple determination denoted by R^2 is defined as follows:

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} \quad (4)$$

where;

SSR is sum of squares of regression

SST is the total sum of squares

$\hat{y}_i$ is the prediction value of y_i

$\bar{y}$ is the mean of y

y_i is the actual value of y for observation i

R^2 measures the proportionate reduction of total variation in y . When we look at R^2 values we would like to select models with large R^2 value. There is a pitfall in this criteria, because adding a variable to the model increases R^2 , therefor a model including all variables has the largest R^2 value. That characteristic of this criteria voids its applicability in the subset selection.

2. Maximum Adjusted R^2 or Minimum MSE

Since R^2 keeps increasing with adding new variables in the model, the use of the adjusted coefficient of multiple determination R^2 has been suggested as a criteria, which

takes the number of parameters in the model into account through the degrees of freedom.

$$R_a^2 = 1 - \left(\frac{n-1}{n-p} \right) * \frac{SSE}{SST} = 1 - \frac{\frac{MSE}{SST}}{\frac{n-1}{n-1}} \quad (5)$$

$SST/(n-1)$ is fixed for given y observations, because of this reason R_a^2 increases if MSE decreases. Hence R_a^2 and MSE are equivalent criteria (Netter and others, 1990:446). Minimum MSE or Maximum R_a^2 is one of the widely used criteria in practice.

3. Minimum Mallow C_p

Mallow C_p is an approximation of the Mean Squared Error of Prediction (MSEP). It is calculated using the following equation:

$$C_p = \frac{SSR}{S^2} - n + 2 * p \quad (6)$$

where:

SSR is the regression sum of squares

S^2 is an estimate of the variance

n is number of data points (sample size)

p is number of parameters (including intercept)

There are two applications of this criteria in the subset selection.

Application 1. The theoretical expected value of C_p is p . Thus, the C_p value of models with little bias will

tend to be $C_p \sim p$. The model with C_p value closest to p will be selected as the "best" model.

Application 2. The other application of the C_p criteria in the subset selection is to choose the model having Minimum C_p value. The models with small C_p values have small total mean squared error, therefore the model with the smallest C_p is selected to be the best.

The application #1 is hard to automate, while application #2 is easy to automate. For this reason application #2 is employed in this thesis.

4. Minimum PRESSp or Minimum Sp Criterion

PRESSp (Prediction Sum of Squares) like the C_p criterion is an approximation of the MSEp. Its formula is:

$$Sp = \frac{SSE}{(n-p) * (n-p-2)} \quad (7)$$

where:

SSE is sum of squares of error

n is the sample size

p is the number of parameters

Models with small Sp values are considered to be good candidate models because they have small prediction errors. This method has received considerable praise in recent years.

Stepwise Procedures

These procedures do not involve evaluating all-possible regression equations. The stepwise procedures discussed in this thesis effort are:

1. Backward Elimination Procedure (BE)
2. Forward Selection Procedure (FS)
3. Stepwise Regression Method

1. Backward Elimination Procedure

This method is a suboptimal procedure, since it doesn't calculate all-possible models. The steps in this procedure can be described as follows:

- a. A regression equation containing all variables is calculated.

- b. The partial F-test is computed for every variable in the model treated as though it is the last variable to enter the regression equation.

- c. The lowest partial F-test value, F_L , is compared with a preselected significant level F_0 .

- d. If $F_L < F_0$ remove the corresponding variable from consideration and go to step b.

- e. If $F_L > F_0$ conclude that this is the final model.

2. Forward Selection Procedure

Backward elimination procedure begins with the largest model and subsequently reduces the number of variables in the model until a decision is reached on the equation to be used. The forward selection procedure works from the other direction. It begins with the null model. The steps can be summarized as follows:

- a. Calculate first order linear equation for each of the variables in the pool.
- b. Calculate the F^* statistic for each model and the X variable with the largest F^* value is the candidate to enter the model, say X_4 .
- c. If F^* of $X_4 > F_0$, the variable is entered into the model.
- d. Now, linear equations with two variables, where X_4 is one of the pair is fit and the corresponding F^* values are computed.
- e. If the largest F^* value $> F_0$, then these variables are entered into the model. Steps d and e are followed for the other variables until $F^* < F_0$.
- f. If $F^* < F_0$, it is concluded that the previous model is the final model.

3. Stepwise Regression Method

There is a pitfall in Backward Elimination and Forward

Selection procedures. A variable which is significant at one step may turn out to be insignificant in the following steps and because there is no elimination for the variables already in the model for Forward Selection procedure, that variable which turned to be insignificant will be left in the model. The same problem occurs for the Backward Elimination procedure. A variable which was eliminated because it was considered insignificant may turn out to be significant in the succeeding steps, but because of no addition of the variables to the model it is not possible to include that significant variable in the final model. The stepwise regression method was developed to eliminate this problem. This method is a combination of the previously mentioned two methods. It can be considered as a Forward Selection method employing Backward Elimination procedure after the addition of a new variable in the model. Miller refers to these methods as a "cheap" methods, because it is not guaranteed that the final model is the "best".

Miller's Method

The idea of augmenting the variable pool with known extraneous random variables was first introduced to the

literature by Miller in 1984. Later in 1990 Miller provided more detailed information about this idea (Miller, 1990:84-85). Woollard in 1993 applied this augmentation procedure and named this method Miller's Method (Woollard, 1993:21-22). In this method Woollard augmented the variable pool with an equal number of "dummy", known extraneous random variables. These random variables are generated from standard normal distribution. Later Forward Selection procedure was employed until the first known extraneous random variable was included in the model. Subset selection procedure stopped and discards the model including the known extraneous random variable, and used the previous model. The other variables not included in the model were considered insignificant.

IV. METHODOLOGY AND MODEL DEVELOPMENT

OBJECTIVE

The goal of this thesis research is to introduce and study a modification of Miller's Method which we call the Modified-Miller's method (MM method), study two step subset selection procedures, one of which applies Miller's Method in the first step and employs Minimum MSE or Minimum Sp or Minimum Cp in the second step and the other of which employs Minimum MSE or Minimum Sp or Minimum Cp in the first step and applies Miller's Method in the second step. The results of all techniques will be compared including the results of the previous simulation studies done by Hansen in 1988 and by Woollard in 1993. In the researches mentioned above, Minimum MSE, Minimum Sp, Minimum Cp and Miller's method were studied.

LIMITATIONS

Two kinds of limitations were experienced in this simulation study:

- (1) Limitations based on the assumptions of the

applied technique.

Since least squares regression is extensively employed in this thesis research the result and the conclusions are valid within the limitations of these assumptions. These assumptions can be summarized as follows: data is assumed to be representative of the population, the error term is assumed to be independent and identically distributed with mean zero and variance σ^2 (Woollard, 1993:25)

(2) Computer limitations.

In this thesis research over 80,000 models were examined. Because of the allocation problems which occurred in the computer system, the research area was restricted and the Modified Miller's method was run with fewer replications than what is recommended.

OVERVIEW

The methodology and approach in this thesis will be consistent with that used by Hansen and Woollard so that it will be possible to compare the results of the methods employed in this research with the results of the previous two studies. This thesis effort aims to extend the process

of finding better subset selection methods started by Hansen in 1988 and later continued by Woollard in 1993. It is possible to divide this research effort mainly into four areas of focus:

- (1) Data generation.
- (2) Introduction and the development process of the Modified-Miller's method.
- (3) Model selection by employing different subset selection methods.
- (4) Generation of performance measurements and evaluation of the subset selection methods.

The data used in this study was generated by Hansen and later corrected by Woollard. These data sets were generated with known statistical properties.

For the application of the Modified-Miller's method SAS Stepwise procedure and FORTRAN routines were employed and in the two stage model selection process SAS all-subsets and Stepwise procedures were used along with the FORTRAN codes that pick the best model among the generated ones.

Two different performance measurements were calculated. The first one, designated PM, calculates the percentage of the correct variables selected in the given model. The second performance measure, designated FM, calculates the percentage of the perfect models (for this study the perfect

model consist of set {X1 X2 X3}) to the number of all models.

For the evaluation of the employed methods, experimental design was constructed based on the RSM (Response Surface Methodology) to determine the impact of the specific statistical properties of the data.

OVERVIEW OF DATA GENERATION PROCESS

The data used in this simulation study was generated by Hansen in 1988 (Hansen 1988:34-37). Later in 1993 Woollard made a few corrections in the original data (Woollard 1993:27-31). The data used in this simulation study is the revised version. An overview of that data generation process is given in the following section.

The data needed for this simulation study was created with certain characteristics considering the design point. The design points were established according to the RSM (Response Surface Methodology). This approach to data generation provides the advantage of knowing the characteristics of the data. Those characteristics in the experimental design are called factors and in this study

they are given in the next section.

Factors

In the data generation for this simulation study six potentially significant statistical properties were considered as factors. For each factor low and high settings were established, therefor a 2^6 full factorial design was used. The six factors employed in this design were:

1. The Number of Extraneous Variables.

Extraneous variables are those independent variables which are in the variable pool, but which actually are independent of the dependent variable and act as noise if included in the prediction model. In this design, at the low setting the number of extraneous variables is 1 and at the high setting, 3. They are denoted E1, E2 and E3.

2. The Correlation Between Real Predictors.

Real predictors are denoted X1, X2, X3 and X4. These are independent variables used in generation of the dependent variable. At low setting the variables are independent from each other, in another words orthogonal

(having zero correlation) and at high setting they are highly correlated with a correlation of 0.9.

3. The Variance of The Extraneous Variables.

The low setting is 1 and the high setting is 100.

4. The Variance of The Real Predictors.

The low setting is 1 and the high setting is 100.

5. The Sample Size.

The low setting for the sample size is 10 and the high setting is 20. Sample size less than 10 cannot be used in computing S_p criteria (Hansen, 1988:35-36)

6. The Variance of the Error Term.

The low setting for the variance of the error term is 0.0625 and the high setting is 0.25.

Order	Factor Description	Values		Factor Symbol	
		Low	High	Low	High
1	Number of extr.var.	1.0	3.0	a	A
2	Corr. of real pred.	0.0	0.9	b	B
3	σ^2 of extr. var.	1.0	100.0	c	C
4	σ^2 of real pred.	1.0	100.0	d	D
5	Sample Size	10.0	20.0	e	E
6	σ^2 of the error term	0.0625	0.25	f	F

Table 4-1.

Summary table for the factors used in data generation.

In order to make the representation of factors easy throughout the thesis, factors are expressed with the respective factor symbol.

Design Point	Data Files	Factor Settings
1	01.dat	abcdef
2	02.dat	Abcdef
3	03.dat	aBcdef
4	04.dat	ABcdef
5	05.dat	abCdef
6	06.dat	AbCdef
7	07.dat	aBCdef
8	08.dat	ABCdef
.	.	.
.	.	.
.	.	.
63	63.dat	aBCDEF
64	64.dat	ABCDEF

Table 4-2.

Table 4-2 shows the order of factors according to the design point. Capital letter shows high settings, lower case letters represent low settings.

Data Generation

The data for real predictors and the dependent variable were generated from the following equation:

$$Y_i = X1_i + X2_i + X3_i + X4_i + \epsilon_i \quad (8)$$

where;

Y_i is the response (dependent variable)

$X1_i, \dots, X4_i$ are randomly generated predictors

ϵ_i is the noise term

In order to have a more realistic simulation of the real world, after the generation of Y_i , $X1_i$, $X2_i$, $X3_i$, $X4_i$ with the characteristics of the design point i , the $X4_i$ term was dropped from the variable pool. In the real world we may not be aware of a significant term or we may be aware of it but not able to collect data about it and include it in the variable pool. In addition to these real predictors 1 or 3 extraneous variables have been included in the variable pool. The number of generated data for a specific design point is consistent with the sample size factor of this

design point. The error term (ϵ_i) is also included in the data generation for the better representation of the real world.

Data Sets

For each design point 60 data sets have been generated. It is possible to consider each set as a sample from a population. According to the sample size factor settings in the experimental design, this sampling is done with a sample of size 10 for low settings and a sample of size 20 for high settings.

PERFORMANCE MEASUREMENTS

In regression analysis the first concern is to find a model having no extraneous variables, and the second concern is to find a model including the right number of real predictors in order to minimize deviation from the perfect

model. In the real world it is not possible to know what is the perfect model, but it can be described as a model which best describes the whole population. To achieve these goals, in this research effort two types of measurements were employed in order to evaluate the success of the applied subset selection method.

1. Performance Measurement for Real Predictors (PM).

This measurement deals with the first concern; having no extraneous variables or in other words having only real predictors in the selected model. The equation for the PM value was set as follows;

$$PM = \frac{\text{number of correct variables chosen}}{\text{number of variables chosen}} \quad (9)$$

In the real world it is not possible to know which variables are real and which are extraneous, therefor it is not possible to compute the PM value. But in the simulation, where these factors are under control, PM values are computed to evaluate the applied subset selection criteria. PM takes into account the number of the extraneous variables

along with the number of the real predictors. It is worse to select a model with only two predictors of which one is extraneous, than it is to select a model containing four variables of which one is extraneous (Woollard, 1993:34). The PM values for these two cases are;

$$\text{Case one : } Y1 = X1 E1 \quad \rightarrow \quad PM = \frac{1}{2} = 0.5$$

$$\text{Case two : } Y2 = X1 X2 X3 E1 \quad \rightarrow \quad PM = \frac{3}{4} = 0.75$$

Each model has one extraneous variable, but the second model is logically preferable to the first since it includes a larger number of real predictors, and the same conclusion was arrived at with the computed PM value.

One can think of a model containing only one predictor and it is a real one; and compare it with a model containing four predictors, three of which are real with the fourth one being an extraneous variable, and a third model containing

three predictors, all of which are real. The PM values are;

$$\text{Model number 1 : } PM_1 = \frac{1}{1} = 1.00$$

$$\text{Model number 2 : } PM_2 = \frac{3}{4} = 0.75$$

$$\text{Model number 3 : } PM_3 = \frac{3}{3} = 1.00$$

At first this may not look reasonable. One may ask "Is the first model superior to the second model with the first and third models being equivalent?". In terms of the extraneous variables the answer is "Yes". These results are consistent with the goal of PM measurement. This performance measurement is designed to eliminate the extraneous variables. The PM consideration of the model

consisting of only {X1} and model consisting of {X1, X2, X3} are equivalent has been compensated for in the Full Model (FM) measurement.

2. Performance Measurement for the Full Model (FM).

This measurement deals with the second concern: finding a model which includes the right number of real predictors in order to get less deviation from the perfect model. For this study the perfect model is {X1, X2, X3}, since the data was generated from $Y_i = X1_i + X2_i + X3_i + X4_i + e_i$ and only {X1, X2, X3} were in the variable pool. Therefor, models different then {X1, X2, X3} have been ignored. The equation for this performance measurement is as follows;

$$FM = \frac{\text{number of full models}}{\text{number of all models}} \quad (10)$$

The following example is given to demonstrate the computation of the FM values used to evaluate the performance of the applied methods in this thesis effort:

For each design point 60 data sets were established. Employing any of the subset selection criteria used in this study provides us with a model, which is assumed to be the best one for that specific data set. Therefor for each design point there are 60 models on hand. (Null models,

having no independent predictors, are also considered as a model.) Assume that 35 of these 60 models are {X1 X2 X3}. The FM value for this given design point is $35/60 = 0.583$.

RESULTS OF PREVIOUS SIMULATION STUDIES

In the previous two subset selection simulation studies two performance measurements were employed.

1. Performance measure (PM) for the percentage of correct variables.

The PM values described in previous sections are used by Hansen to evaluate the performance of Minimum MSE, Minimum Sp, and Minimum Cp (Hansen, 1988:39). In 1993 Woollard used the same performance measurement to evaluate Minimum MSE, Minimum Sp, Minimum Cp, and Miller's Method (Woollard, 1993:39). In his research, Woollard reran the three methods studied by Hansen after the correction of the data files. Because of this correction in the data files, only Woollard's PM results were covered in this section. The experimental design used to examine the behavior of the employed subset selection criteria to the statistical

characteristics of the data will be explained in detail in chapter V, but in this stage it can be told that this experimental design is based on RSM, where the following equations are the models of PM, given characteristics (factors) of the data. Woollard's results for PM measurement are (Woollard, 1993:40):

Minimum MSE.

$$\begin{aligned}
 PM_{MSE} = & 0.78 - 0.10 (A) + 0.0023 (D) + 0.006 (E) + 0.0062 (F) \\
 & + 0.003 (AE) + 0.003 (AF) - 0.003 (DF) - 0.006 (EF) \\
 & - 0.003 (AEF)
 \end{aligned}$$

Minimum Sp.

$$\begin{aligned}
 PM_{Sp} = & 0.85 - 0.07 (A) + 0.002 (D) + 0.007 (F) + 0.002 (AD) \\
 & + 0.006 (AE) - 0.003 (DE) - 0.007 (AF) - 0.002 (DF) \\
 & - 0.005 (EF) - 0.002 (ADF) - 0.006 (AEF)
 \end{aligned}$$

Minimum CP.

$$PM_{CP} = 0.84 - 0.07(A) + 0.003(D) + 0.007(E) + 0.008(F) \\ + 0.002(AD) + 0.006(AE) + 0.003(DE) + 0.008(AF) \\ - 0.003(DF) - 0.005(EF) - 0.002(ADF) - 0.006(AEF)$$

Miller's Method.

$$PM_{Miller} = 0.88 - 0.04(A) + 0.01(B) + 0.02(E) + 0.01(AB) \\ + 0.008(AE) - 0.008(BE) - 0.008(BCEF)$$

Method	Minimum	Minimum	Minimum	Miller's
Factors	MSE	Sp	Cp	Method
A(# of ext. var)	-0.1	-0.7	-0.07	-0.04
B (ind.corr)	0	0	0	+0.01
C(σ^2 of ext.var.)	0	0	0	0
D(σ^2 of ind.var.)	+0.0023	+0.002	+0.003	0
E(sample size)	+0.006	+0.007	+0.007	+0.02
F(error term)	+0.0062	+0.007	+0.008	0
Intercept	+0.78	+0.85	+0.84	+0.88

Table 4-3.

Main factor Coefficients of Effects by Method for PM measurement

Performance Measure of Model Accuracy

For the performance measure of the model accuracy, Hansen and Woollard used Theoretical Mean Square Error of Prediction (TMSEP) (Hansen, 1988:44) and (Woollard, 1993:50). TMSEP is defined as follows;

$$TMSEP_{k,m} = \frac{\sum_{t=1}^{n_k} (y_t - \bar{y}_{k,m})^2}{n_k - p_{k,m}} \quad (11)$$

where:

$TMSEP_{k,m}$ is the TMSEP for data set k using the subset selection technique m

y_t is the theoretical conditional mean of Y calculated from the model that data was generated from.

$\bar{y}_{k,m}$ is the predicted value of Y using technique m to the data set k

n_k is the sample size of the data set k

$p_{k,m}$ is the number of predictors in the model applying technique m to the data set k

In this research for the performance of the model

accuracy the Full Model (FM) measurement is employed. The reasons for employing this performance measurement are;

(1) Calculation of TMSEP is based on sampled data $(Y_{k,m}, n_k, p_{k,m})$, which means that this performance measure was influenced by the sampling error. In this performance measurement there is a risk to end up with a model including many extraneous variables and have an overfitted model.

(2) In the real world, it is not possible to calculate FM value because the perfect model (model including all real predictors and no extraneous ones) is not known, and actually the goal of the regression analysis is to find that perfect model or the closest one. In this simulation study the perfect model is already available, because the data was generated from the model established by the researcher. Using this advantage FM values are easily computed employing equation (10). FM measurement is very conservative, since it is considering models consisting of only $\{X_1, X_2, X_3\}$, but this performance measure is free from sampling error.

As it was mentioned in the beginning of this section, Hansen and Woollard employed TMSEP for model accuracy. In order to compare the results of the previous studies with the results of the subset selection technique applied in

this research, Woollard model accuracy computations were rerun according to FM criteria. The results were shown and compared with the results of the new techniques in chapter V.

NEW SUBSET SELECTION METHODS STUDIED IN THIS RESEARCH

Seven different approaches were examined in this thesis effort for the subset selection. The first six of them are two stage subset selection technique and the seventh one is Modified Miller's Method (MM Method). These subset selection technique can be listed as follows:

1. Miller's & MSE Method (MLRMSE method)
2. Miller's & Sp Method (MLRSp method)
3. Miller's & Cp Method (MLRCp method)
4. MSE & Miller's Method (MSEMLR method)
5. Sp & Miller's Method (SpMLR method)
6. Cp & Miller's Method (CpMLR method)
7. Modified Miller's Method (MM method)

Two Stage Subset Selection Process Where Miller's Method is Used for Screening.

The first three methods are two stage subset selection techniques where, in the first stage Miller's method is employed as a screening method and in the second stage Minimum MSE or Minimum Sp or Minimum Cp is employed as a final subset selection technique to the variable pool obtained by the first stage (Miller's Method). This procedure can be illustrated as follows:

Assume that the original variable pool includes X1, X2, X3, E1, E2 and E3. Miller's Method is applied as a screening process in the first stage and the model obtained include X1, X2, X3 and E2 variables. In the second stage for the final subset selection process Minimum MSE, Minimum Sp or Minimum Cp is employed using variable pool X1, X2, X3 and E2.

Two Stage Subset Selection Process Where Miller's Method is Used For Final Subset Selection.

The second three two stage subset selection methods listed in page 39, which are the 4th, 5th and 6th methods

employs Minimum MSE or minimum S_p or Minimum C_p for screening in the first stage and for the final subset selection process in the second stage Miller's Method is employed.

