Survey of Income and Program Participation

CREATING SIPP LONGITUDINAL FILES
USING OSIRIS IV

by

Marita Servais
University of Michigan

No. 8715 37

December 1987

TOPIC CONTRIBUTED

This paper was prepared for presentation at the 1987 Meeting of the American Statistical Association, August 17-20, 1987. The paper was included in a topic contribution session, "Accessing Large and Complex Data Sets: SIPP As A Case Study", sponsored by the Statistical Computing Section of the American Statistical Association.

Table of Contents

CREATING SIPP LONGITUDINAL FILES USING OSIRIS.IV

Marita Servais, The University of Michigan

Institute for Social Research, P. O. Box 1248, Ann Arbor, MI 41806-1248

The 1984 SIPP (Survey of Income and Program Participation) panel data is distributed in nine person-level cross-sectional data files that are not in a form suitable for longitudinal analysis. Features of the SIPP relevant to the construction of longitudinal records are described. The construction of three longitudinal person-level files using facilities available in a standard software package are discussed. The first introduces the OSIRIS.IV file merging command. The second creates a longitudinal record allowing for "skipped" waves with a subset of variables from each of the nine waves for all persons present in at least one wave of the panel. The third creates a 1984 calendar-year file for all persons in a sample household during the calendar year 1984, "lining-up" reference months. And in conclusion, issues that need to be considered by the analyst who wishes to construct a data file suitable for longitudinal analysis are discussed.[1]

I. The SIPP Design

A. Description of Study

The 1984 SIPP Panel consisted of about 26,000 housing units of which about 21,000 were eligible for interview. There were 53,726 people (adults and children) in the sample households at the time of the initial interview. The panel was divided into four sub-samples of approximately equal size called "rotation groups".

Interviewing began in October 1983 for Rotation Group 1, in November for Rotation Group 2, in December for Rotation Group 3 and in January 1984 for Rotation Group 4. Each rotation group was interviewed at four month intervals for the duration of the panel. Rotation Groups 1 and 2 were interviewed nine times. Rotation Groups 3 and 4 were interviewed eight times.

Each interview contained basic "core" information. Information was obtained for the period of the four months prior to the month of interview. These months are called "reference months". The third thru ninth interview schedules also contained a set of supplemental questions or topical modules that were included only once.

People in the original sample households were followed and interviewed at subsequent times if they moved out into other households or moved out to form households of their own. The people with whom they subsequently shared living quarters were interviewed only when they co-resided with a member of the original panel. There were 64,928 persons present for one or more interviews over the course of the study.

Each cycle of interviews using the same interview schedule, administered to the entire sample with the exceptions noted below, was called a "wave". Rotation Group 4 was not interviewed during Wave 2; Rotation Group 3 was not interviewed Wave 8. Interviewing concluded in July 1986 with Rotation Group 2. The following table illustrates the interviewing schedule.

B. Distribution Data Files

The SIPP public use rectangular data files from the 1984 panel contain one record for each person present in a sample household at the time of interview (or for a portion of the previous four month period if present at the previous interview). Household, family, and sub-family information is included on the person record.

The data file for a wave contains between approximately 32,000 to 59,000 records (depending on the wave) with a logical record length of 5,352 for the core data only. These data files are distributed on two to three (6,250 bpi) tapes per wave.

While the format of the SIPP rectangular distribution files is useful for analysts wishing to do cross-sectional analysis, analysts wishing to do many types of longitudinal analysis are faced with data management tasks to produce a record suitable for analysis.

Month of Interview by Wave for Rotation Groups

		Rotation Group		
Wave	One	Two	Three	Four
1	Oct 83	Nov 83	Dec 83	Jan 84
2	Feb 84	Mar 84	Apr 84	......
3	Jun 84	Jul 84	Aug 84	May 84
4	Oct 84	Nov 84	Dec 84	Sep 84
5	Feb 85	Mar 85	Apr 85	Jan 85
6	Jun 85	Jul 85	Aug 85	May 85
7	Oct 85	Nov 85	Dec 85	Sep 85
8	Feb 86	Mar 86	......	Jan 86
9	Jun 86	Jul 86	Apr 86	May 86

