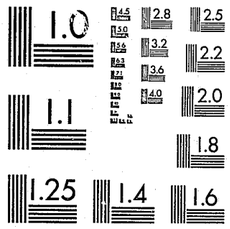

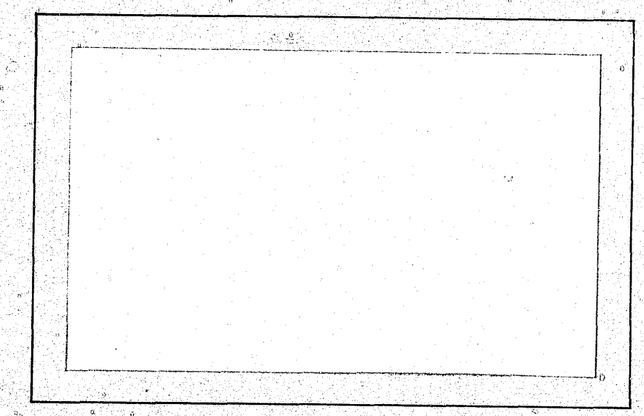

DEPARTMENT
OF
STATISTICS

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

RESTRUCTURING OF THE NCS LONGITUDINAL DATA SET

by

Wm. F. Eddy

S.E. Fienberg

N.S. Prescott

Technical Report No. 274

First Draft

Department of Statistics

Carnegie-Mellon University

Pittsburgh, PA 15213

December, 1982

This material is based in part on work performed under Contract No. 81-IJ-CX-0087 with the National Institute of Justice, Office of Justice Assistance, Research, and Statistics, U.S. Department of Justice. Points of view and opinions stated herein are those of the authors and do not necessarily represent the official position or policies of the U.S. Department of Justice.

1. Introduction

In this memorandum we describe a methodology for converting the National Crime Survey Quarterly Collection Files to longitudinal files. To implement this methodology a special program to convert the quarterly collection files to ASCII files, that can be run on a VAX 11/780, was developed. The program is listed in Appendix A.

2. Logical Structure of the Public Use Files

The National Crime Survey data is made available by the Inter-University Consortium for Political and Social Research (ICPSR) in OSIRIS IV data sets.[1] An OSIRIS IV data set consists of two components, a dictionary to be used in conjunction with OSIRIS IV software and a separate data file. OSIRIS IV software is designed to be used on IBM Hardware and operating systems, so to use the data files on a DEC VAX it is necessary to restructure the files. First, we describe the logical format of OSIRIS IV data files.

These files are stored in a hierarchical structure containing 3 types of records, each of a different length. The three types of records are: 1) Household records, which contain demographic information about the household; 2) Person records, which contain information about each person (over age 12) in the household, and when appropriate, household incident questions; 3) Incident records, which contain details about personal or household victimizations. The structure may be visualized as a tree with the household record as the root. The first level of branches are person records associated with the household. The second level of branches are incident records associated with particular persons within the household.

Each housing unit in the sample is assigned a unique identification number. The tree structures are arranged sequentially in the order of their identification numbers. Within the structure a household record is followed by a person record (if the household has any) which is, in turn, followed by the incident records of that person, if any. This pattern is repeated

[1] OSIRIS IV is the proprietary software package of the Survey Research Center of the Institute for Social Research in Ann Arbor, Michigan.

until all household members and their associated incident records are reported. This is followed by a new structure beginning with a household record. The following is a particular example.

Record #	Record Description		
1	Household 1		
2	Household 1	Person 1	
3	Household 1	Person 1	Incident 1
4	Household 1	Person 2	
5	Household 2		
6	Household 2	Person 1	
7	Household 2	Person 2	
8	Household 2	Person 2	Incident 1
9	Household 2	Person	Incident 2
10	Household 3		

In this example, the first household has 2 individuals, the first individual reported 1 incident. The second household also has 2 individuals and the second individual reported 2 incidents.

