A COMPARISON TO TWO METHODS USED FOR RANKING TASK EXPOSURE LEVELS USING SIMULATED DATA

CAPT COSTANTINO JOSEPH

UNIVERSITY OF OKLAHOMA HSC

THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

13. ABSTRACT (Maximum 200 words)
A COMPARISON OF TWO METHODS USED FOR
RANKING TASK EXPOSURE LEVELS
USING SIMULATED MULTI-TASK DATA

A THESIS
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
degree of
Master of Science

BY
JOSEPH COSTANTINO
Oklahoma City, Oklahoma
1999
A COMPARISON OF TWO METHODS USED
FOR RANKING TASK EXPOSURE LEVELS
USING SIMULATED MULTI-TASK DATA

APPROVED BY:

Margaret L. Phillips, Ph.D., CIH, Chair
Nurtan A. Esmen, Ph.D.
David L. Johnson, Ph.D., CIH

Thesis Committee
COPYRIGHT

by

JOSEPH COSTANTINO

December 17, 1999
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION AND LITERATURE REVIEW</td>
<td>1</td>
</tr>
<tr>
<td>II. PURPOSE AND SCOPE</td>
<td>8</td>
</tr>
<tr>
<td>III. METHODS AND MATERIALS</td>
<td>9</td>
</tr>
<tr>
<td>IV. RESULTS</td>
<td>14</td>
</tr>
<tr>
<td>V. DISCUSSION AND CONCLUSION</td>
<td>28</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>31</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>33</td>
</tr>
<tr>
<td>Appendix A</td>
<td>33</td>
</tr>
<tr>
<td>Appendix B</td>
<td>37</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1. Probability of correctly ranking each task: Type 1 distribution with J = 20 and GSD = 4</td>
<td>15</td>
</tr>
<tr>
<td>2. Probability of correctly ranking each task: Type 2 distribution with J = 20 and GSD = 4</td>
<td>16</td>
</tr>
<tr>
<td>3. Probability of correctly ranking each task: Type 1 distribution with J = 20 and GSD = 2</td>
<td>17</td>
</tr>
<tr>
<td>4. Probability of correctly ranking each task: Type 2 distribution with J = 20 and GSD = 2</td>
<td>18</td>
</tr>
<tr>
<td>5. Probability of correctly ranking each task: Type 1 distribution with J = 100 and GSD = 4</td>
<td>19</td>
</tr>
<tr>
<td>6. Probability of correctly ranking each task: Type 2 distribution with J = 100 and GSD = 4</td>
<td>20</td>
</tr>
<tr>
<td>7. Probability of correctly ranking each task: Type 1 distribution with J = 100 and GSD = 2</td>
<td>21</td>
</tr>
<tr>
<td>8. Probability of correctly ranking each task: Type 2 distribution with J = 100 and GSD = 2</td>
<td>22</td>
</tr>
<tr>
<td>9. Number of runs that the highest two tasks were correctly ranked</td>
<td>24</td>
</tr>
<tr>
<td>10. Number of runs that were correctly ranked</td>
<td>25</td>
</tr>
<tr>
<td>11. Misclassification indices</td>
<td>26</td>
</tr>
<tr>
<td>12. Correlation coefficients</td>
<td>27</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION AND LITERATURE REVIEW

The role of the occupational hygienist is to protect the health and well-being of workers and the public through anticipation, recognition, evaluation, and control of hazards arising in or from the workplace. During the evaluation phase, an occupational hygienist may collect air samples to quantify workers' exposure levels. Documentation of exposure-related factors, or determinants, is an important aspect of both the sampling event and the comprehensive exposure evaluation. Some examples of determinants include: worker location, raw materials and equipment used, engineering controls, environmental conditions, and task. Researchers have studied determinants of exposure to identify factors that are associated with an increase or decrease in exposure levels.

Determinants are observed and recorded during experimental and observational studies. In most cases, experimental studies are designed to evaluate the effectiveness of engineering controls. For example, studies have assessed the effect of various ventilation configurations on oxygen levels in confined spaces, exposure levels of worker's handling a flour additive powder, and exposure to a pesticide surrogate during spraying in a greenhouse.

Observational studies attempt to identify the effect of various determinants on
exposure levels under actual working conditions. For jobs consisting of a variety of tasks at different locations, the occupational hygienist may find it useful to identify those tasks with high exposure levels. The identification of these tasks allows the hygienist to take preventive measures to reduce these exposure levels through the implementation of engineering controls, changes in work practices, or personal protective equipment.

Several sampling strategies have been used to identify determinants of exposure such as: area sampling, full-shift sampling, and task-specific sampling. In an ideal situation, the occupational hygienist would conduct an exposure assessment for each individual task to rank them according to exposure levels; however, this is not cost effective, nor does it allow for use of existing multi-task exposure assessment data. Many studies have used a task-specific sampling strategy. Task sampling was conducted during a highway construction project in an attempt to establish baseline exposure levels to such tasks as: digging trenches, paving asphalt, and grinding road cover. The study used the project budget to define construction stages (i.e., earthworks, drainage) and then the tasks within each stage were identified. Exposures to noise, dust, and asphalt fumes were measured on operating engineers and laborers. A printing plant study evaluated the influence of task and duration on solvent exposures. Instantaneous samples were collected during tasks performed by seven offset press operators over a three day period. Maximum solvent concentrations were measured during a plate change task. One advantage of the task-specific strategy is that peak exposures may be identified,
particularly if direct-reading instrumentation is used.

The area sampling approach places monitoring equipment close to the sources of concern.\textsuperscript{11-13} Area samples of antineoplastic agents were collected in an outpatient oncology clinic and pharmacy.\textsuperscript{11} In these two areas antineoplastic agents were prepared and administered. Personal sampling was not feasible because of the large air sample volumes required to meet minimum detection limits. To quantify ambient concentrations of dust generated from wood-working machines, area samples were collected in three factories.\textsuperscript{12} The area sampling results identified the cross-cut saw, horizontal belt sander, and the plate saw as the machines generating maximum concentrations in each respective factory. Studies using area sampling only are limited because personal exposures are usually underestimated and the worker’s behavior cannot be evaluated.\textsuperscript{2}

In practice, much industrial hygiene sampling has been compliance driven. The compliance sampling strategy usually uses worst-case monitoring with a focus on exposures during the time of the survey. An attempt is made to identify the maximum-exposed workers in a group. One or a few measurements are taken and simply compared with the occupational exposure limit.\textsuperscript{14} A comprehensive exposure assessment strategy would include an evaluation of all potential hazards so that exposure levels are characterized for all workers, on all days. This strategy is not used very often due to the high cost of collecting and analyzing so many samples. The compliance strategy is reactive while the comprehensive strategy is proactive. Current emphasis in exposure
assessment is focused on moving beyond "compliance management" to "risk management." Since most occupational exposure limits are full-shift time-weighted averages (TWAs), existing exposure assessment data consists largely of full-shift or partial-shift samples, often spanning multiple tasks.

Four studies have attempted to identify high exposure tasks using multi-task sampling data. These studies constructed multiple linear regression models to estimate the relationships between the tasks performed and measured exposure levels. Multiple regression analysis is a mathematical technique used to determine the relationships between a dependent variable and multiple independent variables. However, the results of the data analyses in these studies were not validated by comparison with simultaneous single-task sampling results. Therefore, there is no way to know if these models were accurate.

The objectives of a bakery study were to measure full-shift exposure to inhalable dust in bakeries and define the determinants of full-shift exposure. The study used a cross-sectional design with one exposure measurement from each individual in the recruited bakeries. Ninety-six workers, employed in seven different bakeries, participated in the study. Two side-by-side full-shift inhalable dust samples were obtained from each study participant. Multiple linear regression was used to identify the combination of independent variables that had the best ability to explain full-shift inhalable dust exposure levels. The multiple regression model was tested for violation of the assumptions of
regression analysis. Assumptions of homoscedasity and linearity were tested via
graphical methods. The model indicated which tasks were associated with increasing or
decreasing exposure levels. The tasks which the model predicted to be associated with
increasing exposures included: dough-forming, bread and bun production, and flour
pouring and dusting.

The main objective of a pig farm study\textsuperscript{17} was to use modeling to obtain a more
valid measure of long-term average exposure for epidemiologic purposes. The model
was constructed using a limited number of measurements and by using surrogate
measures of exposure. This study suggested that most long-term average exposure
estimates are imprecise due to intraindividual variability and a limited number of
measurements. In a group of 198 Dutch pig farmers, exposure to endotoxins was
measured on one workday in the summer and one workday in the winter. In the summer
and winter, the farmers were requested to complete a diary on time spent in different
activities during the day of the exposure measurement and the following six days. Time
spent in each activity was recorded. In a subgroup of six farmers, exposure
measurements were performed nearly monthly during a one-year period. Farm
characteristics such as number of animals, feeding methods, heating and ventilation, type
of floor, and bedding material were recorded during walk-through surveys. The data set
contained 95 distinct variables. The multiple linear regression analysis identified those
tasks (ear tagging, teeth cutting) which were associated with increased exposure levels.
A rubber manufacturing industry study\textsuperscript{18} assessed chemical exposures in ten plants. Personal exposures to airborne particulates, rubber fumes and solvents, and dermal contamination were measured. Information on tasks performed, ventilation characteristics, and production variables were used in multiple regression models to identify those factors which affected exposure levels. Model adequacy was tested with standard regression techniques such as residual plots and outlier detection. The multiple regression analysis predicted which tasks (cleaning, weighing, jointing) were statistically significantly associated with higher exposure levels.

A lumber mill study quantified metals exposures of saw filers.\textsuperscript{19} Observations of tasks, locations, and activities were recorded in ten minute intervals. A stepwise multiple linear regression model was used to identify the determinants of exposure. Maximum exposure levels to cobalt and chromium were associated with wet carbide grinding and knife grinding, respectively.

An alternative method to rank task exposures using time-weighted average (TWA) samples has recently been studied.\textsuperscript{21} This method, referred to as the P-screen method, was evaluated using simulated data. (See Materials and Methods for details on data simulation and the P-screen methodology.) The P-screen method ranked the two highest exposure tasks correctly 100\% of the time, if the number of samples was adequate and the task distributions were not highly overlapped. The performance of the model improved with decreasing task distribution geometric standard deviation (GSD),
increased spacing of the task distributions, and an increase in the number of samples. The model proved to be most useful for stratifying exposure levels into high, medium, and low categories.

Several other data analysis methods have been used to identify determinants of exposure such as: arithmetic means, geometric means, Analysis of Variance, and Kruskal-Wallis. However, none of these methods can rank tasks using multi-task data with limited information on task times. Only the P-screen method and the multiple linear regression method have this capability. Therefore, it is appropriate to compare the performance of these two methods. The focus of this study was to conduct a side-by-side comparison of these two methods using simulated multi-task data.
CHAPTER II

PURPOSE AND SCOPE

This study compared the performance of two methods, P-screen and multiple linear regression, at ranking task exposures. Monte Carlo methods using simulated data were used to assess performance of the methods under a variety of experimental conditions.
CHAPTER III

METHODS AND MATERIALS

The existing computer program used for the P-screen method study\textsuperscript{21} was modified to incorporate the multiple linear regression analysis method. The program was written in Microsoft QuickBASIC (see Appendix B). The following parameters were controllable by the programmer: task GSDs, spacing between task median concentrations, and number of samples. The number of tasks was arbitrarily fixed at six. Subject to these parameters, task durations and task concentrations were created through random number generation. The simulated data consisted of multi-task TWA concentrations. This study evaluated how performance of the methods were affected by three experimental parameters: (1) number of samples ($J = 20$ or $100$), (2) nominal GSD ($\sigma_g = 2 \pm 0.5$ or $\sigma_g = 4 \pm 0.5$), and (3) overlap between distributions ($20 - 80\%$ overlap). The overlap parameter was determined by the GSD and the task medians; two different task median spacings were already integrated in the program so that each experiment evaluated the performance for two sets of task distributions. The task distributions were log-normally distributed. The Type 1 task distribution medians were separated by a factor of two with actual task medians equal to $1, 2, 4, 8, 16, \text{and} 32$, in arbitrary units. The Type 2 task distribution medians were separated by a factor of $2^{1.5}$ with actual task medians equal to $0.5, 1.4, 4, 11.3, 32,$ and $90.5$, in arbitrary units.
Generation of Simulated Data

The following is a summary of the steps to generate simulated TWA data.\textsuperscript{21}

1. The task time matrix $\theta$ ("T" task times for each of "J" samples) was randomly generated. Thirty task time matrices were generated for each experiment. In each sample, at least one task time was randomly assigned a zero value (i.e., the task does not occur). All other task times $\theta_y$ were assigned random times as a discrete fraction of the sampling time (1/96, 2/96, ..., 96/96), such that the sum of the task times was equal to one.

2. The task concentrations $C_{ij}$ were created. These values were randomly selected from established task concentration distributions. One hundred task concentration matrices were generated for each task time matrix (30 task time matrices X 100 concentration matrices = 3000 trials per experiment).

3. Time-weighted averages $C_j$ for each sample are calculated by $\sum \theta_y C_y$

P-screen Method

The following is a summary of the steps taken to estimate task median concentrations using the P-screen method.\textsuperscript{21}

1. The natural logarithm of each time-weighted average sample concentration ($C_j$) was taken.
2. The P-screen matrix was set up for each sample where 0 = task performed during sample and 1 = task not performed during sample.

3. The P-screen matrix transpose was multiplied by \( \log C_j \) to get the raw P-sum vector for each sample. This step aggregated TWA sample concentrations based on the non-occurrence of a particular task during that sample. The P-sum vector was normalized by dividing each element by the number of samples contributing to that element.

4. Task time weights were estimated for each sample by assigning \( \hat{\theta}_n \) equal to \( 1/m \) where \( m \) tasks were performed during the sample for at least 5% of the sample period. All other estimated task times are set equal to zero.