Modified Miller's Method (MM Method)

The basic idea in the Miller's Method is this: since extraneous variables have no relation with the dependent variable in reality, the correlation should be zero; because of the sampling error the sample correlation matrix may not show zero correlation between dependent and extraneous variables, and applied techniques for finding the best model may include extraneous variables in the final model. In order to minimize this undesirable effect, the variable pool is augmented with known extraneous random variables. When the subset selection procedure (Stepwise with Forward selection was used in Miller's Method) first picks one of these known extraneous random variables, the selection procedure stops and the previous model (model including no known extraneous random variables) is selected as the final model. In reality Miller's Method is a simulation of extraneous variables, because randomly generated known random variables are exposed to the same sampling error that

extraneous variables are; for small sample size this error is expected to be large and with the increased sample size this error is expected to decrease. Because Miller's Method is a simulation, like other simulation technique it should be run multiple times. The specific reason for the need to run more than once is this: Assume that because of the sampling error the first variable picked by the selection procedure could be one of the known extraneous random variables. In this case the selection procedure stops and the final model is concluded to be the null model. With multiple runs the effect of this undesired result will be minimized. Modifications in the Miller's method lead us to a new method called Modified-Miller's method (MM method). In the application of the MM method there are four questions which need to be answered.

- (1) What is the best number of known extraneous random variables to be used in the augmentation of the variable pool?
- (2) How many runs should be performed?
- (3) How to select the best model?
- (4) How to decide when the best model is the null model? (The null model is best when the pool is all extraneous. The question is how to conclude that all variables in the pool are extraneous.)

The answers follow:

1. Number of known extraneous random variables
(augmentation coefficient).

Assume that there is a pool of 4 variables, 3 of them are real predictors and 1 is extraneous (which we can know only in a simulation). This pool is augmented with 4 known extraneous random variables. Suppose that the subset selection process has reached the stage where the selected variables are all real ones and that all extraneous variable, including the augmented ones, have the same chance of being picked next. Then, the conditional probability that the extraneous variable is included in the model is $1/(1+4)=0.20$ and the probability that one of the known extraneous random variables is chosen is $4/(1+4)=0.80$. Now instead of augmenting with 4 let us augment the pool with 8 known extraneous variables. The probability that the extraneous variable is included will be $1/(1+8)=0.11$ and the probability that one of the known extraneous random variables is chosen is $8/(1+8)=.89$. The critical point in the decision on the number of known random variables is this: if there are fewer known extraneous random variables than necessary, the probability of having extraneous variables in the model is high. If there are more known extraneous random variables than are needed, the probability

that some the real predictors are not included in the model is high. In order to find the optimum number of known extraneous random variables each data set was augmented by 0.5, 1.0, 1.5 and 2 times the number of variables a specific data set has.

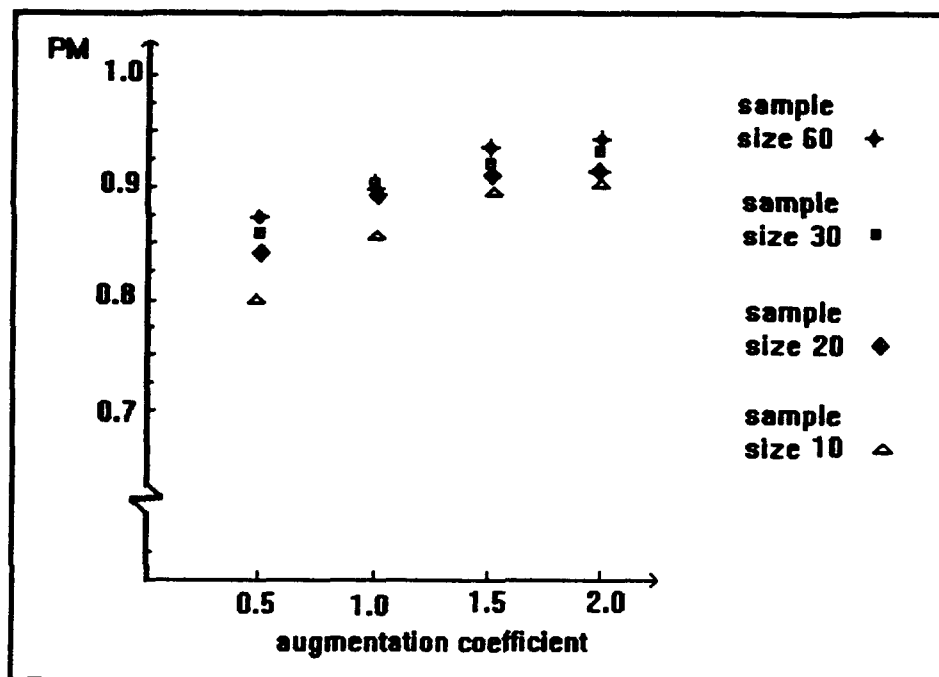
These numbers are designated as augmentation coefficients. The MM method was run for each set with each augmentation. To be able to examine the effect of the sample size more effectively, data sets of 30 and 60 samples were obtained by joining 3 sets of sample size 10 or 20 of the original data. Therefor, $64 \times 60 = 3840$ data sets of size 10 or 20 and $64 \times 20 = 1280$ data sets of size 30 or 60, as total 5120 data sets were examined for each augmentation coefficient. For the measurement of the performance PM and FM measurements were employed.

The PM values of the MM method for different augmentation coefficients was given in the Table 4-4. For better representation, the data in column 5 (average PM values) was graphed (Graph 4-1). The data for this column is the average of the average PM values of 1 and 3 extraneous cases. Therefor, only sample sizes were considered, because in the real world only this factor is under the control of the experimenter.

Smpl. Size	Augm. coeff.	Avrg. PM val. of models obtained from sets having 1 ext. var.	Avrg. PM val. of models obtained from sets having 3 ext. var.	Avrg. PM values
10	0.5	0.8737217	0.745335	0.8095536
	1.0	0.9042049	0.8105713	0.8573881
	1.5	0.9249750	0.8391179	0.8820465
	2.0	0.9350211	0.8576428	0.8963320
20	0.5	0.8971385	0.8017612	0.8494499
	1.0	0.9286529	0.8675144	0.8980837
	1.5	0.9474303	0.9038717	0.9256510
	2.0	0.9232731	0.9127891	0.9180311
30	0.5	0.8885906	0.8405158	0.8645532
	1.0	0.9361144	0.8884765	0.9122955
	1.5	0.9331758	0.9241160	0.9286459
	2.0	0.9513009	0.9421828	0.9467419
60	0.5	0.9039543	0.8608187	0.8823865
	1.0	0.9473114	0.8706017	0.9089566
	1.5	0.9581975	0.9120995	0.9351485
	2.0	0.9648063	0.9301523	0.9474793

Table 4-4.

The PM values of models obtained by using different augmentation coefficients.



Graph 4-1.

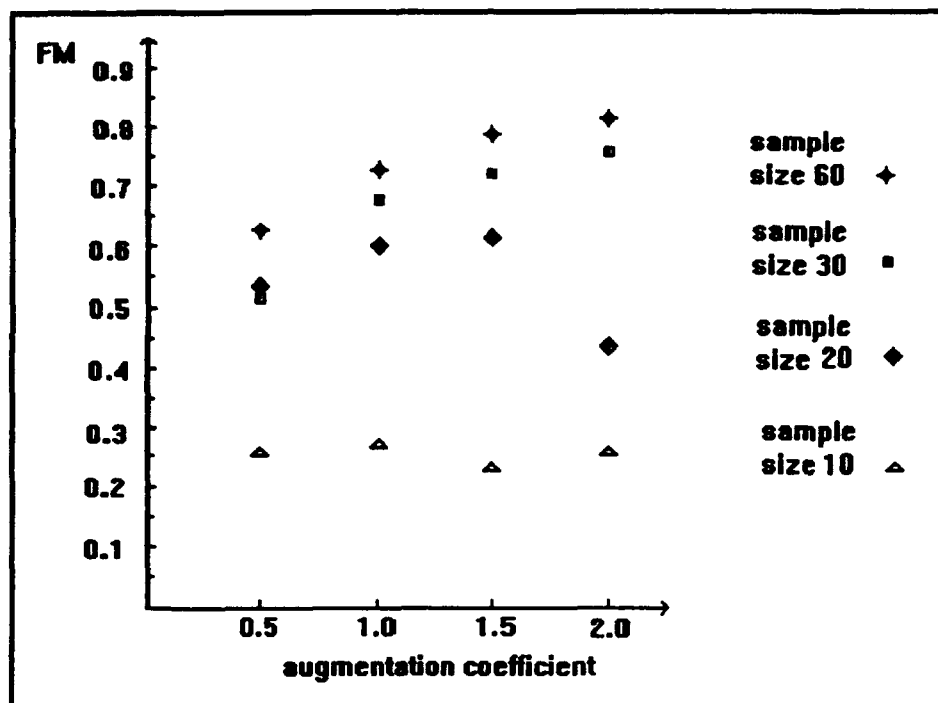
The PM values of Table 4-4 column 5 were graphed to demonstrate the relation between sample size, the augmentation coefficient and PM measurement.

The FM values of MM method were displayed in the table 4-5. The average FM values (column 5) were graphed (Graph 4-2) considering sample size and the augmentation coefficients.

Sampl. size	Augm. coeff.	Avrg. FM val. of models obtained from sets having 1 ext. var.	Avrg. FM val. of models obtained from sets having 3 ext. var.	Avrg. FM values
10	0.5	0.320	0.192	0.256
	1.0	0.322	0.197	0.260
	1.5	0.288	0.171	0.230
	2.0	0.335	0.164	0.250
20	0.5	0.600	0.473	0.537
	1.0	0.670	0.530	0.600
	1.5	0.684	0.551	0.618
	2.0	0.278	0.579	0.429
30	0.5	0.565	0.491	0.528
	1.0	0.709	0.644	0.677
	1.5	0.671	0.778	0.724
	2.0	0.750	0.756	0.753
60	0.5	0.725	0.523	0.624
	1.0	0.794	0.656	0.725
	1.5	0.825	0.753	0.789
	2.0	0.825	0.803	0.814

Table 4-5.

The FM values for models obtained for different augmentation coefficients.



Graph 4-2.

This graph demonstrates the FM values of Table 4-5 column 5.

The results of PM value measurement shows that when the number of known extraneous random variables increases the PM value keeps increasing, but this is not a constant increase, and diminishing returns were observed.

The results of FM measurement show that the optimum number of known random variables depends on the sample size.

This is a good result, since the experimenter in the real world can only control the sample size. For 1 and 3 extraneous variables it was observed that for sample size 10 the maximum FM value occurs for number of known extraneous random variables which is the same with the number of variables in the pool. For sample size 20 maximum FM value occurs for 1.5 augmentation coefficient. For sample size 30 and 60 maximum FM value occurs near 2 augmentation coefficient.

2. Run Numbers

Since the MM method is a kind of simulation each run can be considered as one sample. The more runs, the more accurate is the final model. The recommended number of runs for MM method can be computed using the expected value and the standard deviation of the Binomial Distribution. The conditions for Binomial Distributions are:

1. The experiment consists of a sequence of n trials, where n is fixed in advance of the experiment.
2. Each trial can result in one of the same two possible outcomes, denoted by success (S) or failure (F).
3. Trials are independent.
4. The probability of success is constant from trial to trial.

In MM method run number is set ahead of time, say n (condition 1 holds). Each run can end up with null model (success) or with a non-null model (failure), so, condition 2 is also satisfied. In MM method each run is independent from the others thus, condition 3 is satisfied. The probability of having null model ($P(\text{null}) = \text{number of known extraneous variables} / \text{all variables in the augmented pool}$) is constant for each run so, condition 4 holds.

This decision process for run number using binomial distribution is demonstrated as follows:

Assume that all the variables in the given set are extraneous and the number of variables in the set is V . This set is augmented with r number of known extraneous variables. Thus, the probability of having null model is,

$$P(\text{null}) = \frac{r}{V+r}$$

The mean and the variance for a Binomial Distribution can be computed as follows:

$$E(S) = np \qquad V(S) = np(1-p) \qquad (12)$$

where:

S is the number of successes.

n is the number of trials.

p is the probability of success in a trial.

Now, computing the mean and the variance and the standard deviation for the given case yields:

$$Bin(n, \frac{r}{V+r})$$

$$E(null) = n(\frac{r}{V+r})$$

$$V(null) = n(\frac{r}{V+r}) (\frac{V}{V+r})$$

$$\sigma = \frac{\sqrt{nrV}}{V+r}$$

Since observations should be positive, the 3 standard deviation to the left of the mean value should be at least zero;

$$\mu - 3\sigma = \frac{nr}{V+r} - 3 \frac{\sqrt{nVr}}{V+r} > 0$$

$$\therefore nr > 3\sqrt{nVr}$$

$$n^2 r^2 > 9nVr$$

$$n > 9 \left(\frac{V}{r} \right)$$

If the standard deviation is reduced from 3 to 2 or 1, then, the required number of runs will be less but, this will reduce the confidence of concluding that the model is null or non-null. For this reason the decision of number of runs is left to the experimenter based on his decision of confidence level.

In this research 6 runs were used because of dealing with large number of models and having computer allocation problems, with the tradeoff having less confidence level. For 6 runs $64 \times 60 \times 6 = 23040$ models were examined. For demonstration purpose 8 data sets with 60 runs each were used and the results were compared with the same data sets with 6 runs in chapter VI.

3. Finding The Best Model

The procedure for finding the best model is demonstrated with the help of the following example. The data used in this example comes from design point 14 and data set 31.

Model	Frequency
$y = x_2$	7
$y = x_2 x_3$	3
$y = x_1 x_2 x_3$	18
$y = x_1 x_2 x_3 E_3$	3
$y = x_1 x_2 x_3 E_3 E_2$	8
Null Model	21

Table 4-6.

In this example the sample size is 10 and the variable pool $\{X_1, X_2, X_3, E_1, E_2, E_3\}$ consists of 6 variables. This pool was augmented by 6 known random variables and the MM Method was run 60 times. The most frequent model is the null model. Let us assume for a moment that all variables

in the pool are extraneous. The probability of having a null model is $P(0)=0.50$. In the given example the ratio of null models to all models is $21/60=0.35$. Since this ratio is much smaller than 0.50 and since run number is large enough (60) it is possible to conclude that the final model is not null. The second most frequent model is $\{X_1, X_2, X_3\}$ therefor, it is concluded that the best model consists of set $\{X_1, X_2, X_3\}$.

The steps in the best model selection can be summarized as follows;

- Find the most frequent model. If this is not the null model than it is the best one.
- If the most frequent model is the null model then compute the theoretical probability of having the null model with the assumption that all variables in the pool are extraneous. If the run number is small and the theoretical probability and the actual ratio of the null models to all models is close than calculate the confidence interval of your model (described in the null model section).
- If the ratio of null models is much less than theoretical value then select the second most frequent model as the best model.

4. Null Model

In some cases it would be hard to decide whether to neglect the most frequent null model and consider the second most frequent one as the best model. In such a case where theoretical ratio and the empirical ratio of the null model is close, especially for small sample size, confidence interval should be considered.

For the demonstration purposes the confidence interval for the MM method results in Table 4-6 is computed. Assume all the variables in the pool are extraneous. Then;

$$P(null) = \frac{6}{12} = .5$$

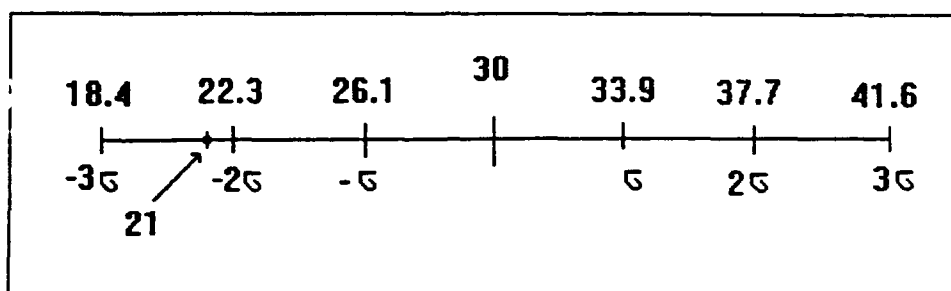
Now, the experiment turns to be a Binomial experiment because all the conditions of the Binomial Distribution are satisfied. For the given example (Table 4-6), the expected value, the variance and the standard deviation of null models is:

$$E(null) = 60 * 0.5 = 30$$

$$V(null) = 60 * 0.5 (1 - 0.5) = 15$$

$$\sigma_{null} = 3.872983$$

These are the theoretical values assuming that all the variables in the pool are extraneous. Now, compare it with the empirical null value displayed in Graph 4-3.



Graph 4-3.

The farther to the left is the empirical value of null models from $E(\text{null})$, the higher is the confidence in concluding that the assumption of all of the variables extraneous is wrong. In the given example the deviation from the expected value is larger than two standard deviations. Therefore, with very high confidence it can be concluded that the assumption of all variables extraneous is wrong. Cross validation of this decision can be done by computing the probabilities of having 21 or less.

$$P(\text{null} \leq 21) = \sum \binom{60}{k} (0.5)^k (0.5)^{60-k} = 0.014$$

This small probability (0.014) of having 21 or less also indicates that the assumption is wrong.

If comparison is made between Miller's Method and Modified Miller's Method, it is possible to say that Miller's Method is a special case of Modified Miller's Method where, the augmentation coefficient is fixed at 1 and the number of runs is one.

V. Experimental Design and the Results

In this chapter two 2^6 full factorial experimental designs were established, where PM and FM values calculated for each method at each design point were used as the response value. The results of these experiments helped determine the effect of factors (statistical properties of the data) on the performance of the applied subset selection method.

Experiment

The experiment conducted in this research is the same used by Hansen and Woollard and was established according to the RSM (Response Surface Methodology) understanding. When using RSM it is convenient to use coded factors (variables). The following equation provides translating a variable from uncoded space to the coded space.

$$Z_i = \frac{X_i - \frac{High_i + Low_i}{2}}{\frac{High_i}{2} - \frac{Low_i}{2}} \quad (13)$$

where;

X_i is the variable in uncoded space of factor i

Z_i is the variable in coded space of factor i

$High_i$ is the upper bound of the uncoded variable X_i

Low_i is the lower bound of the uncoded variable X_i

The following table shows the factors in uncoded space and the corresponding factors in the coded space.

Variable description	Uncoded variable	Uncoded		Coded variable	Coded	
		Low	High		Low	High
Num. of Extr. variables	A	1.0	3.0	Z_1	-1	1
Corr. of Ind. variables	B	0.0	0.9	Z_2	-1	1
Variance of Extr. variables	C	1.0	100	Z_3	-1	1
Variance of Ind. variables	D	1.0	100	Z_4	-1	1
Sample Size	E	10.0	20.0	Z_5	-1	1
Variance of Error Term	F	.0625	0.25	Z_6	-1	1

Table 5-1.

Variable Coding for Response Surface Methodology

In order to obtain the effect of interactions along with the main factors a 2^6 full factorial experimental design was set up. The interaction terms are simply the product of the corresponding coded main factor. Therefor, the design matrix is a 64×64 matrix where entries are -1 and 1.

PM Equations

By editing the PM.DAT file created by Woollard using STATISTIX 4.0, a 2^6 full factorial design matrix was augmented by the PM values calculated for each subset selection method at each design point. This data file was used as an input to the SAS file where the stepwise regression procedure was run to generate the following equations.

$$PM_{MLRMSE} = 0.891 - 0.050A - 0.016EF - 0.022AEF$$

$$PM_{MLRSP} = 0.895$$

$$PM_{MLRCP} = 0.882 - 0.061A - 0.019AEF$$

$$PM_{MSBLR} = 0.873 - 0.055A + 0.021E - 0.009DE \\ - 0.010ADE$$

$$PM_{SPMLR} = 0.952 - 0.021A + 0.013B + 0.022E + 0.10AE \\ - 0.010BE$$

$$PM_{CPMLR} = 0.951 - 0.021A + 0.012B + 0.024E \\ + 0.012AE - 0.010BE$$

$$PM_{MM} = 0.929 - 0.021A + 0.008B + 0.024E \\ + 0.010AE + 0.007ACDEF$$

The following table includes the main factor coefficient of effects by methods for PM values of Woollard research and the results of the methods examined in this thesis effort.

FACTORS	μ	A	B	C	D	E	F
METHODS							
MSE	.780	-.100	0	0	.002	.006	.006
Sp	.850	-.070	0	0	.002	.007	.007
Cp	.840	-.070	0	0	.003	.007	.008
Miller's	.880	-.040	.010	0	0	.020	0
MLRMSE	.891	-.050	0	0	0	0	0
MLRSp	.895	0	0	0	0	0	0
MLRCp	.882	-.060	0	0	0	0	0
MSEMLR	.873	-.055	0	0	0	.021	0
SpMLR	.952	-.021	.013	0	0	.022	0
CpMLR	.951	-.021	.012	0	0	.022	0
MM	.929	-.021	.008	0	0	.024	0

Table 5-2.

Summary Table of the Effects of Main Factors for PM measurement.

Results for PM Measurement

1. Intercept (μ)

The highest mean PM value was calculated for the SpMLR Method (0.952), second the CpMLR Method (0.951). The MM Method has the third largest PM value (0.929) and the smallest PM value is that of the MSE method (0.78).

2. Factor A (Number of Extraneous Variables)

All methods are influenced by the number of extraneous variables, except the MLRSp method. The PM value decreases with increases in the number of extraneous variables. The most influenced method is the MSE method (-0.100). The MM Method is one of the least influenced (-0.021).

3. Factor B (Correlation of Independent Variables)

Miller's Method, SpMLR, CpMLR and MM methods are influenced by this factor.

4. Factor C (Variance of Extraneous Variables)

None of the methods are influenced by this factor.

5. Factor D (Variance of Independent Variables)

This factor has an influence on MSE, Sp and Cp methods, but it is a small (0.002-0.003).

6. Factor E (Sample Size)

All methods, except MLRMSE, MLRSp and MLRCp are influenced by this factor. The MM Method is the most influenced one (0.024).

7. Factor F (Error Term)

Only MSE, Sp and Cp methods are influenced by this factor.