3. Physical Structure of the Public Use Files

As discussed above, the OSIRIS IV data set contains three types of records. Each type is a different length. When these records are stored on magnetic tape they are grouped into larger physical records, called blocks. The blocks have approximately constant length, so it is necessary, at times, to break the last logical record, within a block, into two pieces. The first piece is placed at the end of one block, the remaining part is placed at the beginning of the next block. This broken record is said to be "spanned" across the two blocks. Since the blocks do not have a fixed length, but only a maximum length, they are referred to as "variable" length blocks.

Because the survey instrument changed in 1977, ICPSR has divided the data into 2 groups,

(1973–1976) and (1977–Present). In the 1973–1976 period, the household records are 231 characters, the person records are 102 characters, and the incident records are 310 characters. The blocks are not more than 7854 characters in length. In the 1977–present period, the household records are 235 characters, the person records are 118 characters, and the incident records are 328 characters. The blocks are not more than 7924 characters in length.

This discussion is not important for OSIRIS IV users, because OSIRIS IV handles the processing and I/O operations. For non-OSIRIS IV IBM users this discussion becomes more relevant, but IBM operating system software can still operate on the OSIRIS IV data sets. For non-IBM users (including DEC VAX users) this discussion is critical. The OSIRIS IV tape format is not in a form that can necessarily be processed by a non-IBM operating system.

Character data on an IBM machine is usually stored in a code known as EBCDIC (Extended Binary Coded Decimal Interchange Code). Character data on non-IBM machines is usually stored in a code known as ASCII (American Standard Code for Information Interchange). Thus it is necessary to convert the characters in the public use files from EBCDIC to ASCII. Additionally, because the OSIRIS IV tape format cannot necessarily be processed on non-IBM operating systems, it is necessary to reformat the records simultaneously.

On an IBM system, the format of the physical records written in variable blocked spanned format is as follows: the first word (four 8-bit bytes) in each block contains information describing the block, the remainder of the block contains logical records. Each record is preceded by a word of information describing the record. Both block and record descripter words are written in binary form. The first two bytes of a block or record descriptor word contain the length of the block or logical record. This length includes the length of the descriptor word. The second two bytes of a record descriptor word indicate (among other things) if the record is spanned across 2 or more blocks.

Because the block and record descriptor words are written in binary and the data is written in EBCDIC, it is not possible to convert the tape from EBCDIC to ASCII and then reformat

the records. An example will illustrate the format. For illustrative purposes we take the blocksize to be 300 characters. In "Variable Blocked Spanned Format" the first 3 blocks in our previous example (from the 1973–1976 period) would appear on a magnetic tape as follows:

Block 1

	BCW	RCW	Record 1 231 characters	RCW	Record 2 57 characters	
bytes	0	4	8	239	243	300

Block 2

	BCW	RCW	Record 2 45 characters	RCW	Record 3 243 characters	
bytes	0	4	8	53	57	300

Block 3

	BCW	RCW	Record 3 67 char.	RCW	Record 4 102 char.	RCW	Record 5 115 char.	
bytes	0	4	8	75	79	181	185	300

As we can see, the second record is spanned across the first and second block, the third record is spanned across the second and third block and the fifth record is spanned across the third and fourth block, etc.

Now we describe the Fortran program which converts the IBM EBCDIC Variable Blocked Spanned file to an ASCII file. In this program we will read each physical record (block) and break it into logical records for output. When necessary, spanned records must be reunited and output as one logical record. Also, EBCDIC to ASCII translation must be done simultaneously. The format of the output records is determined by the host operating system. In our case this is DEC VAX/VMS. We chose to make our output files conform to the ANSI (American National Standards Institute) format for magnetic tapes. The record format is variable with a maximum logical record length of 328 and a blocksize of 2048. An outline of

the program follows.

1. Initialize

2. Read a physical block

3. For each logical record in the block a. Convert from EBCDIC to ASCII; b. Write the logical record

4. If there is no more data in the block, go to step 2

The actual details are considerably more complicated. In particular, the handling of spanned records requires some care. The program is listed in Appendix A.