5. The P-screen matrix transpose for each sample was multiplied by the \( \hat{\theta}_n \) matrix to get the P-screened \( \hat{\theta} \) matrix. The P-screened \( \hat{\theta} \) matrix was normalized by dividing each row by the number of samples contributing to that row.

6. The normalized P-screened \( \hat{\theta} \) matrix was inverted.

7. The normalized P-sum vector was multiplied by the inverted P-screened \( \hat{\theta} \) matrix to get estimated task median concentrations (\( \log c_i^{med} \)) for each sample.

8. Estimated task median concentration rankings were compared with actual rankings.
Multiple Linear Regression

The module written for the multiple linear regression produced regression coefficients for each task based on the model:

\[ \ln c = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_r X_r. \]  

where: \( c \) was the TWA sample concentration
\( \beta_0 \) was the intercept
\( \beta_s \) were the regression coefficients
\( X_s \) were dichotomous variables representing the occurrence (\( X_i = 1 \)) or nonoccurrence (\( X_i = 0 \)) of the task

The intercept and regression coefficients were obtained by linear algebra methods using the following equations

\[ \iff LRA \cdot X = LRB \]  
\[ \implies X = LRA^{-1} \cdot LRB \]

where the LRA matrix and the LRB vector were as defined in Appendix A and \( X \) was the vector of regression coefficients. The LRA matrix was set up based on the occurrence or nonoccurrence (<5% of the sample period) of a task. This matrix was only changed when a new task time matrix was generated. The LRB vector was set up based on the sample concentration and the task occurrence or nonoccurrence (see Appendix A for an example). The LRAB matrix (7x8) was composed of the LRA matrix (7x7) and the LRB vector (7x1). The LRAB matrix was sent to the MATSOLV1 subroutine for solution to obtain the regression coefficients. The magnitude and orientation of the regression coefficients were used to rank task concentrations. The regression coefficients were also
used to estimate single-task median concentrations. As each module was incorporated into the existing program, a spreadsheet was used to validate randomly selected runs for at least two different task time matrices.
A review of the summary statistics showed that the performance of the two methods was very similar. Summary statistics that were calculated included: (1) the probability of correctly ranking each task (Tables 1 - 8), (2) the number of runs that the highest two tasks were correctly ranked (Table 9), and (3) the number of correctly ranked runs (Table 10). As expected, model performance decreased as overlap and GSD increased and as the number of samples decreased.

A misclassification index (MI) was calculated to compare "how bad" the ranking was for each of the 3000 runs. Misclassification indices were computed for each data set (task time matrix) and the overall index for the entire experiment (Table 11). An MI of 0 indicated a correctly ranked run, while an MI of 1 indicated a worst-case ranking (i.e., reversed order 6, 5, 4, 3, 2, 1). The MI was calculated as follows:

\[ MI = \frac{\sum |\text{actual rank} - \text{assigned rank}|}{18} \]  

A One-way Analysis of Variance (ANOVA) of the MIs was conducted to test the null hypothesis that the MIs between the 30 task time matrices were all equal. All of the One-way ANOVAs resulted in a rejection of the null hypothesis. The results of this analysis
Table 1

Probability of Correctly Ranking Each Task:
Type 1 Distribution with J=20 and GSD=4.

Multiple Linear Regression

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.336</td>
<td>0.264</td>
<td>0.189</td>
<td>0.123</td>
<td>0.070</td>
<td>0.018</td>
</tr>
<tr>
<td>2</td>
<td>0.290</td>
<td><strong>0.267</strong></td>
<td>0.190</td>
<td>0.147</td>
<td>0.077</td>
<td>0.028</td>
</tr>
<tr>
<td>True</td>
<td>3</td>
<td>0.244</td>
<td>0.219</td>
<td><strong>0.210</strong></td>
<td>0.176</td>
<td>0.101</td>
</tr>
<tr>
<td>Rank</td>
<td>4</td>
<td>0.092</td>
<td>0.162</td>
<td>0.228</td>
<td><strong>0.266</strong></td>
<td>0.189</td>
</tr>
<tr>
<td>5</td>
<td>0.027</td>
<td>0.063</td>
<td>0.123</td>
<td>0.184</td>
<td><strong>0.330</strong></td>
<td>0.273</td>
</tr>
<tr>
<td>6</td>
<td>0.010</td>
<td>0.026</td>
<td>0.060</td>
<td>0.103</td>
<td>0.232</td>
<td><strong>0.569</strong></td>
</tr>
</tbody>
</table>

P-screen Method

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>0.360</strong></td>
<td>0.260</td>
<td>0.175</td>
<td>0.118</td>
<td>0.065</td>
<td>0.021</td>
</tr>
<tr>
<td>2</td>
<td>0.264</td>
<td><strong>0.261</strong></td>
<td>0.205</td>
<td>0.145</td>
<td>0.084</td>
<td>0.040</td>
</tr>
<tr>
<td>True</td>
<td>3</td>
<td>0.227</td>
<td>0.230</td>
<td><strong>0.217</strong></td>
<td>0.173</td>
<td>0.099</td>
</tr>
<tr>
<td>Rank</td>
<td>4</td>
<td>0.105</td>
<td>0.152</td>
<td>0.204</td>
<td><strong>0.249</strong></td>
<td>0.203</td>
</tr>
<tr>
<td>5</td>
<td>0.034</td>
<td>0.066</td>
<td>0.137</td>
<td>0.197</td>
<td><strong>0.301</strong></td>
<td>0.264</td>
</tr>
<tr>
<td>6</td>
<td>0.010</td>
<td>0.030</td>
<td>0.061</td>
<td>0.118</td>
<td>0.247</td>
<td><strong>0.534</strong></td>
</tr>
</tbody>
</table>
Table 2

Probability of Correctly Ranking Each Task:
Type 2 Distribution with J=20 and GSD=4.

Multiple Linear Regression

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.320</td>
<td>0.268</td>
<td>0.194</td>
<td>0.142</td>
<td>0.067</td>
<td>0.009</td>
</tr>
<tr>
<td>2</td>
<td>0.301</td>
<td>0.264</td>
<td>0.212</td>
<td>0.137</td>
<td>0.069</td>
<td>0.017</td>
</tr>
<tr>
<td>True</td>
<td>0.269</td>
<td>0.235</td>
<td>0.220</td>
<td>0.165</td>
<td>0.081</td>
<td>0.031</td>
</tr>
<tr>
<td>Rank</td>
<td>0.089</td>
<td>0.171</td>
<td>0.235</td>
<td>0.300</td>
<td>0.176</td>
<td>0.029</td>
</tr>
<tr>
<td>5</td>
<td>0.017</td>
<td>0.050</td>
<td>0.113</td>
<td>0.191</td>
<td>0.413</td>
<td>0.217</td>
</tr>
<tr>
<td>6</td>
<td>0.004</td>
<td>0.012</td>
<td>0.026</td>
<td>0.065</td>
<td>0.195</td>
<td>0.698</td>
</tr>
</tbody>
</table>

P-screen Method

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.340</td>
<td>0.273</td>
<td>0.185</td>
<td>0.131</td>
<td>0.058</td>
<td>0.013</td>
</tr>
<tr>
<td>2</td>
<td>0.269</td>
<td>0.271</td>
<td>0.219</td>
<td>0.142</td>
<td>0.077</td>
<td>0.022</td>
</tr>
<tr>
<td>True</td>
<td>0.265</td>
<td>0.227</td>
<td>0.226</td>
<td>0.169</td>
<td>0.075</td>
<td>0.039</td>
</tr>
<tr>
<td>Rank</td>
<td>0.099</td>
<td>0.158</td>
<td>0.222</td>
<td>0.271</td>
<td>0.202</td>
<td>0.047</td>
</tr>
<tr>
<td>5</td>
<td>0.023</td>
<td>0.055</td>
<td>0.120</td>
<td>0.203</td>
<td>0.376</td>
<td>0.222</td>
</tr>
<tr>
<td>6</td>
<td>0.003</td>
<td>0.016</td>
<td>0.028</td>
<td>0.084</td>
<td>0.213</td>
<td>0.657</td>
</tr>
</tbody>
</table>
Table 3

Probability of Correctly Ranking Each Task:
Type 1 Distribution with \( J=20 \) and GSD=2.

**Multiple Linear Regression**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.333</td>
<td>0.296</td>
<td>0.181</td>
<td>0.141</td>
<td>0.048</td>
<td>0.001</td>
</tr>
<tr>
<td>2</td>
<td>0.451</td>
<td>0.212</td>
<td>0.182</td>
<td>0.116</td>
<td>0.038</td>
<td>0.002</td>
</tr>
<tr>
<td>True Rank</td>
<td>3</td>
<td>0.169</td>
<td>0.293</td>
<td>0.295</td>
<td>0.174</td>
<td>0.061</td>
</tr>
<tr>
<td>4</td>
<td>0.046</td>
<td>0.177</td>
<td>0.257</td>
<td>0.345</td>
<td>0.145</td>
<td>0.030</td>
</tr>
<tr>
<td>5</td>
<td>0.002</td>
<td>0.019</td>
<td>0.063</td>
<td>0.160</td>
<td>0.504</td>
<td>0.253</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.004</td>
<td>0.022</td>
<td>0.064</td>
<td>0.204</td>
<td>0.706</td>
</tr>
</tbody>
</table>

**P-screen Method**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.356</td>
<td>0.302</td>
<td>0.173</td>
<td>0.120</td>
<td>0.047</td>
<td>0.002</td>
</tr>
<tr>
<td>2</td>
<td>0.426</td>
<td>0.246</td>
<td>0.158</td>
<td>0.126</td>
<td>0.039</td>
<td>0.005</td>
</tr>
<tr>
<td>True Rank</td>
<td>3</td>
<td>0.173</td>
<td>0.287</td>
<td>0.337</td>
<td>0.159</td>
<td>0.036</td>
</tr>
<tr>
<td>4</td>
<td>0.044</td>
<td>0.141</td>
<td>0.250</td>
<td>0.383</td>
<td>0.146</td>
<td>0.036</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.017</td>
<td>0.060</td>
<td>0.153</td>
<td>0.524</td>
<td>0.245</td>
</tr>
<tr>
<td>6</td>
<td>0.001</td>
<td>0.007</td>
<td>0.023</td>
<td>0.058</td>
<td>0.207</td>
<td>0.704</td>
</tr>
</tbody>
</table>
Table 4

Probability of Correctly Ranking Each Task:
Type 2 Distribution with J=20 and GSD=2.

### Multiple Linear Regression

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.307</td>
<td>0.272</td>
<td>0.186</td>
<td>0.189</td>
<td>0.045</td>
<td>0.001</td>
</tr>
<tr>
<td>2</td>
<td>0.427</td>
<td>0.208</td>
<td>0.192</td>
<td>0.147</td>
<td>0.025</td>
<td>0.000</td>
</tr>
<tr>
<td>True</td>
<td>0.209</td>
<td>0.300</td>
<td><strong>0.303</strong></td>
<td>0.150</td>
<td>0.035</td>
<td>0.003</td>
</tr>
<tr>
<td>Rank</td>
<td>0.056</td>
<td>0.210</td>
<td>0.261</td>
<td><strong>0.355</strong></td>
<td>0.108</td>
<td>0.010</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.010</td>
<td>0.054</td>
<td>0.133</td>
<td><strong>0.624</strong></td>
<td>0.179</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.005</td>
<td>0.026</td>
<td>0.163</td>
<td><strong>0.806</strong></td>
</tr>
</tbody>
</table>

### P-screen Method

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.338</td>
<td>0.287</td>
<td>0.173</td>
<td>0.156</td>
<td>0.045</td>
<td>0.001</td>
</tr>
<tr>
<td>2</td>
<td>0.414</td>
<td>0.226</td>
<td>0.170</td>
<td>0.160</td>
<td>0.029</td>
<td>0.001</td>
</tr>
<tr>
<td>True</td>
<td>0.200</td>
<td>0.312</td>
<td><strong>0.325</strong></td>
<td>0.133</td>
<td>0.025</td>
<td>0.005</td>
</tr>
<tr>
<td>Rank</td>
<td>0.046</td>
<td>0.165</td>
<td>0.279</td>
<td><strong>0.376</strong></td>
<td>0.117</td>
<td>0.016</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.008</td>
<td>0.047</td>
<td>0.142</td>
<td><strong>0.616</strong></td>
<td>0.186</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.001</td>
<td>0.006</td>
<td>0.034</td>
<td>0.169</td>
<td><strong>0.791</strong></td>
</tr>
</tbody>
</table>
Table 5

Probability of Correctly Ranking Each Task:
Type 1 Distribution with J=100 and GSD=4.

**Multiple Linear Regression**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.615</td>
<td>0.255</td>
<td>0.100</td>
<td>0.028</td>
<td>0.002</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.254</td>
<td>0.391</td>
<td>0.264</td>
<td>0.081</td>
<td>0.010</td>
<td>0.001</td>
</tr>
<tr>
<td>True Rank</td>
<td>0.110</td>
<td>0.258</td>
<td>0.380</td>
<td>0.219</td>
<td>0.033</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.021</td>
<td>0.092</td>
<td>0.228</td>
<td>0.545</td>
<td>0.110</td>
<td>0.003</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.004</td>
<td>0.028</td>
<td>0.124</td>
<td>0.770</td>
<td>0.073</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.003</td>
<td>0.074</td>
<td>0.923</td>
</tr>
</tbody>
</table>

**P-screen Method**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.613</td>
<td>0.260</td>
<td>0.099</td>
<td>0.027</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.258</td>
<td>0.406</td>
<td>0.245</td>
<td>0.082</td>
<td>0.008</td>
<td>0.001</td>
</tr>
<tr>
<td>True Rank</td>
<td>0.105</td>
<td>0.247</td>
<td>0.394</td>
<td>0.224</td>
<td>0.031</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.024</td>
<td>0.084</td>
<td>0.237</td>
<td>0.554</td>
<td>0.099</td>
<td>0.002</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.003</td>
<td>0.025</td>
<td>0.110</td>
<td>0.791</td>
<td>0.070</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
<td>0.003</td>
<td>0.069</td>
<td>0.928</td>
</tr>
</tbody>
</table>
Table 6

Probability of Correctly Ranking Each Task:
Type 2 Distribution with J=100 and GSD=4.