C. Identifiers

The SIPP has constructed a set of identifiers to uniquely identify each person's record in each wave and to provide the possibility of linking information about the same person from different waves to form a cross-wave record for a person. A person may be identified by a combination of three variables: SU-ID, a nine-digit number which identifies the original sample household of which a person was a member or with which a person later became associated (for persons not present at the time of the first interview). PP-ENTRY, a two-digit number which within SU-ID identifies the household at which the person first entered the panel — the first digit indicates the wave in which the address was first assigned, the second digit sequentially numbers within the wave multiple households derived from the original sample household. This number will be "11" for all persons present at the time of the initial interview. PP-PNUM a three-digit number which within SU-ID and PP-ENTRY identifies a person. Persons present at the first interview have person numbers beginning with one, 101, 102, etc., persons present for the first time at the second interview have person numbers beginning with two, 201, 202, etc., and so forth for subsequent waves.

The values for this set of variables is constant for a person across all waves of the panel. This set of variables allows matching a person's record from Wave 1 with the same person's record from Wave 2 and subsequent waves.[2]

II. Matching Two or More Waves of Data

To begin our task at a simple level, we will consider the matching and merging of variables for persons from three waves of data, Wave 1, Wave 5 and Wave 9, when all rotation groups were interviewed.

A person present during Wave 1 may not be present in a responding household in Waves 5 or 9. And conversely, a person not present for Wave 1 may have joined a sample household by Wave 5 or 9. Illustrated below are abbreviated data matrices for Waves 1, 5, and 9.

The Wave 1 data matrix consists of four records, three from SU-ID #1, persons 101, 102, and 103 and one from SU-ID #2, person 101.[3] Values for data variables are represented here graphically as "x's".

Wave 1 Data Records N=4

SU-ID	PP-PNUM	DATA
1	101	x x x
1	102	x x x
1	103	x x x
2	101	x x x

The Wave 5 data matrix consists of five records. just two from SU-ID #1. persons 101 and 102. as person 103 is no longer present, and three from SU-ID #2, person 101 and two new persons 501 and 502. Values for data variables are represented as "y's".

Wave 5 Data Records N=5

SU-ID	PP-PNUM	DATA
1	101	y y y
1	102	y y y
2	101	y y y
2	501	y y y
2	502	y y y

In the Wave 9 data matrix SU-ID #1 has three records because person 101 has left while person 103 has returned and person 901 has joined the household. SU-ID #2 has two records as person 501 has left the sample. Values for data variables are represented as "z's".

Wave 9 Data Records N=5

SU-ID	PP-PNUM	DATA
1	102	z z z
1	103	z z z
1	901	z z z
2	101	z z z
2	502	z z z

The fourth data matrix illustrates the merged record that results from the joining of a person's record from Wave 1 with that person's record from Waves 5 and 9. Dashes indicate where padding needs to occur to fill in variables for persons not present during a wave. One record exists in the combined data file for any person who was present in a sample household for one or more waves. Persons 1-102 and 2-201 were present for all three waves. Person 1-101 was present in Waves 1 and 5 but not Wave 9. Person 1-103 was present in Waves 1 and 9 but not Wave 5. Person 1-901 just joined a sample household in Wave 9 and was not present for Waves 1 or 5. Person 2-501 was just present for Wave 5 and person 2-502 was present for Waves 5 and 9 but not Wave 1.

Combined Waves 1, 5 and 9 Data Records N=7

		WAVE 1	WAVE 5	WAVE 9
SU-ID	PP-PNUM	DATA	DATA	DATA
1	101	x x x	y y y	– – –
1	102	x x x	y y y	z z z
1	103	x x x	– – –	z z z
1	901	– – –	– – –	z z z
2	101	x x x	y y y	z z z
2	501	– – –	y y y	– – –
2	502	– – –	y y y	z z z

In order to produce this matrix, our program must be able to create a combined output data record with with data from the Waves 1, 5, and 9 data records by identifying matching input records from the Waves 1, 5 and 9 input matrices and padding as needed if a person was not present for a wave. To do this in OSIRIS.IV[4] the following commands would suffice:

```
1)  &UPDATE
2)    COMBINING THREE WAVES OF DATA
3)    PRINT=OUTD PAD=ZERO
4)    INFI=W1 V=@(LIST) ID=@(IDVARS) OPT RENU=1001
5)    INFI=W5 V=@(LIST) ID=@(IDVARS) OPT RENU=5001
6)    INFI=W9 V=@(LIST) ID=@(IDVARS) OPT RENU=9001
7)  &END
```