FM Equations

By editing the PM.DAT file created by Woollard, a 2^6 full factorial design matrix was augmented with the FM values calculated for each subset selection method at each design point. This data file was used as an input to the SAS file where a stepwise regression procedure was run to generate the following equations.

$$FM_{MSE} = 0.384 - 0.0155A + 0.078E$$

$$FM_{Sp} = 0.495 - 0.101A + 0.175E$$

$$FM_{CP} = 0.487 - 0.102A + 0.032D + 0.173E + 0.032DF$$

$$FM_{Miller} = 0.433 - 0.064A + 0.061B + 0.042D + 0.036BD \\ + 0.173E - 0.025BE - 0.021BDE - 0.030F \\ + 0.019ABF + 0.027DF + 0.029BDF$$

$$FM_{MLRMSE} = 0.406 - 0.078A + 0.063B - 0.034AB + 0.035D \\ + 0.027BD + 0.170E - 0.031AE - 0.032F \\ + 0.035DF$$

$$FM_{MLRSP} = 0.416 - 0.082A + 0.062B - 0.033AB + 0.040D \\ + 0.027BD + 0.170E - 0.031AE - 0.032F \\ + 0.035DF$$

$$FM_{MLRCP} = 0.390 - 0.099A + 0.058B - 0.032AB + 0.034D \\ - 0.025AD + 0.032BD + 0.177E - 0.039AE \\ - 0.030F + 0.034DF$$

$$FM_{MSEMLR} = 0.443 - 0.083A + 0.062B + 0.031D + 0.034BD \\ + 0.170E - 0.031F + 0.034DF$$

$$FM_{SpMLR} = 0.499 - 0.035A + 0.074B + 0.061D + 0.042BD \\ + 0.247E - 0.028BDE - 0.050F + 0.025DF \\ + 0.028BDF$$

$$FM_{CPMLR} = 0.503 - 0.041A + 0.076B + 0.058D + 0.038BD \\ + 0.250E - 0.027BDE - 0.041F + 0.033DF$$

$$FM_{MM} = 0.485 - 0.061A + 0.085B + 0.043D + 0.042BD \\ + 0.215E - 0.023BDE - 0.032F + 0.030DF \\ + 0.027BDF$$

The following table includes the main factor coefficient of effects by method for FM values.

FACTORS	μ	A	B	C	D	E	F
METHODS							
MSE	.384	-.155	0	0	0	.078	0
Sp	.495	-.101	0	0	0	.178	0
Cp	.487	-.102	0	0	.032	.178	0
Miller's	.433	-.064	.061	0	.042	.173	-.030
MLRMSE	.406	-.078	.063	0	.035	.170	-.032
MLRSp	.416	-.082	.062	0	.040	.187	-.030
MLRCp	.390	-.099	.058	0	.034	.177	-.030
MSEMLR	.443	-.083	.062	0	.031	.170	-.031
SpMLR	.499	-.035	.074	0	.061	.247	-.050
CpMLR	.503	-.041	.076	0	.058	.250	-.041
MM	.485	-.061	.085	0	.043	.215	-.032

Table 5-3.

Summery table of FM values of the effects of the main factors.

Results of FM Measurement

1. Intercept (μ)

The highest mean FM value was calculated for CpMLR Method (0.503). This method was followed with very small difference by FM value calculated for SpMLR method. The MM Method has the fifth large FM value, but the difference between the first and fifth one is not too large.

2. Factor A (Number of Extraneous Variables)

All methods are influenced by the number of extraneous variables. The least influenced one is SpMLR (-0.035) and the most influenced method is MSE (-0.155). The MM Method is the third least influenced method (-0.061).

3. Factor B (Correlation of Independent Variables)

Only MSE, Sp and Cp methods are not influenced by this factor. The most affected one is the MM Method (0.085).

4. Factor C (Variance of the Extraneous Variables)

None of the methods are influenced by this factor.

5. Factor D (Variance of Independent Variables)

All methods, except MSE and Sp are influenced by this factor. The most affected one is SpMLR Method (0.061)

6. Factor E (Sample Size)

All methods are influenced by this factor and it is the dominant one. The most affected method is CpMLR (0.250) and MM Method is the third most affected with coefficient 0.215.

7. Factor F (Error Term)

All methods are influenced by the Error Term except MSE, Sp and Cp methods. The most influenced one is SpMLR Method.

Analysis Based on Mean and Standard Deviation of PM and FM Values

In order to get better insight of the performance of the examined methods the mean and the standard deviation of their PM and FM values are calculated, and the results are given in the table 5-4.

	PM Values		FM Values	
	Mean	St.Dev.	Mean	St.Dev.
Min. MSE	0.7790	0.0985	0.3840	0.1894
Min. Sp	0.8500	0.0772	0.4940	0.2277
Min. Cp	0.8430	0.0765	0.4857	0.2260
Miller's	0.8770	0.0526	0.4320	0.2176
MLRMSE	0.8908	0.0694	0.4059	0.2242
MLRSp	0.8948	0.1205	0.4158	0.2418
MLRCp	0.8820	0.0767	0.3903	0.2381
MSEMLR	0.8733	0.0664	0.4425	0.2219
SpMLR	0.9517	0.0441	0.4992	0.2891
CpMLR	0.9513	0.0432	0.5033	0.2921
MM Mth.	0.9286	0.0391	0.4849	0.2618

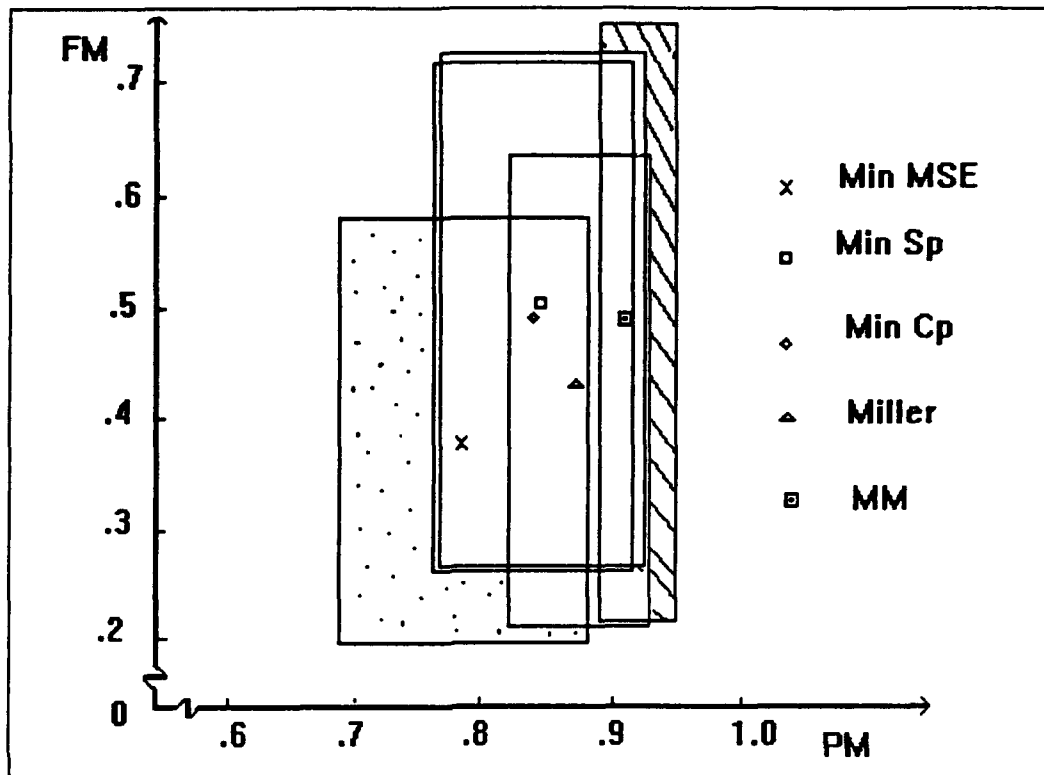
Table 5-4.

The mean and standard deviation of PM and FM value of the employed subset selection technique.

In order to have clear and understandable graphs, the analysis of the methods according to their PM and FM mean value and standard deviation is performed in four steps. The width of the boxes drawn in the following graphs represent the standard deviation of the PM values, and the height of the boxes represent the standard deviation of the FM values. The character in the center of a box represents the mean value of the performance measurements of the studied technique. For example the mean PM value of Minimum MSE method displayed in Table 5-4 is 0.7790 and mean FM value is 0.3840. It is displayed in (0.7790,0.3840) coordinates of PM-FM plane as x. Standard deviation of PM value is displayed as a horizontal line beginning from $0.7790 - 0.0985 = 0.6805$ and ending at $0.7790 + 0.0985 = 0.8775$. The same way the vertical line was computed, which demonstrates the one standard deviation from the mean in FM values.

Step 1: Analysis of one step subset selection method

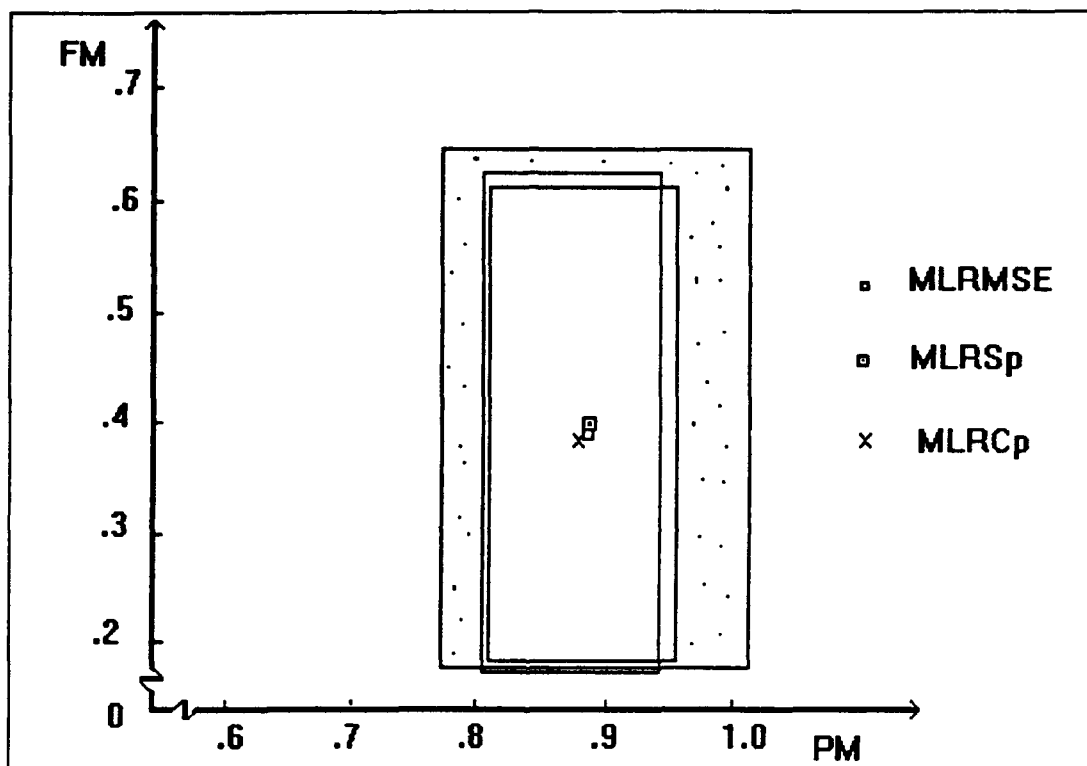
This analysis considers Minimum MSE, Minimum Sp, Minimum Cp, Miller's method and the MM method. The results are displayed in Graph 5-1.



Graph 5-1.

Considering PM measurement it is observed that MM method performs much better than other methods, since it has the largest mean and the smallest standard deviation. Considering the FM measurements of Sp, MM and Cp methods one can tell that they have almost the same mean value. The variance of MM method is higher than the others. But considering the two measurements together, MM method looks as the best one. For the tradeoff having large variance in FM measurement it performs wonderful in PM measurement.

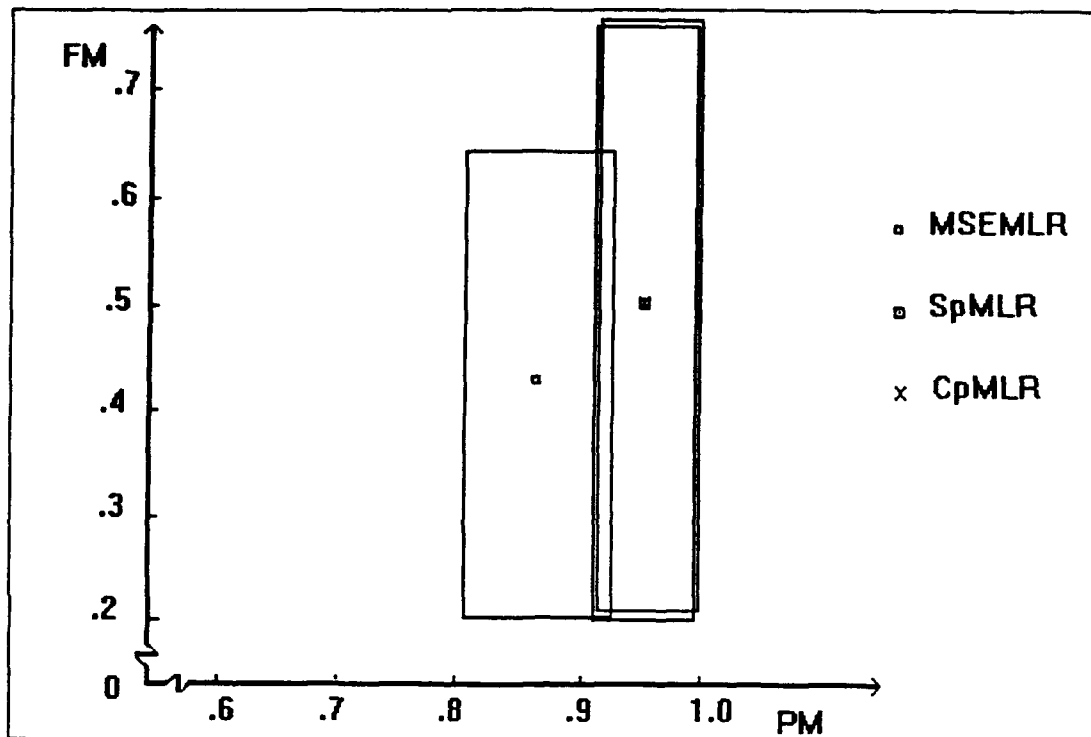
Step 2: Analysis of two stage subset selection methods, where Miller's method is used for screening. This analysis is displayed in Graph 5-2.



Graph 5-2.

Considering both performance measurements, none of the methods perform clearly better than the others. But MLRMSE method can be selected as the best one since it has the smallest standard deviation.

Step 3: Analysis of two stage subset selection methods, where Miller's method is used for final model selection. Analysis is displayed in Graph 5-3.

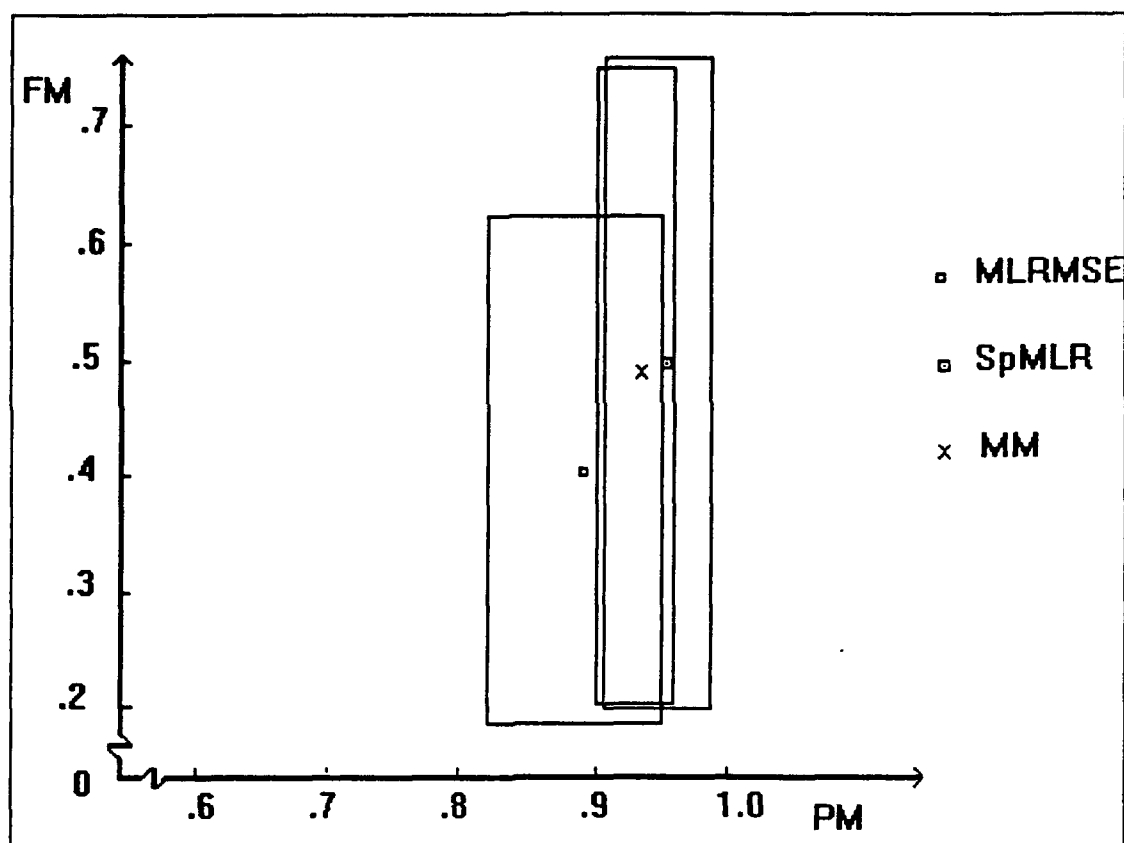


Graph 5-3.

It is observed that SpMLR and CpMLR methods performed almost the same and are each better than the MSEMLR method. So, for the comparison in the step 4 either of them can be selected, let's say SpMLR method.

Step 4: Search for the best of the best subset selection methods.

In this analysis the methods which are considered the best in the previous three steps are compared. The analysis is displayed in Graph 5-4.



Graph 5-4.

According to this final graph SpMLR method (CpMLR is equivalent) performs the best. The mean values of SpMLR method is little bit higher than the mean of the MM method, but the standard deviation of MM method is slightly smaller than the standard deviation of SpMLR method. Because MM method is a one stage method and is not considering all-possible subsets, for larger pool of variables it could be preferable to the SpMLR method, which would require much computational time and effort.

VI. Conclusion and Recommendations

For Further Research

Conclusion

Objective. The objectives of this thesis research were: (1) introduce and study a modification of Miller's method which we call the modified Miller's method (MM method), (2) study two step subset selection procedures, one of which applies Miller's method in the first step and employs Minimum MSE or Minimum Sp or Minimum Cp in the second step and other of which employs Minimum MSE or Minimum Sp or Minimum Cp in the first step and applies Miller's Method in the second step. (3) Compare the results of all techniques, including the results of methods studied by Hansen in 1988 and Woollard in 1993 (Minimum MSE, Minimum Sp, Minimum Cp and Miller's method).

Techniques Studied. Techniques studied in this research: (1) Applying Miller's method in the first step for screening and Minimum MSE or Minimum Sp or Minimum Cp in

the second step. These techniques were named MLRMSE, MLRSp and MLRCp, (2) Applying Minimum MSE or Minimum SP or Minimum Cp in the first step and Miller's method in the second step. These technique were named MSEMLR, SpMLR and CpMLR, and, (3) Modified-Miller's Method (MM method) was implemented for the subset selection.

Methodology. In the two stage subset selection method applied in this research, essentially two approaches have been used. In the first one Miller's method was employed for screening and Minimum MSE, or Minimum Sp or Minimum Cp was used for final model selections. In the second approach data were screened by Minimum MSE or Minimum Sp or Minimum Cp and Miller's method was used for final model selection. Modifications were done in Miller's method by augmenting the variable pool with known extraneous variables where the number of known extraneous variables was based upon sample size of the original data, and multiple runs were done where the data of original variables are fixed and the data of known extraneous variables are changed for each run. This new procedure was named Modified-Miller's method (MM method). For the evaluation of the performance of each method two performance measurements were employed. The first one called performance Measurement (PM), measured the

percentage of correct variables in a model. The second measurement called Full Model (FM) measured the percentage of the number of models consisting of {X1 X2 X3} to the number of all models generated.

Response surface methodology was employed to observe the influence of statistical characteristics of the data to the performance of each method. It is observed that for PM measurement SpMLR method has the highest mean value (0.952) and CpMLR method is second with mean 0.951. The MM method performed the third best with mean value 0.929. The worst performing method is Minimum MSE method with mean 0.780. MLRSp method was not influenced by any of the factors, and factor C (variance of extraneous variables) has no influence on any method. The factor A (Number of extraneous variables) is one of the domain factors. The most influence it has is upon Minimum MSE method with coefficient -0.100 and the small influence upon SpMLR CpMLR and MM method with coefficient -0.021 for all. The other domain factor is Sample size (factor E). The most influenced method is MM method with value 0.024 and MLRMSE, MLRSp and MLRCp method were not influenced by this factor. Thus, considering these influence of the characteristics of the data, SpMLR, CpMLR and MM method can be used for screening the extraneous variables and their performance can be

increased with the increased sample size. If there is not much insight about the characteristics of the data on hand, the MLRSp method can be employed for screening the variables with the tradeoff having less mean performance value. In terms of finding the right number of real predictors, FM measurement indicates that CpMLR and SpMLR performs the best with mean values 0.503 and 0.499 respectively. The worst performing method is Minimum MSE with mean (0.384). The MM method performed fairly good with mean 0.485. The sample size (factor E) is the domain factor for getting better results. The most influenced methods by this factor are CpMLR, SpMLR and MM methods (0.250, 0.247 and 0.215 respectively). Therefor, if increasing the sample size is easily possible employing one of these methods will provide better results for finding the right number of real predictors.

For the evaluation of the influence of factors considering both PM and FM measurement, mean values and their standard deviations were computed. In terms of the mean values and standard deviation criteria it is observed that SpMLR and CpMLR methods performing the best. The MM method is the next one with a very small difference. It should be considered that in this thesis research because of the computer allocation problems MM method is applied with 6

runs, which is smaller than what was recommended. In order to observe the influence of this limitation 8 random set of data including 3 extraneous were selected and MM method was employed with run=60. The comparative results of having six runs and having sixty runs were given in the Table 6-1.