Multiple Linear Regression

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.572</td>
<td>0.267</td>
<td>0.128</td>
<td>0.030</td>
<td>0.003</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.259</td>
<td>0.383</td>
<td>0.264</td>
<td>0.090</td>
<td>0.004</td>
<td>0.000</td>
</tr>
<tr>
<td>True Rank</td>
<td>0.146</td>
<td>0.269</td>
<td>0.372</td>
<td>0.195</td>
<td>0.018</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.022</td>
<td>0.081</td>
<td>0.223</td>
<td>0.618</td>
<td>0.055</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.013</td>
<td>0.066</td>
<td>0.902</td>
<td>0.019</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.019</td>
<td>0.981</td>
</tr>
</tbody>
</table>

P-screen Method

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.575</td>
<td>0.264</td>
<td>0.127</td>
<td>0.032</td>
<td>0.002</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.269</td>
<td>0.390</td>
<td>0.253</td>
<td>0.086</td>
<td>0.002</td>
<td>0.000</td>
</tr>
<tr>
<td>True Rank</td>
<td>0.129</td>
<td>0.267</td>
<td>0.381</td>
<td>0.208</td>
<td>0.014</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.026</td>
<td>0.078</td>
<td>0.232</td>
<td>0.918</td>
<td>0.018</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.001</td>
<td>0.008</td>
<td>0.055</td>
<td>0.917</td>
<td>0.018</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.018</td>
<td>0.982</td>
</tr>
</tbody>
</table>
Table 7

Probability of Correctly Ranking Each Task:
Type 1 Distribution with J=100 and GSD=2.

**Multiple Linear Regression**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.728</td>
<td>0.242</td>
<td>0.028</td>
<td>0.002</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.241</td>
<td>0.607</td>
<td>0.129</td>
<td>0.023</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>0.030</td>
<td>0.134</td>
<td>0.667</td>
<td>0.167</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>0.016</td>
<td>0.175</td>
<td>0.773</td>
<td>0.034</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.003</td>
<td>0.035</td>
<td>0.957</td>
<td>0.007</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.007</td>
<td>0.993</td>
</tr>
</tbody>
</table>

**P-screen Method**

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.680</td>
<td>0.275</td>
<td>0.041</td>
<td>0.003</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.286</td>
<td>0.579</td>
<td>0.111</td>
<td>0.024</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>0.032</td>
<td>0.132</td>
<td>0.661</td>
<td>0.174</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>0.013</td>
<td>0.186</td>
<td>0.774</td>
<td>0.025</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
<td>0.025</td>
<td>0.968</td>
<td>0.006</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.006</td>
<td>0.994</td>
</tr>
</tbody>
</table>
Table 8

Probability of Correctly Ranking Each Task:
Type 2 Distribution with J=100 and GSD=2.

Multiple Linear Regression

Assigned Rank

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.653</td>
<td>0.267</td>
<td>0.072</td>
<td>0.007</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.280</td>
<td>0.523</td>
<td>0.162</td>
<td>0.035</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>True</td>
<td>0.063</td>
<td>0.189</td>
<td>0.582</td>
<td>0.166</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Rank</td>
<td>0.004</td>
<td>0.020</td>
<td>0.184</td>
<td>0.779</td>
<td>0.013</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.013</td>
<td>0.987</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

P-screen Method

Assigned Rank

<table>
<thead>
<tr>
<th>Assigned Rank</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.580</td>
<td>0.308</td>
<td>0.096</td>
<td>0.017</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.343</td>
<td>0.477</td>
<td>0.143</td>
<td>0.037</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>True</td>
<td>0.074</td>
<td>0.183</td>
<td>0.571</td>
<td>0.172</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Rank</td>
<td>0.004</td>
<td>0.032</td>
<td>0.191</td>
<td>0.768</td>
<td>0.005</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.005</td>
<td>0.994</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.999</td>
</tr>
</tbody>
</table>
indicated that at least one MI data set was not equal to any others in each experiment. Therefore, there were statistically significant differences in performance between task time matrices (using MIs).

Correlation coefficients of the MIs (using 3000 computed values) for the two methods were also calculated for each experiment (Table 12). The MI correlation coefficients ranged from 0.66 - 0.86. Correlation increased with increasing number of samples, overlap, and GSD. The magnitude of the correlation coefficients indicated that although there was a tendency for both methods to misclassify the same runs, the degree of misclassification was not the same for both methods.

A review of the estimated task medians showed that the P-screen method was positively biased on the higher concentration tasks and the multiple linear regression method was also positively biased but for the lower concentration tasks. The estimated task medians for the regression method were “compressed” and the P-screen estimates were “expanded.”
Table 9

Number of runs (out of 3000) that the highest two tasks were correctly ranked.

<table>
<thead>
<tr>
<th># Samples</th>
<th>Type 1 Distribution</th>
<th>Type 2 Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD=2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Regression: 2872</td>
<td>Regression: 2960</td>
</tr>
<tr>
<td></td>
<td>P-Screen: 2905</td>
<td>P-Screen: 2983</td>
</tr>
<tr>
<td>20</td>
<td>Regression: 1469</td>
<td>Regression: 1862</td>
</tr>
<tr>
<td></td>
<td>P-Screen: 1517</td>
<td>P-Screen: 1830</td>
</tr>
<tr>
<td>GSD=4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Regression: 2308</td>
<td>Regression: 2705</td>
</tr>
<tr>
<td></td>
<td>P-Screen: 2371</td>
<td>P-Screen: 2750</td>
</tr>
<tr>
<td>20</td>
<td>Regression: 843</td>
<td>Regression: 1171</td>
</tr>
<tr>
<td></td>
<td>P-Screen: 743</td>
<td>P-Screen: 1043</td>
</tr>
</tbody>
</table>
Table 10

Number of runs (out of 3000) that were correctly ranked.

<table>
<thead>
<tr>
<th>Type 1 Distribution</th>
<th>Type 2 Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD=2</td>
<td>GSD=2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td># Samples</td>
<td># Samples</td>
</tr>
<tr>
<td>Regression</td>
<td>Regression</td>
</tr>
<tr>
<td>P-Screen</td>
<td>P-Screen</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>1377</td>
<td>1173</td>
</tr>
<tr>
<td>1313</td>
<td>1031</td>
</tr>
<tr>
<td>135</td>
<td>162</td>
</tr>
<tr>
<td>180</td>
<td>177</td>
</tr>
<tr>
<td>542</td>
<td>601</td>
</tr>
<tr>
<td>547</td>
<td>618</td>
</tr>
<tr>
<td>79</td>
<td>100</td>
</tr>
<tr>
<td>66</td>
<td>91</td>
</tr>
<tr>
<td>Type 1 Distribution</td>
<td>Type 2 Distribution</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GSD=2</strong></td>
<td><strong>GSD=2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td># Samples</td>
<td># Samples</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression</td>
<td>Regression</td>
</tr>
<tr>
<td>0.0766</td>
<td>0.0938</td>
</tr>
<tr>
<td>0.2880</td>
<td>0.2774</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>P-Screen</td>
<td>P-Screen</td>
</tr>
<tr>
<td>0.0813</td>
<td>0.1050</td>
</tr>
<tr>
<td>0.2745</td>
<td>0.2663</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GSD=4</strong></td>
<td><strong>GSD=4</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td># Samples</td>
<td># Samples</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression</td>
<td>Regression</td>
</tr>
<tr>
<td>0.1646</td>
<td>0.1538</td>
</tr>
<tr>
<td>0.3694</td>
<td>0.3361</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>P-Screen</td>
<td>P-Screen</td>
</tr>
<tr>
<td>0.1599</td>
<td>0.1506</td>
</tr>
<tr>
<td>0.3794</td>
<td>0.3452</td>
</tr>
</tbody>
</table>

*Average misclassification indices for 3000 runs.
Table 12
Correlation Coefficients*

<table>
<thead>
<tr>
<th>Type 1 Distribution</th>
<th>Type 2 Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSD=2</td>
<td>GSD=2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Correlation</th>
<th></th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.749</td>
<td>100</td>
<td>0.706</td>
</tr>
<tr>
<td>20</td>
<td>0.724</td>
<td>20</td>
<td>0.661</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Correlation</th>
<th></th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.863</td>
<td>100</td>
<td>0.842</td>
</tr>
<tr>
<td>20</td>
<td>0.784</td>
<td>20</td>
<td>0.756</td>
</tr>
</tbody>
</table>

*Correlation coefficients of the two methods using 3000 misclassifications indices.
CHAPTER V

DISCUSSION AND CONCLUSION

Multiple linear regression, used in many professions for many years, has been accepted as the preferred data analysis method to establish relationships between independent and dependent variables. The observation (based on the summary statistics) that the P-screen method performed similarly may be of interest to anyone conducting data analysis because each method has assumptions and limitations. Assumptions of multiple regression include: the model is linear, and the error terms are independent, have constant variance, and are normally distributed. The P-screen method will not work for those jobs that require each task to be performed to complete the job (assembly line). For each sample, at least one task cannot be performed so that the number of unknowns (task medians) equals the number of simultaneous equations to be solved. Those samples for which all tasks were performed would have to be removed from the data set prior to analysis.

An understanding of each of the methods' assumptions and limitations allows someone to add to his data analysis "tool box." Furthermore, the comparative similar performance may indicate that a "revised P-screen" or some other alternative method may exist which can outperform the multiple linear regression or at least may not be subject to the same assumptions and limitations.
While the estimated versus actual ranking probabilities (Tables 1 - 8) tracked very closely, it would be necessary to conduct several replicates of the experiment to determine whether the small differences in performance (summary statistics) are statistically significant. The correlation coefficients of the MIs suggested that one method may perform better under certain conditions; however, the analyses conducted during this study did not identify these factors.

All statistical data analysis methods have limitations. It is important for the occupational hygienist to understand these limitations. The ability to check the results using two methods (as compared to only checking model assumptions of one method) could provide a more complete understanding of the data. In particular, this could be the case if the specific conditions under which one method performs better were known.

Important occupational health decisions are made based on the results of data analysis such as: whether or not to install engineering controls, whether or not personal protective equipment should be required, and whether the operation is in regulatory agency compliance. Any opportunities to improve data analysis techniques should be explored since the improvements may further protect the worker’s health. As an occupational health program strategy moves away from being compliance driven towards a comprehensive exposure assessment program, data analysis will become a more integral part of the hygienist’s duties. Use of a task ranking method could help advance an occupational health program in two ways: (1) it could identify high exposure tasks from...
historical multi-task data, and (2) it could reduce the total number of samples required to comprehensively assess all potential hazards. Any improvements in data analysis techniques could enhance the decision making of these programs.
BIBLIOGRAPHY


APPENDIX A

ALGORITHMS OF MULTIPLE LINEAR REGRESSION
MATRIX AND VECTORS WITH AN EXAMPLE OF A SOLUTION
\[ LRA = m \left( \sum x_1 \sum x_2 \sum x_3 \sum x_4 \sum x_5 \sum x_6 \right) \]

\[ \sum x_1 \sum x_1^2 \sum x_1 x_2 \sum x_1 x_3 \sum x_1 x_4 \sum x_1 x_5 \sum x_1 x_6 \]

\[ \sum x_2 \sum x_2 x_1 \sum x_2^2 \sum x_2 x_3 \sum x_2 x_4 \sum x_2 x_5 \sum x_2 x_6 \]

\[ \sum x_3 \sum x_3 x_1 \sum x_3 x_2 \sum x_3^2 \sum x_3 x_4 \sum x_3 x_5 \sum x_3 x_6 \]

\[ \sum x_4 \sum x_4 x_1 \sum x_4 x_2 \sum x_4 x_3 \sum x_4^2 \sum x_4 x_5 \sum x_4 x_6 \]

\[ \sum x_5 \sum x_5 x_1 \sum x_5 x_2 \sum x_5 x_3 \sum x_5 x_4 \sum x_5^2 \sum x_5 x_6 \]

\[ \sum x_6 \sum x_6 x_1 \sum x_6 x_2 \sum x_6 x_3 \sum x_6 x_4 \sum x_6 x_5 \sum x_6^2 \]

\[ LRB = \sum y \]

\[ \sum x_1 y \]

\[ \sum x_2 y \]

\[ \sum x_3 y \]

\[ \sum x_4 y \]

\[ \sum x_5 y \]

\[ \sum x_6 y \]

\[ \alpha \]

\[ \beta_1 \]

\[ \beta_2 \]

\[ X = \beta_3 \]

\[ \beta_4 \]

\[ \beta_5 \]

\[ \beta_6 \]

34
EXAMPLE OF A MULTIPLE LINEAR REGRESSION SOLUTION.

THE SOLUTION SHOWS THE STEPS REQUIRED (AS DESCRIBED IN METHODS AND MATERIALS) TO OBTAIN AN INTERCEPT AND REGRESSION COEFFICIENTS FOR A SCENARIO WITH TEN SAMPLES AND SIX TASKS.
<table>
<thead>
<tr>
<th>TASK OCCUR</th>
<th>LRA</th>
<th>LRA -1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1 0 1</td>
<td>10 4 7 8 9 2 9</td>
<td>-3 -6E-17 -4E-16 -3E-16 -2 -1 -1</td>
</tr>
<tr>
<td>0 1 1 1 1 1</td>
<td>4 4 2 4 4 0 4</td>
<td>-6E-16 1 0.5 -0.5 -0.5 0.5 3E-16</td>
</tr>
<tr>
<td>0 0 1 0 1 1</td>
<td>7 2 7 5 7 1 6</td>
<td>-2E-15 0.5 1 -5E-17 -1 6.2E-16 5E-16</td>
</tr>
<tr>
<td>1 0 1 1 0 1</td>
<td>8 4 5 8 7 2 8</td>
<td>-1E-15 -0.5 -1E-16 1.5 1E-15 -0.5 -1</td>
</tr>
<tr>
<td>0 1 0 1 0 0</td>
<td>9 4 7 7 9 1 8</td>
<td>-2 -0.5 -1 1E-16 3 1 5E-17</td>
</tr>
<tr>
<td>0 1 1 1 0 1</td>
<td>2 0 1 2 1 2 2</td>
<td>-1 0.5 2.6E-16 -0.5 1 1.5 3E-16</td>
</tr>
<tr>
<td>0 1 0 1 0 1</td>
<td>9 4 6 8 8 2 9</td>
<td>-1 0 0 -1 0 0 2</td>
</tr>
</tbody>
</table>

Sum of X: 4 7 8 9 2 9
Sum of X^2: 4 7 8 9 2 9

Sample concentrations:
- 2.495781: 28.1
- 3.384271: 9.65
- 4.246408: 19.4
- 3.33527: 22.6
- 1.024439: 23.8
- 1.03136: 7.63
- 2.905559: 25.2
- 4.322038
- 2.531066
- 2.78459

Solution

\[ LRB \]

\[ X : \]

<table>
<thead>
<tr>
<th>Rank</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>0.033</td>
<td>5 2</td>
</tr>
<tr>
<td>0.4569</td>
<td>4 3</td>
</tr>
<tr>
<td>0.1422</td>
<td>1 4</td>
</tr>
<tr>
<td>0.7111</td>
<td>6 5</td>
</tr>
<tr>
<td>0.374</td>
<td>2 6</td>
</tr>
</tbody>
</table>
APPENDIX B

COMPUTER PROGRAM USED TO COMPARE THE TWO METHODS
REM LRST1005.BAS
REM This program uses more finely quantized task duration inputs (1/96 of shift) and
  censors tasks with duration < 5% of sample period

REM Task concentration distribution is log-normal but dependent on averaging time.
REM The underlying short-time distributions are assumed to be second-order stationary
REM Successive short-time concentrations during a task are assumed to be totally
  uncorrelated.
REM The nominal GM and GSD are for the task taking 1/6 of shift.
REM GSDs of task distributions may differ from each other within specified bounds.

