A Simulation of Industry and Occupation Codes in 1970 and 1980 U.S. Census

by

M. Didem Avcioglu-Ayturk

A Masters Project Submitted to the Faculty
Of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Master of Science
in
Applied Statistics
by

____
June 2005

APPROVED:

____

Professor Balgobin Nandram, Major Advisor

____

Professor Bogdan Vernescu, Department Head

ABSTRACT

A SIMULATION OF INDUSTRY AND OCCUPATION CODES IN 1970 and 1980 U.S. CENSUS

Classification systems change from census to census for a variety of reasons. The change from 1970 U.S Census to 1980 U.S Census classification was so dramatic that studying the changes and making comparisons are too complicated and expensive.

Treating the actual census results as unknown, we simulated a new Census data base reflecting the real situation in 1970 & 1980 classification systems. One of our objective is to explain the process by which codes change so that the researchers can better understand how the new data bases were created. The second objective is to show how this newly created data base is then used to study the comparability of the two classification systems.

In this project we do not attempt any estimative or predictive inference. We simply simulate the industry and occupation codes in the U.S. Census public-use samples via a model similar to the one used for multiple imputation.

To My beloved Father and Mother…

ACKNOWLEDGEMENTS

First and foremost I would like to express sincere appreciation and my deepest gratitude to my supervisor, Prof. Balgobin Nandram, for his excellence in scientific guidance, insight and his constant support and patience throughout the course of this research and making its completion possible. I know that I have been very lucky to have had an opportunity to experience his unbelievably broad knowledge.

My special thanks to my faithful friends, Jasraj Kohli, Elizabeth Teixeira, Ashley Moraski and Alina Ursan who supported me in numerous ways. Thanks to all of you.

I would also like to thank the Department of Mathematical Sciences for providing me with the facilities and necessary equipments that I utilized during my work. I must also acknowledge the faculty of our department for their help in my education.

My precious parents Mete and Umit, thank you very much for everything. Thanks a lot for your everlasting love, tenderheartedness and for your invaluable advices.

Finally, my deepest thanks go to my husband Engin, my best friend, who urged me on by way of his untiring support and seemingly unlimited belief in me. Thank you for all your guidance, timely suggestions, patience and your support.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Inroduction

Industry and occupation classification systems change from census to census for a variety of reasons. These reasons might be the results of adding new industries and occupations; and/or deleting the declining industries and occupations, or combining them. Then, it is difficult to make comparisons and study changes over time.

For example, there exists substantial difference in coding system of 1970 and 1980 in the industry and occupation (I/O) codes in U.S. census public-use samples (PUS). This difference is so substantial that it is often difficult to estimate trends between the 1970s and 1980s using I/O based statistics. Even in the majority of cases, I/O codes used in 1980 cannot be mapped directly into comparable 1970 codes. Comparability of I/O codes is important to assess for several areas of social science researches, for legal purposes, and for economic and social policy usages.

Double coding millions of records by hand would be too expensive, therefore, in 1983, U.S. Bureau of the Census began effort to recalibrate the I/O codes on 1970 to the 1980 standard. Their research report in 1989, and follow up paper in 1991 represent the most extensive application of multiple imputation. Note that, multiple imputation, as opposed to single imputation, enables analyses of the imputed data sets to reflect variance in classification due to the imputation procedure. They used Bayesian strategy and showed how conventional maximum likelihood methods are difficult to perform.

The goal of this project is to create 1970-1980 Census data base that can be used to study the comparability of industry and occupation classification systems. In order to establish a basis for I/O comparability a set of records coded under the old (1970) and revised (1980) classification systems is needed. For this purpose, we first start with generating the predictor matrix. In this matrix, each row represents the groups, and the columns show the selected set of demographic predictors (age, sex, race, etc.). Then from this matrix, we randomly select 200 cases and create a subsample of 1970 and 1980 schemes. This subsample, consisting of each independently drawn parameter vectors, is first used to predict the possible 1980 industry codings (binary variable) and then used for predicting any given 1970 industry code.

For each of 200 cases, first we find the probability that a person classified in the industry according to the 1980 classification scheme, then we generate the n_i's, representing the sample size in the ith group (i=1,…200) where they were combined in g groups. Then using these informations we predict the possible 1980 codings. Once 1980 coding scheme is generated using the relevant subsample of the double-coded sample, corresponding subsample of the 1970 codings is predicted, and tested.

In Section 2 of Chapter 1, we give brief summary about the factors that make the difference in coding systems. We discuss previously work in the area, census technical papers and efforts. We try to present basic approaches used for recalibrating the 1970 industry and occupation codes in census public-use samples to the 1980 standard.

In Chapter 2 we describe the methodology and computational approach used for relating the industry and occupation codes and comparisons within two different census classifications. In Chapter 3 we present results from the simulations that we performed in creating and matching the coding schemes. Finally, we present our conclusion, a discussion and comparison of the coding systems.

1.2 Literature Survey

There is a great deal of interest among social science researchers in dealing with the problems of data noncomparability. The very changes that are the object of study ordinarily engender changes in the way the data necessary to analyze change, are collected. Industrial and Occupational (I/O) data are prime examples. Occupation refers to the characteristics of a job which is the function fulfilled and the tasks performed to accomplish it. Industry refers to the goods or services produced by the enterprise within which the job is performed.

Each decade the U.S. Census Bureau collects relatively detailed data on the industrial and occupational composition of labor force. But for each census, the detailed industrial and occupational classifications are changed to reflect notable variations among industries and occupations at the time of data collection. Thus, new industries and occupations are added and declining industries and occupations are deleted. Moreover, categories are sometimes recombined in complex ways.

As an example, the 1960 census contained no specific occupation category for the computer programmers, including the 8,700 or so programmers in the labor force at that time in the "professional, technical, and kindred workers, etc." category. By 1970 there were 263,000 computer specialists, divided into three categories in the 1970 census classification: "computer programmers", "computer system analysts", and "computer specialists" (U.S. Bureau of The Census 1973). In 1980 the classification was changed again, this time number of categories reduced to two: "computer systems analysts and scientists" (a subcategory of mathematical and computer scientists) and "computer programmers" (a subcategory of technicians, except health, engineering and science) (U.S. Bureau of the Census 1981). The 1970 category "computer programmers" was split between the two 1980 categories, but the two other 1970 categories mapped only into the 1980 category "computer system analysts and scientists" (Vines and Priebe, 1988).

Many other recombinations were even more complex, and less than one third of the occupation categories in the 1970 detailed occupation classification mapped into a single category in the 1980 detailed occupation classification. Moreover, the categories least likely to change tended to be those involving small numbers of workers, so that less than 15 percent of the labor force in 1970 was in a category that mapped into a single 1980 category (Vines and Priebe, 1988).

Over the past several censuses, changes in the classification of detailed industries have not been as extensive as changes in the detailed occupation classification (because of the adoption of the Standard Industrial Classification prior to the 1940 census), but they are still substantial. About 24 percent of the 1970 industry categories mapped into more than one 1980 category classification, and the non-matching categories included about 36 percent of the 1970 labor force.

While the changes over a 10 years' period in the detailed classification schemes for industry and occupation have little practical consequence for cross-sectional analysis, they pose severe difficulty for many kinds of cross-temporal analysis. For example, it is hard to assess adequately from census data the effect of a decade of affirmative action, since it is not easy to answer accurately questions as whether the proportion of woman in management has changed between 1970 and 1980. The difficulty is that there is no way of knowing whether a given job would be counted as managerial occupation in one scheme but not in the other. Indeed, a person who had the same job in 1970 and 1980 might have moved into or out of a detailed occupational category that we would be willing to count as managerial occupation simply because of a change in the occupational coding scheme. Similar problems arose in assessing the changes in the industrial distribution of labor force. For example, assessment of the claim that jobs in secondary industries are increasingly occupied by women, minorities, and immigrants depend on consistent classification of industries over time.

The Census Bureau has been concerned with the problem of cross-temporal comparability of industry and occupation classifications for a very long time. The census of 1900, for example, includes a comparison of occupation data from 1820 to 1900, although these comparisons were somewhat limited (U.S. Bureau of the Census 1904). In the early 1940's, Alba Edwards (Edwards, 1943) undertook a major monographic study on occupational comparability since 1870. In addition to data for 1940, Edwards' monograph provided detailed occupational data for 1930, reclassified into the 1940 classification, and comparable data for 1870 through 1930, classified according to the 1930 classification. This was followed by a similar effort by Kaplan and Casey (1958), which provided detailed occupational distributions for males and females for 1900 through 1950, classified according to the 1950 classification. The Edwards, Kaplan and Casey monographs provide extremely useful data because they permit the estimates of historical change in the occupational structure that were uncontaminated by changes in the classification scheme. But, they permit no detailed analysis of change in the occupational composition of the labor force with respect to characteristics other than sex.

Subsequent to these efforts, census technical papers were published analyzing the change in the industry and occupation classification systems between 1950 and 1960 (Priebe, 1968), between 1960 and 1970 (Priebe et al. 1972), and between 1970 and 1980 (Vines and Priebe, 1988). In each case, a subsample of census return was "double coded". That is, they were coded with the industry and occupation classification for the subsequent census in addition to the one used initially. This provided two kinds of information: the distribution of the labor force for two successive census years coded into the same classification, and a map relating the categories in one classification to the categories in the other. Like the earlier exercises, however, these were very limited. The estimates of industrial and occupational change were at the national level, broken down only by sex. And the mapping between classifications was at an aggregate level, showing for one year the distribution of the categories for the other year that mapped into it. These distributions can not tell us anything about the fate of any particular job. What mapping does tells us is that, for example, of those classified as working in "health services" in 1970, 11 percent would have been classified as working in "offices of health practitioners", 61 percent as working in "health services", 5 percent as working in "job training and vocational rehabilitation services", and 23 percent as working in "administration of human resources programs", according to the 1980 detailed classification of industries. But the map does not tell us how to assign each 1970 worker in "health services" to a particular 1980 category, which would be necessary, for example, to compare Public-Use Microdata Sample (PUMS) data on individuals from the two censuses.

The inherent difficulty of comparing industry and occupation distributions based on different classification schemes has not deterred analysts from attempting to do so. In the absence of a principled way of converting data from one classification scheme to another, however, analysts have had no choice but to resort to a variety of ad hoc matching schemes (Treiman and Terrell, 1975; Williams, 1976; Pampel, Land, and Felson, 1977; Synder, Hayward, and Hudis, 1978; Blau and Hendricks, 1979; Rumberger 1981).

A major difficulty with the ad hoc schemes commonly employed, apart from their lack of standardization, is that they treat the recalibration process as error free. Thus, inferences about changes over time in industry or occupation characteristics will generally appear stronger than warranted. Moreover, the amount of error is likely to vary substantially and in unknowable ways, in different parts of the classification scheme. The result is that the analyst of social change is ordinarily hard pressed to know the extent to which observed differences in industrial and occupational data reflect true changes in social structure and the extent to which they represent classification error.

There are only two ways out of this dilemma. One is to return to the original data and recode them with the new classification scheme. In the case of PUMSs from the U.S. census, this costs too much. The other is to develop a statistically principled way of converting data from one classification scheme to another, a method that will be relatively accurate and permit an assessment of the degree of error entailed in the conversion process.

There are several reports on evaluating the accuracy with which industry codes assigned to 1970 data can be recalibrated from the 1970 classification to 1980 classification. The basic approach derives from work on the theory of "multiple imputation", proposed by Rubin (1978) to handle problems of missing data in surveys and developed in a number of subsequent publications (Rubin, 1987; Rubin and Schenker, 1986). The general strategy is to predict or impute the missing values from the relationships existing among variables in those cases without missing data. However, instead of obtaining a single imputation of the missing values, imputations were repeated for a number of times and a range of estimates corresponding to the distribution of responses were created in the complete data. These multiple imputations can then be combined to produce an overall best estimate and to compute standard error statistics that reflect both the usual variability inherent in samples and the additional variability due to the imputation process.

As we said before there are several efforts made to create a new Census data base that can be used to study comparability of industry and occupation classification systems. One of the main projects is done by U.S. Bureau of the Census which had begun in 1983, aimed to recalibrate industry and occupation codes on 1970 census public-use samples (PUS's) to the 1980 standard.

The project here shows a work of simulating the industry and occupation codes in the U.S. Census public-use samples via a model similar to the one used for multiple imputation.

CHAPTER 2

SIMULATIONS OF 1980 CODES and 1970 CODES

This chapter represents how to simulate industry and occupation codes in U.S. Census public-use samples via a model particular to the one used for multiple imputation (see Clogg, Rubin, Schenker, Schultz and Weidman (1991)).

2.1 General Considerations

Suppose we have a sample of size N. Our goal is to predict a binary (0-1) variable Y based on a K-dimensional vector of predictors, X. Further, assume that there are I possible distinct values of X ($X_1, \ldots\ldots, X_I$) which means that the sample can be divided into I groups. Let Y_{ij} denote the value of Y for the jth unit in group I (i=1,….,I; j=1,….,n_i), where n_i is the sample size in the ith group and $\sum_i n_i = N$. The Y's are representing the 1980 codes and they are used to predict the corresponding codes in 1970. The X shows a set of predictors like age, race, and sex, and are available in both 1970 and 1980.

2.2 Choice of Predictors

The predictors (components of X) used to predict 1980 industry codes are selected from the 1991 report of the Census Bureau. These predictors show some selected demographic and personal characteristics of respondents. The characteristics used as predictors are age, sex, race, sex and race interaction, class of worker, residence in a Standard Metropolitan Statistical Area, education and hours worked per week. The values of the predictors are coded as +1, 0, or -1 according to the levels of these predictors, so that coefficient values refer to deviations from means. Table 2.2.1 shows the selected predictors used for the industry imputations.