Des. Pnt.-Data Set	60 Replications	6 Replications
06-03	X3 X1 X2	X3 X1 X2
06-08	X2 X3 X1	X2 X3 X1
34-03	X1	X1 E2 X2 E1
40-03	X1 X2	X1 X2
46-26	X1 E3 E1	X1 E3 E1
50-53	X2	X2
54-03	X2 E3	X2 E3
60-10	X2 X1 X3	X2 X1 X3
PM Value	0.854	0.791
PM Value	0.375	0.375

Table 6-1.

In the Table 6-1 the rows are labeled with the design point and data set, columns are labeled with the number of runs.

According to the results of Table 6-1, the only difference is observed for the results of design point 34 and set 3, therefor, having six runs is not a bad choice but having sixty runs provides better results.

Considering the results of Table 6-1, the MM method is expected to perform better if it were applied with the recommended run numbers. Therefor, it would be hard to tell that CpMLR or SpMLR performs better than MM method. One other reason to favor the MM method is its having a smaller standard deviation. The last and the most important reason to consider the MM method as the best one is the application of either the CpMLR or the SpMLR methods is hard for a variable pool of large size.

Recommendations For Further Research

This research can address the following further researches:

1. To expand the number of factors under consideration.

To test the behavior and the performance of the methods compared in this research to different statistical

characteristics not included in this research such as negative correlation, larger sample size and variance, may give better insight of the phenomena. The researcher should be warned that he could have computer allocation problems. To eliminate that problem he could reduce number of sets from 60 to, say 30.

2. Running MM method for large run number.

It will be easy to run MM method for large number of replication, say 30. This results can be compared with the results of methods studied in this thesis.

3. Applying MM method to a large set.

Applying MM method to a pool of variables where the number of variables is large, say 25. An embellishment of this could be applying MM method 5 times to 5 different combination of 5 variables chosen from variable pool of size 25.

Definition of Flow Chart Symbols



represents FORTRAN and SAS codes

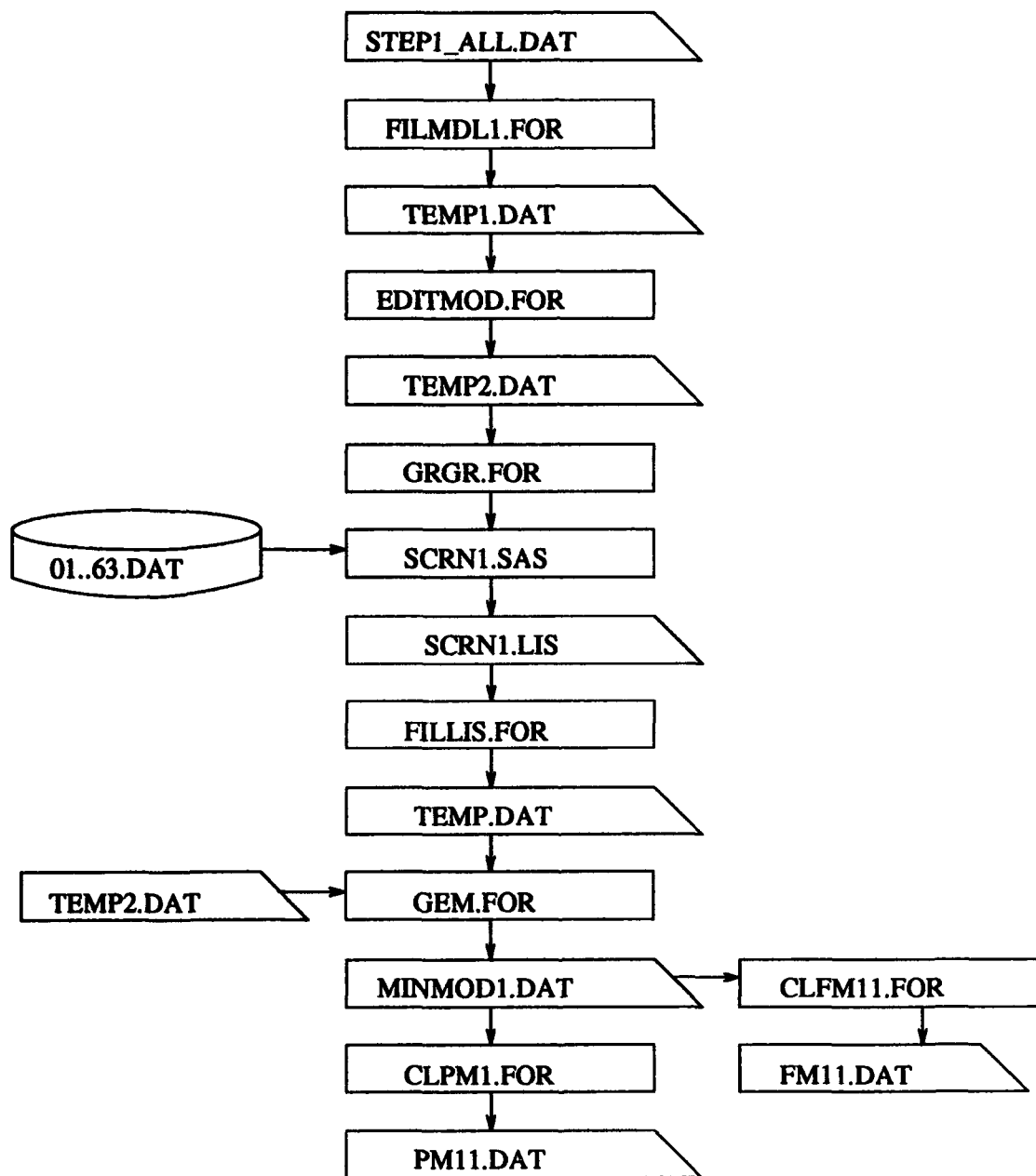


represents FORTRAN *.dat files and SAS *.lis files

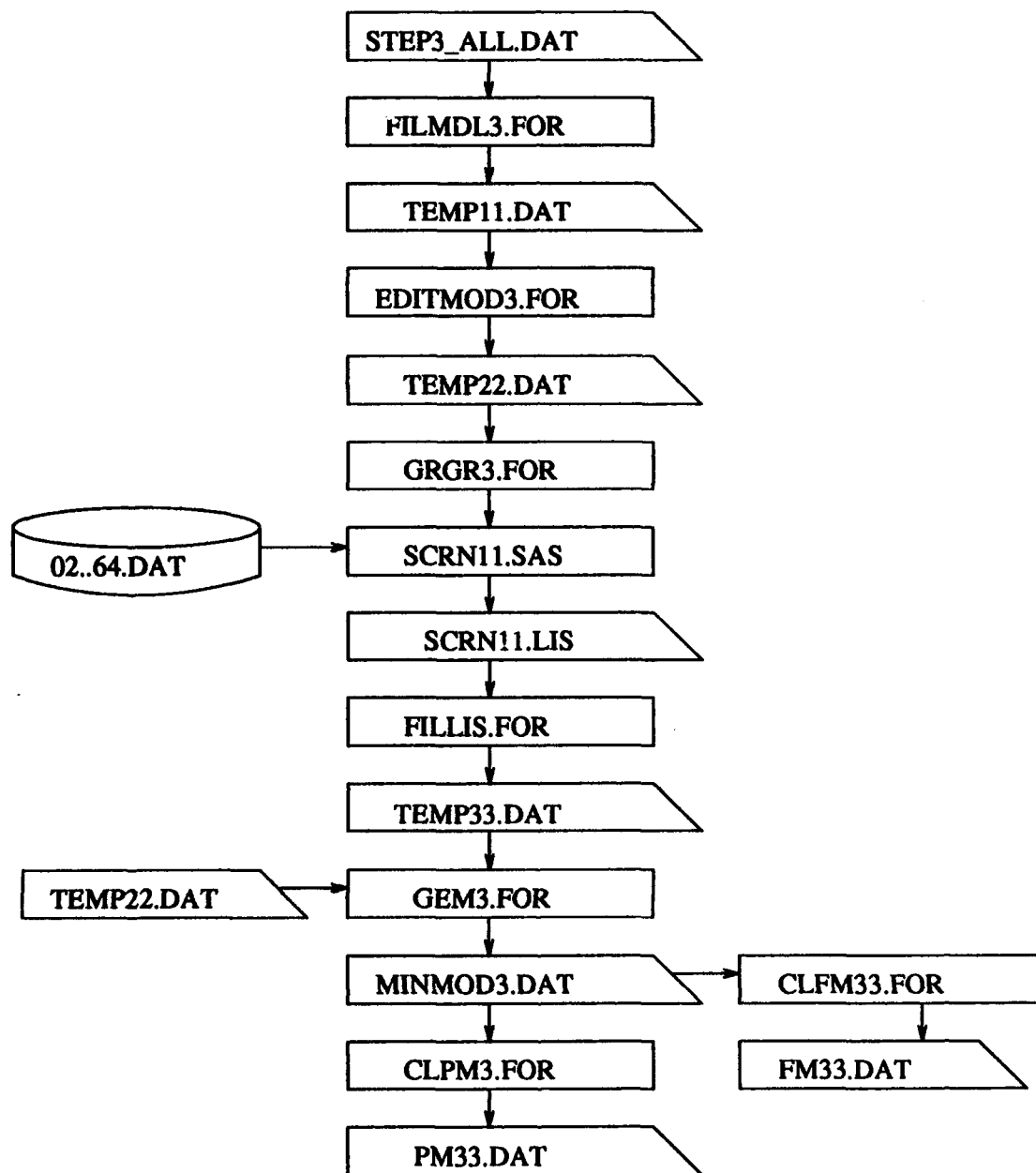


represents the generated data files

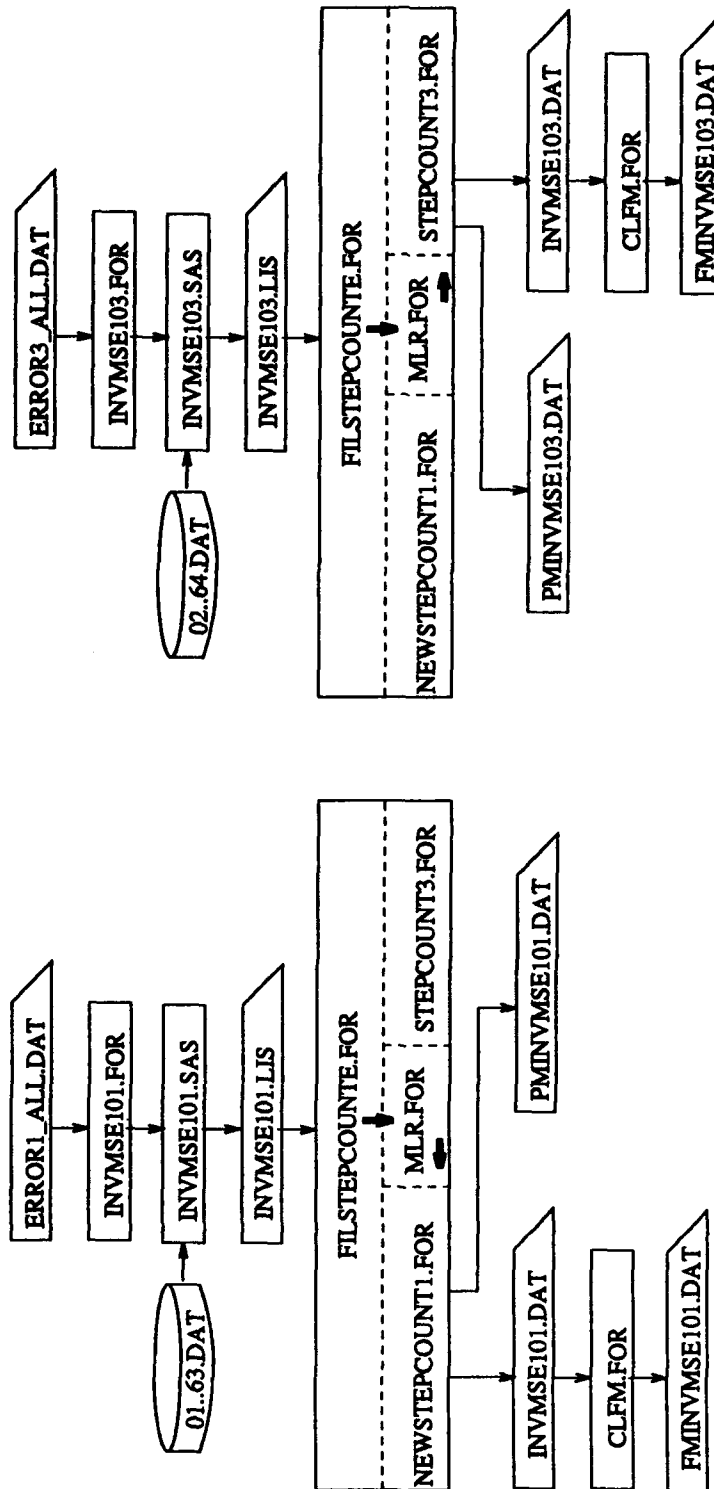
**APPENDIX A : Flow chart for two stage subset selection where Miller's method is used for screening
(1 extraneous variable)**



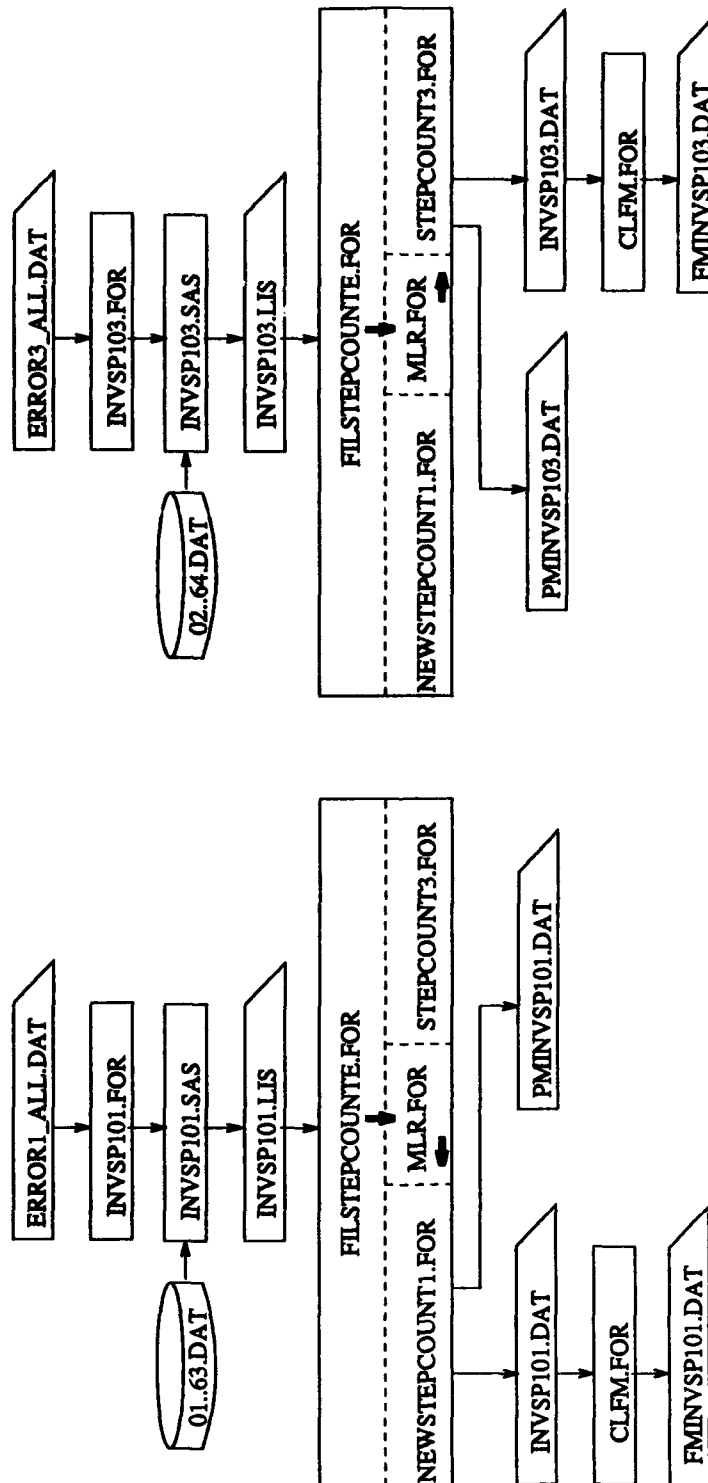
APPENDIX B : Flow chart for two stage subset selection where Miller's method is used for screening
(3 extraneous variable)



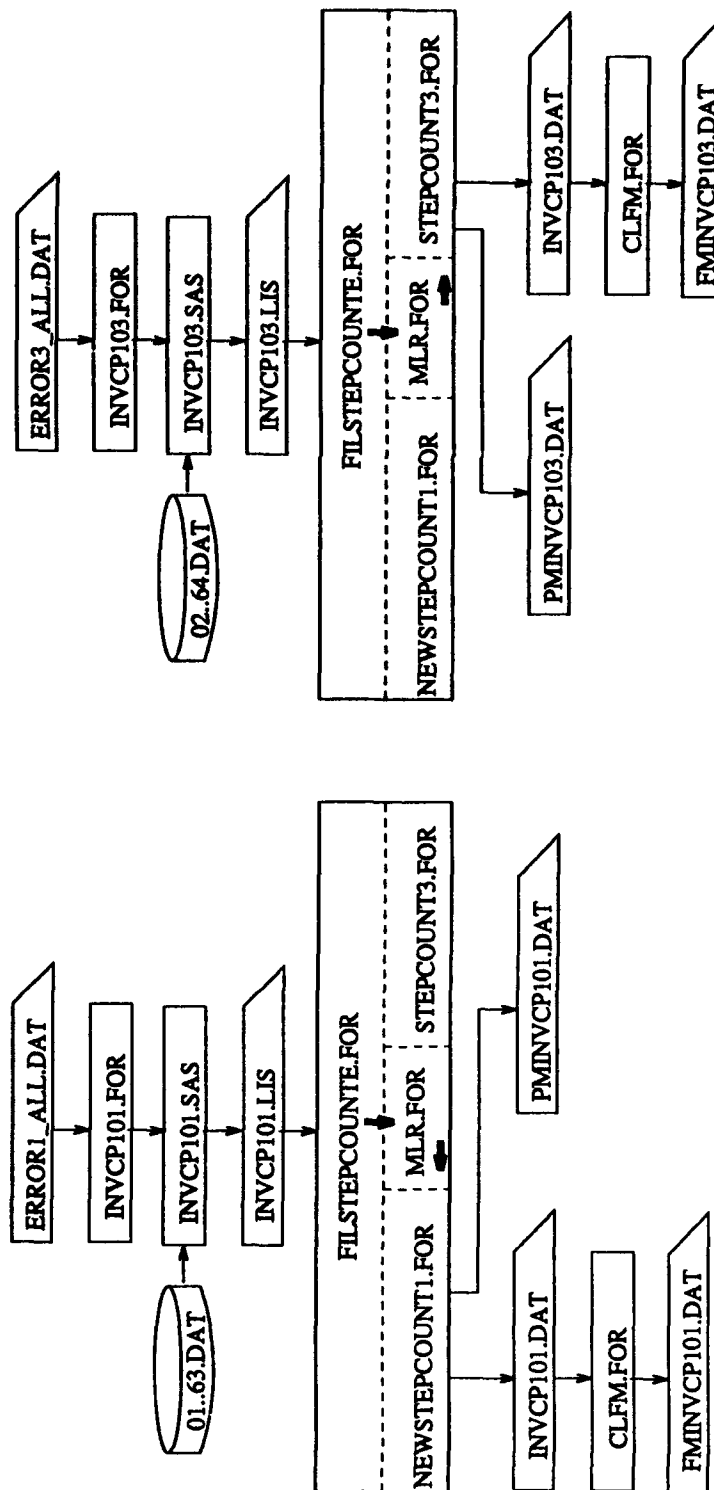
APPENDIX C : Flow chart for two stage subset selection where Minimum MSE is used for screening and Miller's method is used for final model selection