REM In this version the GSDs of the 1/6 time weighted task concentration distributions
  are allowed to differ from nominal value within some defined bounds

REM Samples are taken randomly from 5 different workers, each of whom has
  characteristicly higher or lower than median exposure for each task
REM Uses estimated task thetas to calculate p-screened theta-hat matrix

REM The program allows multiple (i.set) task duration matrices to be used, with (cyc)
  replications for each theta matrix.
REM The task duration inputs are generated using the "tasktime" subroutine

REM Assignment of exposures
  DEFSTR U
  DEFINT I-K
  DIM estvact(6, 6, 2)
  DIM workereffect(5)
  DIM weight(100, 6)
  DIM esam(100, 2)
  DIM th(6)
  DIM theta(100, 6)
  DIM m(6, 2)
  DIM time(100, 6)
  DIM ord(100, 6)
  DIM ran(100, 6)
  DIM pthat(6, 6)
  DIM cumrank(6, 2)
  DIM averank(6, 2)
  DIM worker(100)
  DIM p(100, 2)
DIM pr(6, 2)
DIM q(6, 2)
DIM r(6, 2)
DIM a(6, 7)
DIM pscreen(6, 100)
DIM zer(6)
DIM GSD(2)
DIM sigtask(6, 2)
DIM sigdiff(6, 2)
DIM rantask(6)
DIM taskord(6)
DIM arithmean(6, 2)
DIM arithvar(6, 2)
DIM TASKOCCUR(100, 6)
DIM y(100), LRA(7, 7), LRB(7, 2)
DIM LRAB(7, 8), LRx(7)
DIM LRq(6, 2), LRr(6, 2)
DIM LRcumrank(6, 2), LRaverank(6, 2)
DIM LRqtemp(7, 2)
DIM mlr(3000, 2), mps(3000, 2)
DIM LRmedian(6, 2)

REM $DYNAMIC
DIM x(6)
REM $DYNAMIC
DIM h(6, 7)
REM $DYNAMIC
DIM xx(6)
REM $DYNAMIC
DIM i.to(6)
REM $DYNAMIC
DIM i.from(6)
REM $DYNAMIC
DIM b(6, 7)
REM $DYNAMIC
DIM c(6)
i.fail = 0
u.fail = ""
REM create range of task GSDs
sig1 = 2: sig2 = 2
sigdev = 0
FOR i.t = 1 TO 6
   sigdiff(i.t, 1) = sig1 - sigdev * (1.4 - i.t * .4)
sigdiff(i.t, 2) = sig2 - sigdev * (1.4 - i.t * .4)
NEXT i.t
OPEN "c:\Qb45\results\PSresult.txt" FOR OUTPUT AS #3
OPEN "c:\Qb45\results\LRresult.txt" FOR OUTPUT AS #2
OPEN "c:\Qb45\results\setmisav.txt" FOR OUTPUT AS #4
OPEN "c:\Qb45\results\runsmis.txt" FOR OUTPUT AS #5
OPEN "c:\Qb45\results\medians1.txt" FOR OUTPUT AS #6
OPEN "c:\Qb45\results\medians2.txt" FOR OUTPUT AS #7
WRITE #6, "TYPE -1", " 1, 2, 4, 8, 16, 32 "
WRITE #7, "TYPE -2", "0.5, 1.4, 4, 11.3, 32, 90.5 "
WRITE #4, "Dataset, Avg&var mlr1, Avg&var mlr2, Avg&var mpsl, Avg&var mps2"
WRITE #5, "Run, mlr1, mlr2, mps1, mps2"

REM read work distributions:
CLS
sfac = 0
FOR i = 1 TO 5
   workereffect(i) = sfac - LOG(l)
sfac = sfac + (LOG(l))/2
NEXT i
runs = 0: dataset = 0: cumhi1 = 0: cumhi2 = 0
LRcumhi1 = 0: LRcumhi2 = 0
cummlr1 = 0: cummlr2 = 0: cummps1 = 0: cummps2 = 0

FOR i.set = 1 TO 30
   dataset = dataset + 1
   WRITE #3,
   WRITE #3, "Dataset ", dataset
   WRITE #2,
   WRITE #2, "Dataset ", dataset

ERASE exam
ERASE th
ERASE theta
ERASE m
FOR i.sample = 1 TO 100
s = 0
FOR i.task = 1 TO 6
IF weight(i.sample, i.task) > 4.8 THEN
s = s + 1
TASKOCCUR(i.sample, i.task) = 1
ELSE
TASKOCCUR(i.sample, i.task) = 0
pscreen(i.task, i.sample) = 1
zer(i.task) = zer(i.task) + 1
END IF
NEXT i.task
thet = 1 / s
FOR i = 1 TO 6
IF weight(i.sample, i) > 4.8 THEN
theta(i.sample, i) = thet
ELSE \( \theta(i, \text{sample}, i) = 0 \)
END IF
NEXT \( i \)
NEXT \( i, \text{sample} \)

REM generate p-screened theta hats and mmedians - Medians are absolute.
REM The generated medians are used for comparison:
WRITE #3, "P-screened theta hat matrix"
FOR \( i, \text{tr} = 1 \) TO 6
FOR \( i, \text{tc} = 1 \) TO 6
FOR \( i, \text{s} = 1 \) TO 100
\( p\hat{\theta}(i, \text{tr}, i, \text{tc}) = p\hat{\theta}(i, \text{tr}, i, \text{tc}) + \theta(i, \text{s}, i, \text{tc}) \times \text{pscreen}(i, \text{tr}, i, \text{s}) \)
NEXT \( i, \text{s} \)
\( p\hat{\theta}(i, \text{tr}, i, \text{tc}) = p\hat{\theta}(i, \text{tr}, i, \text{tc}) / \text{zer}(i, \text{tr}) \)
NEXT \( i, \text{tc} \)
WRITE #3, \( p\hat{\theta}(i, \text{tr}, 1), p\hat{\theta}(i, \text{tr}, 2), p\hat{\theta}(i, \text{tr}, 3), p\hat{\theta}(i, \text{tr}, 4), p\hat{\theta}(i, \text{tr}, 5), \)
\( p\hat{\theta}(i, \text{tr}, 6) \)
\( m(i, \text{tr}, 1) = 2 \times (i, \text{tr} - 1) \)
NEXT \( i, \text{tr} \)

REM random assignment of GSD to task
FOR \( i, \text{ran} = 1 \) TO 6
\( i, \text{ord} = \text{taskord}(i, \text{ran}) \)
\( \text{sigtask}(i, \text{ord}, 1) = \text{sigdiff}(i, \text{ran}, 1) \)
\( \text{sigtask}(i, \text{ord}, 2) = \text{sigdiff}(i, \text{ran}, 2) \)
NEXT \( i, \text{ran} \)
FOR \( i, \text{tr} = 1 \) TO 6
\( \text{arithmean}(i, \text{tr}, 1) = m(i, \text{tr}, 1) \times \text{EXP}((\text{LOG}(\text{sigtask}(i, \text{tr}, 1)) / 2) \)
\( \text{arithvar}(i, \text{tr}, 1) = 16 \times (\text{arithmean}(i, \text{tr}, 1) ^ 2) \times (\text{EXP}((\text{LOG}(\text{sigtask}(i, \text{tr}, 1))) ^ 2) - 1) \)
\( m(i, \text{tr}, 2) = 2 \times (1.5 \times i, \text{tr} - 2.5) \)
\( \text{arithmean}(i, \text{tr}, 2) = m(i, \text{tr}, 2) \times \text{EXP}((\text{LOG}(\text{sigtask}(i, \text{tr}, 2))) / 2) \)
\( \text{arithvar}(i, \text{tr}, 2) = 16 \times (\text{arithmean}(i, \text{tr}, 2) ^ 2) \times (\text{EXP}((\text{LOG}(\text{sigtask}(i, \text{tr}, 2))) ^ 2) - 1) \)
NEXT \( i, \text{tr} \)

REM Setting up the LRA matrix
REM First Row
PRINT "The LRA Matrix"
LRA(1, 1) = 100
FOR \( J = 1 \) TO 100
\( \text{ATEMP12} = \text{TASKOCCUR}(J, 1) \)
LRA(1, 2) = ATEMP12 + LRA(1, 2)
ATEMP13 = TASKOCCUR(J, 2)
LRA(1, 3) = ATEMP13 + LRA(1, 3)
ATEMP14 = TASKOCCUR(J, 3)
LRA(1, 4) = ATEMP14 + LRA(1, 4)
ATEMP15 = TASKOCCUR(J, 4)
LRA(1, 5) = ATEMP15 + LRA(1, 5)
ATEMP16 = TASKOCCUR(J, 5)
LRA(1, 6) = ATEMP16 + LRA(1, 6)
ATEMP17 = TASKOCCUR(J, 6)
LRA(1, 7) = ATEMP17 + LRA(1, 7)
NEXT J

LRA(2, 1) = LRA(1, 2)
LRA(3, 1) = LRA(1, 3)
LRA(4, 1) = LRA(1, 4)
LRA(5, 1) = LRA(1, 5)
LRA(6, 1) = LRA(1, 6)
LRA(7, 1) = LRA(1, 7)

REM 'LRA' matrix diagonal elements, except (1,1)

FOR J = 1 TO 100
ATEMP22 = TASKOCCUR(J, 1) ^ 2
LRA(2, 2) = ATEMP22 + LRA(2, 2)
ATEMP33 = TASKOCCUR(J, 2) ^ 2
LRA(3, 3) = ATEMP33 + LRA(3, 3)
ATEMP44 = TASKOCCUR(J, 3) ^ 2
LRA(4, 4) = ATEMP44 + LRA(4, 4)
ATEMP55 = TASKOCCUR(J, 4) ^ 2
LRA(5, 5) = ATEMP55 + LRA(5, 5)
ATEMP66 = TASKOCCUR(J, 5) ^ 2
LRA(6, 6) = ATEMP66 + LRA(6, 6)
ATEMP77 = TASKOCCUR(J, 6) ^ 2
LRA(7, 7) = ATEMP77 + LRA(7, 7)
NEXT J

REM Intermultiplied elements

FOR J = 1 TO 100

REM
ATEMP23 = TASKOCCUR(J, 1) * TASKOCCUR(J, 2)
LRA(2, 3) = ATEMP23 + LRA(2, 3)
ATEMP24 = TASKOCCUR(J, 1) * TASKOCCUR(J, 3)
LRA(2, 4) = ATEMP24 + LRA(2, 4)
ATEMP25 = TASKOCCUR(J, 1) * TASKOCCUR(J, 4)
LRA(2, 5) = ATEMP25 + LRA(2, 5)
ATEMP26 = TASKOCCUR(J, 1) * TASKOCCUR(J, 5)
LRA(2, 6) = ATEMP26 + LRA(2, 6)
ATEMP27 = TASKOCCUR(J, 1) * TASKOCCUR(J, 6)
LRA(2, 7) = ATEMP27 + LRA(2, 7)
ATEMP34 = TASKOCCUR(J, 2) * TASKOCCUR(J, 3)
LRA(3, 4) = ATEMP34 + LRA(3, 4)
ATEMP35 = TASKOCCUR(J, 2) * TASKOCCUR(J, 4)
LRA(3, 5) = ATEMP35 + LRA(3, 5)
ATEMP36 = TASKOCCUR(J, 2) * TASKOCCUR(J, 5)
LRA(3, 6) = ATEMP36 + LRA(3, 6)
ATEMP37 = TASKOCCUR(J, 2) * TASKOCCUR(J, 6)
LRA(3, 7) = ATEMP37 + LRA(3, 7)
ATEMP45 = TASKOCCUR(J, 3) * TASKOCCUR(J, 4)
LRA(4, 5) = ATEMP45 + LRA(4, 5)
ATEMP46 = TASKOCCUR(J, 3) * TASKOCCUR(J, 5)
LRA(4, 6) = ATEMP46 + LRA(4, 6)
ATEMP47 = TASKOCCUR(J, 3) * TASKOCCUR(J, 6)
LRA(4, 7) = ATEMP47 + LRA(4, 7)
ATEMP56 = TASKOCCUR(J, 4) * TASKOCCUR(J, 5)
LRA(5, 6) = ATEMP56 + LRA(5, 6)
ATEMP57 = TASKOCCUR(J, 4) * TASKOCCUR(J, 6)
LRA(5, 7) = ATEMP57 + LRA(5, 7)
ATEMP67 = TASKOCCUR(J, 5) * TASKOCCUR(J, 6)
LRA(6, 7) = ATEMP67 + LRA(6, 7)