Table 2.2.1 Predictor Variables

Predictor	***Values***
Sex	-1 if male 1 if female
Race	-1 if black 1 if non-black
Sex*Race	Sex x Race
Age1	-1 if 16 ≤ age ≤24 0 if age ≥ 40 1 if 25 ≤ age ≤39
Age2	-1 if 16 ≤ age ≤24 0 if 25 ≤ age ≤39 or ≥ 60 1 if 40 ≤ age ≤59
Class of Worker = 1 (Class-1)	-1 if private industry 0 if self-employed or without pay 1 if government
Class of Worker = 2 (Class-2)	-1 if private industry 0 if government 1 if self-employed or without pay
Metropolitan Residence (Metro)	-1 if in metropolitan area 1 otherwise
Education	-1 if high school or less 1 if at least one year in college
Hours worked per week (Hours)	-1 if hours per week ≤ 34 1 if otherwise

2.3 Model and Algorithm

In this project, although 1970 and 1980 codings are known, for modeling purposes true values are treated as unknown. In order to represent the information that we have about 1970 and 1980 codings, first we generate the X, predictor matrix.

The predictor matrix is 5184 x 10 where each row represents the groups and the columns show the selected set of demographic predictors. Since we work with 6 predictors with two levels and 4 predictors with 3 levels, we have totally I=5184 groups ($2^6 x 3^4$). We select randomly 200 groups to represent the 1970-1980 mappings.

In reality, 1970 coding data and 1980 coding data are not the same. Some cases in 1970 coding are not represented in 1980 coding scheme and likewise some cases in 1980 cannot be mapped to 1970 scheme. In order to model the relation between these two coding schemes, we put our main interest into the part where these mappings can be made, represented by 200 cases as a subsample.

In this randomly selected sample, these 200 cases for 1970 industry code have two different codings in the corresponding set of 1980 codes. This subsample, consisting of each independently drawn parameter vectors, is first used to predict the possible 1980 industry codings and then used for predicting any given 1970 industry code.

For each of 200 cases, let π_i's represent the probability that a person classified in the industry according to the 1980 classification scheme. Here, π_i's are assumed to be distributed as randomly drawn beta variates.

$$\pi_i \sim Beta\left(\frac{\tau\gamma_i \exp(X_i'\beta)}{1+\gamma_i \exp(X_i'\beta)}, \quad \frac{\tau}{1+\gamma_i \exp(X_i'\beta)} \right), \quad i = 1,.....,200, \tag{2.3.1}$$

$$p(\pi_i | \alpha, \beta) \propto \pi_i^{\alpha-1} (\pi_i)^{\beta-1} \tag{2.3.2}$$

where $\alpha = \frac{\tau\gamma_i \exp(X_i'\beta)}{1+\gamma_i \exp(X_i'\beta)}$ and $\beta = \frac{\tau}{1+\gamma_i \exp(X_i'\beta)}$.

Here, γ_i's are flattening constants (technical note: adding flattening constants to the observed frequencies is a sufficient condition for obtaining a unique maximizer for β) generated from randomly drawn Gamma variates,

$$\gamma_i \sim Gamma\left(\alpha, \quad \frac{\alpha}{\exp(\beta_0)} \right), \quad i = 1,....,200, \tag{2.3.3}$$

$$p(\gamma_i | \alpha, \beta_0) \propto \gamma_i^{\alpha-1} (\gamma_i)^{\frac{\alpha}{\exp(\beta_0)}-1}. \tag{2.3.4}$$

Note that we used α value to be equal to 2 and τ value to be 100 in calculations. β_0, intercept value, is also included in the model.

The vector $\beta = (\beta_1,......\beta_{10})$ represents the vector of parameters. The values for this parameter vector are selected from the 1991 report of the Census Bureau, as a result of several fitted logistic regressions. Table 2.3.1 shows the values for this vector. Some coefficient estimates for the predictors are so small that they might be considered as insignificant and one might wonder why they have been included in the model.

Table 2.3.1 Values for the parameter vector based on fitted logistic regressions

Predictor	$\hat{\beta}$
Constant (β_0)	0.436
Sex	0.297
Race	-0.138
Sex*Race	0.009
Age1	-0.289
Age2	0.529
Class of Worker = 1	0.002
Class of Worker = 2	-0.585
Metropolitan Residence	0.238
Education	0.228
Hours worked per week	0.181

The n_i's, representing the sample size in the ith group (i=1,…,200) for 1980 scheme, are combined in g groups and then randomly drawn from a Binomial distirbution with parameters n=3 and with success probability, p=0.50.

$$N_1,.......,N_g \sim Binomial(n, p) \ , \tag{2.3.5}$$

where

$$P(N_i = n_i | n, p) = \binom{n}{n_i} p^{n_i} (1-p)^{n-n_i} \quad n_i = 0,.....,n \,. \tag{2.3.6}$$

Note that it is feasible to assume small sample sizes for the groups because in reality most of the n_i's are zero.

However, in order to relate 1980 coding scheme with corresponding 1970 coding, these combined group totals have to be same. Assuming that $\sum_{i=1}^{g} N_i = t$, we find the joint pdf of sampling probabilities for g grouped n_i's by conditioning on group total t. Here, note that group total t is worked as sufficient statistics. By this way, the joint probability density function of sampling totals conditioned on group total t, $P\left(N_1 = n_1, ...N_g = n_g \middle| \sum_{i=1}^{g} N_i = t\right)$, resulted with hypergeometric distribution.

In order to show this, let us assume that g=2. Then the probability that $N_1, N_2 \sim Binomial(n, p)$ given group total gives us,

$$P\left(N_1 = n_1, N_2 = n_2 \middle| \sum_{i=1}^{2} N_i = t\right) = \frac{P(N_1 = n_1, N_2 = n_2 | N_1 + N_2 = t)}{P(N_1 + N_2 = t)}$$

$$= \frac{P(N_1 = n_1, N_2 = t - n_2)}{P(N_1 + N_2 = t)}$$

$$= \frac{\binom{n}{n_1} p^{n_1} (1-p)^{n-n_1} \binom{n}{t-n_1} p^{t-n_1} (1-p)^{n-t+n_1}}{\binom{2n}{t} p^t (1-p)^{n-t}} \tag{2.3.7}$$

$$= \frac{\binom{n}{n_1}\binom{n}{t-n_1}}{\binom{2n}{t}}.$$

Similarly, if we take g equal to 3, the joint probability density function of sampling totals conditioned on group total t will be,

$$P\left(N_1 = n_1, N_2 = n_2, N_3 = n_3 \middle| \sum_{i=1}^{3} N_i = t \right)$$

$$= P\left(N_3 = n_3 \middle| N_1 = n_1, N_2 = n_2, \quad N_1 + N_2 + N_3 = t\right)$$

$$* P\left(N_2 = n_2 \middle| N_1 = n_1, \quad N_1 + N_2 + N_3 = t\right) * P\left(N_1 = n_1 \middle| N_1 + N_2 + N_3 = t\right)$$

$$= \frac{P(N_1 = n_1, N_1 + N_2 + N_3 = t)}{P(N_1 + N_2 + N_3 = t)}$$

$$= \frac{P(N_1 = n_1, N_2 + N_3 = t - N_1)}{P(N_1 + N_2 + N_3 = t)}$$

$$= \frac{\binom{n}{n_1} p^{n_1} (1-p)^{n-n_1} \binom{2n}{t-n_1} p^{t-n_1} (1-p)^{2n-t+n_1}}{\binom{3n}{t} p^t (1-p)^{3n-t}} \tag{2.3.8}$$

$$= \frac{\binom{n}{n_1}\binom{2n}{t-n_1}}{\binom{3n}{t}}.$$

Continuing in this way the joint probability density function of g group of sampling totals conditioned on group total t can be found as,

$$P\left(N_1 = n_1, \ldots., N_g = n_g \middle| \sum_{i=1}^{g} N_i = t \right)$$

$$= \frac{P\left(N_1 = n_1, \ldots\ldots N_{g-1} = n_{g-1}, N_g = t - \sum_{i=1}^{g-1} N_i \right)}{P\left(\sum_{i=1}^{g} N_i = t \right)}$$

$$= \frac{\prod_{i=1}^{g-1}\binom{n}{n_i} p^{n_i}(1-p)^{n-n_i}\binom{m}{t-\sum_{i=1}^{g-1} n_i} p^{t-\sum_{i=1}^{g-1} n_i}(1-p)^{m-t+\sum_{i=1}^{g-1} n_i}}{\prod_{i=1}^{g}\binom{n}{n_i} p^{n_i}(1-p)^{n-n_i}}$$

$$= \frac{\prod_{i=1}^{g-1}\binom{n}{n_i}\binom{n}{t-\sum_{i=1}^{g-1} n_i}}{\binom{ng}{t}}. \tag{2.3.9}$$

In this step rather than directly using the independent binomial trials in generating the ith sample totals, using the information on grouped total, t, we are able to generate g groups of sample totals by using the hypergeometric distribution.

After obtaining the n_i's, we are ready to predict the possible 1980 industry codings, where Y_{ij}'s are represented by randomly drawn binomial variates,

$$Y_{ij} \sim Binomial(n_i,\ \pi_i)\ ,\ i = 1,\ldots,I\ ,\ j = 1,\ldots,n_i\ , \tag{2.3.10}$$

$$P(Y_{ij} = y_{ij} | n_i, \pi_i) = \binom{n_i}{y_{ij}} \pi_i^{y_{ij}} (1-\pi_i)^{n_i - y_{ij}} \quad y_{ij} = 0,1 \quad n_i = 0,1,2,3. \tag{2.3.11}$$

Once a model predicting a given 1980 codes is generated, each set of 1980 codes for the corresponding subsample of the 1970 PUS is imputed as follows:

First, in order to represent the 1970 codings, we randomly draw a subsample of 200 observations from the predictor matrix. This subsample, consisting of each independently drawn parameter vectors, is different from the subsample created for 1980 scheme. Then for each 200 cases, we draw randomly π_i^*'s, representing the probability that a person classified in the industry according to the 1970 classification scheme, where they are assumed to be distributed as

$$\pi_i^* \sim Beta\left(\frac{\tau\gamma_i^* \exp(X_i'^* \beta)}{1+\gamma_i^* \exp(X_i'^* \beta)}, \quad \frac{\tau}{1+\gamma_i^* \exp(X_i'^* \beta)} \right). \tag{2.3.12}$$

These π_i^*'s are different from the π_i's calculated for 1980 scheme in a way that in estimating the γ_i^*'s from the gamma distribution we worked with different β_0.

After generating the classification probabilities, π_i^*'s, we are left with one more step before completing the 1970 coding scheme. That step is where we generate the n_i's, sample size in the ith group. We again worked with sampling probabilities for g grouped n_i's by conditioning on group total t, similar to the process that we used in creating the 1980 scheme. However, this time we choose to work with different success probability rather than using 0.5.

After obtaining the sample totals for the groups, Y_{ij}^*'s, possible 1970 industry codings are generated from randomly drawn binomial variates.

$$Y_{ij}^* \sim Binomial\left(n_i^*,\ \pi_i^*\right)\ ,\ i=1,...,I\ ,\ j=1,...,n_i^*. \tag{2.3.13}$$

CHAPTER 3

RESULTS & DISCUSSIONS

3.1 Distribution of Codes

As we mentioned earlier, to model the relation between 1970 and 1980 coding schemes we generated a subsample consisting of 200 cases. Then this randomly selected subsample is used to predict the codings for the two censuses, where each census scheme is assumed to be represented by two different industry codings, 0 or 1. It should be noted that, first we predicted the 1980 coding scheme then we used that information to predict the corresponding 1970 coding. In this section we give the results of these codings and show how they distributed according to the 1970 & 1980 censuses.

Each of the 200 cases represents a profile of an observational unit. For example, a case represented by coding "-1 -1 0 -1 -1 1 1" refers to a black female at the age between 16-24, working in the private industry with at least 35 hours per week and resides in a metropolitan area, as described previously in Table 2.3.1. Note that 200 cases are randomly selected (with no replacement) from the predictor matrix, where it holds all kinds of possible profile combinations.

We do not know how many observations are represented by each case, that is, we do not have the exact knowledge about the sample sizes for each group. Therefore, we generated those samples by restricting the group sizes at most with 3 units, in other words, we fixed the total sample size to 300 and then conditioning on this total we randomly generate the group sample sizes from a hypergeometric distribution (See (2.3.9)). Note that it is feasible to assume small sample sizes for the groups because in reality most of the sample sizes are either zero or close to zero.

The probabilities that observations (people) classified in the industry according to the 1980 scheme, generated from a beta distribution (See (2.3.1)), are then combined with the information that we gained from the sample sizes in order to predict the 1980 coding for each unit in 300 samples using the binomial variates. Within the same manner, we were also able to construct the 1970 classification system. The main difference between two codings is merging from the random generation process of flattening constants and sample sizes.

In random generation process of flattening constants, γ_i's, where we generated them from a gamma distribution (See (2.3.3)), for 1980 coding we worked with β_0 value equal to 0.436 where as in 1970's coding we worked with 4 different values of β_0 (See Table 3.1.1). This difference in β_0 values is affecting the shape parameter of the gamma distribution and therefore creating a difference in generation of flattening constants.

On the other hand, in generating the sample sizes we worked with different success probabilities. For 1980 scheme we used p=0.5, and for 1970's sample sizes we used 4 different values of p (See Table 3.1.1). However, as future results showed that, using different p values has no effect on the distribution of the codings.

Table 3.1.1 is constructed based on 100 iterations of coding process discussed above. With respect to the selected values of β_0 and p, first row denoted by #1 shows the number of cases coded with 1 (average of 100 runs for 200 groups) in 1970 coding schemes. Second row (%1) shows the percentage of those cases coded as 1 in the 1970 samples, and third row (SD) shows the standard deviations. Last column shows all those results for 1980 classification where we assumed β_0 equal to 0.436 and p=0.5. Note that, the runs (selected) used in constructing the table results are given in appendix A.III.