4. Hierarchical to Longitudinal Reorganization

The quarterly collection files are, in theory, simply a copy of the data collected during each calendar quarter. There are many practical problems in trying to collate the data for a particular household from the various collection quarters. The fundamental difference between the ICPSR files and longitudinal files is the organization. We have chosen to divide the longitudinal data into 3 separate files; one for the household records, one for the person records, and one for the incident records.

Each of these files may be regarded as a rectangular array with each row being either a household, person, or incident record. The structure of the three longitudinal files for the same three particular households used in the example follow:

Household file

Record #		
1	Household 1	
2	Household 2	
3	Household 3	

Person file

Record #		
1	Household 1	Person 1
2	Household 1	Person 2
3	Household 2	Person 1
4	Household 2	Person 2

Incident file

Record #			
1	Household 1	Person 1	Incident 1
2	Household 2	Person 2	Incident 1
3	Household 2	Person 2	Incident 2

The major advantage of data stored in this type of organization is that it is accessible by all computers which can process ANSI standard magnetic tapes and standard statistical programs because of its rectangular structures. Since the rows of data in each file are stored in the same order in each file (with respect to the internal identification number), cross-referencing among 2 or 3 of the files is possible. Although we can still think of the data in the context of a household-person-incident hierarchy, it is simpler to conceptualize questions and then to design programs to operate on rectangular files.

In most cases a household is interviewed every six months for a total of 7 interviews. At

each interview information is collected about the past six months. The information collected during an interview constitutes a cross-sectional record. For each interview there will be one household record, some person records and if necessary, 1 or more incident records. All of these records for a household collected over the 7 interview period will be collected and referred to as a longitudinal record. Actually, this record consists of three longitudinal records -- one for the households, one for the person and one for the incidents. Each household is assigned an identification code. This code indicates (among other things), when the household entered the sample, therefore, the period in which the household is to be considered in the sample. The cross-sectional records contain what we will call secondary fields, such as, sex, age, and race. The secondary field information can be used when trying to connect a cross-sectional record to an existing longitudinal record.

Cross-sectional records also have cross-reference fields which provide a link among the different types of records (household, person, incident) within a household for a single interview period. Examples of cross-reference fields are: number of incidents reported by a household and number of household members 12 years of age or older.

A multi-pass process will be used to create the longitudinal files. The actual process will be different for each of the three types of files, but for simplicity we will illustrate the general procedure which is slightly modified for each of the three files. Let us further assume that some longitudinal records exist and we are in the process of adding a new month of data. Actually, this will be the case, once the first month of data is placed in the longitudinal file. The first pass will deal with two types of cross-sectional records:

1. Household rotated into the sample for the first time. It can be determined from the household identification code (as discussed above) when a household is to be rotated into the sample. This type of record has no match in the longitudinal file, so the record is just added to the file.

2. Household previously in survey. In theory, records of this type have matches in the longitudinal file. The internal identification code will facilitate the matching of these types of records. When two records are matched, the secondary fields are also checked to see that they agree.

After the first pass some records from the ICPSR files will remain. These records will be dealt with in pass two as follows:

1. The longitudinal records which should have a match in the cross-sectional file and the remaining cross-sectional records will be searched to find potential matches. If, for example, the two records differ by 1 digit and the secondary fields match, the two records will be united. The cross reference fields can be utilized here.

2. A statistical analysis on the many secondary fields can be performed, to help in matching the remaining records.

The third pass will try to match the remaining records by hand.