The first line invokes the OSIRIS.IV command UPDATE that merges two or more files.[5] The second line is a label. It is printed on the output and has no other function. The third line is a parameter statement that requests the printing of the new output dictionary, PRINT=OUTD, and the padding of the output record with zeros for portions of the output record that are missing because the person was not present during Wave 1, 5, or 9, PAD=ZERO. The fourth thru sixth lines are update statements which specify actions to be taken.[6] INFI=W1, INFI=W5 and INFI=W9 specify the suffix that will be used to identify the input data matrix for Waves 1, 5, and 9 respectively. ID=@(IDVARS) specifies the variables that will be used to match a person's record from a wave with the person's record from other waves. V=@(LIST) specifies the list of variables to be taken from the data file.[7] OPT specifies that an output record is to be written even if a record for the person is not present in a wave. RENU=1001, RENU=5001 and RENU=9001 specify that the variables from Waves 1, 5, and 9 will be numbered, respectively, starting with 1001, 5001 and 9001 in the output data record.[8] The seventh line terminates this job step.[9]

III. Merging Nine Waves of Data Allowing for Skipped-Waves

The steps outlined in the previous section made no allowance for "skipped" waves. Between the Wave 1 and Wave 5 data collection there was a sixteen month interval for Rotation Groups 1, 2 and 3 and a twelve month interval for Rotation Group 4 which skipped Wave 2. Similarly between the Wave 5 and Wave 9 data collection there was a sixteen month interval for for Rotation Groups 1, 2 and 4 and a twelve month interval for Rotation Group 3 which skipped Wave 8.

If we wished to merge data from the first three *interviews* which occurred at four month intervals for all four rotation groups (as opposed to the first three *waves*), we would need data from Waves 1, 2 and 3 for Rotation Groups 1 thru 3 and from Waves 1, 3 and 4 for Rotation Group 4. We will need to fetch data for the second interview for Rotation Group 4 from Wave 3 because Rotation Group 4's second interview was Wave 3 as it was not interviewed during Wave 2. A schematic representation of records with merged data from the first three interviews would look like this:

Rotation Group	First Interview	Second Interview	Third Interview
1	Wave 1	Wave 2	Wave 3
2	Wave 1	Wave 2	Wave 3
3	Wave 1	Wave 2	Wave 3
4	Wave 1	Wave 3	Wave 4

We would need to get data for the first interview from the Wave 1 data matrix for all rotation groups, for the second interview from the Wave 2 data matrix for Rotation Groups 1 thru 3 and from the Wave 3 data matrix for Rotation Group 4, and for the third interview from the Wave 3 data matrix for Rotation Groups 1 thru 3 and from the Wave 4 data matrix for Rotation Group 4.

The procedure for merging all nine waves of data is similar. Since Rotation Group 4 was not interviewed during Wave 2 and Rotation Group 3 was not interviewed at Wave 8, if we were to merge all waves without allowing for these skipped waves the following records would result:

Rotation	Wave								
Group	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
1	W1	W2	W3	W4	W5	W6	W7	W8	W9
2	W1	W2	W3	W4	W5	W6	W7	W8	W9
3	W1	W2	W3	W4	W5	W6	W7	..	W9
4	W1	..	W3	W4	W5	W6	W7	W8	W9

and it would look like persons in Rotation Groups 3 and 4 had missed an interview when in fact they had been in responding households for the entire panel. What we would like instead is a record where the interval between succeeding *interviews* is four months for all rotation groups. For Rotation Group 4 we will need to create a record that has the Wave 3 data moved into the second position, the Wave 4 data into the third position, and so forth and for Rotation Group 3 we will need to create a record that has the Wave Nine data moved into the eighth position:

Rotation	Interview								
Group	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
1	W1	W2	W3	W4	W5	W6	W7	W8	W9
2	W1	W2	W3	W4	W5	W6	W7	W8	W9
3	W1	W2	W3	W4	W5	W6	W7	W9	..
4	W1	W3	W4	W5	W6	W7	W8	W9	..

For convenience we will have the variables for the first interview begin with variable 1001, for the second interview begin with variable 2001, and so forth, with variables for the ninth interview beginning with variable 9001.