Table 3.1.1 Simulated values of the 1970 & 1980 codings

			P					
			0.45	**0.475**	**0.5**	**0.525**	**0.55**	
β_0	0.09		**1970**	**1970**	**1970**	**1970**	**1970**	**1980**
		# 1	74	72	76	73	73	130
		% 1	0.248	0.241	0.253	0.242	0.242	0.432
		SD	0.036	0.024	0.029	0.026	0.026	0.030
	0.218		**1970**	**1970**	**1970**	**1970**	**1970**	**1980**
		# 1	78	79	82	79	80	130
		% 1	0.261	0.263	0.274	0.263	0.266	0.432
		SD	0.032	0.025	0.03	0.027	0.028	0.030
	0.436		**1970**	**1970**	**1970**	**1970**	**1970**	**1980**
		# 1	90	90	93	90	90	130
		% 1	0.30	0.302	0.312	0.30	0.301	0.432
		SD	0.029	0.027	0.029	0.030	0.029	0.030
	0.64		**1970**	**1970**	**1970**	**1970**	**1970**	**1980**
		# 1	100	102	117	101	101	130
		% 1	0.334	0.340	0.391	0.337	0.336	0.432
		SD	0.033	0.024	0.028	0.029	0.032	0.030
	0.872		**1970**	**1970**	**1970**	**1970**	**1970**	**1980**
		# 1	115	115	117	114	114	130
		% 1	0.382	0.382	0.391	0.381	0.379	0.432
		SD	0.031	0.026	0.028	0.031	0.034	0.030

From Table 3.1.1, it is clear that a change in p values is not effecting the distribution of ones in 1970 samples. However, ascending values of β_0 had an increasing impact on the results. The main difference that we can observe from this table is that, there exists a tendency of 1980 samples coded as 1 compared to 1970 samples. However, this table is not adequate to give detailed information about the marginal distribution with respect to the predictors. Next section is devoted to show the marginal distributions of predictors under the 1970 and 1980 industry coding scheme and comparisons based on them.

3.2 Comparisons of Codes for 1970 and 1980 Considering the Marginal Distributions

To make the comparisons between the 1970 and 1980 classification systems, the following tables are constructed based on the model procedure and computational approach described in sections 2.4 and 3.1, respectively. Table 3.2.1 presents the results, based on 100 iterations, for the sample with the same codes in 1970 and 1980 broken down by sex. For the 1970 classification system, as we mentioned earlier, we used 4 different values of β_0. Last two rows of the table show the mean and standard deviation results for this consideration. Here, standard deviation (SD) describes the between variability in the frequencies.

Table 3.2.1 Multiple codes of 1980 compared with actual 1970 codings (Marginal distributions by sex)

SEX			1980 code Male (-1)		1980 code Female (1)	
			0 coded	**1 coded**	**0 coded**	**1 coded**
			95	59	73	73
1970 code	β_0	0.09	124	29	104	43
		SD	12.81	9.69	9.03	5.91
		0.436	117	36	96	51
		SD	11.6	8.29	9.65	5.22
		0.872	104	49	82	65
		SD	11.82	8.53	9.08	5.32
		Average	106	38	103	54
		SD	10.46		7.37	

Remark: Note that, zero and one codings represent two different industry codings.

From Table 3.2.1, it can be seen that 95 males are coded as zero in 1980 classification, where 106 males (average) coded as 0 in corresponding 1970 coding system. The between variability in the frequencies are high.

Table 3.2.2 represents the distribution results, for the the sample with the same codes in 1970 and 1980 according to the race. By Table 3.2.2, we can say that 69 colored cases are coded as one in 1980 classification; where as only 53 colored people (average) coded as 1 the 1970 coding system. We can also observe how the distribution of non-black people is resulted according to these classification systems.

Table 3.2.2 Multiple codes of 1980 compared with actual 1970 codings (Marginal distributions by race)

RACE			1980 code Black (-1)		1980 code Non-black (1)	
			0 coded	**1 coded**	**0 coded**	**1 coded**
			84	68	84	64
1970 code	β_0	0.09	116	42	112	30
		SD	5.77	10.45	11.46	4.87
		0.436	107	51	106	37
		SD	6.64	10.38	10.14	4.86
		0.872	92	65	94	49
		SD	6.47	10.84	9.46	6.06
		Average	105	53	104	39
		SD	8.43		7.81	

Table 3.2.3 shows coding distributions according to the two different classifications for the residence in a Standard Metropolitan Statistical Area. The 89 of the cases residing in a metropolitan area are coded as zero in 1980 classification, where as 113 of the cases coded as 0 in the 1970 coding system. Additionally, comparing the residence results within each classification we can observe the residence preference movement from non-metropolitan areas to the metropolitan areas.

Table 3.2.3 Multiple codes of 1980 compared with actual 1970 codings (Marginal distributions by metropolitan residence)

METROPOLITAN RESIDENCE			1980 code Metro (-1)		1980 code Non-Metro (1)	
			0 coded	**1 coded**	**0 coded**	**1 coded**
			89	59	78	74
1970 code	β_0	0.09	121	32	107	40
		SD	13.03	8.3	4.89	9.19
		0.436	115	39	98	49
		SD	11.34	7.04	4.53	10.36
		0.872	102	51	84	63
		SD	9.6	9.63	3.75	12.14
		Average	113	41	96	51
		SD	9.82		7.48	

Table 3.2.4 represents the distribution results, for the sample with the same codes in 1970 and 1980 according to the education. According to the table, 87 of the cases having at most a high school degree are coded as zero in 1980 classification, where as 106 of them coded as 0 in the 1970 classification.

Table 3.2.4 Multiple codes of 1980 compared with actual 1970 codings (Marginal distributions by education)

EDUCATION			1980 code High sch. or less (-1)		1980 code At least 1 year college (1)	
			0 coded	**1 coded**	**0 coded**	**1 coded**
			87	59	81	73
1970 code	β_0	0.09	114	29	114	43
		SD	8.67	4.97	9.28	9.86
		0.436	107	37	106	51
		SD	6.89	4.58	10.02	9.78
		0.872	96	47	90	67
		SD	5.4	4.68	9.19	10.65
		Average	106	38	103	54
		SD	5.86		9.8	

Moreover, comparing the education results between classifications, we observed that more people started to get higher education degrees. The change in standard deviation from 5.86 to 9.8 is also validating this result.

Table 3.2.5 gives the results for the sample (with the same codes in 1970 and 1980) according to the number of hours worked per week. The 87 of the cases working at most 34 hours per week are coded as zero in 1980 classification, where as 105 of them coded as 0 in the 1970 classification. One interesting result is to note that changes in classification systems in the number of hours worked per week (Table 3.2.5) and in the education (Table 3.2.4) are closely related. This might be a result of existence of close relationship between those predictor variables.

Table 3.2.5 Multiple codes of 1980 compared with actual 1970 codings (Marginal distributions by hours worked per week)

HOURS WORKED PER WEEK (HWPW)			**1980 code HWPW ≤ 34 (-1)**		**1980 code HWPW ≥ 35 (1)**	
			0 coded	**1 coded**	**0 coded**	**1 coded**
			87	56	80	77
1970 code	β_0	0.09	115	32	113	40
		SD	10.64	5.64	12.83	7.82
		0.436	106	40	107	47
		SD	10.42	5.92	13.19	7.77
		0.872	93	53	93	61
		SD	12.23	6.63	12.88	7.7
		Average	105	42	104	49
		SD	8.58		10.37	

All Table results are validating the change in the structure of the industries. For example, Table 3.2.1 is showing the change in the sexual structure of industries.

To sum up, we can state that the industry classification systems of 1970 and 1980 censuses is successfully simulated. The changes in those classifications are clearly observed. In order to recalibrate the 1970 classification system to the 1980 standard, several iterations were made and as a result of those runs we clearly observed the changes in the structure of the industries considering the selected demographic predictors.

For future work, since we have all the information for 1980, we can fit the model to the 1980 data. We have information about the individuals and their covariates for 1970 data, but not any information on their codes. Then 1970 can be predicted using the model fitted for 1980 data. This will constitute a full simulation study.

REFERENCES

Blau, F.D., and Wallace, E.H. (1979). Occupational Segregation by Sex: Trends and Prospects. Journal of Human Resources. Vol. 14, 197-210.

Clogg, C.C., Rubin, D.B., Schenker, N., Schultz, B., and Weidman L. (1991). Multiple Imputation of Industry and Occupation Codes in Census Public-Use Samples Using Bayesian Logistic Regression. Journal of the American Statistical Association. Vol. 86, No.413, 68-78.

Edwards, A.M. (1943). Sixteen Census of the United States. Population Comparative Occupational Statistics for the United States, 1887 to 1940: A Comparison of the 1930 and 1940 Census Occupation and Industry Classifications and Statistics; a Comparable Series of Occupation Statistics, 1887 t0 1930; and a Socio-Economic Grouping of the Labor Force, 1910 to 1940. Washington, DC: U.S. Government Printing Office

Kaplan, D.L., and Casey, M.C. (1958). Occupational Trends in the United States 1900 to 1950. Working Paper No.5. Washington, DC: U.S. Bureau of the Census.

Pampel, F.C., Land, K.C., and Felson, M. (1977). A Social Indicator Model of Changes in the Occupational Structure of the United States, 1947-1974. American Sociological Review. Vol. 42, 951-964.

Priebe, J.A. (1968). Changes between the 1950 and 1960 Occupation and Industry Classifications with Detailed Adjustments of 1950 Data to the 1960 Classifications. Bureau of the Census Technical Paper No. 18. Washington, DC: U.S. Government Printing Office.

Priebe, J.A., Heinkel, J. and Greene, S. (1972). 1970 Occupation and Industry Classification Systems in Terms of their 1960 Occupation and Industry Elements. Washington, DC: U.S. Government Printing Office.

Rubin, D.B. (1978). Multiple Imputations in Sample Surveys-A Phenomenological Bayesian Approach to Nonresponse. Proceedings of the

Survey Research Methods Section, American Statistical Association. Washington, DC: American Statistical Association.

Rubin, D.B. (1983). Progress Report on Project for Multiple Imputation of 1980 Codes, unpublished manuscript distributed to the U.S. Bureau of the Census, the National Science Foundation, and the Social Science Research Council.

Rubin, D.B., and Schenker, N. (1986). Multiple Imputation for Interval Estimation from Simple Random Samples with Ignorable Nonresponse. Journal of the American Statistical Association. Vol. 81, 366-374.

Rubin, D.B., and Schenker, N. (1987a). Interval Estimation From Multiply Imputed Data: A Case Study Using Census Agriculture Industry Codes. Journal of Official Statistics. Vol.3, 375-387.

Rubin, D.B., and Schenker, N. (1987b). Logit Based Interval Estimation for Binomial Data Using the Jeffreys Prior. Sociological Methodology, ed. C.C. Clogg. Washington, DC: American Sociological Association.

Rumberger, R. (1981). Changing Skill Requirements of Jobs in the U.S. Ecenomy. Industrial and Labor Relations Review. Vol. 34, 578-590.

Schenker, N., and Treiman, D.J. (1989). Analysis of Public-Use Data with Multiply Imputed Industry and Occupation Codes. UCLA Statistics Series. No.39.

Schenker, N., and Treiman, D.J., and Weidman, L. (1993). Analyses of Public Use Decennial Census Data with Multiply Imputed Industry and Occupation Codes. Applied Statistics. Vol. 42, No.3, 545-556.

Synder, D., Hayward, M.D., and Hudis, P.M. (1978). The Location of Change in the Sexual Structure of Occupations, 1950-1970: Insights from Labor Market Segmentation Theory. American Journal of Sociology. Vol. 84, 706-717.

Treiman, D.J., and Terrell, K. (1975). Women, Work, and Wages-Trends in the Female Occupational Structure Since 1940. Social Indicator Models, edited by K.C. Land and S. Spilerman. New York: Russell Sage. 157-199.

Treiman, D.J., and Rubin, D.B. (1983). Multiple Imputation of Categorical Data to Achieve Calibrated Public-Use Samples, unpublished proposal to the National Science Foundation.

Treiman, D.J., Bielby, W.T., and Cheng, M.T. (1988). Evaluating a Multiple-Imputation Method for Recalibrating 1970 U.S. Census Detailed Industry Codes to the 1980 Standard. Sociological Methodology. Vol. 18, 309-345.

U.S. Bureau of the Census. (1940). Twelfth Census of the United States 1900. Special Reports. Occupations at the Twelfth Census. Washington, DC: U.S. Government Printing Office.

U.S. Bureau of the Census. (1973). Census of the Population: 1970. Subject Reports. Final Report PC(2)-7A. Occupational Characteristics. Washington, DC: U.S. Government Printing Office.

U.S. Bureau of the Census. (1976). 1970 Census of Population and Housing: Procedural History. Report PHC(R)1. Washington, DC: U.S. Government Printing Office.

U.S. Bureau of the Census. (1981). Census of the Population Alphabetical Index of Industries and Occupations. 2d ed. Washington, DC: U.S. Government Printing Office.

U.S. Bureau of the Census. (1983). Detailed Occupation and Years of School Completed by Age, for the Civilian Labor Force by Sex, Race, and Spanish Origin: 1980, Supplementary Report PC80-S1-8. Washington, DC: U.S. Government Printing Office.

U.S. Bureau of the Census. (1984). Detailed Occupation of the Experienced Civilian Labor Force by Sex for the United States and Regions: 1980 and 1970, Supplementary Report PC80-S1-15. Washington, DC: U.S. Government Printing Office.

Vines, P., and Priebe, J.A. (1988). The Relationship between the 1970 and 1980 Industry and Occupation Classification Systems. Washington, DC: U.S. Government Printing Office.

Weidman, L. (1989). Statistical Research Division Report Series: Final Report. Industry and Occupation Imputation. Washington, DC: U.S. Government Printing Office.

Williams, G. (1976). Trends in Occupational Differentiation by Sex. Demography. Vol.16, 73-88.

APPENDIX A

F77 SOURCE CODE

Note: Most of the subroutines are obtained from the Applied Statistics Library (StatLib) of Carnegie Mellon University, Pittsburgh.