NEXT J

LRA(3, 2) = LRA(2, 3)
LRA(4, 2) = LRA(2, 4)
LRA(4, 3) = LRA(3, 4)
LRA(5, 2) = LRA(2, 5)
LRA(5, 3) = LRA(3, 5)
LRA(5, 4) = LRA(4, 5)
LRA(6, 2) = LRA(2, 6)
LRA(6, 3) = LRA(3, 6)
LRA(6, 4) = LRA(4, 6)
LRA(6, 5) = LRA(5, 6)
LRA(7, 2) = LRA(2, 7)
LRA(7, 3) = LRA(3, 7)
LRA(7, 4) = LRA(4, 7)
LRA(7, 5) = LRA(5, 7)
LRA(7, 6) = LRA(6, 7)

REM SIMULATION WITH SOLUTIONS
REM Cycles

a$(1) = "Type -1": a$(2) = "Type -2"
LRctl = 0: LRct2 = 0: LRhi1 = 0: LRhi2 = 0
ctl = 0: ct2 = 0: cyc = 0: hi1 = 0: hi2 = 0
setmlr1 = 0: setmlr2 = 0: setmps1 = 0: setmps2 = 0

FOR i.cyc = 1 TO 100

REM generate exposure distributions for type 1 and type 2
RANDOMIZE (VAL(RIGHT$(TIME$, 2)))
WRITE #2, "TASKOCCUR Matrix"
WRITE #3, "Worker, Task Durations and Sample Concentrations"
FOR i.s = 1 TO 100
s1 = 0: s2 = 0: l2 = LOG(2)
worker(i.s) = INT(5 * RND) + 1
effect = workereffect(worker(i.s))
FOR i.t = 1 TO 6
p = RND: q = RND
norm = SQR(-2 * LOG(p)) * COS(6.28318 * q)
IF weight(i.s, i.t) > 0 THEN
sgl = LOG(EXP(SQR(LOG((arithvar(i.t, 1) / weight(i.s, i.t)) / (arithmean(i.t, 1) ^ 2) + 1))) - .5 + RND): sg2 = LOG(EXP(SQR(LOG((arithvar(i.t, 2) / weight(i.s, i.t)) / (arithmean(i.t, 2) ^ 2) + 1))) - .5 + RND)
median1 = LOG(arithmean(i.t, 1)) - (sgl ^ 2) / 2: median2 = LOG(arithmean(i.t, 2)) - (sg2 ^ 2) / 2
END IF
s1 = s1 + weight(i.s, i.t) * EXP((sg1 * norm + effect + median1)) / 96
s2 = s2 + weight(i.s, i.t) * EXP((sg2 * norm + effect + median2)) / 96
NEXT i.t

45
WRITE #3, worker(i.s), weight(i.s, 1), weight(i.s, 2), weight(i.s, 3), weight(i.s, 4),
weight(i.s, 5), weight(i.s, 6), s1, s2
esam(i.s, 1) = LOG(s1)
esam(i.s, 2) = LOG(s2)
WRITE #2, TASKOCCUR(i.s, 1), TASKOCCUR(i.s, 2), TASKOCCUR(i.s, 3),
TASKOCCUR(i.s, 4), TASKOCCUR(i.s, 5), TASKOCCUR(i.s, 6), esam(i.s, 1),
esam(i.s, 2)
NEXT i.s
ERASE LRB

REM Generating the LRB vector
REM
PRINT "The LRB Vector"
FOR i = 1 TO 2
FOR J = 1 TO 100
B1TEMP = esam(J, i)
LRB(1, i) = B1TEMP + LRB(1, i)
B2TEMP = esam(J, i) * TASKOCCUR(J, 1)
LRB(2, i) = B2TEMP + LRB(2, i)
B3TEMP = esam(J, i) * TASKOCCUR(J, 2)
LRB(3, i) = B3TEMP + LRB(3, i)
B4TEMP = esam(J, i) * TASKOCCUR(J, 3)
LRB(4, i) = B4TEMP + LRB(4, i)
B5TEMP = esam(J, i) * TASKOCCUR(J, 4)
LRB(5, i) = B5TEMP + LRB(5, i)
B6TEMP = esam(J, i) * TASKOCCUR(J, 5)
LRB(6, i) = B6TEMP + LRB(6, i)
B7TEMP = esam(J, i) * TASKOCCUR(J, 6)
LRB(7, i) = B7TEMP + LRB(7, i)
NEXT J
NEXT i

WRITE #2, "LRA Matrix", "LRB Matrix 1&2 "
FOR i = 1 TO 7
WRITE #2, LRA(i, 1), LRA(i, 2), LRA(i, 3), LRA(i, 4), LRA(i, 5), LRA(i, 6), LRA(i, 7),
LRB(i, 1), LRB(i, 2)
NEXT i
REM CREATING THE LRAB MATRIX TO SEND TO MATSOLV1

FOR i.type = 1 TO 2
FOR i = 1 TO 7
FOR ii = 1 TO 7
LRAB(i, ii) = LRA(i, ii)
NEXT ii
NEXT i
FOR iii = 1 TO 7
LRAB(iii, 8) = LRB(iii, i.type)
NEXT iii

GOSUB Matsolv1:

REM Assigning the solution vector LRx to LRq Matrix
FOR i.x = 1 TO 7
LRqtemp(i.x, i.type) = LRx(i.x)
NEXT i.x
NEXT i.type

REM Calculating the task median concentrations LR method
FOR ii = 1 TO 2
FOR i = 2 TO 7
LRmedian(i - 1, ii) = EXP(LRqtemp(1, ii) + LRqtemp(i, ii))
NEXT i
NEXT ii

REM Changing the 7 element solution matrix to 6 - eliminating the intercepts
FOR ii = 1 TO 2
FOR i = 1 TO 6
LRq(i, ii) = LRqtemp(i + 1, ii)
NEXT i
NEXT ii

REM Ranking the Beta coefficients
FOR ii = 1 TO 2
FOR i = 1 TO 6
s = 1
FOR J = 1 TO 6
IF LRq(i, ii) > LRq(J, ii) THEN
s = s + 1
END IF
NEXT J
LRr(i, ii) = s
LRcumrank(i, ii) = LRcumrank(i, ii) + s
NEXT i
NEXT ii

WRITE #2, a$(1), LRq(1, 1), LRq(2, 1), LRq(3, 1), LRq(4, 1), LRq(5, 1), LRq(6, 1)
WRITE #2, a$(2), LRr(1, 2), LRr(2, 2), LRr(3, 2), LRr(4, 2), LRr(5, 2), LRr(6, 2)
WRITE #2, a$(2), LRr(1, 2), LRr(2, 2), LRr(3, 2), LRr(4, 2), LRr(5, 2), LRr(6, 2)
WRITE #2, a$(2), LRr(1, 2), LRr(2, 2), LRr(3, 2), LRr(4, 2), LRr(5, 2), LRr(6, 2)

REM Calculating the misclassification of each run for the LR method"

FOR ii = 1 TO 2
mllrsu = 0
FOR i = 1 TO 6
mllrtemp = ABS(i - LRr(i, ii))
mllrsu = mllrsu + mllrtemp
NEXT i
mllr(runs + 1, ii) = mllrsu / 18
NEXT ii

REM Count correctly ranked cycles

d2 = 0
FOR i = 1 TO 6
d2 = d2 + (LRr(i, 1) - i) * (LRr(i, 1) - i)
NEXT i
IF d2 = 0 THEN
LRct1 = LRct1 + 1
END IF
IF LRr(6, 1) = 6 AND LRr(5, 1) = 5 THEN
LRhi1 = LRhi1 + 1
END IF
d2 = 0
FOR i = 1 TO 6
d2 = d2 + (LRr(i, 2) - i) * (LRr(i, 2) - i)
NEXT i
IF d2 = 0 THEN
LRct2 = LRct2 + 1
END IF
IF LRr(6, 2) = 6 AND LRr(5, 2) = 5 THEN
LRhi2 = LRhi2 + 1
END IF
FOR i.type = 1 TO 2
FOR i.act = 1 TO 6
FOR i.est = 1 TO 6
IF LRr(i.act, i.type) = i.est THEN
LRestvact(i.act, i.est, i.type) = LRestvact(i.act, i.est, i.type) + 1
END IF
NEXT i.est
NEXT i.act
NEXT i.type
REM find the overall variances:
s1 = 0: s2 = 0: s12 = 0: s22 = 0
FOR i = 1 TO 100
s1 = s1 + esam(i, 1)
s2 = s2 + esam(i, 2)
s12 = s12 + esam(i, 1) * esam(i, 1)
s22 = s22 + esam(i, 2) * esam(i, 2)
NEXT i
V(1) = (s12 - s1 * s1 / 100) / 99
V(2) = (s22 - s2 * s2 / 100) / 99
REM Calculate Scaled values
FOR i = 1 TO 2
FOR ii = 1 TO 100
p(ii, i) = esam(ii, i) / SQR(V(i))
NEXT ii
NEXT i
REM Calculate Sums
WRITE #3, "P sums"
FOR i = 1 TO 6
s1 = 0: s2 = 0: s3 = 0
FOR ii = 1 TO 100
REM test weight(i.sample, i.task) sum if it is less than 5% of sample period if not skip
IF weight(ii, i) < 4.8 THEN
    s1 = s1 + p(ii, 1)
    s2 = s2 + p(ii, 2)
    s3 = s3 + 1
END IF
NEXT ii
pr(i, 1) = s1 / s3
pr(i, 2) = s2 / s3
WRITE #3, pr(i, 1), pr(i, 2)
NEXT i
REM compute results
FOR i.spread = 1 TO 2
FOR i.t = 1 TO 6
FOR i.c = 1 TO 6
    a(i.t, i.c) = pthat(i.t, i.c)
NEXT i.c
a(i.t, 7) = pr(i.t, i.spread)
PRINT a(i.t, 1), a(i.t, 2), a(i.t, 3), a(i.t, 4), a(i.t, 5), a(i.t, 6), a(i.t, 7)
NEXT i.t
GOSUB Matsolv:
FOR i.t = 1 TO 6
q(i.t, i.spread) = x(i.t)
NEXT i.t
NEXT i.spread
FOR i = 1 TO 6
FOR ij = 1 TO 2
    q(i, ij) = EXP(q(i, ij) * SQR(V(ij)))
PRINT "   Type- ",
PRINT USING "##"; ij;
PRINT "#####.###"; m(i, ij); q(i, ij);
NEXT ij
PRINT
NEXT i
REM ranking the calculated task concentrations
FOR ii = 1 TO 2
FOR i = 1 TO 6
s = 1
FOR J = 1 TO 6
IF q(i, ii) > q(J, ii) THEN
s = s + 1
END IF
NEXT J
r(i, ii) = s
cumrank(i, ii) = cumrank(i, ii) + s
NEXT i
NEXT ii

WRITE #3, a$(1), q(1, 1), q(2, 1), q(3, 1), q(4, 1), q(5, 1), q(6, 1)
WRITE #3, a$(1), r(1, 1), r(2, 1), r(3, 1), r(4, 1), r(5, 1), r(6, 1)
WRITE #3, a$(2), q(1, 2), q(2, 2), q(3, 2), q(4, 2), q(5, 2), q(6, 2)
WRITE #3, a$(2), r(1, 2), r(2, 2), r(3, 2), r(4, 2), r(5, 2), r(6, 2)

WRITE #6, "PScreen", q(1, 1), q(2, 1), q(3, 1), q(4, 1), q(5, 1), q(6, 1)
WRITE #6, "Lin reg", LRmedian(1, 1), LRmedian(2, 1), LRmedian(3, 1),
LRmedian(4, 1), LRmedian(5, 1), LRmedian(6, 1)

WRITE #7, "PScreen", q(1, 2), q(2, 2), q(3, 2), q(4, 2), q(5, 2), q(6, 2)
WRITE #7, "Lin reg", LRmedian(1, 2), LRmedian(2, 2), LRmedian(3, 2),
LRmedian(4, 2), LRmedian(5, 2), LRmedian(6, 2)

REM Calculating the misclassification of the PS method

FOR ii = 1 TO 2
mpssum = 0
FOR i = 1 TO 6
mpstemp = ABS(i - r(i, ii))
mpssum = mpssum + mpstemp
NEXT i
mps(runs + 1, ii) = mpssum / 18
NEXT ii
WRITE #5, runs + 1, mlr(runs + 1, 1), mlr(runs + 1, 2), mps(runs + 1, 1), mps(runs + 1, 2)

REM Count correctly ranked cycles

51
d2 = 0
FOR i = 1 TO 6
  d2 = d2 + (r(i, 1) - i) * (r(i, 1) - i)
NEXT i
IF d2 = 0 THEN
  ct1 = ct1 + 1
END IF
IF r(6, 1) = 6 AND r(5, 1) = 5 THEN
  hil = hi1 + 1
END IF

FOR i = 1 TO 6
  d2 = d2 + (r(i, 2) - i) * (r(i, 2) - i)
NEXT i
IF d2 = 0 THEN
  ct2 = ct2 + 1
END IF
IF r(6, 2) = 6 AND r(5, 2) = 5 THEN
  hi2 = hi2 + 1
END IF