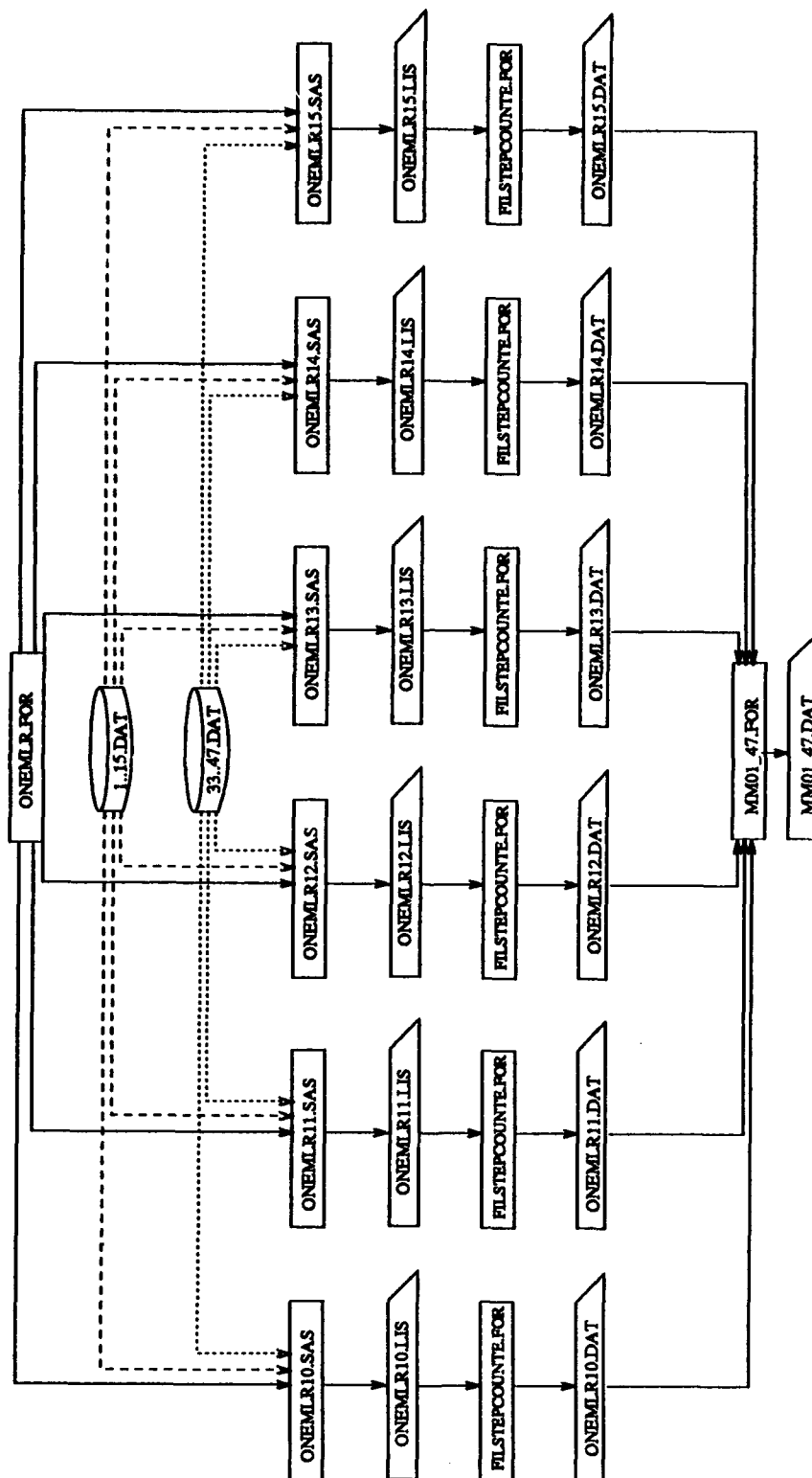
APPENDIX D : Flow chart for two stage subset selection where Minimum Sp is used for screening and
Miller's method is used for final model selection



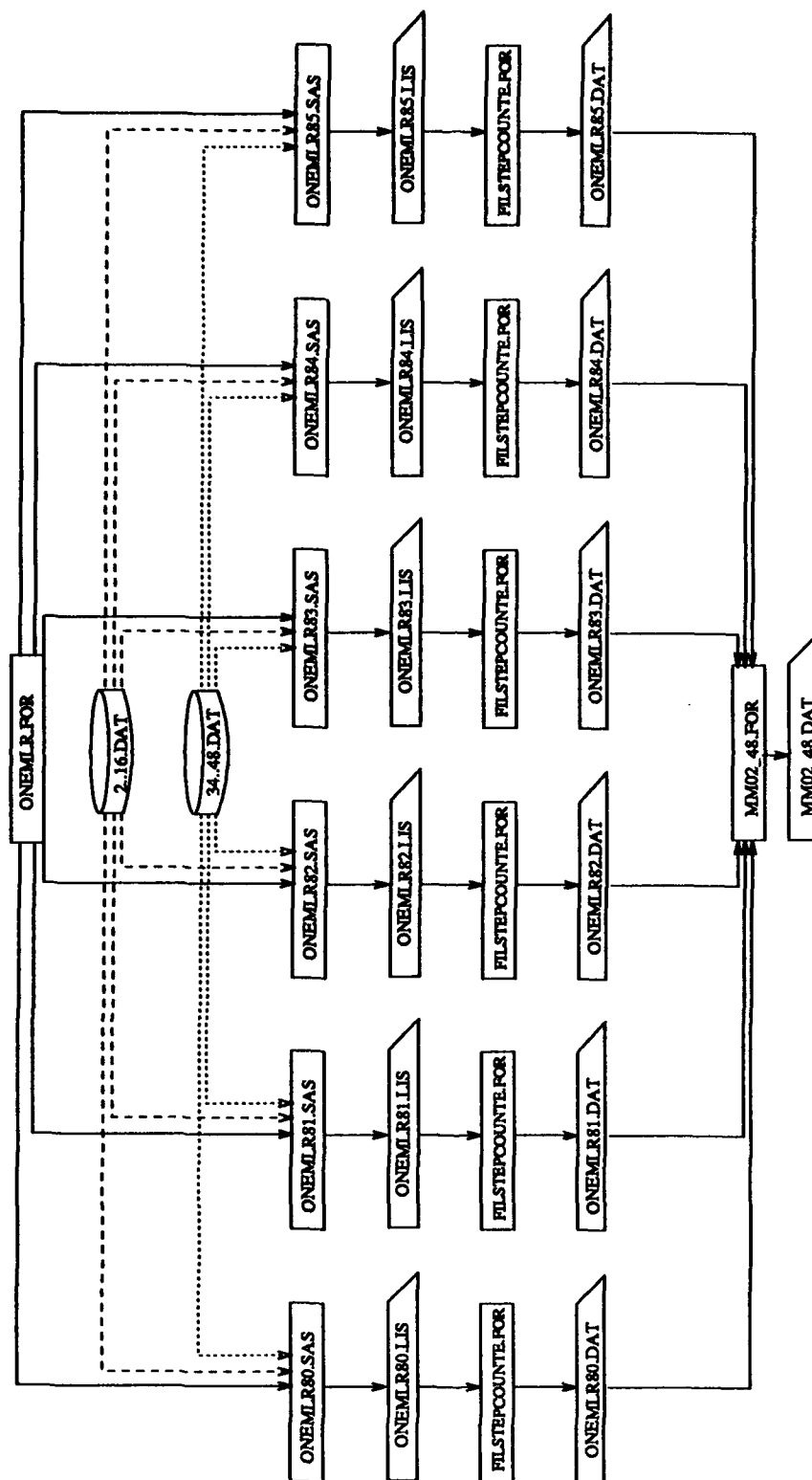
APPENDIX E : Flow chart for two stage subset selection where Minimum Cp is used for screening and Miller's method is used for final model selection



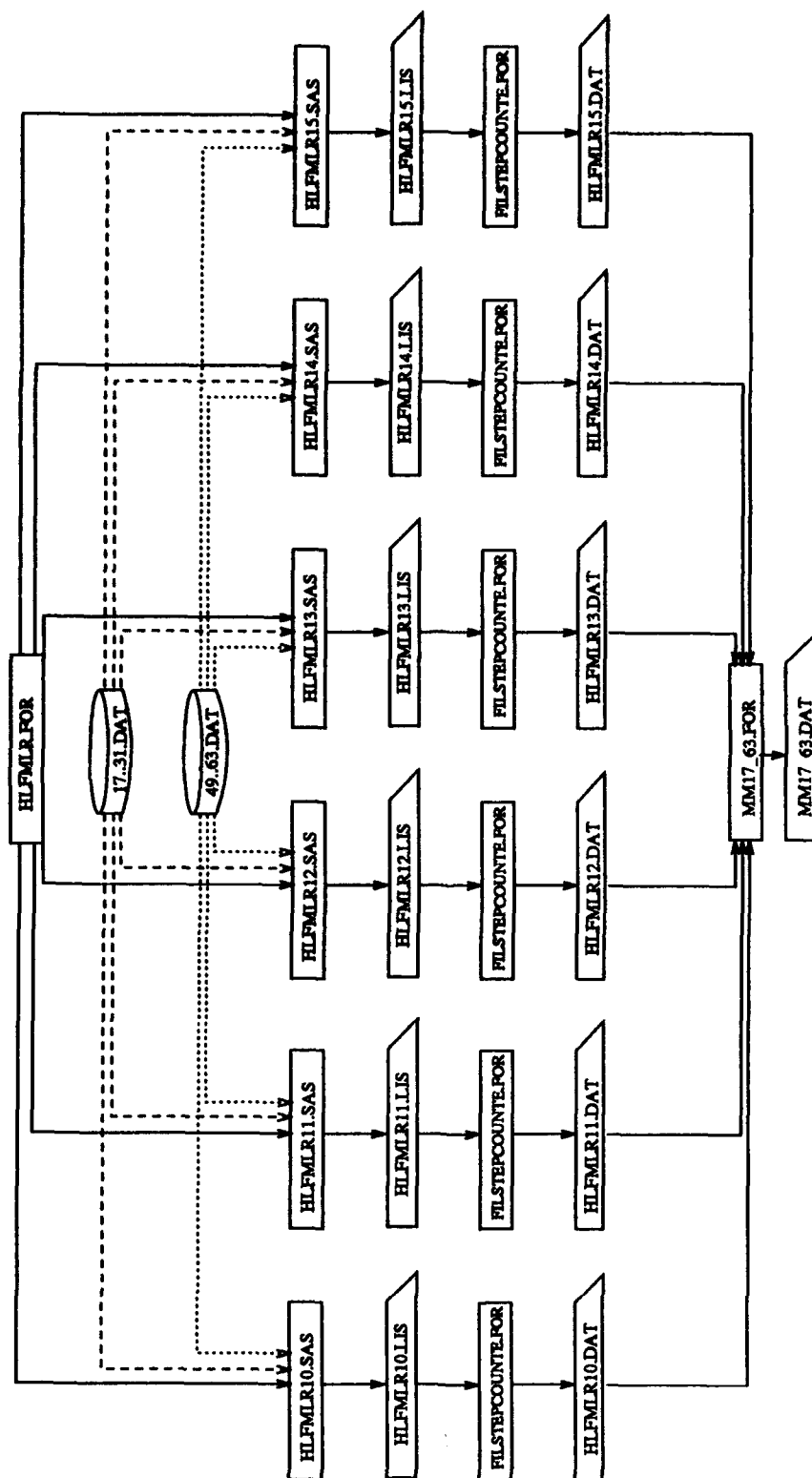
APPENDIX F : Flow chart for MM method for sample size 10 and 1 extraneous



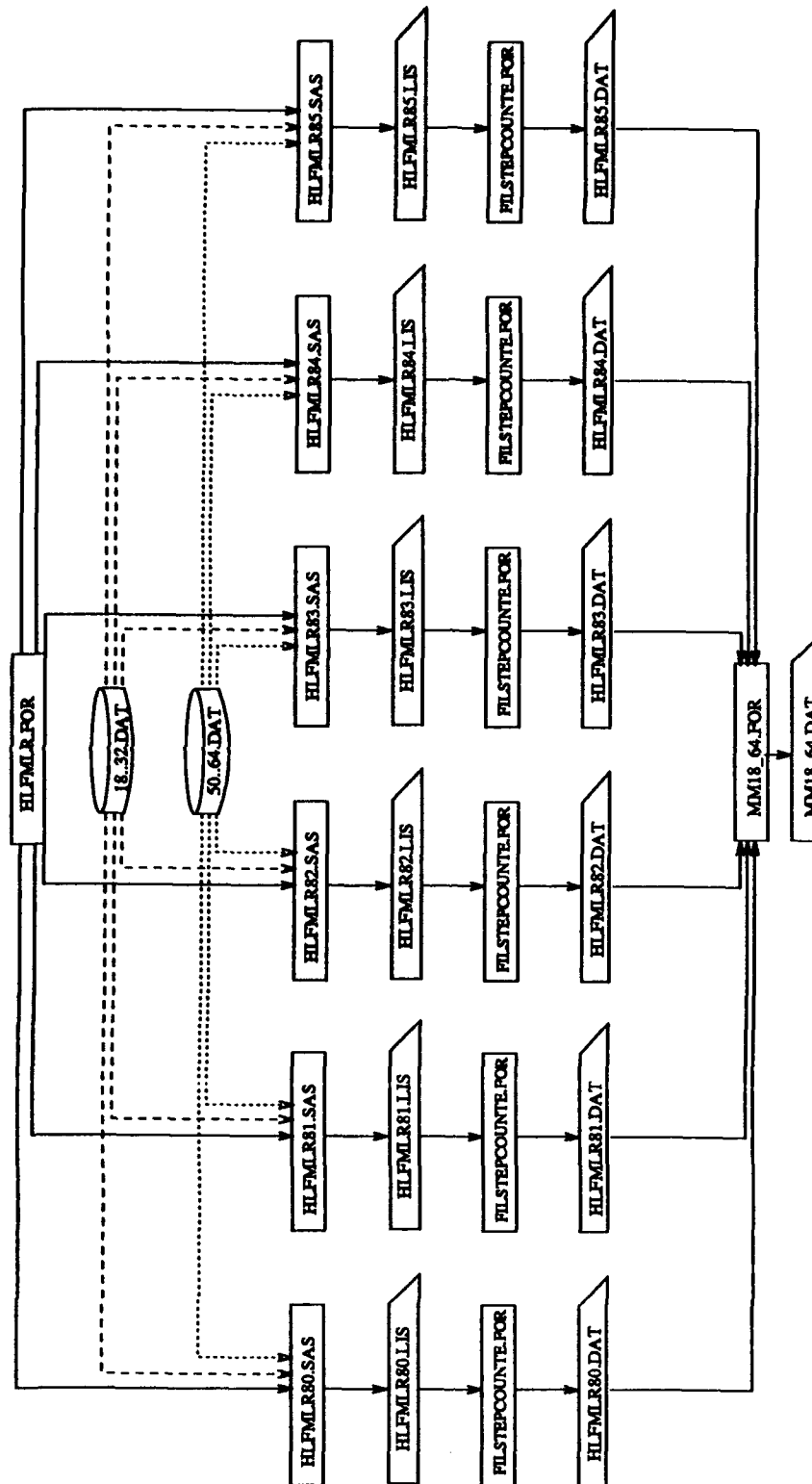
APPENDIX G : Flow chart for MM method for sample size 10 and 3 extraneous



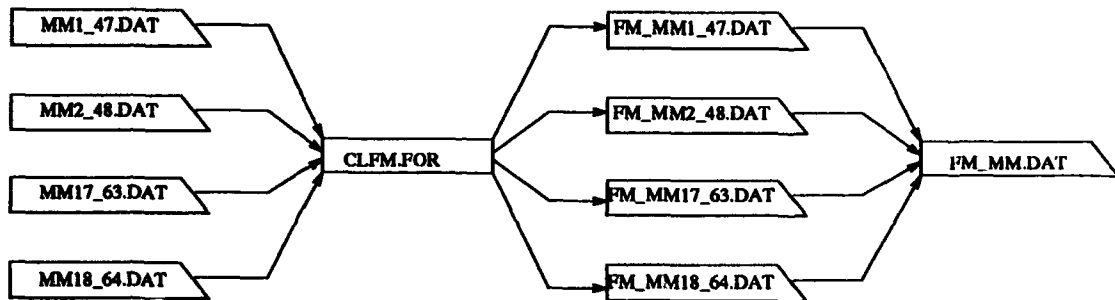
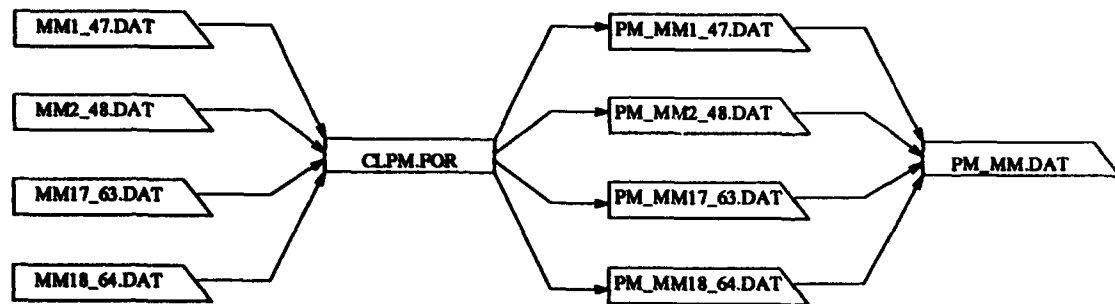
APPENDIX H : Flow chart for MM method for sample size 20 and 1 extraneous



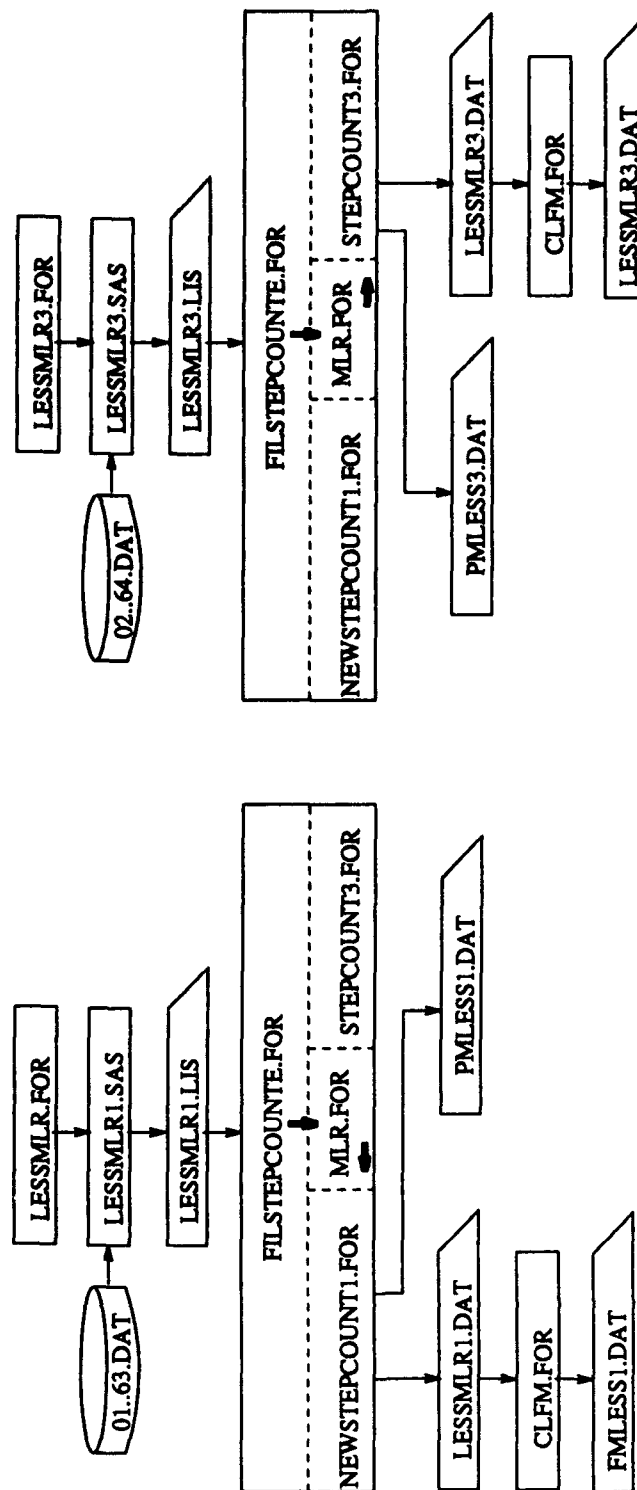
APPENDIX I : Flow chart for MM method for sample size 20 and 3 extraneous



APPENDIX J : Combining different data files and finding PM and FM value for all design points



APPENDIX K : Flow chart for Miller's method with augmentation coefficient 0.5



Note : For coefficients 1, 1.5 and 2 change "LESS" to "ONE", "HLF" and "DBL" respectively

Appendix L:

FORTRAN Programs

CLFM11.FOR	97
CLFM33.FOR	98
CLPM1.FOR	99
CLPM3.FOR	101
EDITMOD.FOR	103
EDITMOD.FOR	104
FILLIS.FOR	105
FILMD1.FOR	108
FILMDL3.FOR	111
FILSTEPCOUNT1.FOR	114
GEM.FOR	120
GEM3.FOR	122
GRGR.FOR	124
GRGR3.FOR	125
HLFMLR.FOR	126
INVMSE103.FOR	127
INVSP101.FOR	130
INVSP103.FOR	132
LESSMLR.FOR	135
MM1_47.FOR	136
MM17_63.FOR	137
MM18_64.FOR	139
MM2_48.FOR	141
NEWSTEPCOUNT1.FOR	143
ONEMLR.FOR	150
STEPCOUNT3.FOR	151

***** CLFM11.FOR *****

```

character*80 line
character*3 t
character*11 m
open(unit=1,file='minmod1.dat',status='old')
open (unit=2,file='fm11.dat',status='new')
do 10 k=1,63,2
  fmmse=0
  fmsp=0
  fmcp=0
  do 20 j=1,180
    read (1,500,end=200) line
    t= line(2:4)
    m= line(62:72)
    IF (t.EQ.'MSE') goto 15
    IF (t.EQ.'Sp ') goto 25
    IF (t.EQ.'Cp ') goto 35

15    IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  '))
      +fmmse=fmmse+1
      IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  '))
      +fmmse=fmmse+1
      IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  '))
      +fmmse=fmmse+1
      goto 20

25    IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  ')) fmsp=fmsp+1
      IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  ')) fmsp=fmsp+1
      IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  ')) fmsp=fmsp+1
      goto 20

35    IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  ')) fmcp=fmcp+1
      IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  ')) fmcp=fmcp+1
      IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  '))
      +fmcp=fmcp+1

20    CONTINUE
      vmse=fmmse/60
      vsp=fmsp/60
      vcp=fmcp/60
      write(2,*) k, vmse ,vsp, vcp
10    continue
500    format(A80)
200    stop
      end

```

***** CLFM33.FOR *****

```

      character*80 line
      character*3 t
      character*11 m
      open(unit=1,file='minmod3.dat',status='old')
      open (unit=2,file='fm33.dat',status='new')
      do 10 k=2,64,2
        fmmse=0
        fmsp=0
        fmcp=0
        do 20 j=1,180
          read (1,500,end=200) line
          t= line(2:4)
          m= line(62:72)
          IF (t.EQ.'MSE') goto 15
          IF (t.EQ.'Sp ') goto 25
          IF (t.EQ.'Cp ') goto 35
15      IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  '))
          +fmmse=fmmse+1
          IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  '))
          +fmmse=fmmse+1
          IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  '))
          +fmmse=fmmse+1
          goto 20
25      IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  ')) fmsp=fmsp+1
          IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  ')) fmsp=fmsp+1
          IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  ')) fmsp=fmsp+1
          goto 20
35      IF((m.EQ.'X1 X2 X3  ').OR.(m.EQ.'X1 X3 X2  ')) fmcp=fmcp+1
          IF((m.EQ.'X3 X1 X2  ').OR.(m.EQ.'X3 X2 X1  ')) fmcp=fmcp+1
          IF((m.EQ.'X2 X1 X3  ').OR.(m.EQ.'X2 X3 X1  '))
          +fmcp=fmcp+1
20      CONTINUE
          vmse=fmmse/60
          vsp=fmsp/60
          vcp=fmcp/60
          write(2,*) k, vmse ,vsp, vcp
10      continue
500      format(A80)
200      stop
      end