In order to do this our program will have to get variables for the second interview from Wave 2 for Rotation Groups 1 thru 3 and from Wave 3 for Rotation Group 4, to get variables for the third interview from Wave 3 for Rotation Groups 1 thru 3 and from Wave 4 for Rotation Group 4, and so forth. Similarly it will have to get data for the eighth interview from Wave 8 for Rotation Groups 1 and 2 and from Wave 9 for Rotation Groups 3 and 4. It will also have to pad the ninth interview for Rotation Groups 3 and 4. The following command lines would do this using OSIRIS.IV:

```
1)  &UPDATE
2)    13: INCLUDE @(SU-ROT)=1-3
3)    4: INCLUDE @(SU-ROT)=4
4)    12: INCLUDE @(SU-ROT)=1-2
5)    34: INCLUDE @(SU-ROT)=3-4
6)    MERGE W1-W9, CREATE FILE
7)    PRINT=OUTD PAD=ZERO
    &
8)    INFI=W1 V=@(LIST) ID=@(IDVARS) OPT RENU=1001
    &
9)    INFI=W2 V=@(LIST) ID=@(IDVARS) OPT RENU=2001
10)   INFI=W3 V=@(LIST) ID=@(IDVARS) OPT RENU=2001 FILT=4
    &
11)   INFI=W3 V=@(LIST) ID=@(IDVARS) OPT RENU=3001 FILT=13
12)   INFI=W4 V=@(LIST) ID=@(IDVARS) OPT RENU=3001 FILT=4
    &
13)   INFI=W4 V=@(LIST) ID=@(IDVARS) OPT RENU=4001 FILT=13
14)   INFI=W5 V=@(LIST) ID=@(IDVARS) OPT RENU=4001 FILT=4
    &
15)   INFI=W5 V=@(LIST) ID=@(IDVARS) OPT RENU=5001 FILT=13
16)   INFI=W6 V=@(LIST) ID=@(IDVARS) OPT RENU=5001 FILT=4
    &
17)   INFI=W6 V=@(LIST) ID=@(IDVARS) OPT RENU=6001 FILT=13
```

```
18)    INFI=W7 V=@(LIST) ID=@(IDVARS) OPT RENU=6001 FILT=4
    &
19)    INFI=W7 V=@(LIST) ID=@(IDVARS) OPT RENU=7001 FILT=13
20)    INFI=W8 V=@(LIST) ID=@(IDVARS) OPT RENU=7001 FILT=4
    &
21)    INFI=W8 V=@(LIST) ID=@(IDVARS) OPT RENU=8001 FILT=12
22)    INFI=W9 V=@(LIST) ID=@(IDVARS) OPT RENU=8001 FILT=34
    &
23)    INFI=W9 V=@(LIST) ID=@(IDVARS) OPT RENU=9001 FILT=12
    &
 24) &END
```

The first line invokes the OSIRIS.IV command UPDATE that merges two or more files. Lines 2 thru 5 establish filters that will be used to select records of specified sets of rotation groups for differential processing. Line 2 will select the records of Rotation Groups 1 thru 3, line 3 will select the records of Rotation Group 4, line 4 of Rotation Groups 1 and 2, and line 5 of Rotation Groups 3 and 4. The sixth line is a label. The seventh line is a parameter statement which specifies that the output dictionary will be printed, PRINT=OUTD, and that output records for persons not present for one or more waves will have a output record written with the variables for the missing waves padded with zeros, PAD=ZERO. Lines 8 thru 23 are update statements which specify groups of variables that need to be obtained from the input data files to create the output data record. INFI=Wn specifies the suffix that will be used to identify the input data matrix. ID=@(IDVARS) specifies the variables to be used to match a person's records from one wave with that person's records from another wave to form a merged output record for the person. V=@(LIST) specifies the variables to be obtained from the input record. LIST needs to have been previously specified. OPT specifies that an output record is to be written for each person even if they were not present in one or more waves. FILT=4, FILT=13, FILT=12, FILT=34 specify the number of the filter which will select, respectively, records of people in Rotation

Group 4, Rotation Groups 1 thru 3, Rotation Groups 1 and 2, and Rotation Groups 3 and 4, from input data matrices. RENU=1001, RENU=2001, RENU=3001, etc. specify how the variables specified in the update statement are to be numbered in the output data record. The twenty-fourth line terminates the job step.