A.I : Source Code For Simulations

```
real x(500,0:10),beta(0:10),gam(500),
     +  pie1(500),pie2(500)

      integer ix(10000,10),inds(10000),indns(10000),
     + n1(500),n2(500),iy1(500,100),iy2(500,100)

      beta(0) = .436
      beta(1) = .297
      beta(2) = -.138
      beta(3) = .009
      beta(4) = .238
      beta(5) = .228
      beta(6) = .181
      beta(7) = -.289
      beta(8) = .529
      beta(9) = .002
      beta(10) = -.585

      alpha = 10

      tau = 100

      isum = 0
      do i1=-1,1,2
      do i2=-1,1,2
      do i3=-1,1,2
      do i4=-1,1,2
      do i5=-1,1,2
      do i6=-1,1,2
      do i7=-1,1
      do i8=-1,1
      do i9=-1,1
      do i10=-1,1
```

```
          isum = isum + 1
          ix(isum,1) = i1
          ix(isum,2) = i2
          ix(isum,3) = i3
          ix(isum,4) = i4
          ix(isum,5) = i5
          ix(isum,6) = i6
          ix(isum,7) = i7
          ix(isum,8) = i8
          ix(isum,9) = i9
          ix(isum,10) = i10
         end do
         end do
         end do
         end do
         end do
         end do
         end do
         end do
         end do
         end do

         idum = -5

         n = isum
         k = 2304

         ng = 200
         np = 10

         do 1000 its=1,100

         call random(n,ng,idum,inds,indns)

         do i=1,ng
           x(i,0) = 1
          write(6,'(12i5)') i,inds(i),(ix(inds(i),j),j=1,10)
          do j=1,10
           x(i,j) = ix(inds(i),j)
          end do
         end do

         write(6,*) 'ng n',ng,n

         pp =.5
         nb = 3
         nt = 300
         n1(1) = 2
         do i=2,ng
           isum = 0
           do k=1,i-1
            isum = isum + n1(k)
           end do
           ntc = nt-isum
           nn1 = nb
           nn2 = nb*(ng-i)
111        nval1 = bnldev(pp,nn1,idum)
           nval2 = bnldev(pp,nn2,idum)
           nvals = nval1 + nval2
           if(nvals .ne. ntc) go to 111
           n1(i) = nval1
         end do
         isumt = 0
         do i=1,ng
          isumt = isumt + n1(i)
          write(6,*) i,n1(i),isumt
```

```
      end do
      pp =.45
      nb = 3
      nt = 300
      n2(1) = 1
      do i=2,ng
        isum = 0
        do k=1,i-1
         isum = isum + n2(k)
        end do
        ntc = nt-isum
        nn1 = nb
        nn2 = nb*(ng-i)
112     nval1 = bnldev(pp,nn1,idum)
        nval2 = bnldev(pp,nn2,idum)
        nvals = nval1 + nval2
        if(nvals .ne. ntc) go to 112
        n2(i) = nval1
      end do
      isumt1 = 0
      isumt2 = 0
      do i=1,ng
       isumt1 = isumt1 + n1(i)
       isumt2 = isumt2 + n2(i)
      end do

      do i=1,ng
       asum = 0.
       do k=1,np
        asum = asum + x(i,k)*beta(k)
       end do
       call gamdev(idum,alpha,dran)
       gam(i) = dran/(alpha/(.5*exp(beta(0))))
       dum = gam(i)*exp(asum)/(1+gam(i)*exp(asum))
       alp1 = dum*tau
       alp2 = (1-dum)*tau
       call gamdev(idum,alp1,dran1)
       call gamdev(idum,alp2,dran2)
       pie1(i) = dran1/(dran1+dran2)

       call gamdev(idum,alpha,dran)
       gam(i) = dran/(alpha/(.25*exp(beta(0))))
       dum = gam(i)*exp(asum)/(1+gam(i)*exp(asum))
       alp1 = dum*tau
       alp2 = (1-dum)*tau
       call gamdev(idum,alp1,dran1)
       call gamdev(idum,alp2,dran2)
       pie2(i) = dran1/(dran1+dran2)
      end do

      do i=1,ng
       do j=1,n1(i)
        iy1(i,j) = 0
        uu = ran1(idum)
        if(uu .le. pie1(i)) iy1(i,j) = 1
       end do
       do j=1,n2(i)
        iy2(i,j) = 0
        uu = ran1(idum)
        if(uu .le. pie2(i)) iy2(i,j) = 1
       end do
       isum1 = 0
       do j=1,n1(i)
        isum1 = isum1+iy1(i,j)
       end do
       isum2 = 0
```

```
        do j=1,n2(i)
         isum2 = isum2+iy2(i,j)
        end do

        write(15,'(2i5,4i8)') its,i,n1(i),isum1,n2(i),isum2

       end do

       isum11 = 0
       isum12 = 0
       isum21 = 0
       isum22 = 0
       do i=1,ng
        if(n1(i) .eq. 0 .and. n2(i) .eq. 0) isum11 = isum11 + 1
        if(n1(i) .eq. 0 .and. n2(i) .gt. 0) isum12 = isum12 + 1
        if(n1(i) .gt. 0 .and. n2(i) .eq. 0) isum21 = isum21 + 1
        if(n1(i) .gt. 0 .and. n2(i) .gt. 0) isum22 = isum22 + 1
       end do
       j=1
       write(16,*) its,j,isum11,isum12
       j=2
       write(16,*) its,j,isum21,isum22

       isum1 = 0
       isum2 = 0
       do i=1,ng
       do j=1,n1(i)
        isum1 = isum1 + iy1(i,j)
        isum2 = isum2 + iy2(i,j)
       end do
       end do
       write(16,*) its,isum1,isum2

       isum1 = 0
       isum2 = 0
       isum3 = 0
       isum4 = 0
isum5 = 0
isum6 = 0
       do i=1,ng
        if(x(i,7) .eq. -1) then
         if(n2(i) .gt. 0) then
          do j=1,n2(i)
           if(iy2(i,j) .eq. 0) then
            isum1 = isum1 + 1
           elseif(iy2(i,j) .eq. 1) then
            isum2 = isum2 + 1
          else
            isum3 = isum3 + 1
           end if
          end do
         end if
        elseif(x(i,7) .eq. 1) then
         if(n2(i) .gt. 0) then
          do j=1,n2(i)
           if(iy2(i,j) .eq. 0) then
            isum4 = isum4 + 1
           elseif(iy2(i,j) .eq. 1) then
            isum5 = isum5 + 1
          else
            isum6 = isum6 + 1
           end if
          end do
         end if
        end if
       end do
```

```
      isumt = isum1+isum2+isum3+isum4
      write(101,'(6i8)') isum1,isum2,isum3,isum4,isum5,isum6


 1000 continue

      stop
      end
```