```

***** CLPM1.FOR *****

```

real pmmse,pmisp,pmcp
character*3 mth
character*2 mod(4)
integer var,tvarmse,emse,esp,ecp,tvarsp,tvarcp
open (unit=1,file='minmod1.dat',status='old')
open (unit=2,file='pm11.dat',status='new')
do 5 k=1,63,2
    emse=0
    esp=0
    ecp=0
    tvarmse=0
    tvarsp=0
    tvarcp=0
    do 10 I=1,60
        read(1,500,end=200)mth,var,mod(1),mod(2),mod(3),mod(4)
        if (mth.EQ.'MSE') then
            tvarmse=tvarmse+var
            do 20 j=1,4
                if (mod(j).EQ.'E1') then
                    emse=emse+1
                endif
            continue
        else
            if (mth.EQ.'Sp ') then
                tvarsp=tvarsp+var
                do 30 j=1,4
                    if (mod(j).EQ.'E1') then
                        esp=esp+1
                    endif
            continue
        IF (mth.EQ.'Cp ') then
            tvarcp=tvarcp+var
            do 40 j=1,4
                if (mod(j).EQ.'E1') then
                    ecp=ecp+1
                endif
            continue
        ENDIF
    ENDIF
    ENDIF

```

```

10      continue
      dmse=tvarmse-emse
      dsp= tvarsp-esp
      dcp= tvarcp-ecp
      pmmse= dmse / tvarmse
      pmisp= dsp / tvarsip
      pmcp= dcp / tvarcp
      write (2,*) k,pmmse,pmisp,pmcp
5      continue
500     format (1X,A3,11X,I1,45X,A2,1X,A2,1X,A2,1X,A2)
200     STOP
      END

```

***** CLPM3.FOR *****

```

real pmmse,pmsp,pmcp
character*3 mth
character*2 mod(6)
integer var,tvarmse,emse,esp,ecp,tvarsp,tvarcp
open (unit=1,file='minmod3.dat',status='old')
open (unit=2,file='pm33.dat',status='new')
do 5 k=2,64,2
    emse=0
    esp=0
    ecp=0
    tvarmse=0
    tvarsp=0
    tvarcp=0
    do 10 I=1,60
read(1,500,end=200)
+ mth,var,mod(1),mod(2),mod(3),mod(4),mod(5),mod(6)
        IF (mth.EQ.'MSE') then
            tvarmse=tvarmse+var
            do 20 j=1,6
                if((mod(j).EQ.'E1').OR.(mod(j).EQ.'E2'))
+ .OR.(mod(j).EQ.'E3')) then
                    emse=emse+1
                endif
20      continue
            else
                IF (mth.EQ.'Sp ') then
                    tvarsp=tvarsp+var
                    do 30 j=1,6
                        if((mod(j).EQ.'E1').OR.(mod(j).EQ.'E2'))
+ .OR.(mod(j).EQ.'E3')) then
                            esp=esp+1
                        endif
30      continue
                    else
                        IF (mth.EQ.'Cp ') then
                            tvarcp=tvarcp+var
                            do 40 j=1,4
                                if ((mod(j).EQ.'E1').OR.(mod(j).EQ.'E2'))
+ .OR.(mod(j).EQ.'E3')) then
                                    ecp=ecp+1
                                endif

```

```

40          continue
          ENDIF
          ENDIF
          ENDIF

10      continue
      dmse=tvarmse-emse
      dsp= tvarsp-esp
      dcp= tvarcp-ecp
      pmmse= dmse / tvarmse
      pmisp= dsp / tvarsip
      pmcp= dcp / tvarcp
      write (2,*) k,pmmse,pmisp,pmcp

5      continue
500     format (1X,A3,11X,I1,45X,A2,1X,A2,1X,A2,1X,A2)
200     STOP
      END

```

***** EDITMOD.FOR *****

```
      INTEGER REP ,VAR
      CHARACTER*12 MOD
      OPEN (UNIT=1,FILE='TEMP1.DAT',STATUS='UNKNOWN')
      OPEN (UNIT=2,FILE='TEMP2.DAT',STATUS='NEW')
      COUNT=0
      DO 10 K=1,1920
        READ(1,500,END=200) REP, VAR, MOD
        IF (VAR.EQ.0) THEN
          VAR=1
          MOD='x4'
          COUNT=COUNT+1
        ENDIF
      WRITE (2,500) REP, VAR, MOD
10    CONTINUE
500   FORMAT (5X,I2,12X,I1,6X,A12)
      PRINT*, '# OF ZERO MODELS=',COUNT
200   STOP
      END
```

***** EDITMOD3.FOR *****

```
      INTEGER REP ,VAR
      CHARACTER*12 MOD
      OPEN (UNIT=1,FILE='TEMP11.DAT',STATUS='UNKNOWN')
      OPEN (UNIT=2,FILE='TEMP22.DAT',STATUS='NEW')
      COUNT=0
      DO 10 K=1,1920
         READ(1,500,END=200) REP, VAR, MOD
         IF (VAR.EQ.0) THEN
            VAR=1
            MOD='x4'
            COUNT=COUNT+1
         ENDIF
      WRITE (2,500) REP, VAR, MOD
10    CONTINUE
500   FORMAT (5X,I2,12X,I1,6X,A12)
      PRINT*,'# OF ZERO MODELS=',COUNT
200   STOP
      END
```

***** FILLIS.FOR *****

```

Character*20 NewIn
Character*20 NewOut
Character*80 Line
CHARACTER I, J
Integer Var
Logical VarFlag

5  Continue
   Print *, 'Name of file to examine? (20 char or less;',
+   ' *** to quit)'
   Read (*, '(A20)') NewIn
   If (NewIn(1:1).EQ.'*') GO TO 999
   Print *, 'Output file? (20 char or less)'
   Read (*, '(A20)') NewOut
7  Continue
   Print *, 'Number of extraneous variables?(1 or 3
+ONLY!!)'
   Read (*, '(I1)') Var
   If ((Var.NE.1).AND.(Var.NE.3)) GO TO 7

9  Continue
   VarFlag = (Var.EQ.3)

   Open (unit=10, file=NewIn, status='OLD',
&       iostat=IERROR, err=1000)

   Open (unit=11, file='temp.dat', status='NEW',
&       iostat=IERROR, err=1000)
   open (unit=12, file='temp33.dat', status='NEW',
&       iostat=IERROR, err=1000)

10 Continue
   Read(10,200,END=888) Line
   I = LINE (8:8)
   J = LINE (9:11)

   IF (VarFlag) GO TO 777
   IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN
       WRITE (11,200) LINE
   ELSE

   IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
       WRITE (11,200) LINE
   ELSE

   IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
       WRITE (11,200) LINE

```

```

ELSE

    IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
        WRITE (11,200) LINE
    ENDIF
    ENDIF
    ENDIF
    ENDIF

GO TO 10

777 Continue
    IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN
        WRITE (12,200) LINE
    ELSE
        IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
            WRITE (12,200) LINE
        ELSE
            IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
                WRITE (12,200) LINE
            ELSE
                IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
                    WRITE (12,200) LINE
                ELSE
                    IF ((I.EQ.'5').AND.(J.EQ.' ')) THEN
                        WRITE (12,200) LINE
                    ELSE
                        IF ((I.EQ.'6').AND.(J.EQ.' ')) THEN
                            WRITE (12,200) LINE
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
    ENDIF
    ENDIF

GO TO 10

200 Format (A80)
888 Continue

GO TO 5

999 Continue

```

```

        Print *, 'Processing complete. Program terminated.'
        Stop

* Error trap: *****
1000 Continue
        Print 1100, '+++ ERROR WHILE OPENING FILE +++',
        &          '          error code = ', IERROR
1100 FORMAT(/1X, A/ 1X, A, I8/)
        GO TO 5
*****
        END

```

***** FILMDL1.FOR *****

```

      Character*20 NewIn
      Character*20 NewOut
      Character*80 Line
      CHARACTER  I, J
      Integer Var
      Logical VarFlag

5      Continue
      Print *, 'Name of file to examine? (20 char or less;',
+      ' *** to quit)'
      Read (*, '(A20)') NewIn
      If (NewIn(1:1).EQ.'*') GO TO 999
      Print *, 'Output file? (20 char or less)'
      Read (*, '(A20)') NewOut
7      Continue
      Print *, 'Number of extraneous variables? (1 ONLY!!)'
      Read (*, '(I1)') Var
      If ((Var.NE.1).AND.(Var.NE.3)) GO TO 7

9      Continue
      VarFlag = (Var.EQ.3)

      Open (unit=10, file=NewIn, status='OLD',
&          iostat=IERROR, err=1000)

      Open (unit=11, file='temp1.dat', status='NEW',
&          iostat=IERROR, err=1000)

10     Continue
      Read(10,200,END=888) Line
      I = LINE (20:20)
      J = LINE (21:24)

      IF (VarFlag) GO TO 777

      IF ((I.EQ.'0').AND.(J.EQ.' ')) THEN

          WRITE (11,200) LINE
      ELSE

          IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN

              WRITE (11,200) LINE
          ELSE

```

```

IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
    WRITE (11,200) LINE
ELSE
    IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
        WRITE (11,200) LINE
    ELSE
        IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
            WRITE (11,200) LINE
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

GO TO 10

777 Continue

```

IF ((I.EQ.'0').AND.(J.EQ.' ')) THEN
    WRITE (11,200) LINE
IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN
    WRITE (11,200) LINE
ELSE
    IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
        WRITE (11,200) LINE
    ELSE
        IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
            WRITE (11,200) LINE
        ELSE
            IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
                WRITE (11,200) LINE
            ELSE
                IF ((I.EQ.'5').AND.(J.EQ.' ')) THEN
                    WRITE (11,200) LINE
                ELSE
                    IF ((I.EQ.'6').AND.(J.EQ.' ')) THEN
                        WRITE (11,200) LINE
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```

                ENDIF

                GO TO 10

200   Format (A80)

888   Continue
      Close(10)
      Close(11)

      Print *, 'Filtering complete on ', NewIn, '. COUNTING
BEGUN.'

      GO TO 5

999   Continue

      Print *, 'Processing complete. Program terminated.'
      Stop

* Error trap: *****
1000  Continue
      Print 1100, '+++ ERROR WHILE OPENING FILE +++',
&      '          error code = ', IERROR
1100  FORMAT(/1X, A/ 1X, A, I8/)
      GO TO 5
*****
      END

```

***** FILMDL3.FOR *****

```
Character*20 NewIn
Character*20 NewOut
Character*80 Line
CHARACTER I, J
Integer Var
Logical VarFlag

5  Continue
   Print *, 'Name of file to examine? (20 char or less;',
+   ' *** to quit)'
   Read (*, '(A20)') NewIn
   If (NewIn(1:1).EQ.'*') GO TO 999
   Print *, 'Output file? (20 char or less)'
   Read (*, '(A20)') NewOut
7  Continue
   Print *, 'Number of extraneous variables? (1 ONLY!!!)'
   Read (*, '(I1)') Var
   If ((Var.NE.1).AND.(Var.NE.3)) GO TO 7

9  Continue
   VarFlag = (Var.EQ.3)

   Open (unit=10, file=NewIn, status='OLD',
&       iostat=IERROR, err=1000)

   Open (unit=11, file='temp11.dat', status='NEW',
&       iostat=IERROR, err=1000)

10 Continue
   Read(10,200,END=888) Line
   I = LINE (20:20)
   J = LINE (21:24)

   IF (VarFlag) GO TO 777

   IF ((I.EQ.'0').AND.(J.EQ.' ')) THEN

       WRITE (11,200) LINE
   ELSE

       IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN

           WRITE (11,200) LINE
       ELSE

           IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
```

```

        WRITE (11,200) LINE
    ELSE
        IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
            WRITE (11,200) LINE
        ELSE
            IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
                WRITE (11,200) LINE
            ENDIF
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDIF
GO TO 10
777 Continue
    IF ((I.EQ.'0').AND.(J.EQ.' ')) THEN
        WRITE (11,200) LINE
    ELSE
        IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN
            WRITE (11,200) LINE
        ELSE
            IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN
                WRITE (11,200) LINE
            ELSE
                IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN
                    WRITE (11,200) LINE
                ELSE
                    IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN
                        WRITE (11,200) LINE
                    ELSE
                        IF ((I.EQ.'5').AND.(J.EQ.' ')) THEN
                            WRITE (11,200) LINE
                        ELSE
                            IF ((I.EQ.'6').AND.(J.EQ.' ')) THEN

```

```

WRITE (11,200) LINE

                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF

GO TO 10

200  Format (A80)

888  Continue
      Close(10)
      Close(11)

      Print *, 'Filtering complete on ', NewIn, '. COUNTING BEGUN.'

      GO TO 5

999  Continue

      Print *, 'Processing complete. Program terminated.'
      Stop

* Error trap: *****
1000 Continue
      Print 1100, '+++ ERROR WHILE OPENING FILE +++',
      &      '          error code = ', IERROR
1100 FORMAT(/1X, A/ 1X, A, I8/)
      GO TO 5
*****
      END

```

***** FILSTEPCOUNT.EFOR *****

```

Character*14 NewIn
Character*20 NewOut
Character*80 Line
Character*1 I, J
Character*2 K
Integer Var
Logical VarFlag, BatchFlag

5  Continue
   Print *, 'Interactive(I) or Batch(B) mode? (I or B only):'
   Read (*, '(A1)') Mode
   IF ((Mode.NE.'I').AND.(Mode.NE.'B')) Go to 5
   BatchFlag=(Mode.EQ.'B')
   IF (BatchFlag) Go to 6
   Print *, 'Name of file to examine? (20 char or less;',
+       ' *** to quit)'
   Read (*, '(A20)') NewIn
   If (NewIn(1:1).EQ.'*') GO TO 999
6  Continue
   Print *, 'Output file? (20 char or less)'
   Read (*, '(A20)') NewOut
7  Continue
   Print *, 'Number of extraneous variables? (1 or 3 ONLY!!)'
   Read (*, '(I1)') Var
   If ((Var.NE.1).AND.(Var.NE.3)) GO TO 7

9  Continue
   VarFlag = (Var.EQ.3)

   IF ((VarFlag).AND.(BatchFlag)) THEN
       Open (unit=9, file='Step3_Input.dat', status='OLD',
+         iostat=IERROR, err=1000)

   ELSE
       IF ((.NOT.VarFlag).AND.(BatchFlag)) THEN
           Open (unit=9, file='newstep1_Input.dat', status='OLD',
+             iostat=IERROR, err=1000)

       ENDIF
   ENDIF
11 Continue
   IF (BatchFlag) Read(9, '(A10)', END=666) NewIn
   Print *, 'Filtering begun on ', NewIn, '.  Filtered data ',
+       'is being dumped to MLR.DAT.'

   Open (unit=10, file=NewIn, status='OLD',
&       iostat=IERROR, err=1000)

```

```
Open (unit=11, file='mlr.dat', status='NEW',  
&      iostat=IERROR, err=1000)
```

```
10  Continue  
    Read(10,200,END=888) Line  
    I = LINE (5:5)  
    J = LINE (6:8)  
    K = LINE (4:5)  
  
    IF (VarFlag) GO TO 777  
  
    IF ((I.EQ.'1').AND.(J.EQ.'  ')) THEN  
        WRITE (11,200) LINE  
    ELSE  
  
        IF ((I.EQ.'2').AND.(J.EQ.'  ')) THEN  
            WRITE (11,200) LINE  
        ELSE  
  
            IF ((I.EQ.'3').AND.(J.EQ.'  ')) THEN  
                WRITE (11,200) LINE  
            ELSE  
  
                IF ((I.EQ.'4').AND.(J.EQ.'  ')) THEN  
                    WRITE (11,200) LINE  
                ELSE  
  
                    IF ((I.EQ.'5').AND.(J.EQ.'  ')) THEN  
                        WRITE (11,200) LINE  
                    ELSE  
  
                        IF ((I.EQ.'6').AND.(J.EQ.'  ')) THEN  
                            WRITE (11,200) LINE  
                        ELSE  
  
                            IF ((I.EQ.'7').AND.(J.EQ.'  ')) THEN  
                                WRITE (11,200) LINE  
                            ELSE  
  
                                IF ((I.EQ.'8').AND.(J.EQ.'  ')) THEN  
                                    WRITE (11,200) LINE  
                                ELSE  
  
                                    IF ((I.EQ.'9').AND.(J.EQ.'  ')) THEN  
                                        WRITE (11,200) LINE  
                                    ELSE  
  
                                        IF ((K.EQ.'10').AND.(J.EQ.'  ')) THEN  
                                            WRITE (11,200) LINE  
                                        ELSE
```

```
IF ((K.EQ.'11').AND.(J.EQ.' ')) THEN  
  WRITE (11,200) LINE  
ELSE
```

```
  IF ((K.EQ.'12').AND.(J.EQ.' ')) THEN  
    WRITE (11,200) LINE  
  ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
GO TO 10
```

```
777 Continue  
IF ((I.EQ.'1').AND.(J.EQ.' ')) THEN  
  WRITE (11,200) LINE  
ELSE
```

```
  IF ((I.EQ.'2').AND.(J.EQ.' ')) THEN  
    WRITE (11,200) LINE  
  ELSE
```

```
    IF ((I.EQ.'3').AND.(J.EQ.' ')) THEN  
      WRITE (11,200) LINE  
    ELSE
```

```
      IF ((I.EQ.'4').AND.(J.EQ.' ')) THEN  
        WRITE (11,200) LINE  
      ELSE
```

```
        IF ((I.EQ.'5').AND.(J.EQ.' ')) THEN  
          WRITE (11,200) LINE
```

ELSE

```
IF ((I.EQ.'6').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((I.EQ.'7').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((I.EQ.'8').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((I.EQ.'9').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'10').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'11').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'12').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'13').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'14').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'15').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'16').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'17').AND.(J.EQ.' ')) THEN
  WRITE (11,200) LINE
ELSE
```

```
IF ((K.EQ.'18').AND.(J.EQ.' ')) THEN
```

```

WRITE (11,200) LINE
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
GO TO 10

```

```
200  Format (A80)
```

```
666  Continue
      BatchFlag=.FALSE.
```

```
888  Continue
      IF (BatchFlag) THEN
        Close(10)
        Go to 11
      ELSE
        Close(9)
      
```

```

        Close(10)
        Close(11)
    ENDIF

    Print *, 'Filtering complete.  Analysis of data in',
+         ' MLR.DAT has begun.', ' Analysis results will',
+         ' be dumped to ', Newout, '.'