FOR i.type = 1 TO 2
  FOR i.act = 1 TO 6
    FOR i.est = 1 TO 6
      IF r(i.act, i.type) = i.est THEN
        estvact(i.act, i.est, i.type) = estvact(i.act, i.est, i.type) + 1
      END IF
    NEXT i.est
  NEXT i.act
NEXT i.type

setmlr1 = setmlr1 + mlr(runs + 1, 1)
setmlr2 = setmlr2 + mlr(runs + 1, 2)
setmps1 = setmps1 + mps(runs + 1, 1)
setmps2 = setmps2 + mps(runs + 1, 2)
cyc = cyc + 1
runs = runs + 1
NEXT i.cyc
FOR ii = 1 TO 6
  FOR i = 1 TO 2
    averank(ii, i) = cumrank(ii, i) / cyc
  NEXT i
END FOR
NEXT i
WRITE #3, "task", ii, "average rank - type 1", averank(ii, 1), "average rank - type 2", averank(ii, 2)
PRINT "task"; ii; "average rank - type 1"; averank(ii, 1); "average rank - type 2"; averank(ii, 2)
NEXT ii
WRITE #3, ctl, "of", eye, "Type 1 cycles correct"
WRITE #3, ct2, "of", eye, "Type 2 cycles correct"
WRITE #3, hil, "of", eye, "highest 2 conc. correctly ranked for Type 1"
WRITE #3, hi2, "of", eye, "highest 2 conc. correctly ranked for Type 2"
cumhi1 = cumhi2 + hi1
cumhi2 = cumhi2 + hi2
cumct1 = cumct2 + ct1
cumct2 = cumct2 + ct2
cummpsl = cummpsl + setmpsl
cummps2 = cummps2 + setmps2
IF i.fail = 0 THEN ELSE WRITE #3, u.fail, i.fail, "times"
END IF
PRINT ct1; "of"; cyc; "type 1 cycles correct"
PRINT
PRINT hil; "of"; cyc; "highest 2 conc. correctly ranked for Type 1"
PRINT
PRINT ct2; "of"; cyc; "type 2 cycles correct"
PRINT
PRINT hi2; "of"; cyc; "highest 2 conc. correctly ranked for Type 2"
PRINT
avgmpsl = setmpsl / eye
avgmps2 = setmps2 / eye
REM Calculating the variance of the 100 misclassifications, P screen, per dataset
k = runs - 99
l = runs
eps1 = 0: eps2 = 0: vsumps1 = 0
vsumps2 = 0: vps1 = 0: vps2 = 0
FOR i = k TO l
eps1 = (mps(i, 1) - avgmpsl)^2
eps2 = (mps(i, 2) - avgmps2)^2
REM
\[ vsumps1 = vsumps1 + \text{eps}1 \]
\[ vsumps2 = vsumps2 + \text{eps}2 \]
\[ \text{NEXT } i \]
\[ vpsl = vsumpsl / (\text{eye} - 1) \]
\[ vps2 = vsumps2 / (\text{eye} - 1) \]

\begin{verbatim}
WRITE #3, 
FOR ii = 1 TO 6 
FOR i = 1 TO 2 
LRaverank(ii, i) = LRcumrank(ii, i) / cyc 
\text{NEXT } i 
WRITE #2, "task", ii, "average rank - type 1", LRaverank(ii, 1), "average rank - type 2", LRaverank(ii, 2) 
PRINT "task"; ii; "average rank - type 1"; LRaverank(ii, 1); "average rank - type 2"; LRaverank(ii, 2) 
\text{NEXT } ii 
WRITE #2, LRctl, "of", eye, "Type 1 cycles correct" 
WRITE #2, LRct2, "of", eye, "Type 2 cycles correct" 
WRITE #2, LRhil, "of", eye, "highest 2 cone, correctly ranked for Type 1" 
WRITE #2, LRhi2, "of", eye, "highest 2 cone, correctly ranked for Type 2" 
LRcumhil = LRcumhil + LRhil 
LRcumhi2 = LRcumhi2 + LRhi2 
LRcumctl = LRcumctl + LRctl 
LRcumct2 = LRcumct2 + LRct2 
cummlrl1 = cummlrl1 + setmrl1 
cummlrl2 = cummlrl2 + setmrl2 
\end{verbatim}

IF i.fail = 0 THEN 
ELSE 
WRITE #2, u.fail, i.fail, "times" 
END IF 
PRINT LRctl; "of"; eye; "type 1 cycles correct" 
PRINT 
PRINT LRhil; "of"; eye; "highest 2 cone, correctly ranked for Type 1" 
PRINT 
PRINT LRct2; "of"; eye; "type 2 cycles correct" 
PRINT 
PRINT LRhi2; "of"; eye; "highest 2 cone, correctly ranked for Type 2" 
PRINT 
avgmlrl1 = setmrl1 / cyc 

54
avgmlr2 = setmlr2 / cyc

REM Calculating the variance of the 100 misclassifications LR, per dataset

\[ \text{y} = \text{runs} - 99 \]
\[ \text{z} = \text{runs} \]
\[ \text{emr1} = 0; \text{emr2} = 0; \text{vsummr1} = 0 \]
\[ \text{vsummr2} = 0; \text{vmr1} = 0; \text{vmr2} = 0 \]

FOR i = y TO z
    \[ \text{emr1} = (\text{mlr}(i, 1) - \text{avgmlr1})^2 \]
    \[ \text{emr2} = (\text{mlr}(i, 2) - \text{avgmlr2})^2 \]
    \[ \text{vsummr1} = \text{vsummr1} + \text{emr1} \]
    \[ \text{vsummr2} = \text{vsummr2} + \text{emr2} \]
NEXT i

\[ \text{vmr1} = \text{vsummr1} / (\text{eye} - 1) \]
\[ \text{vmr2} = \text{vsummr2} / (\text{eye} - 1) \]

WRITE #4, dataset, avgmlr1, vmr1, avgmlr2, vmr2, avgmps1, vps1, avgmps2, vps2
WRITE #2,

NEXT i.set

WRITE #3, "RESULTS OF SCREENED TIME MATRIX INVERSION METHOD", \ LRA(1, 1), "SAMPLES"
WRITE #3,
WRITE #3, "Assumes zero autocorrelation"
WRITE #3, "GSD of underlying task distribution for task time weight 1/6 differed from nominal value by +/-", sigdev
WRITE #3,
WRITE #3, "Estimated versus actual rank for", dataset, "datasets with", cyc, "cycles per dataset"
WRITE #3, "Total of", runs, "runs"
WRITE #3, "Worker effect: multiply median concentration by", \ EXP(workereffect(1)), EXP(workereffect(2)), EXP(workereffect(3)), \ EXP(workereffect(4)), EXP(workereffect(5))
WRITE #3,
GSD(1) = sig1: GSD(2) = sig2
FOR i = 1 TO 6
    WRITE #3, "Type", i, " - nominal GSD = ", GSD(i), "+/-", sigdev
FOR i = 1 TO 6
    FOR ii = 1 TO 6
        estvact(i, ii, iii) = estvact(i, ii, iii) / runs

55
WRITE #3, "true rank =", i, estvact(i, 1, iii), estvact(i, 2, iii), estvact(i, 3, iii), estvact(i, 4, iii), estvact(i, 5, iii), estvact(i, 6, iii)
NEXT i
WRITE #3,
NEXT iii
WRITE #3, cumhi1, "of", runs, "highest 2 tasks correctly ranked for Type 1"
WRITE #3, cumhi2, "of", runs, "highest 2 tasks correctly ranked for Type 2"
WRITE #3, cumct1, "of", runs, "Type 1 runs ranked correctly"
WRITE #3, cumct2, "of", runs, "Type 2 runs ranked correctly"
totavgmpl = cummpsl / runs
totavgmpl2 = cummpsl2 / runs
WRITE #3, "Avg misclass for P-screen Type 1 -", totavgmpl, "Type 2 -",
totavgmpl2

CLOSE #3

WRITE #2, "RESULTS OF MULTIPLE LINEAR REGRESSION METHOD",
LRA(1, 1), "SAMPLES"
WRITE #2,
WRITE #2, "Assumes zero autocorrelation"
WRITE #2, "GSD of underlying task distribution for task time weight 1/6 differed
from nominal value by +/-", sigdev
WRITE #2,
WRITE #2, "Estimated versus actual rank for", dataset, "datasets with", eye, "cycles
per dataset"
WRITE #2, "Total of", runs, "runs"
WRITE #2, "Worker effect: multiply median concentration by",
EXP(workereffect(1)), EXP(workereffect(2)), EXP(workereffect(3)),
EXP(workereffect(4)), EXP(workereffect(5))
WRITE #2,
GSD(1) = sig1: GSD(2) = sig2
FOR iii = 1 TO 2
WRITE #2, "Type", iii, " - nominal GSD = ", GSD(iii), "+/-", sigdev
FOR i = 1 TO 6
FOR ii = 1 TO 6
LRestvact(i, ii, iii) = LRestvact(i, ii, iii) / runs
NEXT ii
WRITE #2, "true rank =", i, LRestvact(i, 1, iii), LRestvact(i, 2, iii), LRestvact(i, 3, iii), LRestvact(i, 4, iii), LRestvact(i, 5, iii), LRestvact(i, 6, iii)

56
REM Check here for 6 OR 7 SIZE ESTVACT
NEXT i
WRITE #2,
NEXT iii
WRITE #2, LRcumhi1, "of", runs, "highest 2 tasks correctly ranked for Type 1"
WRITE #2, LRcumhi2, "of", runs, "highest 2 tasks correctly ranked for Type 2"
WRITE #2, LRcumct1, "of", runs, "Type 1 runs ranked correctly"
WRITE #2, LRcumct2, "of", runs, "Type 2 runs ranked correctly"
totavgmlrl = cummlrl / runs
totavgmlr2 = cummlr2 / runs
WRITE #2, "Avg misclass for LR method Type 1 -", totavgmlrl, "Type 2 -", totavgmlr2
CLOSE #2

END

Matsolv:
REM Solution of simultaneous equations
REM THE MAIN SUBROUTINE
REM Definitions

ERASE h
REDIM h(6, 7)
ERASE i.from
REDIM i.from(6)
ERASE i.to
REDIM i.to(6)
ERASE b
REDIM b(6, 7)
ERASE e
REDIM e(6)
ERASE x
REDIM x(6)
ERASE xx
REDIM xx(6)
kr = 6
b.max = 0
FOR i = 1 TO kr
FOR J = 1 TO kr + 1
b(i, J) = a(i, J)
NEXT J
IF ABS(b(i, kr + 1)) > b.max THEN
b.max = ABS(b(i, kr + 1))
END IF
NEXT i
GOSUB Analyse:
CLS
RETURN
REM
REM the main subroutine ends here
REM
condition:
LOCATE 10, 1
COLOR 11
PRINT
"+-*|=*|=*|=|^@...|^@...|^@...|=*|=|^@...|^@...|^@...|=*|=|^@...|^@...
" COLOR 12
LOCATE 15, 7
PRINT "ESTIMATING THE CONDITION NUMBER OF THE COEFFICIENT MATRIX"
LOCATE 16, 22
PRINT "THIS MAY TAKE A SHORT WHILE."
LOCATE 21, 1
PRINT
"+-*|=*|=*|=|^@...|^@...|^@...|^@...|^@...|^@...|^@...|^@...|^@...|^@...|^@...|^@...
" r.min = 1E+20
FOR i = 1 TO kr - 1
FOR J = i + 1 TO kr
r.sum = 0
FOR k = 1 TO kr
r.sum = r.sum + ABS(a(i, k) - a(J, k))
NEXT k
IF r.sum < r.min THEN
r.min = r.sum
END IF
IF r.sum = 1E-10 THEN
u.type = "Singular"
"
J = kr
i = kr - 1
END IF
NEXT J
NEXT i
IF u.type <> "Singular" THEN
  c.min = 1E+20
  FOR i = 1 TO kr - 1
  FOR J = i + 1 TO kr
    c.sum = 0
    FOR k = 1 TO kr
      c.sum = c.sum + ABS(a(k, i) - a(k, J))
    NEXT k
    IF c.sum < c.min THEN
      c.min = c.sum
    END IF
  NEXT i
END IF
IF u.type <> "Singular" THEN
  r.max = 0
  FOR i = 1 TO kr
    r.sum = 0
    FOR k = 1 TO kr
      r.sum = r.sum + ABS(a(i, k))
    NEXT k
    IF r.sum > r.max THEN
      r.max = r.sum
    END IF
  NEXT i
END IF
IF u.type <> "Singular" THEN
  c.max = 0
  FOR i = 1 TO kr
    c.sum = 0
    FOR k = 1 TO kr
      c.sum = c.sum + ABS(a(k, i))
    NEXT k
    IF c.sum > c.max THEN
      c.max = c.sum
    END IF
  NEXT i
END IF
FOR k = 1 TO kr
    c.sum = c.sum + ABS(a(k, i))
NEXT k
IF c.sum > c.max THEN
    c.max = c.sum
END IF
NEXT i
END IF
IF u.type <> "Singular" THEN
    c1 = r.max / r.min
    c2 = c.max / c.min
    IF c1 > c2 THEN
        condition.a = c1
    ELSE
        condition.a = c2
    END IF
END IF
RETURN:
Analyse:
REM the structured analysis program
PRINT "b.max"; b.max
GOSUB Error.crit:
GOSUB Pivot:
IF u.type <> "Singular" THEN
    GOSUB condition:
END IF
IF u.type <> "Singular" THEN
    GOSUB Crout:
ELSE
    PRINT "Singular System"
END IF
IF u.type = "Singular" OR u.type = "Solution" THEN
ELSE
    PRINT "LU Decomposition fails the error test - Trying Gauss-Seidel"
    GOSUB Gauss:
END IF
IF u.type = "Singular" OR u.type = "Solution" THEN
ELSE
PRINT "Gauss-Seidel iteration fails the error test - Trying LU iteration"
GOSUB Iterate:
END IF