I. Appendix A

```
      PROGRAM VBSNEW
      INTEGER I256,NREC(3),FUNC,OLR
      INTEGER*2 IO,CHAN,IOSB(4),BLI,BLIP1
      BYTE EBLOCK(7854)
      CHARACTER ABLOCK(7854),OREC(310),TABLE(256),TAPE*8,MESSAGE*128
      INTEGER SYS$ASSIGN,SYS$QIOW,LIB$SYS_GETMSG,IO$_READVBLK
      INTEGER LIB$MOVTC,LIB$MOVC3
      EXTERNAL IO$_READVBLK
      DATA IN/41/,IOUT/42/,I256/256/,NREC/0,0,0/,TAPE/'TAPE:'/
      OPEN(UNIT=IOUT,RECORDTYPE='VARIABLE',FILE='TAPEO:',
     1     STATUS='NEW',RECL=310,BLOCKSIZE=2048)
      FUNC=%LOC(IO$_READVBLK)
      NBLK=0
C
C     GET THE EBCDIC TO ACSII TRANSLATION TABLE
C
      CALL TRANS(TABLE)
C
C     ALLOCATE AN IO CHANNEL
```



```
C
      NERR=SYS$ASSIGN(TAPE,CHAN,,)
      IF (.not. nerr) THEN
        CALL LIB$SYS_GETMSG(NERR,MLEN,MESSAGE,,)
        TYPE *,' asssign ',MESSAGE(:MLEN)
      END IF
C
C     QUEUE AN IO REQUEST AND WAIT
C     THE REUEST IS FOR A READ OF 1 VIRTUAL BLOCK
C
100   NERR=SYS$QIOW(,%VAL(CHAN),%VAL(FUNC),IOSB,,,EBLOCK,
     1              %VAL(7854),,,,)
      IF (.not. nerr) THEN
        CALL LIB$SYS_GETMSG(NERR,MLEN,MESSAGE,,)
        TYPE *,' qiow ',MESSAGE(:MLEN)

      END IF
      NBLK=NBLK+1
C
C     TRANSLATE EBCDIC TO ACSII
C
      NERR=LIB$MOVTC(%DESCR(EBLOCK),' ',TABLE,ABLOCK)
      IF (.not. nerr) THEN
        CALL LIB$SYS_GETMSG(NERR,MLEN,MESSAGE,,)
        TYPE *,' tran ',MESSAGE(:MLEN)
      END IF
      IQ=IOSB(2)
      IF(IQ.EQ.0)GOTO 999
      I =5
200   BLI=EBLOCK(I)
      IF(BLI.LT.0)BLI=256+BLI
      BLIP1=EBLOCK(I+1)
      IF(BLIP1.LT.0)BLIP1=256+BLIP1
      LNREC = BLI*I256+BLIP1-4
      ISPAN = EBLOCK(I+2)
      I = I + 4
      J = 1
      IF ( ISPAN .EQ. 2 ) J = J + OLR
C
C     MOVE THE ASCII PART OF REC TO OREC
C
      NERR = LIB$MOVC3(LNREC,%REF(ABLOCK(I)),%REF(OREC(J)))
      IF (.not. nerr) THEN
        CALL LIB$SYS_GETMSG(NERR,MLEN,MESSAGE,,)
        TYPE *,' move ',MESSAGE(:MLEN)
      END IF
      I = I + LNREC
```

```
      IF ( ISPAN .NE. 1 ) GO TO 300
      OLR = LNREC
      GO TO 100
300   IF ( ISPAN .EQ. 2 ) LNREC = LNREC + OLR
C
C     WRITE OUT THE LOGICAL RECORD
C
      WRITE(IOUT,2000) (OREC(K),K=1,LNREC)
      IK = (LNREC-231)/79 + 2
      NREC(IK) = NREC(IK) + 1
      IF ( I+4 .GT. IQ ) GO TO 100
      GO TO 200
999   WRITE(6,3000)NREC,NBLK
      CLOSE(UNIT=IOUT)
      STOP