Choosing a random subset of size k of n

```
c     Sheldon Ross (1994, pg. 264-266)
c     Begin ... December 10, 2002 ...
c     End   ... December 10, 2002 ...
c
      subroutine random(n,k,idum,inds,indns)

      integer ind(10000),inds(10000),indn(10000)

      do i=1,n
       ind(i) = 0
      end do

      if(ran1(idum) .lt. k/float(n)) ind(1) = 1
      do i=2,n
       isum = 0
       do j=1,i
        isum = isum + ind(j)
       end do
       if(ran1(idum) .lt. (k-isum)/float(n-i)) ind(i) = 1
      end do

      isum = 0
      isum1 = 0
      isum2 = 0
      do i=1,n
       isum = isum + ind(i)
       if(ind(i) .eq. 1) then
        isum1 = isum1 + 1
        inds(isum1) = i
       elseif(ind(i) .eq. 0) then
        isum2 = isum2 + 1
        indn(isum2) = i
       end if
      end do

      return
      end
c
      function bnldev(pp,n,idum)
c
c     Returns as a floating-point number an integer value that is a
c     random deviate drawn from a binomial distribution of  n
c     trials each of probability pp, using ran1(idum) as a source
c     of uniform random deviate
c     Requires gammln,ran1
c
      integer idum,n
      real bnldev,pp,pi
      parameter (pi=3.141592654)
      integer j,nold
      real am,em,en,g,oldg,p,pc,pclog,plog,pold,sq,t,y,gammln,ran1
      save nold,pold,pc,plog,pclog,en,oldg
      data nold /-1/, pold /-1./
      if(pp.le.0.5)then
        p=pp
      else
        p=1.-pp
```

```
      endif
 am=n*p
 if (n.lt.25)then
     bnldev=0.
     do 11 j=1,n
          if(ran1(idum).lt.p)bnldev=bnldev+1.
11         continue
 elseif (am.lt.1.) then
     g=exp(-am)
     t=1.
     do 12 j=0,n
          t=t*ran1(idum)
                 if (t.lt.g) goto 1
12         continue
     j=n
1     bnldev=j
 else
     if (n.ne.nold) then
          en=n
         oldg=gammln(en+1.)
         nold=n
 endif
 if (p.ne.pold) then
         pc=1.-p
         plog=log(p)
         pclog=log(pc)
         pold=p
  endif
     sq=sqrt(2.*am*pc)
2       y=tan(pi*ran1(idum))
           em=sq*y+am
     if (em.lt.0..or.em.ge.en+1.) goto 2
     em=int(em)
     t=1.2*sq*(1.+y**2)*exp(oldg-gammln(em+1.)
    +      -gammln(en-em+1.)+em*plog+(en-em)*pclog)
     if (ran1(idum).gt.t) goto 2
     bnldev=em
 endif
      if (p.ne.pp) bnldev=n-bnldev
      return
      end
c
c     .... Generate a Gamma deviate ....
c
      SUBROUTINE gamdev(idum,aalp,d)
      REAL aalp
      double precision U,DF,PPCHI2
      DF = 2*aalp
1111    U = RAN1(IDUM)
        if( u .le. .000002 .or. u .ge. .999998 ) go to 1111
        D = ( ppchi2( U,DF,IFAULT ) )/2
      RETURN
      END
c
      FUNCTION RAN1(IDUM)
      DIMENSION R(97)
      PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
      PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
      PARAMETER (M3=243000,IA3=4561,IC3=51349)
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
        IX1=MOD(IC1-IDUM,M1)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IX1,M2)
        IX1=MOD(IA1*IX1+IC1,M1)
```

```
          IX3=MOD(IX1,M3)
          DO 11 J=1,97
            IX1=MOD(IA1*IX1+IC1,M1)
            IX2=MOD(IA2*IX2+IC2,M2)
            R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
11        CONTINUE
          IDUM=1
        ENDIF
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IA2*IX2+IC2,M2)
        IX3=MOD(IA3*IX3+IC3,M3)
        J=1+(97*IX3)/M3
        IF(J.GT.97.OR.J.LT.1)PAUSE
        RAN1=R(J)
        R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
        RETURN
        END
c
c
         double precision function ppchi2(p, v, ifault)
c
c          Algorithm AS 91   Appl. Statist. (1975) Vol.24, P.35
c
c          To evaluate the percentage points of the chi-squared
c          probability distribution function.
c
c          p must lie in the range 0.000002 to 0.999998,
c          v must be positive,
c          g must be supplied and should be equal to
c            ln(gamma(v/2.0))
c
c       Incorporates the suggested changes in AS R85 (vol.40(1),
c       pp.233-5, 1991) which should eliminate the need for the limited
c       range for p above, though these limits have not been removed
c       from the routine.
c       If IFAULT = 4 is returned, the result is probably as accurate as
c       the machine will allow.
c
c       Auxiliary routines required: PPND = AS 111 (or AS 241) and
c       GAMMAD = AS 239.
c
        integer maxit
        parameter (maxit = 20)
        double precision p, v, g, gammad, ppnd, aa, e, zero, half, one,
     $   two, three, six, pmin, pmax, c1, c2, c3, c4, c5, c6, c7,
     $   c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19,
     $   c20, c21, c22, c23, c24, c25, c26, c27, c28, c29, c30,
     $   c31, c32, c33, c34, c35, c36, c37, c38, a, b, c, ch, p1, p2,
     $   q, s1, s2, s3, s4, s5, s6, t, x, xx
c
        data            aa,       e,    pmin,     pmax
     $     /0.6931471806d0, 0.5d-6, 0.000002d0, 0.999998d0/
        data zero, half, one, two, three, six
     $     /0.0d0, 0.5d0, 1.0d0, 2.0d0, 3.0d0, 6.0d0/
        data        c1,     c2,     c3,     c4,     c5,     c6,
     $              c7,     c8,     c9,    c10,    c11,    c12,
     $             c13,    c14,    c15,    c16,    c17,    c18,
     $             c19,    c20,    c21,    c22,    c23,    c24,
     $             c25,    c26,    c27,    c28,    c29,    c30,
     $             c31,    c32,    c33,    c34,    c35,    c36,
     $             c37,    c38/
     $          0.01d0, 0.222222d0, 0.32d0, 0.4d0, 1.24d0, 2.2d0,
     $          4.67d0, 6.66d0, 6.73d0, 13.32d0, 60.0d0, 70.0d0,
     $          84.0d0, 105.0d0, 120.0d0, 127.0d0, 140.0d0, 1175.0d0,
     $         210.0d0, 252.0d0, 2264.0d0, 294.0d0, 346.0d0, 420.0d0,
     $         462.0d0, 606.0d0, 672.0d0, 707.0d0, 735.0d0, 889.0d0,
```

```
     $              932.0d0,  966.0d0,  1141.0d0,  1182.0d0,  1278.0d0,
1740.0d0,
     $       2520.0d0, 5040.0d0/
      gind = v/2.
      g = gammln( gind )
c
c        test arguments and initialise
c
      ppchi2 = -one
      ifault = 1
      if (p .lt. pmin .or. p .gt. pmax) return
      ifault = 2
      if (v .le. zero) return
      ifault = 0
      xx = half * v
      c = xx - one
c
c        starting approximation for small chi-squared
c
      if (v .ge. -c5 * log(p)) goto 1
      ch = (p * xx * exp(g + xx * aa)) ** (one/xx)
      if (ch .lt. e) goto 6
      goto 4
c
c        starting approximation for v less than or equal to 0.32
c
    1 if (v .gt. c3) goto 3
      ch = c4
      a = log(one-p)
    2 q = ch
      p1 = one + ch * (c7+ch)
      p2 = ch * (c9 + ch * (c8 + ch))
      t = -half + (c7 + two * ch) / p1 - (c9 + ch * (c10 +
     $  three * ch)) / p2
      ch = ch - (one - exp(a + g + half * ch + c * aa) *
     $  p2 / p1) / t
      if (abs(q / ch - one) .gt. c1) goto 2
      goto 4
c
c        call to algorithm AS 111 - note that p has been tested above.
c  AS 241 could be used as an alternative.
c
    3 x = ppnd(p, if1)
c
c        starting approximation using Wilson and Hilferty estimate
c
      p1 = c2 / v
      ch = v * (x * sqrt(p1) + one - p1) ** 3
c
c        starting approximation for p tending to 1
c
      if (ch .gt. c6 * v + six)
     $   ch = -two * (log(one-p) - c * log(half * ch) + g)
c
c
c        call to algorithm AS 239 and calculation of seven term
c        Taylor series
c
    4 do 7 i = 1, maxit
      q = ch
      p1 = half * ch
      p2 = p - gammad(p1, xx, if1)
      if (if1 .eq. 0) goto 5
c
      ifault = 3
      return
```

```
    5 t = p2 * exp(xx * aa + g + p1 - c * log(ch))
      b = t / ch
      a = half * t - b * c
      s1 = (c19 + a * (c17 + a * (c14 + a * (c13 + a * (c12 +
     $  c11 * a))))) / c24
      s2 = (c24 + a * (c29 + a * (c32 + a * (c33 + c35 *
     $  a)))) / c37
      s3 = (c19 + a * (c25 + a * (c28 + c31 * a))) / c37
      s4 = (c20 + a * (c27 + c34 * a) + c * (c22 + a * (c30 +
     $  c36 * a))) / c38
      s5 = (c13 + c21 * a + c * (c18 + c26 * a)) / c37
      s6 = (c15 + c * (c23 + c16 * c)) / c38
      ch = ch + t * (one + half * t * s1 - b * c * (s1 - b *
     $  (s2 - b * (s3 - b * (s4 - b * (s5 - b * s6))))))
      if (abs(q / ch - one) .gt. e) goto 6
    7 continue
      ifault = 4
c
    6 ppchi2 = ch
      return
      end
c
C
        DOUBLE PRECISION FUNCTION PPND(P,IER)
C
C       ALGORITHM AS 111, APPL.STATIST., VOL.26, 118-121, 1977.
C
C       PRODUCES NORMAL DEVIATE CORRESPONDING TO LOWER TAIL AREA = P.
C
C See also AS 241 which contains alternative routines accurate to
C about 7 and 16 decimal digits.
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DATA SPLIT/0.42D0/
        DATA A0,A1,A2,A3/2.50662823884D0,-18.61500062529D0,
     1  41.39119773534D0,-25.44106049637D0/, B1,B2,B3,B4/
     2  -8.47351093090D0,23.08336743743D0,-21.06224101826D0,
     3            3.13082909833D0/,      C0,C1,C2,C3/-2.78718931138D0,-
2.29796479134D0,
     4  4.85014127135D0,2.32121276858D0/, D1,D2/3.54388924762D0,
     5  1.63706781897D0/
  DATA ZERO/0.D0/, ONE/1.D0/, HALF/0.5D0/
C
        IER = 0
        Q = P-HALF
        IF (ABS(Q).GT.SPLIT) GO TO 10
C
C       0.08 < P < 0.92
C
        R = Q*Q
        PPND  =  Q*(((A3*R  +  A2)*R  +  A1)*R  +  A0)/((((B4*R  +  B3)*R  +
B2)*R
     1         + B1)*R + ONE)
        RETURN
C
C       P < 0.08 OR P > 0.92, SET R = MIN(P,1-P)
C
   10   R = P
        IF (Q.GT.ZERO) R = ONE-P
        IF (R.LE.ZERO) GO TO 20
        R = SQRT(-LOG(R))
        PPND = (((C3*R + C2)*R + C1)*R + C0)/((D2*R + D1)*R + ONE)
        IF (Q.LT.ZERO) PPND = -PPND
        RETURN
   20   IER = 1
        PPND = ZERO
```

```
        RETURN
        END
c
c
 DOUBLE PRECISION FUNCTION GAMMAD(X, P, IFAULT)
C
C ALGORITHM AS239  APPL. STATIST. (1988) VOL. 37, NO. 3
C
C Computation of the Incomplete Gamma Integral
C
C Auxiliary functions required: ALNGAM = logarithm of the gamma
C function, and ALNORM = algorithm AS66
C
 INTEGER       IFAULT
 DOUBLE PRECISION PN1, PN2, PN3, PN4, PN5, PN6, X, TOL, OFLO,
    *          XBIG, ARG, C, RN, P, A, B, ONE, ZERO, ALNGAM,
    *          AN, TWO, ELIMIT, PLIMIT, ALNORM, THREE, NINE
 PARAMETER (ZERO = 0.D0, ONE = 1.D0, TWO = 2.D0, OFLO = 1.D+37,
    *          THREE = 3.D0, NINE = 9.D0, TOL = 1.D-14, XBIG = 1.D+8,
    *          PLIMIT = 1000.D0, ELIMIT = -88.D0)
 EXTERNAL ALNGAM, ALNORM
C
 GAMMAD = ZERO
C
C Check that we have valid values for X and P
C
 IF (P .LE. ZERO .OR. X .LT. ZERO) THEN
   IFAULT = 1
   RETURN
 END IF
 IFAULT = 0
 IF (X .EQ. ZERO) RETURN
C
C Use a normal approximation if P > PLIMIT
C
 IF (P .GT. PLIMIT) THEN
   PN1 = THREE * SQRT(P) * ((X / P) ** (ONE / THREE) + ONE /
    *          (NINE * P) - ONE)
   GAMMAD = ALNORM(PN1, .FALSE.)
   RETURN
 END IF
C
C If X is extremely large compared to P then set GAMMAD = 1
C
 IF (X .GT. XBIG) THEN
   GAMMAD = ONE
   RETURN
 END IF
C
 IF (X .LE. ONE .OR. X .LT. P) THEN
C
C Use Pearson's series expansion.
C (Note that P is not large enough to force overflow in ALNGAM).
C No need to test IFAULT on exit since P > 0.
C
   ARG = P * LOG(X) - X - ALNGAM(P + ONE, IFAULT)
   C = ONE
   GAMMAD = ONE
   A = P
  40     A = A + ONE
   C = C * X / A
   GAMMAD = GAMMAD + C
   IF (C .GT. TOL) GO TO 40
   ARG = ARG + LOG(GAMMAD)
   GAMMAD = ZERO
   IF (ARG .GE. ELIMIT) GAMMAD = EXP(ARG)
```

```
C
  ELSE
C
C Use a continued fraction expansion
C
    ARG = P * LOG(X) - X - ALNGAM(P, IFAULT)
    A = ONE - P
    B = A + X + ONE
    C = ZERO
    PN1 = ONE
    PN2 = X
    PN3 = X + ONE
    PN4 = X * B
    GAMMAD = PN3 / PN4
   60     A = A + ONE
    B = B + TWO
    C = C + ONE
    AN = A * C
    PN5 = B * PN3 - AN * PN1
    PN6 = B * PN4 - AN * PN2
    IF (ABS(PN6) .GT. ZERO) THEN
      RN = PN5 / PN6
      IF (ABS(GAMMAD - RN) .LE. MIN(TOL, TOL * RN)) GO TO 80
      GAMMAD = RN
    END IF
C
    PN1 = PN3
    PN2 = PN4
    PN3 = PN5
    PN4 = PN6
    IF (ABS(PN5) .GE. OFLO) THEN
C
C Re-scale terms in continued fraction if terms are large
C
      PN1 = PN1 / OFLO
      PN2 = PN2 / OFLO
      PN3 = PN3 / OFLO
      PN4 = PN4 / OFLO
    END IF
    GO TO 60
   80     ARG = ARG + LOG(GAMMAD)
    GAMMAD = ONE
    IF (ARG .GE. ELIMIT) GAMMAD = ONE - EXP(ARG)
  END IF
C
  RETURN
  END
c
c
      double precision function alnorm(x,upper)
c
c         Algorithm AS66 Applied Statistics (1973) vol22 no.3
c
c       Evaluates the tail area of the standardised normal curve
c       from x to infinity if upper is .true. or
c       from minus infinity to x if upper is .false.
c
      double precision zero,one,half
      double precision con,z,y,x
      double precision p,q,r,a1,a2,a3,b1,b2,c1,c2,c3,c4,c5,c6
      double precision d1,d2,d3,d4,d5
      logical upper,up
c*** machine dependent constants
      double precision ltone,utzero
      data zero/0.0d0/, one/1.0d0/, half/0.5d0/
      data ltone/7.0d0/,utzero/18.66d0/
```

```
      data con/1.28d0/
      data p/0.398942280444d0/,q/0.39990348504d0/,r/0.398942280385d0/
      data a1/5.75885480458d0/,a2/2.62433121679d0/,a3/5.92885724438d0/
      data b1/-29.8213557807d0/,b2/48.6959930692d0/
      data c1/-3.8052d-8/,c2/3.98064794d-4/,c3/-0.151679116635d0/
      data c4/4.8385912808d0/,c5/0.742380924027d0/,c6/3.99019417011d0/
      data d1/1.00000615302d0/,d2/1.98615381364d0/,d3/5.29330324926d0/
      data d4/-15.1508972451d0/,d5/30.789933034d0/
c
      up=upper
      z=x
      if(z.ge.zero)goto 10
      up=.not.up
      z=-z
   10 if(z.le.ltone.or.up.and.z.le.utzero)goto 20
      alnorm=zero
      goto 40
   20 y=half*z*z
      if(z.gt.con) goto 30
c
      alnorm=half-z*(p-q*y/(y+a1+b1/(y+a2+b2/(y+a3))))
      goto 40
   30 alnorm=r*dexp(-y)/(z+c1+d1/(z+c2+d2/(z+c3+d3/(z+c4+d4/(z+c5+d5/
     2   (z+c6))))))
   40 if(.not.up)alnorm=one-alnorm
      return
      end
c
      DOUBLE PRECISION FUNCTION ALNGAM(XVALUE, IFAULT)
C
C     ALGORITHM AS245  APPL. STATIST. (1989) VOL. 38, NO. 2
C
C     Calculation of the logarithm of the gamma function
C
      INTEGER IFAULT
      DOUBLE PRECISION ALR2PI, FOUR, HALF, ONE, ONEP5, R1(9), R2(9),
     +         R3(9), R4(5), TWELVE, X, X1, X2, XLGE, XLGST, XVALUE,
     +         Y, ZERO
C
C     Coefficients of rational functions
C
      DATA R1/-2.66685 51149 5D0, -2.44387 53423 7D1,
     +        -2.19698 95892 8D1,  1.11667 54126 2D1,
     +         3.13060 54762 3D0,  6.07771 38777 1D-1,
     +         1.19400 90572 1D1,  3.14690 11574 9D1,
     +         1.52346 87407 0D1/
      DATA R2/-7.83359 29944 9D1, -1.42046 29668 8D2,
     +         1.37519 41641 6D2,  7.86994 92415 4D1,
     +         4.16438 92222 8D0,  4.70668 76606 0D1,
     +         3.13399 21589 4D2,  2.63505 07472 1D2,
     +         4.33400 02251 4D1/
      DATA R3/-2.12159 57232 3D5,  2.30661 51061 6D5,
     +         2.74647 64470 5D4, -4.02621 11997 5D4,
     +        -2.29660 72978 0D3, -1.16328 49500 4D5,
     +        -1.46025 93751 1D5, -2.42357 40962 9D4,
     +        -5.70691 00932 4D2/
      DATA R4/ 2.79195 31791 8525D-1, 4.91731 76105 05968D-1,
     +         6.92910 59929 1889D-2, 3.35034 38150 22304D0,
     +         6.01245 92597 64103D0/
C
C     Fixed constants
C
      DATA ALR2PI/9.18938 53320 4673D-1/, FOUR/4.D0/, HALF/0.5D0/,
     +     ONE/1.D0/, ONEP5/1.5D0/, TWELVE/12.D0/, ZERO/0.D0/
C
C     Machine-dependant constants.
```

```
C     A table of values is given at the top of page 399 of the paper.
C     These values are for the IEEE double-precision format for which
C     B = 2, t = 53 and U = 1023 in the notation of the paper.
C
      DATA XLGE/5.10D6/, XLGST/1.D+305/
C
      X = XVALUE
      ALNGAM = ZERO
C
C     Test for valid function argument
C
      IFAULT = 2
      IF (X .GE. XLGST) RETURN
      IFAULT = 1
      IF (X .LE. ZERO) RETURN
      IFAULT = 0
C
C     Calculation for 0 < X < 0.5 and 0.5 <= X < 1.5 combined
C
      IF (X .LT. ONEP5) THEN
 IF (X .LT. HALF) THEN
   ALNGAM = -LOG(X)
   Y = X + ONE
C
C     Test whether X < machine epsilon
C
   IF (Y .EQ. ONE) RETURN
 ELSE
   ALNGAM = ZERO
   Y = X
   X = (X - HALF) - HALF
 END IF
 ALNGAM = ALNGAM + X * ((((R1(5)*Y + R1(4))*Y + R1(3))*Y
     +                      + R1(2))*Y + R1(1)) / ((((Y + R1(9))*Y +
R1(8))*Y
     +                + R1(7))*Y + R1(6))
 RETURN
      END IF
C
C     Calculation for 1.5 <= X < 4.0
C
      IF (X .LT. FOUR) THEN
 Y = (X - ONE) - ONE
 ALNGAM = Y * ((((R2(5)*X + R2(4))*X + R2(3))*X + R2(2))*X
     +              + R2(1)) / ((((X + R2(9))*X + R2(8))*X + R2(7))*X
     +              + R2(6))
 RETURN
      END IF
C
C     Calculation for 4.0 <= X < 12.0
C
      IF (X .LT. TWELVE) THEN
 ALNGAM = ((((R3(5)*X + R3(4))*X + R3(3))*X + R3(2))*X + R3(1)) /
     +            ((((X + R3(9))*X + R3(8))*X + R3(7))*X + R3(6))
 RETURN
      END IF
C
C     Calculation for X >= 12.0
C
      Y = LOG(X)
      ALNGAM = X * (Y - ONE) - HALF * Y + ALR2PI
      IF (X .GT. XLGE) RETURN
      X1 = ONE / X
      X2 = X1 * X1
      ALNGAM = ALNGAM + X1 * ((R4(3)*X2 + R4(2))*X2 + R4(1)) /
     +              ((X2 + R4(5))*X2 + R4(4))
```

```
      RETURN
      END
c
      FUNCTION GAMMLN(XX)
      REAL*8 COF(6),STP,HALF,ONE,FPF,X,TMP,SER
      DATA COF,STP/76.18009173D0,-86.50532033D0,24.01409822D0,
     *    -1.231739516D0,.120858003D-2,-.536382D-5,2.50662827465D0/
      DATA HALF,ONE,FPF/0.5D0,1.0D0,5.5D0/
      X=XX-ONE
      TMP=X+FPF
      TMP=(X+HALF)*LOG(TMP)-TMP
      SER=ONE
      DO 11 J=1,6
        X=X+ONE
        SER=SER+COF(J)/X
11    CONTINUE
      GAMMLN=TMP+LOG(STP*SER)
      RETURN
      END
```