IF u.type = "Solution" OR u.type = "Best solution" THEN
GOSUB Results:
END IF

RETURN

Pivot:
REM This is pivoting and equilisation programme
FOR in.dex = 1 TO kr
IF in.dex < kr THEN
a.max = 0
FOR i = in.dex TO kr
FOR J = in.dex TO kr
IF ABS(a(i, J)) > a.max THEN
a.max = ABS(a(i, J))
i.m = i: j.m = J
END IF
NEXT J
NEXT i
REM Exchange rows
FOR i = 1 TO kr + 1
SWAP a(in.dex, i), a(i.m, i)
NEXT i
i.to(in.dex) = in.dex
i.from(in.dex) = j.m
FOR J = 1 TO kr
SWAP a(J, in.dex), a(J, j.m)
NEXT J
END IF
REM equilise
a.max = 0
FOR i = 1 TO kr
IF ABS(a(in.dex, i)) > a.max THEN
a.max = ABS(a(in.dex, i))
END IF
NEXT i
IF a.max < 1E-22 THEN
u.type = "Singular"
PRINT "Failed as singular"
PRINT "in.dex"
in.dex = kr
ELSE
FOR i = 1 TO kr + 1
a(in.dex, i) = a(in.dex, i) / a.max
NEXT i
END IF
NEXT in.dex
RETURN

Error.crit:
REM error criteria
e.error = 1.45E-06
RETURN

Matrix:
FOR i = 1 TO kr
FOR J = 1 TO kr + 1
IF i = J THEN
ic = i - 1: pf = 1
ELSEIF i < J THEN
ic = i - 1: pf = h(i, i)
ELSE
ic = J - 1: pf = 1
END IF
pu = 0
FOR k = 1 TO ic
pu = pu + h(i, k) * h(k, J)
NEXT k
h(i, J) = (a(i, J) - pu) / pf
IF i = J AND ABS(h(i, i)) < 1E-22 THEN
PRINT " Singular"
u.type = "Singular"
i = kr
J = kr + 1
END IF
NEXT J, i
RETURN

Crout:
REM LU decomposition - One pass
   u.method = "L/U Decomposition"
GOSUB Matrix:
IF u.type <> "Singular" THEN
   x(kr) = h(kr, kr + 1)
   FOR i = kr - 1 TO 1 STEP -1
      p0 = 0
      FOR J = i TO kr - 1
         p0 = p0 + h(i, J + 1) * x(J + 1)
      NEXT J
      x(i) = h(i, kr + 1) - p0
   NEXT i
END IF
GOSUB res.check:
   average.error = e.sum / kr
   rms.error = SQR(e.ssq / kr)
   error.maximum = e.max
   IF e.max < e.error THEN
      u.type = "Solution"
   ELSE
      GOSUB Failed:
   END IF
REM related to singular end if
   END IF
RETURN
Failed:
REM prints the failure
   CLS
   i.fail = i.fail + 1
   u.fail = "Results failed to meet the error criterion"
   WRITE #3, "Maximum residual =", error.maximum
RETURN
res.check:
REM Check the results and print error values
REM swap back the solution vector
   e.max = 0
   FOR i = kr - 1 TO 1 STEP -1
      SWAP x(i.to(i)), x(i.from(i))
   NEXT i
REM check results with the original matrix
e.sum = 0: e.ssq = 0
FOR i = 1 TO kr
  e(i) = -b(i, kr + 1)
  FOR j = 1 TO kr
    e(i) = e(i) + b(i, j) * x(j)
  NEXT j
  IF ABS(e(i)) > e.max THEN
    e.max = ABS(e(i))
  END IF
  e.sum = e.sum + e(i)
  e.ssq = e.ssq + e(i) * e(i)
NEXT i
Rel.errmax = condition.a * e.max / b.max
Rel.errmin = e.max / b.max / condition.a
RETURN
Swapback:
REM swap back
FOR i = 1 TO kr - 1
  SWAP x(i.to(i)), x(i.from(i))
NEXT i
RETURN
Gauss:
REM Gauss Seidel iteration
u.method = "Gauss-Seidel Iteration"
GOSUB Swapback
e.init = e.max
i.sei = 0
FOR i = 1 TO kr
  xx(i) = x(i)
  NEXT i
u.dur = "git"
DO
  i.sei = i.sei + 1
  FOR i = 1 TO kr
    aso = a(i, kr + 1)
    FOR j = 1 TO kr
      IF i <> j THEN
        aso = aso - a(i, j) * x(j)
      END IF
    NEXT j
  END IF
  NEXT i
x.new = aso / a(i, i)
x(i) = x.new
NEXT i
GOSUB res.check:
IF e.max < e.rror THEN
  u.dur = "dur"
ELSEIF e.max > e.init THEN
  u.dur = "dur"
GOSUB Swapback:
FOR i = 1 TO kr
  x(i) = xx(i)
NEXT i
GOSUB res.check:
ELSE
  e.init = e.max
GOSUB Swapback:
FOR i = 1 TO kr
  xx(i) = x(i)
NEXT i
  u.dur = "git"
END IF
IF i.sei > 2 * kr THEN
  u.dur = "dur"
END IF
LOOP UNTIL u.dur = "dur"
average.error = e.sum / kr
rms.error = SQR(e.ssq / kr)
error.maximum = e.max
IF e.max < e.rror THEN
  u.type = "Solution"
ELSE
  GOSUB Failed:
END IF
RETURN
Iterate:
REM Lu decomposition with iteration
  u.method = "L/U Iteration"
GOSUB recalc:
  u.lue = "go"
e.old = e.max
FOR i = 1 TO kr
xx(i) = x(i)
NEXT i
DO
ed.max = 0
x.max = 0
GOSUB Matrix:
IF u.type <> "Singular" THEN
e(kr) = h(kr, kr + 1)
FOR i = kr - 1 TO 1 STEP -1
pi0 = 0
FOR J = i TO kr - 1
pi0 = pi0 + h(i, J + 1) * e(J + 1)
NEXT J
e(i) = h(i, kr + 1) - pi0
NEXT i
ELSE
FOR i = 1 TO kr
e(i) = .1
NEXT i
END IF
FOR i = 1 TO kr
x(i) = x(i) + e(i)
IF ABS(e(i)) > ed.max THEN
  ed.max = ABS(e(i))
END IF
IF ABS(xx(i)) > x.max THEN
  x.max = ABS(xx(i))
END IF
NEXT i
PRINT e.max
IF (ed.max / x.max) > .5 THEN
PRINT "The system is practically singular within the working accuracy"
PRINT "Failed as type 2 singular"
FOR i = 1 TO kr
x(i) = xx(i)
NEXT i
GOSUB res.check:
u.lue = "stop"
ELSE
GOSUB res.check:
IF e.max < e.error THEN
  u.lue = "stop"
ELSEIF e.max >= e.old THEN
  u.lue = "stop"
END IF
END IF
LOOP UNTIL u.lue = "stop"

average.error = e.sum / kr
rms.error = SQR(e.ssq / kr)
error.maximum = e.max
IF e.max < e.error THEN
  u.type = "Solution"
ELSE
  u.type = "Best solution"
END IF
RETURN

recalc:
REM swap back
  FOR i = 1 TO kr - 1
    SWAP x(i.to(i)), x(i.from(i))
  NEXT i
REM recalculates the constant vector for L/U iteration and overwrites
  FOR i = 1 TO kr
    FOR J = 1 TO kr
      a(i, kr + 1) = a(i, kr + 1) - a(i, J) * x(J)
    NEXT J
  NEXT i
RETURN

Results:

CLS
LOCATE 2, 5

IF u.type = "Solution" THEN
COLOR 10
LOCATE 10, 10
PRINT "*************** SOLUTION IS OBTAINED *****************
COLOR 15
LOCATE 5, 2
PRINT ""
ELSE
COLOR 12
LOCATE 6, 10
PRINT "THE BEST AVAILABLE SOLUTION WITHIN THE ERROR
SPECIFIED"
COLOR 14
PRINT " The solution can only be improved by using ";
PRINT "a solution program with 
COLOR 13
PRINT " DOUBLE PRECISION ";
COLOR 14
PRINT "arithmetic 
PRINT ""
COLOR 12
LOCATE 10, 10
PRINT "*************** CURRENT SOLUTION ******************
END IF
GOSUB Show:

RETURN

Show:
REM prints results on the screen
CLS
LOCATE 5, 12
PRINT "Method of Calculation : ";
COLOR 11
PRINT u.method
WRITE #3, "calculated by", u.method
RESULTS (;
IF u.type = "Solution" THEN
PRINT " Solution shown is within the specified error :) ";
ELSE
PRINT " Results shown failed to meet the error criterion :) ";
END IF
klef = kr MOD 3
klim = kr - klef
IF klim > 2 THEN
FOR i = 1 TO klim
PRINT " X("; i; ") = ";
PRINT USING "##.######AAAAM"; x(i);
IF (i MOD 3) = 0 THEN
PRINT""
END IF
NEXT i
ELSE
klim = 0
END IF
IF klef = 0 THEN
ELSE
FOR i = klim + 1 TO kr
PRINT " X("; i; ") = ";
PRINT USING "##.######AAAA"; x(i);
NEXT i
END IF
PRINT ""
PRINT
"+++++++++++++++++++++++++++++++++++++++++++++++++++
Error analysis :
PRINT " Maximum residual Calculated : ";
PRINT USING "##.#####^\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
PRINT USING "##.#######"; condition.a
PRINT ""
PRINT USING "##.#######"; Rel.errmin;
PRINT " .LE. Relative Error .LE. ";
PRINT USING "##.#######"; Rel.errmax
PRINT ""
FOR i = 1 TO kr
PRINT "Residual of equation "; i; " = ";
PRINT USING "##.#######"; e(i)
NEXT i
CLS
RETURN

Matsolv1:
REM Solution of simultaneous equations for Mult Linear Regression
REM THE MAIN SUBROUTINE
REM Definitions

ERASE h
REDIM h(7, 8)
ERASE i.from
REDIM i.from(7)
ERASE i.to
REDIM i.to(7)
ERASE b
REDIM b(7, 8)
ERASE e
REDIM e(7)
ERASE LRx
REM   REDIM LRx(7)
ERASE xx
REDIM xx(7)
kr = 7
b.max = 0
FOR i = 1 TO kr
  FOR j = 1 TO kr + 1
    b(i, j) = LRAB(i, j)
  NEXT j
IF ABS(b(i, kr + 1)) > b.max THEN
b.max = ABS(b(i, kr + 1))
END IF
NEXT i
GOSUB LRAnalyse:
CLS
RETURN
REM
REM the main subroutine ends here
REM
LRcondition:

LOCATE 10, 1
COLOR 11
PRINT
"+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...
COLOR 12
LOCATE 15, 7
PRINT "ESTIMATING THE CONDITION NUMBER OF THE COEFFICIENT MATRIX"
LOCATE 16, 22
PRINT "THIS MAY TAKE A SHORT WHILE."
LOCATE 21, 1
PRINT
"+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...+-*@...
r.min = 1E+20
FOR i = 1 TO kr - 1
FOR J = i + 1 TO kr
r.sum = 0
FOR k = 1 TO kr
r.sum = r.sum + ABS(LRAB(i, k) - LRAB(J, k))
NEXT k
IF r.sum < r.min THEN
r.min = r.sum
END IF
IF r.sum = 1E-10 THEN
u.type = "Singular"
J = kr
i = kr - 1
END IF
NEXT J
NEXT i
IF u.type <> "Singular" THEN
c.min = 1E+20
FOR i = 1 TO kr - 1
FOR J = i + 1 TO kr
  c.sum = 0
  FOR k = 1 TO kr
   c.sum = c.sum + ABS(LRAB(k, i) - LRAB(k, J))
  NEXT k
  IF c.sum < c.min THEN
    c.min = c.sum
  END IF
END IF
IF c.sum = 1E-10 THEN
  u.type = "Singular"
  J = kr
  i = kr - 1
END IF
NEXT J
NEXT i
END IF
IF u.type <> "Singular" THEN
  r.max = 0
  FOR i = 1 TO kr
   r.sum = 0
   FOR k = 1 TO kr
    r.sum = r.sum + ABS(LRAB(i, k))
   NEXT k
   IF r.sum > r.max THEN
    r.max = r.sum
   END IF
  NEXT i
END IF
IF u.type <> "Singular" THEN
  c.max = 0
  FOR i = 1 TO kr
   c.sum = 0
   FOR k = 1 TO kr
    c.sum = c.sum + ABS(LRAB(k, i))
   NEXT k
   IF c.sum > c.max THEN
c.max = c.sum
END IF
NEXT i
END IF
IF u.type <> "Singular" THEN
    c1 = r.max / r.min
    c2 = c.max / c.min
    IF c1 > c2 THEN
        condition.a = c1
    ELSE
        condition.a = c2
    END IF
END IF
RETURN:
LRAnalyse:
REM the structured analysis program
PRINT "b.max"; b.max
GOSUB LRError.crit:
GOSUB LRPivot:
IF u.type <> "Singular" THEN
    GOSUB LRcondition:
END IF
IF u.type <> "Singular" THEN
    GOSUB LRCrout:
ELSE
    PRINT "Singular System"
END IF
IF u.type = "Singular" OR u.type = "Solution" THEN
ELSE
    PRINT "LU Decomposition fails the error test - Trying Gauss-Seidel"
    GOSUB LRGauss:
END IF
IF u.type = "Singular" OR u.type = "Solution" THEN
ELSE
    PRINT "Gauss-Seidel iteration fails the error test - Trying LU iteration"
    GOSUB LRiterate:
END IF
IF u.type = "Solution" OR u.type = "Best solution" THEN
GOSUB LRResults:
END IF