The combined effect of these specifications works like this. Variables for the first interview will be taken from the Wave 1 data file for all rotation groups and will be numbered beginning with variable 1001 in the output data record. Variables for the second interview will taken from the Wave 2 data file for all rotation groups included in Wave 2, Rotation Groups 1 thru 3, and from the Wave 3 data file for Rotation Group 4. They will be numbered beginning with variable 2001 in the output data record. Variables for the third interview will be taken from the Wave 3 data file for Rotation Groups 1 thru 3 and from the Wave 4 data file for Rotation Group 4. They will be numbered beginning with variable 3001 in the output data record. The program continues in a similar manner for the fourth thru the seventh interviews. Variables for the eighth interview will be taken from the Wave 8 data file for Rotation Groups 1 and 2 and from the Wave 9 data file for Rotation Groups 3 and 4. They will be numbered beginning with variable 8001 in the output data record. Variables for the ninth interview will be taken from the Wave 9 data file for Rotation Groups 1 and 2 and will be padded with zeros for Rotation Groups 3 and 4 which were not interviewed for a ninth time. They will be numbered beginning with variable 9001 in the output data record.

IV. Creating a 1984 Calendar Year Record

SIPP's staggered interviewing schedule means that Reference Month 1 for the Wave 1 refers to June 1983 for Rotation Group 1, to July 1983 for Rotation Group 2, to August 1983 for Rotation Group 3 and to September 1983 for Rotation Group 4. The longitudinal data record that was described in the previous section had data aligned so that variables about the first reference month described conditions about a different calendar month for persons in different rotation groups. For some types of analysis it may be desirable to have the data arranged so that a given set of variables refer to the same calendar month for all rotation groups. We will examine the steps required to construct a data record with data relating to reference months for 1984 in the same relative position in the record for persons from any reference group.

Data for reference months in 1984 is found in the public use rectangular files for Waves 2, 3, 4 and 5. The data for calendar year 1984 for the four rotation groups is located in the files as follows:

Rotation Group		Month of Interview	Reference Month			
			One	Two	Three	Four
1	Wave 2	Feb 84	...	...	...	Jan
2	Wave 2	Mar 84	...	...	Jan	Feb
3	Wave 2	Apr 84	...	Jan	Feb	Mar
4	Wave 3	May 84	Jan	Feb	Mar	Apr
1	Wave 3	Jun 84	Feb	Mar	Apr	May
2	Wave 3	Jul 84	Mar	Apr	May	Jun
3	Wave 3	Aug 84	Apr	May	Jun	Jul
4	Wave 4	Sep 84	May	Jun	Jul	Aug
1	Wave 4	Oct 84	Jun	Jul	Aug	Sep
2	Wave 4	Nov 84	Jul	Aug	Sep	Oct
3	Wave 4	Dec 84	Aug	Sep	Oct	Nov
4	Wave 5	Jan 85	Sep	Oct	Nov	Dec
1	Wave 5	Feb 85	Oct	Nov	Dec	...
2	Wave 5	Mar 85	Nov	Dec	...	...
3	Wave 5	Apr 85	Dec	...	...	...

We wish to rearrange the data to construct a data record with the following layout for persons in a sampie household during any portion of the calendar 1984 from any rotation group:

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

For convenience we will have the reference month variables for January begin with variable 1001, those for February begin with variable 2001, and so forth, with those for December beginning with variable 12001.

The following table shows where data for each month in 1984 will have to come from in the public use rectangular data files. For instance information about January 1984 would come from Wave 2 Reference Month 4 for Rotation Group 1, from Wave 2 Reference Month 3 for Rotation Group 2, from Wave 2 Reference Month 2 for Rotation Group 3 and from Wave 3 Reference Month 1 for Rotation Group 4.