APPENDIX B

SIMULATED FILES

B.I: Predictor Matrix for 200 cases (randomly selected from a run)

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
-1	-1	-1	-1	1	-1	0	0	0	-1
-1	-1	-1	-1	1	1	-1	0	0	-1
-1	-1	1	-1	1	-1	0	-1	1	0
-1	-1	1	-1	1	-1	1	-1	1	1
-1	-1	1	1	-1	1	0	0	0	0
-1	1	-1	-1	-1	-1	1	-1	1	0
-1	1	-1	-1	-1	1	0	1	-1	1
-1	1	-1	-1	1	-1	0	0	0	1
-1	1	-1	-1	1	1	1	0	1	-1
-1	1	-1	1	-1	-1	0	0	-1	0
-1	1	-1	1	-1	1	-1	1	1	-1
-1	1	-1	1	1	-1	0	-1	-1	1
-1	1	1	-1	-1	1	-1	-1	-1	-1
-1	1	1	1	-1	1	-1	1	-1	0
-1	1	1	1	-1	1	0	-1	0	0
-1	1	1	1	1	-1	1	0	0	-1
-1	1	1	1	1	1	-1	1	1	-1
1	-1	-1	-1	1	-1	-1	0	1	1
1	-1	-1	-1	1	1	1	0	-1	0
1	-1	-1	1	-1	1	0	0	1	0
1	-1	-1	1	1	1	-1	1	0	-1
1	-1	1	-1	1	-1	0	-1	-1	0
1	-1	1	1	-1	-1	1	0	0	0
1	-1	1	1	1	-1	-1	1	0	0
1	-1	1	1	1	1	-1	0	1	1
1	1	-1	1	-1	1	0	-1	1	-1
1	1	1	1	-1	-1	-1	0	-1	0
-1	-1	-1	-1	-1	-1	-1	0	1	-1
-1	-1	-1	1	-1	-1	0	-1	-1	0
-1	-1	-1	1	-1	-1	0	-1	0	0
-1	-1	-1	1	-1	-1	1	1	1	1
-1	-1	-1	1	1	-1	-1	0	-1	-1
-1	-1	-1	1	1	-1	0	-1	-1	1

-1	-1	1	-1	-1	-1	1	0	-1	-1
-1	-1	1	-1	-1	1	1	-1	-1	-1
-1	-1	1	-1	1	-1	0	-1	-1	1
-1	-1	1	-1	1	-1	1	-1	-1	0
-1	-1	1	-1	1	1	-1	0	0	1
-1	-1	1	1	-1	1	1	1	0	0
-1	-1	1	1	1	1	-1	1	0	0
-1	-1	1	1	1	1	1	0	1	1
-1	1	-1	-1	-1	-1	1	0	-1	1
-1	1	-1	-1	-1	-1	1	1	-1	-1
-1	1	-1	-1	1	-1	1	-1	1	0
-1	1	-1	-1	1	-1	1	1	0	-1
-1	1	-1	-1	1	1	-1	0	-1	1
-1	1	-1	-1	1	1	1	1	0	0
-1	1	-1	1	-1	1	0	-1	1	0
-1	1	-1	1	1	-1	0	1	0	0
-1	1	1	-1	-1	-1	1	1	1	1
-1	1	1	-1	-1	1	0	-1	-1	1
-1	1	1	-1	1	-1	-1	-1	0	-1
-1	1	1	-1	1	1	-1	-1	0	-1
-1	1	1	-1	1	1	-1	0	0	1
-1	1	1	-1	1	1	1	0	1	0
-1	1	1	-1	1	1	1	1	1	0
-1	1	1	1	-1	1	-1	1	0	1
-1	1	1	1	-1	1	1	-1	-1	0
-1	1	1	1	-1	1	1	0	1	1
-1	1	1	1	1	-1	0	-1	1	0
-1	1	1	1	1	1	-1	0	1	0
1	-1	-1	-1	-1	1	-1	0	1	-1
1	-1	-1	-1	1	1	0	0	-1	-1
1	-1	-1	1	1	1	-1	1	0	0
1	-1	-1	1	1	1	0	0	0	1
1	-1	-1	1	1	1	1	0	1	0
1	-1	1	-1	-1	-1	-1	1	-1	-1
1	-1	1	-1	-1	1	0	0	1	0
1	-1	1	-1	1	-1	0	-1	-1	1
1	-1	1	1	-1	-1	0	0	-1	1
1	-1	1	1	-1	-1	1	1	0	1
1	-1	1	1	-1	1	1	-1	0	1
1	-1	1	1	1	1	0	0	-1	-1
1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	1	-1	-1	-1	-1	0	0	-1	-1
1	1	-1	-1	-1	-1	1	0	-1	0
1	1	-1	-1	-1	-1	1	0	0	1
1	1	-1	-1	1	-1	-1	-1	0	-1
1	1	-1	-1	1	-1	0	-1	1	1
1	1	-1	-1	1	-1	0	0	0	0
1	1	-1	-1	1	-1	0	1	1	1
1	1	-1	-1	1	-1	1	0	1	1
1	1	-1	1	-1	-1	-1	1	0	1
1	1	-1	1	-1	-1	0	-1	0	0
1	1	-1	1	-1	1	0	-1	0	1
1	1	-1	1	1	-1	0	1	-1	1
1	1	-1	1	1	1	0	0	1	1
1	1	1	-1	-1	-1	0	1	1	-1
1	1	1	-1	-1	1	0	-1	1	0
1	1	1	-1	1	-1	0	1	-1	0
1	1	1	-1	1	-1	1	-1	-1	1
1	1	1	-1	1	1	-1	0	1	0
1	1	1	1	-1	-1	0	-1	1	-1
1	1	1	1	-1	-1	0	0	0	-1
1	1	1	1	-1	1	-1	0	1	0

1	1	1	1	-1	1	0	-1	0	0
1	1	1	1	1	-1	-1	0	0	1
1	1	1	1	1	-1	-1	1	1	0
1	1	1	1	1	1	-1	-1	1	0
1	1	1	1	1	1	0	1	0	-1
-1	-1	-1	-1	-1	1	-1	-1	0	1
-1	-1	-1	-1	-1	1	-1	-1	1	-1
-1	-1	-1	-1	1	-1	-1	1	-1	0
-1	-1	-1	-1	1	1	1	0	-1	1
-1	-1	-1	1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	1	-1	-1	0	1	0
-1	-1	-1	1	1	-1	0	1	-1	0
-1	-1	-1	1	1	-1	1	0	1	0
-1	-1	-1	1	1	-1	1	1	0	1
-1	-1	1	-1	1	1	-1	0	-1	1
-1	-1	1	-1	1	1	-1	1	0	-1
-1	-1	1	1	-1	1	1	1	-1	-1
-1	-1	1	1	1	-1	0	0	1	0
-1	-1	1	1	1	-1	1	1	1	-1
-1	1	-1	-1	-1	-1	1	-1	0	1
-1	1	-1	-1	-1	-1	1	1	1	0
-1	1	-1	-1	-1	1	-1	1	-1	1
-1	1	-1	-1	-1	1	0	1	1	1
-1	1	-1	-1	1	-1	1	1	1	0
-1	1	-1	-1	1	1	0	-1	1	1
-1	1	-1	-1	1	1	0	1	0	-1
-1	1	-1	-1	1	1	1	-1	0	1
-1	1	-1	1	-1	-1	-1	1	1	1
-1	1	-1	1	-1	-1	1	0	-1	0
-1	1	1	-1	-1	-1	0	0	1	0
-1	1	1	-1	-1	1	0	1	1	1
-1	1	1	-1	-1	1	1	0	0	0
-1	1	1	-1	-1	1	1	1	0	-1
-1	1	1	-1	1	-1	-1	-1	0	0
-1	1	1	-1	1	-1	0	1	1	-1
-1	1	1	-1	1	1	1	-1	0	-1
-1	1	1	1	-1	1	1	0	-1	0
-1	1	1	1	1	1	-1	1	-1	-1
1	-1	-1	-1	-1	-1	-1	0	0	-1
1	-1	-1	-1	-1	1	1	0	-1	-1
1	-1	-1	-1	1	-1	1	1	-1	-1
1	-1	-1	-1	1	1	-1	1	-1	0
1	-1	-1	-1	1	1	-1	1	0	0
1	-1	-1	-1	1	1	1	1	1	1
1	-1	-1	1	-1	-1	1	-1	1	1
1	-1	-1	1	-1	-1	1	0	0	-1
1	-1	-1	1	-1	-1	1	0	0	0
1	-1	1	-1	-1	1	0	-1	0	1
1	-1	1	-1	-1	1	1	-1	-1	-1
1	-1	1	-1	1	-1	-1	-1	1	-1
1	-1	1	-1	1	-1	0	0	-1	-1
1	-1	1	-1	1	-1	0	1	-1	-1
1	-1	1	-1	1	-1	0	1	0	0
1	-1	1	1	-1	-1	-1	0	0	-1
1	-1	1	1	-1	-1	-1	0	1	-1
1	-1	1	1	-1	-1	1	1	-1	0
1	-1	1	1	-1	1	-1	-1	-1	-1
1	-1	1	1	-1	1	0	0	-1	1
1	-1	1	1	1	-1	0	1	-1	0
1	-1	1	1	1	-1	1	-1	1	-1
1	1	-1	-1	-1	-1	0	-1	-1	-1
1	1	-1	-1	-1	1	0	0	-1	1

1	1	-1	-1	-1	1	1	0	0	0
1	1	-1	1	-1	1	1	0	-1	0
1	1	1	-1	-1	-1	0	1	0	0
1	1	1	-1	-1	1	-1	0	1	0
1	1	1	-1	-1	1	0	0	-1	0
1	1	1	-1	-1	1	1	0	-1	-1
1	1	1	-1	-1	1	1	1	0	0
1	1	1	-1	1	1	-1	-1	1	-1
1	1	1	-1	1	1	0	1	0	0
1	1	1	-1	1	1	1	-1	0	1
1	1	1	-1	1	1	1	0	-1	0
1	1	1	1	-1	-1	0	0	1	-1
1	1	1	1	-1	-1	1	-1	0	-1
1	1	1	1	1	-1	0	-1	-1	-1
1	1	1	1	1	1	-1	0	-1	1
1	1	1	1	1	1	1	0	1	1
-1	-1	-1	-1	-1	-1	0	0	1	0
-1	-1	-1	-1	-1	1	1	-1	0	0
-1	-1	-1	-1	1	-1	0	-1	-1	1
-1	-1	-1	1	-1	-1	-1	0	1	1
-1	-1	-1	1	-1	-1	-1	1	0	0
-1	-1	1	1	-1	-1	0	-1	0	-1
-1	-1	1	1	1	-1	-1	-1	1	1
-1	-1	1	1	1	1	-1	0	-1	1
-1	-1	1	1	1	1	-1	0	0	1
-1	1	-1	1	1	-1	-1	0	0	1
-1	1	-1	1	1	1	0	0	1	0
-1	1	1	1	-1	-1	-1	1	0	0
-1	1	1	1	-1	1	0	0	0	0
-1	1	1	1	-1	1	1	-1	0	-1
1	-1	-1	-1	1	1	0	0	-1	0
1	-1	-1	1	1	1	0	0	1	1
1	-1	1	-1	-1	1	1	-1	0	0
1	-1	1	-1	1	1	1	0	1	1
1	1	-1	-1	1	-1	1	-1	0	0
1	1	-1	-1	1	-1	1	1	0	1
1	1	-1	-1	1	1	1	0	1	-1
1	1	-1	1	-1	1	0	1	0	-1
1	1	-1	1	1	-1	-1	0	-1	1
1	1	-1	1	1	1	-1	-1	-1	1
1	1	-1	1	1	1	1	-1	0	1
1	1	1	-1	-1	-1	0	0	1	0
1	1	1	1	-1	1	-1	1	-1	0