RETURN

LRPivot:
REM This is pivoting and equilisation programme
    FOR in.dex = 1 TO kr
    IF in.dex < kr THEN
        a.max = 0
    FOR i = in.dex TO kr
        FOR J = in.dex TO kr
            IF ABS(LRAB(i, J)) > a.max THEN
                a.max = ABS(LRAB(i, J))
                i.m = i: j.m = J
            END IF
        NEXT J
    NEXT i
    REM Exchange rows
    FOR i = 1 TO kr + 1
        SWAP LRAB(in.dex, i), LRAB(i.m, i)
    NEXT i
    i.to(in.dex) = in.dex
    i.from(in.dex) = j.m
    FOR J = 1 TO kr
        SWAP LRAB(J, in.dex), LRAB(J, j.m)
    NEXT J
     END IF
    REM equilise
    a.max = 0
    FOR i = 1 TO kr
        IF ABS(LRAB(in.dex, i)) > a.max THEN
            a.max = ABS(LRAB(in.dex, i))
        END IF
    NEXT i
    IF a.max < 1E-22 THEN
        u.type = "Singular"
        PRINT "Failed as singular"
        PRINT "in.dex"
        in.dex = kr
    END IF
ELSE
FOR i = 1 TO kr + 1
LRAB(in.dex, i) = LRAB(in.dex, i) / a.max
NEXT i
END IF
NEXT in.dex
RETURN
LRError.crit:
REM error criteria
c.error = 1.45E-06
RETURN
LRMatrix:
FOR i = 1 TO kr
FOR J = 1 TO kr + 1
IF i = J THEN
ic = i - 1: pf = 1
ELSEIF i < J THEN
ic = i - 1: pf = h(i, i)
ELSE
ic = J - 1: pf = 1
END IF
pu = 0
FOR k = 1 TO ic
pu = pu + h(i, k) * h(k, J)
NEXT k
h(i, J) = (LRAB(i, J) - pu) / pf
IF i = J AND ABS(h(i, i)) < 1E-22 THEN
PRINT " Singular"
i.type = "Singular"
i = kr
J = kr + 1
END IF
NEXT J, i
RETURN
LRCrout:
REM LU decomposition - One pass
u.method = "L/U Decomposition"
GOSUB LRMatrix:
IF u.type <> "Singular" THEN
LRx(kr) = h(kr, kr + 1)
FOR $i = kr - 1$ TO 1 STEP -1
  $p_0 = 0$
  FOR $J = i$ TO $kr - 1$
    $p_0 = p_0 + h(i, J + 1) \times LRx(J + 1)$
  NEXT $J$
  $LRx(i) = h(i, kr + 1) - p_0$
NEXT $i$
GOSUB LRres.check:
  average.error = e.sum / kr
  rms.error = SQR(e.ssq / kr)
  error.maximum = e.max
  IF e.max < e.error THEN
    u.type = "Solution"
  ELSE
    GOSUB LRFailed:
  END IF
REM related to singular end if
END IF
RETURN

LRFailed:
REM prints the failure
  CLS
  i.fail = i.fail + 1
  u.fail = "Results failed to meet the error criterion"
  WRITE #2, "Maximum residual =", error.maximum
RETURN

LRres.check:
REM Check the results and print error values
REM swap back the solution vector
  e.max = 0
  FOR $i = kr - 1$ TO 1 STEP -1
    SWAP LRx(i.to(i)), LRx(i.from(i))
  NEXT $i$
REM check results with the original matrix
  e.sum = 0; e.ssq = 0
  FOR $i = 1$ TO $kr$
    $e(i) = b(i, kr + 1)$
  FOR $J = 1$ TO $kr$
    $e(i) = e(i) + b(i, J) \times LRx(J)$
NEXT J
IF ABS(e(i)) > e.max THEN
  e.max = ABS(e(i))
END IF
e.sum = e.sum + e(i)
e.ssq = e.ssq + e(i) * e(i)
NEXT i
Rel.errmax = condition.a * e.max / b.max
Rel.errmin = e.max / b.max / condition.a
RETURN
LRSwapback:
REM swap back
  FOR i = 1 TO kr - 1
    SWAP LRx(i.to(i)), LRx(i.from(i))
  NEXT i
RETURN
LRGauss:
REM Gauss Seidel iteration
  u.method = "Gauss-Seidel Iteration"
  GOSUB LRSwapback
  e.init = e.max
  i.sei = 0
  FOR i = 1 TO kr
    xx(i) = LRx(i)
  NEXT i
  u.dur = "git"
  DO
    i.sei = i.sei + 1
    FOR i = 1 TO kr
      aso = LRAB(i, kr + 1)
      FOR J = 1 TO kr
        IF i <> J THEN
          aso = aso - LRAB(i, J) * LRx(J)
        END IF
      NEXT J
      x.new = aso / LRAB(i, i)
      LRx(i) = x.new
    NEXT i
    GOSUB LRres.check:
    IF e.max < error THEN
u.dur = "dur"
ELSEIF e.max > e.init THEN
u.dur = "dur"
GOSUB LRSwapback:
FOR i = 1 TO kr
LRx(i) = xx(i)
NEXT i
GOSUB LRres.check:
ELSE
  e.init = e.max
GOSUB LRSwapback:
FOR i = 1 TO kr
xx(i) = LRx(i)
NEXT i
u.dur = "git"
END IF
IF i.sei > 2 * kr THEN
u.dur = "dur"
END IF
LOOP UNTIL u.dur = "dur"
average.error = e.sum / kr
rms.error = SQR(e.ssq / kr)
error.maximum = e.max
IF e.max < e.rror THEN
  u.type = "Solution"
ELSE
GOSUB LRFailed:
END IF
RETURN
LRIterate:
REM Lu decomposition with iteration
  u.method = "L/U Iteration"
GOSUB LRrecalc:
  u.lue = "go"
e.old = e.max
FOR i = 1 TO kr
xx(i) = LRx(i)
NEXT i
DO
ed.max = 0
x.max = 0
GOSUB LRMatrix:
IF u.type <> "Singular" THEN
  e(kr) = h(kr, kr + 1)
  FOR i = kr - 1 TO 1 STEP -1
    pi0 = 0
    FOR J = i TO kr - 1
      pi0 = pi0 + h(i, J + 1) * e(J + 1)
    NEXT J
    e(i) = h(i, kr + 1) - pi0
  NEXT i
ELSE
  FOR i = 1 TO kr
    e(i) = .1
  NEXT i
END IF
FOR i = 1 TO kr
  LRx(i) = LRx(i) + e(i)
  IF ABS(e(i)) > ed.max THEN
    ed.max = ABS(e(i))
  END IF
  IF ABS(xx(i)) > x.max THEN
    x.max = ABS(xx(i))
  END IF
NEXT i
PRINT e.max
IF (ed.max / x.max) > .5 THEN
  PRINT "The system is practically singular within the working accuracy"
  PRINT "Failed as type 2 singular"
  FOR i = 1 TO kr
    LRx(i) = xx(i)
  NEXT i
  GOSUB LRres.check:
  u.lue = "stop"
ELSE
  GOSUB LRres.check:
  IF e.max < e.error THEN
    u.lue = "stop"
  ELSEIF e.max >= e.old THEN
    u.lue = "stop"
FOR i = 1 TO kr
LRx(i) = xx(i)
NEXT i
GOSUB LRres.check:
ELSE
GOSUB LRrecalc:
FOR i = 1 TO kr
xx(i) = LRx(i)
NEXT i
u.lue = "go"
END IF
END IF
LOOP UNTIL u.lue = "stop"

average.error = e.sum / kr
rms.error = SQR(e.ssq / kr)
error.maximum = e.max
IF e.max < error THEN
u.type = "Solution"
ELSE
u.type = "Best solution"
END IF

RETURN
LRrecalc:
REM swap back
FOR i = 1 TO kr - 1
SWAP LRx(i.to(i)), LRx(i.from(i))
NEXT i
REM recalculates the constant vector for L/U iteration and overwrites
FOR i = 1 TO kr
FOR J = 1 TO kr
LRAB(i, kr + 1) = LRAB(i, kr + 1) - LRAB(i, J) * LRx(J)
NEXT J
NEXT i
RETURN
LRResults:
CLS
LOCATE 2, 5
IF u.type = "Solution" THEN
COLOR 10
LOCATE 10, 10
PRINT "**************** SOLUTION IS OBTAINED ****************"
COLOR 15
LOCATE 5, 2
PRINT ""
ELSE
COLOR 12
LOCATE 6, 10
PRINT "THE BEST AVAILABLE SOLUTION WITHIN THE ERROR SPECIFIED"
COLOR 14
PRINT " The solution can only be improved by using ";
PRINT "a solution program with 
COLOR 13
PRINT " DOUBLE PRECISION ";
COLOR 14
PRINT "arithmetic 
PRINT ""
COLOR 12
LOCATE 10, 10
PRINT "**************** CURRENT SOLUTION ****************"
END IF
GOSUB LRShow:
RETURN
LRShow:
REM prints results on the screen
CLS
LOCATE 5, 12
PRINT "Method of Calculation : ";
COLOR 11
PRINT u.method
WRITE #2, "calculated by", u.method
COLOR 15
LOCATE 7, 5
PRINT " RESULTS ( ";
IF u.type = "Solution" THEN
PRINT " Solution shown is within the specified error ) :"
81
ELSE
PRINT "Results shown failed to meet the error criterion."
END IF
klef = kr MOD 3
klim = kr - klef
IF klim > 2 THEN
FOR i = 1 TO klim
PRINT "X("; i; ") = ";
PRINT USING "##.######AAAAA"; LRx(i);
IF (i MOD 3) = 0 THEN
PRINT ""
END IF
NEXT i
ELSE
klim = 0
END IF
IF klef = 0 THEN
ELSE
FOR i = klim + 1 TO kr
PRINT "X("; i; ") = ";
PRINT USING "##.######AAAAA"; LRx(i);
NEXT i
END IF
PRINT ""
PRINT "Error analysis :"
PRINT "Maximum residual Calculated : ";
PRINT USING "##.######AAAAA"; error.maximum
PRINT "Average Residual Calculated : ";
PRINT USING "##.######AAAAA"; average.error
PRINT "Root mean square Residual Calculated : ";
PRINT USING "##.######AAAAA"; rms.error
PRINT "Coefficient Matrix Condition number estimate : ";
PRINT USING "##.##AAAAA"; condition.a
PRINT ""
PRINT USING "##.##AAAAA"; Rel.errmin;
PRINT " .LE. Relative Error .LE. ";
PRINT USING "##.##AAAAA"; Rel.errmax
PRINT ""
FOR i = 1 TO kr
PRINT "Residual of equation "; i; " = ";
PRINT USING "##.######AAAA"; e(i)
NEXT i
CLS
RETURN

tasktime:
REM Generate task durations
REM Generate task assignment matrix
100 ERASE ran
ERASE weight
ERASE ord
ERASE time
FOR ii = 1 TO 100
RANDOMIZE (VAL(RIGHT$(TIME$, 2)))
FOR i = 1 TO 6
ran(ii, i) = RND
NEXT i
NEXT ii
FOR ii = 1 TO 100
FOR i = 1 TO 6
s = 1
FOR J = 1 TO 6
IF ran(ii, i) > ran(ii, J) THEN
s = s + 1
END IF
NEXT J
ord(ii, i) = s
NEXT i
NEXT ii
REM Generate times matrix
FOR ii = 1 TO 100
J = 96
FOR i = 1 TO 4
jj = INT(J * RND)
time(ii, i) = jj
\[ \text{J} = \text{J} - \text{jj} \]
\[ \text{NEXT} \ i \]
\[ \text{time}(\text{ii}, 5) = 96 - \text{time}(\text{ii}, 1) - \text{time}(\text{ii}, 2) - \text{time}(\text{ii}, 3) - \text{time}(\text{ii}, 4) \]
\[ \text{NEXT ii} \]

REM Assign times to tasks
\[ \text{FOR ii} = 1 \ TO \ 100 \]
\[ \text{FOR} \ i = 1 \ TO \ 6 \]
\[ \text{J} = \text{ord}(\text{ii}, i) \]
\[ \text{weight}(\text{ii}, J) = \text{time}(\text{ii}, i) \]
\[ \text{NEXT ii} \]
\[ \text{NEXT i} \]

REM Check for samples with one task excluded
\[ \text{si} = 0; \text{s2} = 0; \text{s3} = 0; \text{s4} = 0; \text{s5} = 0; \text{s6} = 0 \]
\[ \text{FOR ii} = 1 \ TO \ 100 \]
\[ \text{ss} = 0 \]
\[ \text{FOR} \ i = 1 \ TO \ 6 \]
\[ \text{IF} \ \text{weight}(\text{ii}, i) > 0 \ \text{THEN} \]
\[ \text{ss} = \text{ss} + 1 \]
\[ \text{END IF} \]
\[ \text{NEXT i} \]
\[ \text{IF} \ \text{ss} = 5 \ \text{THEN} \]
\[ \text{GOSUB task:} \]
\[ \text{END IF} \]
\[ \text{NEXT ii} \]
\[ \text{IF} \ \text{s1} * \text{s2} * \text{s3} * \text{s4} * \text{s5} * \text{s6} = 0 \ \text{THEN} \]
\[ \text{GOTO 100} \]
\[ \text{END IF} \]

RETURN

\[ \text{task:} \]
\[ \text{IF} \ \text{weight}(\text{ii}, 1) = 0 \ \text{THEN} \]
\[ \text{s1} = \text{s1} + 1 \]
\[ \text{ELSEIF} \ \text{weight}(\text{ii}, 2) = 0 \ \text{THEN} \]
\[ \text{s2} = \text{s2} + 1 \]
\[ \text{ELSEIF} \ \text{weight}(\text{ii}, 3) = 0 \ \text{THEN} \]
s3 = s3 + 1

ELSEIF weight(ii, 4) = 0 THEN
s4 = s4 + 1

ELSEIF weight(ii, 5) = 0 THEN
s5 = s5 + 1

ELSEIF weight(ii, 6) = 0 THEN
s6 = s6 + 1

END IF
RETURN

taskvar:
REM Creates random order for assigning GSD to task
ERASE rantask
ERASE taskord
RANDOMIZE (VAL(RIGHT$(TIME$, 2)))
FOR ii = 1 TO 6
rantask(ii) = RND
NEXT ii
FOR ii = 1 TO 6
s = 1
FOR J = 1 TO 6
IF rantask(ii) > rantask(J) THEN
s = s + 1
END IF
NEXT J
taskord(ii) = s
NEXT ii
RETURN