1000  FORMAT(A2,2X,7850A1)
2000  FORMAT(310A1)
3000  FORMAT(8I7)
      END
      SUBROUTINE  TRANS(TABLE)
      CHARACTER TAB(256),TABLE(256)
      DATA TAB/'00'X,'01'X,'02'X,'03'X,'9C'X,'09'X,'86'X,'7F'X,
     1         '97'X,'8D'X,'8E'X,'0B'X,'0C'X,'0D'X,'0E'X,'0F'X,
     1         '10'X,'11'X,'12'X,'13'X,'9D'X,'85'X,'08'X,'87'X,
     2         '18'X,'19'X,'92'X,'8F'X,'1C'X,'1D'X,'1E'X,'1F'X,
     3         '80'X,'81'X,'82'X,'83'X,'84'X,'0A'X,'17'X,'1B'X,
     4         '88'X,'89'X,'8A'X,'8B'X,'8C'X,'05'X,'06'X,'07'X,
     5         '90'X,'91'X,'16'X,'93'X,'94'X,'95'X,'96'X,'04'X,
     6         '98'X,'99'X,'9A'X,'9B'X,'14'X,'15'X,'9E'X,'1A'X,
     7         '20'X,'A0'X,'A1'X,'A2'X,'A3'X,'A4'X,'A5'X,'A6'X,
     8         'A7'X,'A8'X,'5B'X,'2E'X,'3C'X,'28'X,'2B'X,'21'X,
     9         '26'X,'A9'X,'AA'X,'AB'X,'AC'X,'AD'X,'AE'X,'AF'X,
     1         'B0'X,'B1'X,'5D'X,'24'X,'2A'X,'29'X,'3B'X,'5E'X,
     2         '2D'X,'2F'X,'B2'X,'B3'X,'B4'X,'B5'X,'B6'X,'B7'X,
     3         'B8'X,'B9'X,'7C'X,'2C'X,'25'X,'5F'X,'3E'X,'3F'X,
     4         'BA'X,'BB'X,'BC'X,'BD'X,'BE'X,'BF'X,'C0'X,'C1'X,
     5         'C2'X,'60'X,'3A'X,'23'X,'40'X,'27'X,'3D'X,'22'X,
     6         'C3'X,'61'X,'62'X,'63'X,'64'X,'65'X,'66'X,'67'X,
     7         '68'X,'69'X,'C4'X,'C5'X,'C6'X,'C7'X,'C8'X,'C9'X,
     8         'CA'X,'6A'X,'6B'X,'6C'X,'6D'X,'6E'X,'6F'X,'70'X,
     9         '71'X,'72'X,'CB'X,'CC'X,'CD'X,'CE'X,'CF'X,'D0'X,
     1         'D1'X,'7E'X,'73'X,'74'X,'75'X,'76'X,'77'X,'78'X,
     2         '79'X,'7A'X,'D2'X,'D3'X,'D4'X,'D5'X,'D6'X,'D7'X,
     3         'D8'X,'D9'X,'DA'X,'DB'X,'DC'X,'DD'X,'DE'X,'DF'X,
     4         'E0'X,'E1'X,'E2'X,'E3'X,'E4'X,'E5'X,'E6'X,'E7'X,
     5         '7B'X,'41'X,'42'X,'43'X,'44'X,'45'X,'46'X,'47'X,
     6         '48'X,'49'X,'E8'X,'E9'X,'EA'X,'EB'X,'EC'X,'ED'X,
```



```
     7         '7D'X,'4A'X,'4B'X,'4C'X,'4D'X,'4E'X,'4F'X,'50'X,
     8         '51'X,'52'X,'EE'X,'EF'X,'F0'X,'F1'X,'F2'X,'F3'X,
     9         '5C'X,'9F'X,'53'X,'54'X,'55'X,'56'X,'57'X,'58'X,
     1         '59'X,'5A'X,'F4'X,'F5'X,'F6'X,'F7'X,'F8'X,'F9'X,
     2         '30'X,'31'X,'32'X,'33'X,'34'X,'35'X,'36'X,'37'X,
     3         '38'X,'39'X,'FA'X,'FB'X,'FC'X,'FD'X,'FE'X,'FF'X/
      DO 100 I = 1,256
        TABLE(I)=TAB(I)
100   CONTINUE
      RETURN
      END
```

END