B.II: Detailed Simulation Results for Table 3.1.1

for β_0=0.09, p=045, 0.5 and 0.55

for β_0=0.436, p=045, 0.5 and 0.55

for β_0=0.872, p=045, 0.5 and 0.55

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	135	165	51	124	176
2	141	159	52	150	150
3	126	174	53	127	173
4	127	173	54	137	163
5	137	163	55	134	166
6	134	166	56	130	170
7	130	170	57	130	170
8	130	170	58	124	176
9	123	177	59	150	150
10	150	150	60	127	173
11	127	173	61	137	163
12	137	163	62	134	166
13	134	166	63	130	170
14	130	170	64	130	170
15	130	170	65	123	177
16	124	176	66	150	150
17	150	150	67	127	173
18	127	173	68	137	163
19	137	163	69	134	166
20	134	166	70	130	170
21	130	170	71	130	170
22	130	170	72	123	177
23	123	177	73	150	150
24	150	150	74	127	173
25	127	173	75	137	163
26	137	163	76	134	166
27	134	166	77	130	170
28	130	170	78	130	170
29	130	170	79	123	177
30	124	176	80	150	150
31	150	150	81	127	173
32	127	173	82	137	163
33	137	163	83	134	166
34	134	166	84	130	170
35	130	170	85	130	170
36	130	170	86	123	177
37	124	176	87	150	150
38	150	150	88	127	173
39	127	173	89	137	163
40	137	163	90	134	166
41	134	166	91	130	170
42	130	170	92	130	170
43	130	170	93	123	177
44	123	177	94	150	150
45	150	150	95	127	173
46	127	173	96	137	163
47	137	163	97	134	166
48	134	166	98	130	170
49	130	170	99	130	170
50	130	170	100	124	176

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	70	230	51	88	212
2	82	218	52	77	223
3	92	208	53	79	221
4	79	221	54	82	218
5	82	218	55	75	225
6	75	225	56	64	236
7	64	236	57	54	246
8	54	246	58	88	212
9	88	212	59	77	223
10	77	223	60	79	221
11	79	221	61	82	218
12	82	218	62	75	225
13	75	225	63	64	236
14	64	236	64	54	246
15	54	246	65	88	212
16	88	212	66	77	223
17	77	223	67	79	221
18	79	221	68	82	218
19	82	218	69	75	225
20	75	225	70	64	236
21	64	236	71	54	246
22	54	246	72	88	212
23	88	212	73	77	223
24	77	223	74	79	221
25	79	221	75	82	218
26	82	218	76	75	225
27	75	225	77	64	236
28	64	236	78	54	246
29	54	246	79	88	212
30	88	212	80	77	223
31	77	223	81	79	221
32	79	221	82	82	218
33	82	218	83	75	225
34	75	225	84	64	236
35	64	236	85	54	246
36	54	246	86	88	212
37	88	212	87	77	223
38	77	223	88	79	221
39	79	221	89	82	218
40	82	218	90	75	225
41	75	225	91	64	236
42	64	236	92	54	246
43	54	246	93	88	212
44	88	212	94	77	223
45	77	223	95	79	221
46	79	221	96	82	218
47	82	218	97	75	225
48	75	225	98	64	236
49	64	236	99	54	246
50	54	246	100	88	212

$\mathbf{p_1}$ = 0.5

$\mathbf{p_2}$ = 0.45

β_0 = 0.436

β_0 = 0.09

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	134	166	51	137	163
2	124	176	52	122	178
3	126	174	53	131	169
4	112	188	54	129	171
5	138	162	55	126	174
6	125	175	56	128	172
7	139	161	57	122	178
8	121	179	58	140	160
9	128	172	59	139	161
10	137	163	60	128	172
11	122	178	61	114	186
12	131	169	62	142	158
13	129	171	63	116	184
14	126	174	64	147	153
15	128	172	65	121	179
16	122	178	66	156	144
17	140	160	67	136	164
18	139	161	68	135	165
19	128	172	69	121	179
20	114	186	70	116	184
21	142	158	71	116	184
22	116	184	72	147	153
23	147	153	73	133	167
24	121	179	74	137	163
25	147	153	75	118	182
26	137	163	76	137	163
27	138	162	77	132	168
28	118	182	78	125	175
29	135	165	79	146	154
30	152	148	80	145	155
31	138	162	81	126	174
32	128	172	82	122	178
33	136	164	83	139	161
34	129	171	84	106	194
35	130	170	85	135	165
36	135	165	86	141	159
37	134	166	87	126	174
38	137	163	88	129	171
39	125	175	89	143	157
40	146	154	90	140	160
41	145	155	91	144	156
42	126	174	92	143	157
43	122	178	93	141	159
44	139	161	94	145	155
45	106	194	95	139	161
46	135	165	96	138	162
47	141	159	97	134	166
48	139	161	98	134	166
49	121	179	99	156	144
50	128	172	100	125	175

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	75	225	51	85	215
2	62	238	52	79	221
3	78	222	53	54	246
4	63	237	54	78	222
5	64	236	55	66	234
6	74	226	56	93	207
7	81	219	57	86	214
8	68	232	58	77	223
9	81	219	59	82	218
10	85	215	60	83	217
11	79	221	61	72	228
12	54	246	62	78	222
13	78	222	63	82	218
14	66	234	64	88	212
15	93	207	65	78	222
16	86	214	66	65	235
17	77	223	67	76	224
18	82	218	68	49	251
19	83	217	69	67	233
20	72	228	70	75	225
21	78	222	71	83	217
22	82	218	72	70	230
23	88	212	73	73	227
24	78	222	74	83	217
25	74	226	75	69	231
26	63	237	76	87	213
27	78	222	77	86	214
28	71	229	78	76	224
29	75	225	79	74	226
30	76	224	80	66	234
31	82	218	81	75	225
32	69	231	82	74	226
33	74	226	83	65	235
34	79	221	84	68	232
35	62	238	85	70	230
36	80	220	86	94	206
37	81	219	87	74	226
38	92	208	88	69	231
39	76	224	89	88	212
40	74	226	90	76	224
41	66	234	91	85	215
42	75	225	92	78	222
43	74	226	93	85	215
44	65	235	94	87	213
45	68	232	95	77	223
46	70	230	96	86	214
47	84	216	97	67	233
48	81	219	98	87	213
49	68	232	99	79	221
50	81	219	100	70	230

$\mathbf{p_1}$ = 0.5

$\mathbf{p_2}$ = 0.5

β_0 = 0.436

β_0 = 0.09

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	132	168	51	116	184
2	120	180	52	124	176
3	138	162	53	109	191
4	113	187	54	141	159
5	140	160	55	134	166
6	136	164	56	138	162
7	118	182	57	125	175
8	128	172	58	130	170
9	134	166	59	128	172
10	127	173	60	142	158
11	127	173	61	126	174
12	140	160	62	126	174
13	143	157	63	138	162
14	138	162	64	134	166
15	135	165	65	138	162
16	132	168	66	125	175
17	120	180	67	130	170
18	124	176	68	128	172
19	109	191	69	142	158
20	141	159	70	126	174
21	134	166	71	130	170
22	138	162	72	128	172
23	125	175	73	109	191
24	130	170	74	132	168
25	129	171	75	132	168
26	126	174	76	134	166
27	135	165	77	123	177
28	136	164	78	124	176
29	126	174	79	109	191
30	140	160	80	144	156
31	136	164	81	120	180
32	118	182	82	131	169
33	128	172	83	133	167
34	134	166	84	139	161
35	127	173	85	126	174
36	127	173	86	116	184
37	140	160	87	124	176
38	143	157	88	109	191
39	138	162	89	140	160
40	135	165	90	134	166
41	145	155	91	138	162
42	111	189	92	125	175
43	124	176	93	130	170
44	109	191	94	130	170
45	144	156	95	134	166
46	120	180	96	127	173
47	131	169	97	140	160
48	133	167	98	136	164
49	139	161	99	118	182
50	126	174	100	128	172

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	77	223	51	63	237
2	80	220	52	70	230
3	95	205	53	62	238
4	74	226	54	72	228
5	76	224	55	63	237
6	89	211	56	82	218
7	76	224	57	77	223
8	68	232	58	71	229
9	77	223	59	60	240
10	75	225	60	71	229
11	60	240	61	79	221
12	79	221	62	73	227
13	88	212	63	68	232
14	77	223	64	63	237
15	74	226	65	82	218
16	80	220	66	77	223
17	66	234	67	71	229
18	70	230	68	60	240
19	62	238	69	71	229
20	72	228	70	79	221
21	63	237	71	70	230
22	82	218	72	65	235
23	77	223	73	74	226
24	71	229	74	73	227
25	66	234	75	57	243
26	64	236	76	74	226
27	69	231	77	80	220
28	74	226	78	70	230
29	63	237	79	62	238
30	76	224	80	65	235
31	89	211	81	77	223
32	76	224	82	71	229
33	68	232	83	78	222
34	77	223	84	62	238
35	75	225	85	84	216
36	60	240	86	63	237
37	79	221	87	70	230
38	88	212	88	62	238
39	77	223	89	73	227
40	74	226	90	63	237
41	76	224	91	82	218
42	69	231	92	77	223
43	70	230	93	71	229
44	62	238	94	73	227
45	65	235	95	67	233
46	77	223	96	74	226
47	71	229	97	76	224
48	79	221	98	89	211
49	62	238	99	76	224
50	84	216	100	68	232

$\mathbf{p_1} = 0.5$

$\mathbf{p_2} = 0.55$

$\beta_0 = 0.436$

$\beta_0 = 0.09$

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	135	165	51	124	176
2	141	159	52	150	150
3	126	174	53	127	173
4	127	173	54	137	163
5	137	163	55	134	166
6	134	166	56	130	170
7	130	170	57	130	170
8	130	170	58	124	176
9	123	177	59	150	150
10	150	150	60	127	173
11	127	173	61	137	163
12	137	163	62	134	166
13	134	166	63	130	170
14	130	170	64	130	170
15	130	170	65	123	177
16	124	176	66	150	150
17	150	150	67	127	173
18	127	173	68	137	163
19	137	163	69	134	166
20	134	166	70	130	170
21	130	170	71	130	170
22	130	170	72	123	177
23	123	177	73	150	150
24	150	150	74	127	173
25	127	173	75	137	163
26	137	163	76	134	166
27	134	166	77	130	170
28	130	170	78	130	170
29	130	170	79	123	177
30	124	176	80	150	150
31	150	150	81	127	173
32	127	173	82	137	163
33	137	163	83	134	166
34	134	166	84	130	170
35	130	170	85	130	170
36	130	170	86	123	177
37	124	176	87	150	150
38	150	150	88	127	173
39	127	173	89	137	163
40	137	163	90	134	166
41	134	166	91	130	170
42	130	170	92	130	170
43	130	170	93	123	177
44	123	177	94	150	150
45	150	150	95	127	173
46	127	173	96	137	163
47	137	163	97	134	166
48	134	166	98	130	170
49	130	170	99	130	170
50	130	170	100	124	176

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	88	212	51	100	200
2	99	201	52	94	206
3	106	194	53	91	209
4	91	209	54	100	200
5	100	200	55	86	214
6	86	214	56	84	216
7	84	216	57	74	226
8	74	226	58	100	200
9	100	200	59	94	206
10	94	206	60	91	209
11	91	209	61	100	200
12	100	200	62	86	214
13	86	214	63	84	216
14	84	216	64	74	226
15	74	226	65	100	200
16	100	200	66	94	206
17	94	206	67	91	209
18	91	209	68	100	200
19	100	200	69	86	214
20	86	214	70	84	216
21	84	216	71	74	226
22	74	226	72	100	200
23	100	200	73	94	206
24	94	206	74	91	209
25	91	209	75	100	200
26	100	200	76	86	214
27	86	214	77	84	216
28	84	216	78	74	226
29	74	226	79	100	200
30	100	200	80	94	206
31	94	206	81	91	209
32	91	209	82	100	200
33	100	200	83	86	214
34	86	214	84	84	216
35	84	216	85	74	226
36	74	226	86	100	200
37	100	200	87	94	206
38	94	206	88	91	209
39	91	209	89	100	200
40	100	200	90	86	214
41	86	214	91	84	216
42	84	216	92	74	226
43	74	226	93	100	200
44	100	200	94	94	206
45	94	206	95	91	209
46	91	209	96	100	200
47	100	200	97	86	214
48	86	214	98	84	216
49	84	216	99	74	226
50	74	226	100	100	200

$\mathbf{p_1} = 0.5$

$\mathbf{p_2} = 0.45$

$\beta_{\mathbf{0}} = 0.436$

$\beta_{\mathbf{0}} = 0.436$

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	134	166	51	137	163
2	124	176	52	122	178
3	126	174	53	131	169
4	112	188	54	129	171
5	138	162	55	126	174
6	125	175	56	128	172
7	139	161	57	122	178
8	121	179	58	140	160
9	128	172	59	139	161
10	137	163	60	128	172
11	122	178	61	114	186
12	131	169	62	142	158
13	129	171	63	116	184
14	126	174	64	147	153
15	128	172	65	121	179
16	122	178	66	156	144
17	140	160	67	136	164
18	139	161	68	135	165
19	128	172	69	121	179
20	114	186	70	116	184
21	142	158	71	116	184
22	116	184	72	147	153
23	147	153	73	133	167
24	121	179	74	137	163
25	147	153	75	118	182
26	137	163	76	137	163
27	138	162	77	132	168
28	118	182	78	125	175
29	135	165	79	146	154
30	152	148	80	145	155
31	138	162	81	126	174
32	128	172	82	122	178
33	136	164	83	139	161
34	129	171	84	106	194
35	130	170	85	135	165
36	135	165	86	141	159
37	134	166	87	126	174
38	137	163	88	129	171
39	125	175	89	143	157
40	146	154	90	140	160
41	145	155	91	144	156
42	126	174	92	143	157
43	122	178	93	141	159
44	139	161	94	145	155
45	106	194	95	139	161
46	135	165	96	138	162
47	141	159	97	134	166
48	139	161	98	134	166
49	121	179	99	156	144
50	128	172	100	125	175