    IF (VarFlag) THEN
        Call Stepcount3(NewOut)
    ELSE
        Call newstepcount1(NewOut)
    ENDIF
    GO TO 5

999  Continue

    Print *, 'Processing complete.  Program terminated.'
    Stop

* Error trap: *****
1000 Continue
    Print 1100, '+++ ERROR WHILE OPENING FILE +++',
    &         ' error code = ', IERROR
1100 FORMAT(/1X, A/ 1X, A, I8/)
    GO TO 5
*****
    END

```

***** GEM.FOR *****

```
character*12 mod
character*12 modmse
character*12 modsp
character*12 modcp
character*80 line
character var
integer n,num,nummse,numsp,numcp
real minmse,mse,spms,cpms,sp,cp,msecp,cpsp,msecp,spcp
open (unit=1,file='temp2.dat',status='old')
open (unit=2,file='temp.dat', status='old')
open (unit=3,file='minmod1.dat',status='new')
do 10 I=1,1920
    read (1,500,end=200) line
    var=line(20:20)
    IF (var.EQ.'1') then
        n=1
    else
        IF (var.EQ.'2') then
            n=3
        else
            IF (var.EQ.'3') then
                n=7
            else
                IF (var.EQ.'4') then
                    n=15
                endif
            endif
        endif
    endif
    minmse=1000
    minsp=1000
    mincp=1000
    do 20 j=1,n
        read (2,501,end=200) num,cp,mse,sp,mod
        IF (mse.LE.minmse) then
            minmse=mse
            spms=sp
            cpms=cp
            modmse=mod
            nummse=num
        endif
        IF (sp.LE.minsp) then
            minsp=sp
            msecp=mse
            cpsp=cp
            modsp=mod
            numsp=num
        endif
    enddo
enddo
```

```

endif

IF (cp.LE.mincp) then
mincp=cp
msecp=mse
spcp=sp
modcp=mod
numcp=num
endif
20  continue
    write (3,*) 'MSE', nummse, minmse, spms, cpms, modmse
    write (3,*) 'Sp ', numsp, msesp, minsp, cpsp, modsp
    write (3,*) 'Cp ', numcp, msecp, spcp, mincp, modcp

10  continue
500 format (A80)
501 format (7X,I1,17X,F9.5,3X,F9.7,2X,F10.8,2X,A12)
200 stop
end

```

***** GEM3.FOR *****

```
character*12 mod
character*12 modmse
character*12 modsp
character*12 modcp
character*80 line
character var
integer n,num,nummse,numsp,numcp
real minmse,mse,spms,cpms,sp,cp,msecp,spcp
open (unit=1,file='temp22.dat',status='old')
open (unit=2,file='temp33.dat', status='old')
open (unit=3,file='minmod3.dat',status='new')
do 10 I=1,50000
  read (1,500,end=200) line
  var=line(20:20)
  IF (var.EQ.'1') then
    n=1
  else
    IF (var.EQ.'2') then
      n=3
    else
      IF (var.EQ.'3') then
        n=7
      else
        IF (var.EQ.'4') then
          n=15
        IF (var.EQ.'5') then
          n=31
        IF (var.EQ.'6') then
          n=63
        endif
      endif
    endif
  endif
  endif
  minmse=1000
  minsp=1000
  mincp=1000
  do 20 j=1,n
    read (2,501,end=200) num,cp,mse,sp,mod
    IF (mse.LE.minmse) then
      minmse=mse
      spms=sp
      cpms=cp
      modmse=mod
      nummse=num
    endif
  enddo
enddo
```

```

        IF (sp.LE.minsp) then
        minsp=sp
        msesp=mse
        cpsp=cp
        modsp=mod
        numsp=num
        endif
        IF (cp.LE.mincp) then
        mincp=cp
        msecp=mse
        spcp=sp
        modcp=mod
        numcp=num
        endif
20      continue
        write (3,*) 'MSE', nummse, minmse, spms, cpms, modmse
        write (3,*) 'Sp ', numsp, msesp, minsp, cpsp, modsp
        write (3,*) 'Cp ', numcp, msecp, spcp, mincp, modcp

10      continue
500     format (A80)
501     format (7X,I1,17X,F9.5,3X,F9.7,2X,F10.8,2X,A12)
200     stop
        end

```

***** GRGR.FOR *****

```
CHARACTER*12 GR
OPEN(UNIT=1,FILE= 'TEMP2.dat',STATUS='old')
OPEN(UNIT=2,FILE='SCRN1.SAS',STATUS='new')
DO 40 K=1,63,2
PRINT*, 'K=',K
WRITE(2,*) 'option linesize=80;'
WRITE(2,*) 'filename new ''01.dat'';'
WRITE(2,*) 'data new;'
WRITE(2,*) 'infile new ;'
WRITE(2,*) 'input z y x1 x2 x3 x4 e1;'
DO 50 J=1,60
    READ(1,500,END=200) GR
    WRITE(2,501) J,GR
50  CONTINUE
40  CONTINUE
500  FORMAT(26X,A12)
501  FORMAT(' proc rsquare data=new mse sp cp;' /
1      '   where z=',I2,';' /
2      '   model y=',A12,';' )
200  STOP
END
```

***** GRGR3.FOR *****

```
CHARACTER*12 GR
OPEN(UNIT=1,FILE= 'TEMP22.dat',STATUS='OLD')
OPEN(UNIT=2,FILE='SCRN11.SAS',STATUS='NEW')
DO 40 K=2,64,2
WRITE(2,*) 'option linesize=80;'
WRITE(2,*) 'filename new ' ' ',k,'.dat'';'
WRITE(2,*) 'data new;'
WRITE(2,*) 'infile new ;'
WRITE(2,*) 'input z y x1 x2 x3 x4 e1 e2 e3;'
DO 50 J=1,60
    READ(1,500,END=200) GR
    WRITE(2,501) J,GR
50  CONTINUE
40  CONTINUE
500 FORMAT(26X,A12)
501 FORMAT(' proc rsquare data=new mse sp cp;' /
1      '   where z=' ,I2,';' /
2      '   model y=' ,A12,';' )
200 STOP
END
```

***** HALFMLR.FOR *****

```
open (unit=2,file='hlfmlr80.sas',status='new')
WRITE(2,*) 'option linesize=80 pagesize=57;'
DO 40 K=18,48,2
WRITE(2,*) 'filename new ' ' ',K,'.dat'';'
WRITE(2,*) 'data dataset;'
WRITE(2,*) 'infile new ;'
WRITE(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
WRITE(2,*) 'data randset;'
WRITE(2,*) 'set dataset;'
WRITE(2,*) 'r1=RANNOR(0);'
WRITE(2,*) 'r2=RANNOR(0);'
WRITE(2,*) 'r3=RANNOR(0);'
WRITE(2,*) 'r4=RANNOR(0);'
WRITE(2,*) 'r5=RANNOR(0);'
WRITE(2,*) 'r6=RANNOR(0);'
WRITE(2,*) 'r7=RANNOR(0);'
WRITE(2,*) 'r8=RANNOR(0);'
WRITE(2,*) 'r9=RANNOR(0);'
WRITE(2,*) 'proc stepwise data=randset;'
WRITE(2,*) 'by set;'
WRITE(2,*) 'model y=x1 x2 x3 e1 e2 e3 r1 r2 r3
+ r4 r5 r6 r7 r8 r9 /forward slentry=1;'
40 CONTINUE
END
```

***** INVMSE103.FOR *****

```

character*2 m1,m2,m3,m4,m5,m6
character*3 m
character*80 line
open (unit=1,file='error3_all.dat',status='old')
open (unit=2,file='invmse103.sas',status='new')
do 40 k=2,64,2
  do 50 j=1,60
70    read(1,500,end=200) line
        m= line(2:4)
        m1= line(63:64)
        m2= line(65:66)
        m3= line(67:68)
        m4= line(69:70)
        m5= line(71:72)
        m6= line(73:74)

        if ((m.EQ.'MSE').AND.(m2.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' r1 /forward slentry=1;'
        else

        if ((m.EQ.'MSE').AND.(m3.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,
+      ' r1 r2 /forward slentry=1;'
        else

        if ((m.EQ.'MSE').AND.(m4.EQ.' ')) then

```

```

write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,
+      ' r1 r2 r3/forward slentry=1;'
      else
        if ((m.EQ.'MSE').AND.(m5.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,
+      ' r1 r2 r3 r4 /forward slentry=1;'
      else
        if ((m.EQ.'MSE').AND.(m6.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'r5=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'

```

```

write(2,*) 'where set=',j,';'

write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,' ',m5,
+      ' r1 r2 r3 r4 r5 /forward slentry=1;'
else
    if ((m.EQ.'MSE').AND.(m6.NE.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ',k, '.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'r5=RANNOR(0);'
write(2,*) 'r6=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,' ',m5,'
+      ',m6,' r1 r2 r3 r4 r5 r6 /forward slentry=1;'

    else
        goto 70
    endif
endif
endif
endif
endif
endif
endif
50    continue
40    continue
500  format(A80)
200  stop
end

```

***** INVSP101.FOR *****

```

character*2 m1,m2,m3,m4
character*2 m
character*80 line
open (unit=1,file='error1_all.dat',status='old')
open (unit=2,file='invsp101.sas',status='new')
do 40 k=1,63,2
  do 50 j=1,60
70    read(1,500,end=200) line
        m= line(2:8)
        m1= line(63:64)
        m2= line(65:66)
        m3= line(67:68)
        m4= line(69:70)

        if ((m.EQ.'Sp      ').AND.(m2.EQ.'  ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new ' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' r1 /forward slentry=1;'
        else

          if ((m.EQ.'Sp      ').AND.(m3.EQ.'  ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,
+      ' r1 r2 /forward slentry=1;'
          else

            if((m.EQ.'Sp      ').AND.(m4.EQ.'  ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'

```

```

write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,
+      ' r1 r2 r3 /forward slentry=1;'
      else
        if ((m.EQ.'Sp      ').AND.(m4.NE.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,
+      ' r1 r2 r3 r4 /forward slentry=1;'
      else
        goto 70
      endif
    endif
  endif
endif
50    continue
40    continue
500  format(A80)
200  stop
end

```

***** INVSP103.FOR *****

```

character*2 m1,m2,m3,m4,m5,m6
character*3 m
character*80 line
open (unit=1,file='error3_all.dat',status='old')
open (unit=2,file='invsp103.sas',status='new')
do 40 k=2,64,2
  do 50 j=1,60
70    read(1,500,end=200) line
        m= line(2:8)
        m1= line(63:64)
        m2= line(65:66)
        m3= line(67:68)
        m4= line(69:70)
        m5= line(71:72)
        m6= line(73:74)

        if ((m.EQ.'Sp      ').AND.(m2.EQ.'  ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new ' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' r1 /forward slentry=1;'
        else

            if ((m.EQ.'Sp      ').AND.(m3.EQ.'  ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat'';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,
+      ' r1 r2 /forward slentry=1;'
        else
            if((m.EQ.'Sp      ').AND.(m4.EQ.'  ')) then

```

```

write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,
+   ' r1 r2 r3/forward slentry=1;'
      else

      if ((m.EQ.'Sp      ').AND.(m5.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,
+   ' r1 r2 r3 r4 /forward slentry=1;'

      else

      if ((m.EQ.'Sp      ').AND.(m6.EQ.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'r5=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'

```

```

write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,' ',m5,
+ ' r1 r2 r3 r4 r5 /forward slentry=1;'
else
    if ((m.EQ.'Sp      ').AND.(m6.NE.' ')) then
write(2,*) 'option linesize=80 pagesize=57 ;'
write(2,*) 'filename new' ' ',k,'.dat';'
write(2,*) 'data dataset;'
write(2,*) 'infile new;'
write(2,*) 'input set y x1 x2 x3 x4 e1 e2 e3;'
write(2,*) 'data randset;'
write(2,*) 'set dataset;'
write(2,*) 'r1=RANNOR(0);'
write(2,*) 'r2=RANNOR(0);'
write(2,*) 'r3=RANNOR(0);'
write(2,*) 'r4=RANNOR(0);'
write(2,*) 'r5=RANNOR(0);'
write(2,*) 'r6=RANNOR(0);'
write(2,*) 'proc stepwise data=randset ;'
write(2,*) 'where set=',j,';'
write(2,*) 'model y=',m1,' ',m2,' ',m3,' ',m4,' ',m5,'
',m6,
+ ' r1 r2 r3 r4 r5 r6 /forward slentry=1;'

    else
        goto 70
    endif
endif
endif
endif
endif
endif
50    continue
40    continue
500    format(A80)
200    stop
end

```

***** LESSMLR.FOR *****

```
open (unit=2,file='less30.sas',status='new')
WRITE(2,*) 'option linesize=80 pagesize=57 ;'
DO 40 K=1,63,2
WRITE(2,*) 'filename new ''01.dat'';'
WRITE(2,*) 'data dataset;'
WRITE(2,*) 'infile new ;'
WRITE(2,*) 'input set y x1 x2 x3 x4 e1 ;'
WRITE(2,*) 'data randset;'
WRITE(2,*) 'set dataset;'
WRITE(2,*) 'r1=RANNOR(0);'
WRITE(2,*) 'r2=RANNOR(0);'
WRITE(2,*) 'proc stepwise data=randset;'
WRITE(2,*) 'by set;'
WRITE(2,*) 'model y=x1 x2 x3 e1 r1 r2
+ /forward slentry=1;'
40 CONTINUE
END
```

***** MM1_47.FOR *****

```

character*80 LINE(6)
character*1 NUM(6)
integer z(6)
open(unit=1, file='onemlr10.dat',status='old')
open(unit=2, file='onemlr11.dat',status='old')
open(unit=3, file='onemlr12.dat',status='old')
open(unit=4, file='onemlr13.dat',status='old')
open(unit=5, file='onemlr14.dat',status='old')
open(unit=6, file='onemlr15.dat',status='old')
open(unit=7, file='mm1_47.dat',status='new')
DO 1 v=1,1500
DO 2 s=1,6
READ(s,500,end=200) LINE(s)
NUM(s)=LINE(s) (20:20)
2 CONTINUE
DO 5 j=1,6
z(j)=0
5 CONTINUE

DO 10 k=1,6
IF((NUM(1).EQ.NUM(k)).AND.(LINE(1)(21:23).EQ.' '))
+z(1)=z(1)+1
10 CONTINUE

DO 20 a=1,6
IF((NUM(2).EQ.NUM(a)).AND.(LINE(1)(21:23).EQ.' '))
+z(2)=z(2)+1
20 CONTINUE

DO 30 b=1,6
IF((NUM(3).EQ.NUM(b)).AND.(LINE(1)(21:23).EQ.' '))
+z(3)=z(3)+1
30 CONTINUE

DO 40 c=1,6
IF((NUM(4).EQ.NUM(c)).AND.(LINE(1)(21:23).EQ.' '))
+z(4)=z(4)+1
40 CONTINUE

DO 50 d=1,6
IF((NUM(5).EQ.NUM(d)).AND.(LINE(1)(21:23).EQ.' '))
+z(5)=z(5)+1
50 CONTINUE

DO 60 e=1,6
IF((NUM(6).EQ.NUM(e)).AND.(LINE(1)(21:23).EQ.' '))
+z(6)=z(6)+1
60 CONTINUE

```

***** MM17_63.FOR *****

```

character*80 LINE(6)
character*1 NUM(6)
integer z(6)
open(unit=1, file='hlfmlr10.dat',status='old')
open(unit=2, file='hlfmlr11.dat',status='old')
open(unit=3, file='hlfmlr12.dat',status='old')
open(unit=4, file='hlfmlr13.dat',status='old')
open(unit=5, file='hlfmlr14.dat',status='old')
open(unit=6, file='hlfmlr15.dat',status='old')
open(unit=7, file='mm17_63.dat',status='new')
DO 1 v=1,1500
DO 2 s=1,6
READ(s,500,end=200) LINE(s)
NUM(s)=LINE(s) (20:20)
2 CONTINUE
DO 5 j=1,6
z(j)=0
5 CONTINUE

DO 10 k=1,6
IF((NUM(1).EQ.NUM(k)).AND.(LINE(1)(21:23).EQ.' '))
+ z(1)=z(1)+1
10 CONTINUE

DO 20 a=1,6
IF((NUM(2).EQ.NUM(a)).AND.(LINE(1)(21:23).EQ.' '))
+ z(2)=z(2)+1
20 CONTINUE

DO 30 b=1,6
IF((NUM(3).EQ.NUM(b)).AND.(LINE(1)(21:23).EQ.' '))
+z(3)=z(3)+1
30 CONTINUE

DO 40 c=1,6
IF((NUM(4).EQ.NUM(c)).AND.(LINE(1)(21:23).EQ.' '))
+ z(4)=z(4)+1
40 CONTINUE

DO 50 d=1,6
IF((NUM(5).EQ.NUM(d)).AND.(LINE(1)(21:23).EQ.' '))
+ z(5)=z(5)+1
50 CONTINUE

DO 60 e=1,6
IF((NUM(6).EQ.NUM(e)).AND.(LINE(1)(21:23).EQ.' '))
+ z(6)=z(6)+1

```

```

60      CONTINUE

      BIG=0
      DO 100 f=1,6
      IF (z(f).GT.BIG)  BIG=z(f)
100     CONTINUE

      IF (z(1).EQ.BIG) THEN
      WRITE(7,*) LINE(1)
      ELSE
      IF (z(2).EQ.BIG) THEN
      WRITE(7,*) LINE(2)
      ELSE
      IF (z(3).EQ.BIG) THEN
      WRITE(7,*) LINE(3)
      ELSE
      IF (z(4).EQ.BIG) THEN
      ELSE
      IF (z(5).EQ.BIG) THEN
      WRITE(7,*) LINE(5)
      ELSE
      IF (z(6).EQ.BIG) THEN
      WRITE(7,*) LINE(6)
      ELSE
      WRITE(7,*) 'IT IS TIE.'
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF

1       CONTINUE

500     format(A80)
200     stop
      end

```

***** MM18_64.FOR *****

```

character*80 LINE(6)
character*1 NUM(6)
integer z(6)
open(unit=1, file='hlfmlr80.dat',status='old')
open(unit=2, file='hlfmlr81.dat',status='old')
open(unit=3, file='hlfmlr82.dat',status='old')
open(unit=4, file='hlfmlr83.dat',status='old')
open(unit=5, file='hlfmlr84.dat',status='old')
open(unit=6, file='hlfmlr85.dat',status='old')
open(unit=7, file='mm18_64.dat',status='new')
DO 1 v=1,1500
DO 2 s=1,6
READ(s,500,end=200) LINE(s)
NUM(s)=LINE(s) (20:20)
2  CONTINUE
DO 5 j=1,6
z(j)=0
5  CONTINUE

DO 10 k=1,6
IF((NUM(1).EQ.NUM(k)).AND.(LINE(1)(21:23).EQ.' '))
+ z(1)=z(1)+1
10 CONTINUE

DO 20 a=1,6
IF((NUM(2).EQ.NUM(a)).AND.(LINE(1)(21:23).EQ.' '))
+ z(2)=z(2)+1
20 CONTINUE

DO 30 b=1,6
IF((NUM(3).EQ.NUM(b)).AND.(LINE(1)(21:23).EQ.' '))
+ z(3)=z(3)+1
30 CONTINUE

DO 40 c=1,6
IF((NUM(4).EQ.NUM(c)).AND.(LINE(1)(21:23).EQ.' '))
+ z(4)=z(4)+1
40 CONTINUE

DO 50 d=1,6
IF((NUM(5).EQ.NUM(d)).AND.(LINE(1)(21:23).EQ.' '))
+ z(5)=z(5)+1
50 CONTINUE

DO 60 e=1,6
IF((NUM(6).EQ.NUM(e)).AND.(LINE(1)(21:23).EQ.' '))
+ z(6)=z(6)+1

```

```

60      CONTINUE

      BIG=0
      DO 100 f=1,6
      IF (z(f).GT.BIG)  BIG=z(f)
100     CONTINUE

      IF (z(1).EQ.BIG) THEN
      WRITE(7,*) LINE(1)
      ELSE
      IF (z(2).EQ.BIG) THEN
      WRITE(7,*) LINE(2)
      ELSE
      IF (z(3).EQ.BIG) THEN
      WRITE(7,*) LINE(3)
      ELSE
      IF (z(4).EQ.BIG) THEN
      ELSE
      IF (z(5).EQ.BIG) THEN
      WRITE(7,*) LINE(5)
      ELSE
      IF (z(6).EQ.BIG) THEN
      WRITE(7,*) LINE(6)
      ELSE
      WRITE(7,*) 'IT IS TIE.'
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF

1       CONTINUE

500     format(A80)
200     stop
      end

```

***** MM2_48.FOR *****

```

character*80 LINE(6)
character*1 NUM(6)
integer z(6)
open(unit=1, file='onemlr80.dat',status='old')
open(unit=2, file='onemlr81.dat',status='old')
open(unit=3, file='onemlr82.dat',status='old')
open(unit=4, file='onemlr83.dat',status='old')
open(unit=5, file='onemlr84.dat',status='old')
open(unit=6, file='onemlr85.dat',status='old')
open(unit=7, file='mm2_48.dat',status='new')
DO 1 v=1,1500
DO 2 s=1,6
READ(s,500,end=200) LINE(s)
NUM(s)=LINE(s) (20:20)
2 CONTINUE
DO 5 j=1,6
z(j)=0
5 CONTINUE

DO 10 k=1,6
IF((NUM(1).EQ.NUM(k)).AND.(LINE(1)(21:23).EQ.' '))
+ z(1)=z(1)+1
10 CONTINUE

DO 20 a=1,6
IF((NUM(2).EQ.NUM(a)).AND.(LINE(1)(21:23).EQ.' '))
+ z(2)=z(2)+1
20 CONTINUE

DO 30 b=1,6
IF((NUM(3).EQ.NUM(b)).AND.(LINE(1)(21:23).EQ.' '))
+ z(3)=z(3)+1
30 CONTINUE

DO 40 c=1,6
IF((NUM(4).EQ.NUM(c)).AND.(LINE(1)(21:23).EQ.' '))
+ z(4)=z(4)+1
40 CONTINUE

DO 50 d=1,6
IF((NUM(5).EQ.NUM(d)).AND.(LINE(1)(21:23).EQ.' '))
+ z(5)=z(5)+1
50 CONTINUE

DO 60 e=1,6
IF((NUM(6).EQ.NUM(e)).AND.(LINE(1)(21:23).EQ.' '))
+ z(6)=z(6)+1

```

```

60      CONTINUE

      BIG=0
      DO 100 f=1,6
      IF (z(f).GT.BIG)  BIG=z(f)
100    CONTINUE

      IF (z(1).EQ.BIG) THEN
      WRITE(7,*) LINE(1)
      ELSE
      IF (z(2).EQ.BIG) THEN
      WRITE(7,*) LINE(2)
      ELSE
      IF (z(3).EQ.BIG) THEN
      WRITE(7,*) LINE(3)
      ELSE
      IF (z(4).EQ.BIG) THEN
      ELSE
      IF (z(5).EQ.BIG) THEN
      WRITE(7,*) LINE(5)
      ELSE
      IF (z(6).EQ.BIG) THEN
      WRITE(7,*) LINE(6)
      ELSE
      WRITE(7,*) 'IT IS TIE.'
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF

1      CONTINUE

500    format(A80)
200    stop
      end

```

***** NEWSTEPcount1.FOR *****

SUBROUTINE newstepcount1 (NewOut)

integer num, numvar,h,i,j,k,n,p,q,r,s,t,v,w,x,y,z
integer emiller, varsmiller, cumemiller, cmiller
integer chartmiller(0:3,0:3), ReadCount
real avgvars, avgevars, millerpm
character*2 numchar
character*2 m(12), Model_var, Good_model(5)
character*20 NewOut
character*80 line
logical ModelNotFound, EndOfFile

num=0
emiller=0
varsmiller=0
cumemiller=0
cmiller=0
ReadCount=0
ModelNotFound=.TRUE.
EndOfFile=.FALSE.

10 do 10 i=1,12
m(i)=' '
continue

20 do 20 j=1,4
Good_model(j)=' '
continue

40 do 30 k=0,3
do 40 h=0,3
chartmiller(k,h)=0
continue
30 continue

+ open (unit=11, file='mlr.dat', status='old',
+ iostat=IERROR, err=1000)
+ open (unit=12, file=NewOut, status='new',
+ iostat=IERROR, err=1000)
+ open (unit=13, file='newPMstep1.dat', status='new',
+ iostat=IERROR, err=1000)
write(13,*) ' DESIGNPOINT MILLER'S PM'

ReadCount=ReadCount+1
Read(11,900,end=90) line
numchar=line (4:5)
Model_var=line (10:11)

900

```

Format (A80)
PRINT*, numchar
IF (numchar.EQ.' 1') THEN
    num=1
ELSE
    IF (numchar.EQ.' 2') THEN
        num=2
    ELSE
        IF (numchar.EQ.' 3') THEN
            num=3
        ELSE
            IF (numchar.EQ.' 4') THEN
                num=4
            ELSE
                IF (numchar.EQ.' 5') THEN
                    num=5
                ELSE
                    IF (numchar.EQ.' 6') THEN
                        num=6
                    ELSE
                        IF (numchar.EQ.' 7') THEN
                            num=7
                        ELSE
                            IF (numchar.EQ.' 8') THEN
                                num=8
                            ELSE
                                IF (numchar.EQ.' 9') then
                                    num=9
                                else
                                    IF (numchar.EQ.'10') then
                                        num=10
                                    else
                                        IF (numchar.EQ.'11') then
                                            num=11
                                        else
                                            IF (numchar.EQ.'12') then
                                                num=12
                                            else
                                                Print *, 'Unexpected format in TEMP.DAT: ',
                                                    'Numbers 1,2,3, etc., not found!(1)'
                                                Go to 1300
                                            ENDIF
                                        ENDIF
                                    ENDIF
                                ENDIF
                            ENDIF
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF

```

```

        ENDIF
        ENDIF
        ENDIF

IF (num.NE.1) THEN
    Print *, 'Processing terminated.  Input file ',
+       'in unexpected format: 1st number must be 1.'
    Go to 1300
ELSE
    m(num)=Model_var
ENDIF

Do 60 n=1,63,2
    Write(12,*)' '
    Write(12,*)' '
    Write(12,*)' '
    Write(12,*) '***** DESIGN POINT ',
+       n, '*****'
    Write (12,*)' '
    Write(12,*)'Replication      #Vars      Model'

do 70 p=1,60

80      Continue
        IF (EndOfFile) THEN
            Print *, 'Unexpected file format!  File does not ',
+            'have correct # of design points and reps.'
            Go to 1300
        ENDIF
        ReadCount=ReadCount+1
        Read(11,900,end=90) line
        numchar= line (4:5)
        Model_var= line (10:11)
        IF (numchar.EQ.' 1') THEN
            num=1
        ELSE
            IF(numchar.EQ.' 2') THEN
                num=2
            ELSE
                IF (numchar.EQ.' 3') THEN
                    num=3
                ELSE
                    IF (numchar.EQ.' 4') THEN
                        num=4
                    ELSE
                        IF (numchar.EQ.' 5') THEN
                            num=5
                        ELSE
                            IF (numchar.EQ.' 6') THEN
                                num=6
                            ELSE
                                IF (numchar.EQ.' 7') THEN

```



```

                GO TO 80

120      continue
        IF (numvar.LE.0) Go to 140
        do 130 s=1,numvar
            if(m(s).EQ.'E1') then

                emiller=emiller+1
            endif
130      continue

140      varsmiller=varsmiller+numvar
        cumemiller=cumemiller+emiller
        cmiller=numvar-emiller
        chartmiller(cmiller,emiller)=chartmiller(cmiller,
+          emiller)+1

        do 150 t=1,12
            m(t)=' '
150      Continue

        m(num)=Model_var

        do 160 v=1,4
            Good_model(v)=' '
160      continue

        emiller=0
        numvar=0
        ModelNotFound=.TRUE.

70      continue

        write(12,*) ' '
        write(12,*) '*****',
+          '*****'
        write(12,*) ' '

        IF (varsmiller.GT.0) THEN
            avgvars = real(varsmiller)/60.0
            avgevars = real(cumemiller)/60.0
            millerpm = 1-(avgevars/avgvars)
        ELSE
            avgvars=0
            avgevars=0
            millerpm=0
        ENDIF

        write(12,*) 'The avg number of vars using Miller''s',
+          ' method was ', avgvars

```

```

    write(12,*) 'The avg number of extraneous vars from',
+   ' Miller''s method was', avgevars
    write(12,*) '***** The PM for Miller''s was ',
+   millerpm,' *****'
    write(12,*) ' '
    write(12,*) ' '
    write(12,*) 'Correct Vars (0-3, down) -VS- ',
+   'Extraneous Vars (0-3,across)'

    write(12,*) ' '
    write(12,*) ' Table for Miller''s Method'
    write(12,*) ' '
        do 170 w=0,3
            write(12,*) (chartmiller(w,x),x=0,3)

170      continue
        Write(13,*) n,' ',millerpm

        do 180 y = 0,3
            do 190 z = 0,3
                chartmiller(y,z)=0
190      continue
180      continue

        varsmiller=0
        cumemiller=0

60      Continue
        Close(11)
        Close(12)
        Close(13)
        GO TO 1200

* Error trap: *****

1000  Continue
        Print 1100, '+++ ERROR WHILE OPENING FILE +++',
+   ' error code = ', IERROR
1100  FORMAT(/1X, A/ 1X, A, I8/)

*****

1200  CONTINUE
        Print *,'Counting complete. ', NewOut,' written.'
        Go to 1300

90    Print*, 'End of File encountered at line ',ReadCount
        Print*, 'Design Point:',n,' Replication:',p
        num=1
        Model_var='*'
        EndOfFile=.TRUE.

```

Go to 110

1300 CONTINUE
END

***** ONEMLR.FOR *****

```
open (unit=2,file='onemlr10.sas',status='new')
WRITE(2,*) 'option linesize=80 pagesize=57;'
DO 40 K=1,31,2
WRITE(2,*) 'filename new ' ' ',K,'.dat'';'
WRITE(2,*) 'data dataset;'
WRITE(2,*) 'infile new ;'
WRITE(2,*) 'input set y x1 x2 x3 x4 e1;'
WRITE(2,*) 'data randset;'
WRITE(2,*) 'set dataset;'
WRITE(2,*) 'r1=RANNOR(0);'
WRITE(2,*) 'r2=RANNOR(0);'
WRITE(2,*) 'r3=RANNOR(0);'
WRITE(2,*) 'r4=RANNOR(0);'
WRITE(2,*) 'proc stepwise data=randset;'
WRITE(2,*) 'by set;'
WRITE(2,*) 'model y=x1 x2 x3 e1 r1 r2 r3 r4
+ /forward slentry=1;'
40 CONTINUE
END
```

***** STEPCOUNT3.FOR *****

SUBROUTINE Stepcount3 (NewOut)

integer num, numvar,h,i,j,k,n,p,q,r,s,t,v,w,x,y,z
integer emiller, varsmiller, cumemiller, cmiller
integer chartmiller(0:3,0:3), ReadCount
real avgvars, avgevars, millerpm
character*2 m(19), Model_var, Good_model(7), numchar
character*20 NewOut
character*80 line
logical ModelNotFound, EndOfFile

num=0
emiller=0
varsmiller=0
cumemiller=0
cmiller=0
ReadCount=0
ModelNotFound=.TRUE.
EndOfFile=.FALSE.

10 do 10 i=1,18
m(i)=' '
continue

20 do 20 j=1,6
Good_model(j)=' '
continue

40 do 30 k=0,3
do 40 h=0,3
chartmiller(k,h)=0
continue
30 continue

+ open (unit=11, file='mlr.dat', status='old',
iostat=IERROR, err=1000)
+ open (unit=12, file=NewOut, status='new',
iostat=IERROR, err=1000)
+ open (unit=13, file='PMstep3.dat', status='new',
iostat=IERROR, err=1000)
write(13,*) ' DESIGNPOINT MILLER'S PM'

900 ReadCount=ReadCount+1
Read(11,900,end=90) line
numchar=line (4:5)
Model_var= line (10:11)
Format(A80)

IF (numchar.EQ.' 1') THEN

 num=1

ELSE

 IF (numchar.EQ.' 2') THEN

 num=2

 ELSE

 IF (numchar.EQ.' 3') THEN

 num=3

 ELSE

 IF (numchar.EQ.' 4') THEN

 num=4

 ELSE

 IF (numchar.EQ.' 5') THEN

 num=5

 ELSE

 IF (numchar.EQ.' 6') THEN

 num=6

 ELSE

 IF (numchar.EQ.' 7') THEN

 num=7

 ELSE

 IF (numchar.EQ.' 8') THEN

 num=8

 ELSE

 IF (numchar.EQ.' 9') THEN

 num=9

 ELSE

 IF (numchar.EQ.'10') THEN

 num=10

 ELSE

 IF (numchar.EQ.'11') THEN

 num=11

 ELSE

 IF (numchar.EQ.'12') THEN

 num=12

 ELSE

 IF (numchar.EQ.'13') THEN

 num=13

 ELSE

 IF (numchar.EQ.'14') THEN

 num=14

 ELSE

 IF (numchar.EQ.'15') THEN

 num=15

 ELSE

 IF (numchar.EQ.'16') THEN

 num=16

 ELSE

 IF (numchar.EQ.'17') THEN

 num=17


```

ENDIF
ReadCount=ReadCount+1
Read(11,900,end=90) line
numchar= line (4:5)
Model_var= line (10:11)

IF (numchar.EQ.' 1') THEN
    num=1
ELSE
    IF (numchar.EQ.' 2') THEN
        num=2
    ELSE
        IF (numchar.EQ.' 3') THEN
            num=3
        ELSE
            IF (numchar.EQ.' 4') THEN
                num=4
            ELSE
                IF (numchar.EQ.' 5') THEN
                    num=5
                ELSE
                    IF (numchar.EQ.' 6') THEN
                        num=6
                    ELSE
                        IF (numchar.EQ.' 7') THEN
                            num=7
                        ELSE
                            IF (numchar.EQ.' 8') THEN
                                num=8
                            ELSE
                                IF (numchar.EQ.' 9') THEN
                                    num=9
                                ELSE
                                    IF (numchar.EQ.'10') THEN
                                        num=10
                                    ELSE
                                        IF (numchar.EQ.'11') THEN
                                            num=11
                                        ELSE
                                            IF (numchar.EQ.'12') THEN
                                                num=12
                                            ELSE
                                                IF (numchar.EQ.'13') THEN
                                                    num=13
                                                ELSE
                                                    IF (numchar.EQ.'14') THEN
                                                        num=14
                                                    ELSE
                                                        IF (numchar.EQ.'15') THEN
                                                            num=15
                                                        ELSE

```



```

+           A2,1X,A2,1X,A2,1X,A2)
          Go to 120
        ENDIF
        GO TO 80

120      continue
        IF (numvar.LE.0) Go to 140
        do 130 s=1,numvar
          if ((m(s).EQ.'E1').OR.(m(s).EQ.'E2').OR.
+           (m(s).EQ.'E3')) then
            emiller=emiller+1
          endif
130      continue

140      varsmiller=varsmiller+numvar
          cumemiller=cumemiller+emiller
          cmiller=numvar-emiller
          chartmiller(cmiller,emiller)=chartmiller(cmiller,
+           emiller)+1

        do 150 t=1,18
          m(t)=' '
150      Continue

        m(num)=Model_var

        do 160 v=1,6
          Good_model(v)=' '
160      continue

        emiller=0
        numvar=0
        ModelNotFound=.TRUE.

70      continue

        write(12,*) ' '
        write(12,*) '*****',
+       write(12,*) '*****'

        IF (varsmiller.GT.0) THEN
          avgvars = real(varsmiller)/60.0
          avgevars = real(cumemiller)/60.0
          millerpm = 1-(avgevars/avgvars)
        ELSE
          avgvars=0
          avgevars=0
          millerpm=0

```

```

ENDIF

write(12,*) 'The avg number of vars using Miller''s',
+ ' method was ', avgvars
write(12,*) 'The avg number of extraneous vars from',
+ ' Miller''s method was', avgevars
write(12,*) '***** The PM for Miller''s was ',
+ millerpm, ' *****'
write(12,*) ' '
write(12,*) ' '
write(12,*) 'Correct Vars (0-3, down) -VS- ',

+ 'Extraneous Vars (0-3, across)'
write(12,*) ' '
write(12,*) ' Table for Miller''s Method'
write(12,*) ' '
do 170 w=0,3
    write(12,*) (chartmiller(w,x),x=0,3)

170    continue
    Write(13,*) n, ' ',millerpm

    do 180 y = 0,3
        do 190 z = 0,3
            chartmiller(y,z)=0
190        continue
180    continue

    varsmiller=0
    cumemiller=0

60    Continue
    Close(11)
    Close(12)
    Close(13)
    GO TO 1200

* Error trap: *****

1000 Continue
    Print 1100, '+++ ERROR WHILE OPENING FILE +++',
+ ' error code = ', IERROR
1100 FORMAT(/1X, A/ 1X, A, I8/)

*****

1200 CONTINUE
    Print *, 'Counting complete. ', NewOut, ' written.'
    Go to 1300

90    Print*, 'End of File encountered at line ', ReadCount

```

```
Print*, 'Design Point:', n, '  Replication:', p  
num=1  
Model_var='**'  
EndOfFile=.TRUE.  
Go to 110
```

```
1300 CONTINUE  
END
```

Appendix M:

SAS Programs

HLFMLR80.SAS	160
INVCP101.SAS	162
INVCP103.SAS	164
ONEMLR10.SAS	166
ONEMLR80.SAS	167
LESSMLR.SAS	168
SCRN1.SAS	169
SCR11.SAS	171

***** HLFMLR80.SAS *****

```
option linesize=80 pagesize=57;
filename new '          18.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
r5=RANNOR(0);
r6=RANNOR(0);
r7=RANNOR(0);
r8=RANNOR(0);
r9=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 e2 e3 r1 r2 r3 r4 r5 r6 r7 r8 r9 /forward
slentry=1;
filename new '          20.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
r5=RANNOR(0);
r6=RANNOR(0);
r7=RANNOR(0);
r8=RANNOR(0);
r9=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 e2 e3 r1 r2 r3 r4 r5 r6 r7 r8 r9 /forward
slentry=1;

.
.
.

filename new '          64.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
```

```
set dataset;  
r1=RANNOR(0);
```

***** INCP101.SAS *****

```
option linesize=80 pagesize=57 ;
filename new'          1.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
proc stepwise data=randset ;
where set=          1;
model y=X1 X2 X3 r1 r2 r3 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new'          1.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
proc stepwise data=randset ;
where set=          2;
model y=X1 X2 X3 r1 r2 r3 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new'          1.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
proc stepwise data=randset ;
where set=          3;
model y=X2 X3 E1 r1 r2 r3 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new '          1.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1;
```

```

data randset;
set dataset;
r1=RANNOR(0);
proc stepwise data=randset ;
where set=          4;
model y=X3 r1 /forward slentry=1;

      .          .
      .          .
      .          .

option linesize=80 pagesize=57 ;
filename new'          63.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset ;
where set=          60;
model y=X1 X3 r1 r2 /forward slentry=1;

```

***** INVCP103.SAS *****

```
option linesize=80 pagesize=57 ;
filename new'          2.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset ;
where set=          1;
model y=X1 X2 r1 r2 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new'          2.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
proc stepwise data=randset ;
where set=          2;
model y=X2 X3 E1 E3 r1 r2 r3 r4 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new'          2.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset ;
where set=          3;
model y=X3 E2 r1 r2 /forward slentry=1;
```

```
option linesize=80 pagesize=57 ;
filename new'          2.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
```

```

set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset ;
where set=          4;
model y=X2 X3 r1 r2 /forward slentry=1;
      .              .
      .              .
      .              .
option linesize=80 pagesize=57 ;
filename new'          64.dat';
data dataset;
infile new;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
proc stepwise data=randset ;
where set=          60;
model y=X1 X3 E1 r1 r2 r3/forward slentry=1;

```

***** ONEMLR10.SAS *****

```

option linesize=80 pagesize=57;
filename new '          1.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 r3 r4 /forward slentry=1;
filename new '          3.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 r3 r4 /forward slentry=1;
filename new '          5.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 r3 r4 /forward slentry=1;

```

```

      .      .
      :      :
      .      .

```

```

filename new '          7.dat';
data dataset;
infile new ;

```

***** ONEMLR80.SAS *****

```

option linesize=80 pagesize=57;
filename new '          2.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
r5=RANNOR(0);
r6=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 e2 e3 r1 r2 r3 r4 r5 r6 /forward slentry=1;
filename new '          4.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
r5=RANNOR(0);
r6=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 e2 e3 r1 r2 r3 r4 r5 r6 /forward slentry=1;

```

```

      .      .
      .      .
      .      .

```

```

filename new '          6.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1 e2 e3;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
r3=RANNOR(0);
r4=RANNOR(0);
r5=RANNOR(0);
r6=RANNOR(0);

```

***** LESSMLR.SAS *****

```
option linesize=80 pagesize=57;
filename new '01.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 /forward slentry=1;
filename new '03.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 /forward slentry=1;
filename new '05.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset;
by set;
model y=x1 x2 x3 e1 r1 r2 /forward slentry=1;
```

```
      .      .
      .      .
      .      .
```

```
filename new '07.dat';
data dataset;
infile new ;
input set y x1 x2 x3 x4 e1;
data randset;
set dataset;
r1=RANNOR(0);
r2=RANNOR(0);
proc stepwise data=randset;
```

***** SCRN1.SAS *****

```
option linesize=80;
filename new '01.dat';
data new;
infile new ;
input z y x1 x2 x3 x4 e1;
proc rsquare data=new mse sp cp;
  where z= 1;
  model y=X1 X3      ;
proc rsquare data=new mse sp cp;
  where z= 2;
  model y=x4          ;
proc rsquare data=new mse sp cp;
  where z= 3;
  model y=X2          ;
proc rsquare data=new mse sp cp;
  where z= 4;
  model y=X3          ;
proc rsquare data=new mse sp cp;
  where z= 5;
  model y=X1 X3      ;
proc rsquare data=new mse sp cp;
  where z= 6;
  model y=X1 X3 E1    ;
proc rsquare data=new mse sp cp;
  where z= 7;
  model y=X2 X1 X3    ;
proc rsquare data=new mse sp cp;
  where z= 8;
  model y=X1 X3 X2    ;
proc rsquare data=new mse sp cp;
  where z= 9;
  model y=X2 X1      ;
proc rsquare data=new mse sp cp;
  where z=10;
  model y=X3          ;
proc rsquare data=new mse sp cp;
  where z=11;
  model y=X3          ;
proc rsquare data=new mse sp cp;
  where z=12;
  model y=X1 X2 X3    ;
proc rsquare data=new mse sp cp;
  where z=13;
  model y=X2 X1 X3    ;
proc rsquare data=new mse sp cp;
  where z=58;
  model y=X1 X2 X3    ;
```

```

proc rsquare data=new mse sp cp;
  where z=59;
  model y=X3          ;
proc rsquare data=new mse sp cp;
  where z=60;
  model y=X3          ;

```

```

      .      .
      .      .
      .      .

```

```

option linesize=80;
filename new '63.dat';
data new;
infile new ;
input z y x1 x2 x3 x4 e1;
proc rsquare data=new mse sp cp;
  where z= 1;
  model y=X1 X3 X2      ;
proc rsquare data=new mse sp cp;
  where z= 2;
  model y=X1 X3          ;

```

***** SCRN11 *****

```

option linesize=80;
filename new '          2.dat';
data new;
infile new ;
input z y x1 x2 x3 x4 e1 e2 e3;
proc rsquare data=new mse sp cp;
  where z= 1;
  model y=x4          ;
proc rsquare data=new mse sp cp;
  where z= 2;
  model y=x4          ;
proc rsquare data=new mse sp cp;
  where z= 3;
  model y=X3 E2 X2    ;
proc rsquare data=new mse sp cp;
  where z= 4;
  model y=X3 X2 X1    ;
proc rsquare data=new mse sp cp;
  where z= 5;
  model y=X1 X3 E1    ;
.
.
.
  where z=57;
  model y=X3 X1 E1    ;
proc rsquare data=new mse sp cp;
  where z=58;
  model y=X3 E3 X2    ;
proc rsquare data=new mse sp cp;
  where z=59;
  model y=x4          ;
proc rsquare data=new mse sp cp;
  where z=60;
  model y=X1 X2 X3    ;
option linesize=80;
filename new '          4.dat';
data new;
infile new ;
input z y x1 x2 x3 x4 e1 e2 e3;
proc rsquare data=new mse sp cp;
  where z= 1;
  model y=X2 X3 X1    ;
proc rsquare data=new mse sp cp;
  where z= 2;
  model y=X2 X1 X3    ;

```

Bibliography

- Bowerman, Bruce L. and O'Connell, Richard T. Linear Statistical Models An Applied Approach. Boston: PWS-KENT, 1990.
- Draper, N.R. and Smith, H. Applied Regression Analysis. New York: John Wiley & Sons, 1966.
- Hansen, Capt Ross J. A Comparison of Variable Selection Criteria for Multiple Linear Regression: A Simulation Study. MS Thesis, AFIT/GOR/MA/88D-3. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, DEC 1988.
- Hawkins, Clark A. and Weber Jean E. Statistical Analysis Application to Business and Economics. London: Harper & Row, 1980.
- Hocking, R.R. "The Analysis and Selection of Variables in Linear Regression", Biometrics 32: 1-49 (March 1976).
- Miller, Alan J. "Selection of Subsets of Regression Variables", Journal of Royal Statistical Society, Series A, 147, part 3: 389-425 (1984).
- Miller, Alan J. Subset Selection in Regression. London: Chapman and Hall, 1990.
- Mosteller, Frederick and Tukey, John W. Data Analysis and Regression. London: Addison-Wesley, 1977.
- Neter, John and others Applied Linear Statistical Models. Boston: Irwin, 1990.
- Woollard, Capt David P. A Comparison of Variable Selection Criteria for Multiple Linear Regression: A Second Simulation Study. MS Thesis, AFIT/GOR/ENS/93M-23. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, DEC 1993.

Vita

Lieutenant Ertem Mutlu was born 04 February 1965. In 1980 he entered Military High School in Istanbul. In 1984 he entered Turkish Air Force Academy and graduated in 1988 as a second Lieutenant with a Bachelor of Science degree, majoring in electronics. In 1989 he accomplished Basic Aircraft Maintenance Officer Training in Izmir and was assigned to Bandirma Air Force Base. In 1990, he was assigned to F-16 Aircraft Maintenance Training in Ft Worth Dallas. After working for a year in his previous base in 1992 he was assigned to the Air Force Institute of Technology.

He married former Nilgun Onder in 1988. They have a daughter Niler, 4 years old.

Permanent address: Yesiltepe Mah.
Unal sok. #16
Eskisehir/TURKEY