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	100	200	51	94	206
2	79	221	52	95	205
3	97	203	53	68	232
4	83	217	54	98	202
5	80	220	55	86	214
6	95	205	56	110	190
7	102	198	57	101	199
8	88	212	58	95	205
9	99	201	59	98	202
10	94	206	60	101	199
11	95	205	61	97	203
12	68	232	62	93	207
13	98	202	63	99	201
14	86	214	64	100	200
15	110	190	65	90	210
16	101	199	66	84	216
17	95	205	67	99	201
18	98	202	68	64	236
19	101	199	69	87	213
20	97	203	70	93	207
21	93	207	71	95	205
22	99	201	72	85	215
23	100	200	73	93	207
24	90	210	74	94	206
25	92	208	75	89	211
26	78	222	76	98	202
27	98	202	77	107	193
28	90	210	78	88	212
29	90	210	79	92	208
30	95	205	80	87	213
31	102	198	81	98	202
32	96	204	82	86	214
33	91	209	83	81	219
34	98	202	84	85	215
35	78	222	85	93	207
36	94	206	86	112	188
37	98	202	87	89	211
38	105	195	88	85	215
39	88	212	89	112	188
40	92	208	90	87	213
41	87	213	91	103	197
42	98	202	92	100	200
43	86	214	93	103	197
44	81	219	94	108	192
45	85	215	95	93	207
46	93	207	96	101	199
47	97	203	97	87	213
48	102	198	98	101	199
49	88	212	99	102	198
50	99	201	100	90	210

$\mathbf{p_1}$ = 0.5

$\mathbf{p_2}$ = 0.5

β_0 = 0.436

β_0 = 0.436

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	132	168	51	116	184
2	120	180	52	124	176
3	138	162	53	109	191
4	113	187	54	141	159
5	140	160	55	134	166
6	136	164	56	138	162
7	118	182	57	125	175
8	128	172	58	130	170
9	134	166	59	128	172
10	127	173	60	142	158
11	127	173	61	126	174
12	140	160	62	126	174
13	143	157	63	138	162
14	138	162	64	134	166
15	135	165	65	138	162
16	132	168	66	125	175
17	120	180	67	130	170
18	124	176	68	128	172
19	109	191	69	142	158
20	141	159	70	126	174
21	134	166	71	130	170
22	138	162	72	128	172
23	125	175	73	109	191
24	130	170	74	132	168
25	129	171	75	132	168
26	126	174	76	134	166
27	135	165	77	123	177
28	136	164	78	124	176
29	126	174	79	109	191
30	140	160	80	144	156
31	136	164	81	120	180
32	118	182	82	131	169
33	128	172	83	133	167
34	134	166	84	139	161
35	127	173	85	126	174
36	127	173	86	116	184
37	140	160	87	124	176
38	143	157	88	109	191
39	138	162	89	140	160
40	135	165	90	134	166
41	145	155	91	138	162
42	111	189	92	125	175
43	124	176	93	130	170
44	109	191	94	130	170
45	144	156	95	134	166
46	120	180	96	127	173
47	131	169	97	140	160
48	133	167	98	136	164
49	139	161	99	118	182
50	126	174	100	128	172

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	95	205	51	86	214
2	96	204	52	93	207
3	114	186	53	79	221
4	92	208	54	93	207
5	104	196	55	69	231
6	104	196	56	95	205
7	99	201	57	95	205
8	84	216	58	90	210
9	96	204	59	85	215
10	88	212	60	96	204
11	79	221	61	94	206
12	95	205	62	85	215
13	103	197	63	87	213
14	92	208	64	69	231
15	88	212	65	95	205
16	98	202	66	95	205
17	82	218	67	90	210
18	93	207	68	85	215
19	79	221	69	96	204
20	93	207	70	94	206
21	69	231	71	88	212
22	95	205	72	84	216
23	95	205	73	94	206
24	90	210	74	90	210
25	89	211	75	70	230
26	86	214	76	89	211
27	86	214	77	94	206
28	88	212	78	93	207
29	79	221	79	79	221
30	104	196	80	83	217
31	104	196	81	95	205
32	99	201	82	94	206
33	84	216	83	101	199
34	96	204	84	78	222
35	88	212	85	100	200
36	79	221	86	86	214
37	95	205	87	93	207
38	103	197	88	79	221
39	92	208	89	93	207
40	88	212	90	69	231
41	99	201	91	95	205
42	85	215	92	95	205
43	93	207	93	90	210
44	79	221	94	91	209
45	83	217	95	78	222
46	95	205	96	86	214
47	94	206	97	104	196
48	101	199	98	104	196
49	78	222	99	99	201
50	100	200	100	84	216

$\mathbf{p_1} = 0.5$

$\mathbf{p_2} = 0.55$

$\beta_0 = 0.436$

$\beta_0 = 0.436$

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	135	165	51	124	176
2	141	159	52	150	150
3	126	174	53	127	173
4	127	173	54	137	163
5	137	163	55	134	166
6	134	166	56	130	170
7	130	170	57	130	170
8	130	170	58	124	176
9	123	177	59	150	150
10	150	150	60	127	173
11	127	173	61	137	163
12	137	163	62	134	166
13	134	166	63	130	170
14	130	170	64	130	170
15	130	170	65	123	177
16	124	176	66	150	150
17	150	150	67	127	173
18	127	173	68	137	163
19	137	163	69	134	166
20	134	166	70	130	170
21	130	170	71	130	170
22	130	170	72	123	177
23	123	177	73	150	150
24	150	150	74	127	173
25	127	173	75	137	163
26	137	163	76	134	166
27	134	166	77	130	170
28	130	170	78	130	170
29	130	170	79	123	177
30	124	176	80	150	150
31	150	150	81	127	173
32	127	173	82	137	163
33	137	163	83	134	166
34	134	166	84	130	170
35	130	170	85	130	170
36	130	170	86	123	177
37	124	176	87	150	150
38	150	150	88	127	173
39	127	173	89	137	163
40	137	163	90	134	166
41	134	166	91	130	170
42	130	170	92	130	170
43	130	170	93	123	177
44	123	177	94	150	150
45	150	150	95	127	173
46	127	173	96	137	163
47	137	163	97	134	166
48	134	166	98	130	170
49	130	170	99	130	170
50	130	170	100	124	176

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	109	191	51	120	180
2	119	181	52	119	181
3	125	175	53	116	184
4	116	184	54	127	173
5	127	173	55	113	187
6	113	187	56	111	189
7	111	189	57	95	205
8	95	205	58	120	180
9	121	179	59	119	181
10	119	181	60	116	184
11	116	184	61	127	173
12	127	173	62	113	187
13	113	187	63	111	189
14	111	189	64	95	205
15	95	205	65	121	179
16	120	180	66	119	181
17	119	181	67	116	184
18	116	184	68	127	173
19	127	173	69	113	187
20	113	187	70	111	189
21	111	189	71	95	205
22	95	205	72	121	179
23	121	179	73	119	181
24	119	181	74	116	184
25	116	184	75	127	173
26	127	173	76	113	187
27	113	187	77	111	189
28	111	189	78	95	205
29	95	205	79	121	179
30	120	180	80	119	181
31	119	181	81	116	184
32	116	184	82	127	173
33	127	173	83	113	187
34	113	187	84	111	189
35	111	189	85	95	205
36	95	205	86	121	179
37	120	180	87	119	181
38	119	181	88	116	184
39	116	184	89	127	173
40	127	173	90	113	187
41	113	187	91	111	189
42	111	189	92	95	205
43	95	205	93	121	179
44	121	179	94	119	181
45	119	181	95	116	184
46	116	184	96	127	173
47	127	173	97	113	187
48	113	187	98	111	189
49	111	189	99	95	205
50	95	205	100	120	180

$\mathbf{p_1}$ = 0.5

$\mathbf{p_2}$ = 0.45

$\boldsymbol{\beta_0}$ = 0.436

$\boldsymbol{\beta_0}$ = 0.872

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	134	166	51	137	163
2	124	176	52	122	178
3	126	174	53	131	169
4	112	188	54	129	171
5	138	162	55	126	174
6	125	175	56	128	172
7	139	161	57	122	178
8	121	179	58	140	160
9	128	172	59	139	161
10	137	163	60	128	172
11	122	178	61	114	186
12	131	169	62	142	158
13	129	171	63	116	184
14	126	174	64	147	153
15	128	172	65	121	179
16	122	178	66	156	144
17	140	160	67	136	164
18	139	161	68	135	165
19	128	172	69	121	179
20	114	186	70	116	184
21	142	158	71	116	184
22	116	184	72	147	153
23	147	153	73	133	167
24	121	179	74	137	163
25	147	153	75	118	182
26	137	163	76	137	163
27	138	162	77	132	168
28	118	182	78	125	175
29	135	165	79	146	154
30	152	148	80	145	155
31	138	162	81	126	174
32	128	172	82	122	178
33	136	164	83	139	161
34	129	171	84	106	194
35	130	170	85	135	165
36	135	165	86	141	159
37	134	166	87	126	174
38	137	163	88	129	171
39	125	175	89	143	157
40	146	154	90	140	160
41	145	155	91	144	156
42	126	174	92	143	157
43	122	178	93	141	159
44	139	161	94	145	155
45	106	194	95	139	161
46	135	165	96	138	162
47	141	159	97	134	166
48	139	161	98	134	166
49	121	179	99	156	144
50	128	172	100	125	175

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	125	175	51	112	188
2	101	199	52	118	182
3	124	176	53	95	205
4	109	191	54	121	179
5	105	195	55	111	189
6	121	179	56	128	172
7	125	175	57	119	181
8	113	187	58	126	174
9	123	177	59	123	177
10	112	188	60	127	173
11	118	182	61	121	179
12	95	205	62	122	178
13	121	179	63	117	183
14	111	189	64	120	180
15	128	172	65	113	187
16	119	181	66	119	181
17	126	174	67	126	174
18	123	177	68	89	211
19	127	173	69	111	189
20	121	179	70	120	180
21	122	178	71	114	186
22	117	183	72	110	190
23	120	180	73	121	179
24	113	187	74	120	180
25	122	178	75	109	191
26	110	190	76	123	177
27	121	179	77	129	171
28	106	194	78	112	188
29	109	191	79	114	186
30	122	178	80	123	177
31	125	175	81	121	179
32	119	181	82	103	197
33	113	187	83	114	186
34	128	172	84	102	198
35	105	195	85	119	181
36	120	180	86	136	164
37	116	184	87	112	188
38	126	174	88	108	192
39	112	188	89	131	169
40	114	186	90	108	192
41	123	177	91	122	178
42	121	179	92	118	182
43	103	197	93	125	175
44	114	186	94	132	168
45	102	198	95	112	188
46	119	181	96	114	186
47	121	179	97	109	191
48	125	175	98	126	174
49	113	187	99	130	170
50	123	177	100	112	188

$p_1 = 0.5$

$p_2 = 0.5$

$\beta_0 = 0.436$

$\beta_0 = 0.872$

1980 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	132	168	51	116	184
2	120	180	52	124	176
3	138	162	53	109	191
4	113	187	54	141	159
5	140	160	55	134	166
6	136	164	56	138	162
7	118	182	57	125	175
8	128	172	58	130	170
9	134	166	59	128	172
10	127	173	60	142	158
11	127	173	61	126	174
12	140	160	62	126	174
13	143	157	63	138	162
14	138	162	64	134	166
15	135	165	65	138	162
16	132	168	66	125	175
17	120	180	67	130	170
18	124	176	68	128	172
19	109	191	69	142	158
20	141	159	70	126	174
21	134	166	71	130	170
22	138	162	72	128	172
23	125	175	73	109	191
24	130	170	74	132	168
25	129	171	75	132	168
26	126	174	76	134	166
27	135	165	77	123	177
28	136	164	78	124	176
29	126	174	79	109	191
30	140	160	80	144	156
31	136	164	81	120	180
32	118	182	82	131	169
33	128	172	83	133	167
34	134	166	84	139	161
35	127	173	85	126	174
36	127	173	86	116	184
37	140	160	87	124	176
38	143	157	88	109	191
39	138	162	89	140	160
40	135	165	90	134	166
41	145	155	91	138	162
42	111	189	92	125	175
43	124	176	93	130	170
44	109	191	94	130	170
45	144	156	95	134	166
46	120	180	96	127	173
47	131	169	97	140	160
48	133	167	98	136	164
49	139	161	99	118	182
50	126	174	100	128	172

1970 Codes

Run#	Σ1s	Σ0s	Run#	Σ1s	Σ0s
1	121	179	51	110	190
2	128	172	52	111	189
3	134	166	53	109	191
4	118	182	54	114	186
5	124	176	55	85	215
6	119	181	56	127	173
7	128	172	57	114	186
8	104	196	58	117	183
9	119	181	59	109	191
10	113	187	60	116	184
11	93	207	61	113	187
12	123	177	62	106	194
13	130	170	63	116	184
14	113	187	64	85	215
15	105	195	65	127	173
16	129	171	66	114	186
17	106	194	67	117	183
18	111	189	68	109	191
19	109	191	69	116	184
20	114	186	70	113	187
21	85	215	71	111	189
22	127	173	72	109	191
23	114	186	73	108	192
24	117	183	74	113	187
25	114	186	75	93	207
26	112	188	76	116	184
27	118	182	77	122	178
28	114	186	78	111	189
29	98	202	79	109	191
30	124	176	80	107	193
31	119	181	81	111	189
32	128	172	82	122	178
33	104	196	83	128	172
34	119	181	84	104	196
35	113	187	85	120	180
36	93	207	86	110	190
37	123	177	87	111	189
38	130	170	88	109	191
39	113	187	89	114	186
40	105	195	90	85	215
41	122	178	91	127	173
42	110	190	92	114	186
43	111	189	93	117	183
44	109	191	94	115	185
45	107	193	95	106	194
46	111	189	96	108	192
47	122	178	97	124	176
48	128	172	98	119	181
49	104	196	99	128	172
50	120	180	100	104	196

$\mathbf{p_1} = 0.5$

$\mathbf{p_2} = 0.55$

$\beta_0 = 0.436$

$\beta_0 = 0.872$