Wave and Reference Month by Rotation Groups for Calendar Year 1984

	Rotation Group			
Month	One	Two	Three	Four
Jan	2–4	2–3	2–2	3–1
Feb	3–1	2–4	2–3	3–2
Mar	3–2	3–1	2–4	3–3
Apr	3–3	3–2	3–1	3–4
May	3–4	3–3	3–2	4–1
Jun	4–1	3–4	3–3	4–2
Jul	4–2	4–1	3–4	4–3
Aug	4–3	4–2	4–1	4–4
Sep	4–4	4–3	4–2	5–1
Oct	5–1	4–4	4–3	5–2
Nov	5–2	5–1	4–4	5–3
Dec	5–3	5–2	5–1	5–4

To produce the desired data record the program must put data from Wave 2 Reference Month 4 into the January variables for Reference Group 1, from Wave 2 Reference Month 3 for Rotation Group 2, from Wave 2 Reference Month 2 for Rotation Group 3 and from Wave 3 Reference Month 1 for Rotation Group 4 and so forth. It must also match the January set of variables for a person with the February set of variables for that person and so forth to construct an output record with variables for the entire year for the person. One output record will be constructed for each person in the panel during any portion of the calendar year 1984. The following command lines will do this in one step using OSIRIS.IV:

```
1) &UPDATE
2)    1: INCLUDE @(SU-ROT)=1
3)    2: INCLUDE @(SU-ROT)=2
```

```
4)    3. INCLUDE @(SU-ROT)=3
5)    4. INCLUDE @(SU-ROT)=4
6)    MERGE W2-W5, CREATE FILE LINING UP 1984 MONTHS
7)    PRINT=OUTD PAD=ZERO
   &
8)    INFI=W2 V=@(LIST4) ID=@(IDVARS) OPT RENU=1001 FILT=1
9)    INFI=W2 V=@(LIST3) ID=@(IDVARS) OPT RENU=1001 FILT=2
10)   INFI=W2 V=@(LIST2) ID=@(IDVARS) OPT RENU=1001 FILT=3
11)   INFI=W3 V=@(LIST1) ID=@(IDVARS) OPT RENU=1001 FILT=4
   &
12)   INFI=W3 V=@(LIST1) ID=@(IDVARS) OPT RENU=2001 FILT=1
13)   INFI=W2 V=@(LIST4) ID=@(IDVARS) OPT RENU=2001 FILT=2
14)   INFI=W2 V=@(LIST3) ID=@(IDVARS) OPT RENU=2001 FILT=3
15)   INFI=W3 V=@(LIST2) ID=@(IDVARS) OPT RENU=2001 FILT=4
   &
16)   INFI=W3 V=@(LIST2) ID=@(IDVARS) OPT RENU=3001 FILT=1
17)   INFI=W3 V=@(LIST1) ID=@(IDVARS) OPT RENU=3001 FILT=2
18)   INFI=W2 V=@(LIST4) ID=@(IDVARS) OPT RENU=3001 FILT=3
19)   INFI=W3 V=@(LIST3) ID=@(IDVARS) OPT RENU=3001 FILT=4
   &
20)   INFI=W3 V=@(LIST3) ID=@(IDVARS) OPT RENU=4001 FILT=1
21)   INFI=W3 V=@(LIST2) ID=@(IDVARS) OPT RENU=4001 FILT=2
22)   INFI=W3 V=@(LIST1) ID=@(IDVARS) OPT RENU=4001 FILT=3
23)   INFI=W3 V=@(LIST4) ID=@(IDVARS) OPT RENU=4001 FILT=4
   &
24)   INFI=W3 V=@(LIST4) ID=@(IDVARS) OPT RENU=5001 FILT=1
25)   INFI=W3 V=@(LIST3) ID=@(IDVARS) OPT RENU=5001 FILT=2
26)   INFI=W3 V=@(LIST2) ID=@(IDVARS) OPT RENU=5001 FILT=3
27)   INFI=W4 V=@(LIST1) ID=@(IDVARS) OPT RENU=5001 FILT=4
   &
28)   INFI=W4 V=@(LIST1) ID=@(IDVARS) OPT RENU=6001 FILT=1
29)   INFI=W3 V=@(LIST4) ID=@(IDVARS) OPT RENU=6001 FILT=2
```

```
30)    INFI=W3 V=@(LIST3) ID=@(IDVARS) OPT RENU=6001 FILT=3

31)    INFI=W4 V=@(LIST2) ID=@(IDVARS) OPT RENU=6001 FILT=4

    &

32)    INFI=W4 V=@(LIST2) ID=@(IDVARS) OPT RENU=7001 FILT=1

33)    INFI=W4 V=@(LIST1) ID=@(IDVARS) OPT RENU=7001 FILT=2

34)    INFI=W3 V=@(LIST4) ID=@(IDVARS) OPT RENU=7001 FILT=3

35)    INFI=W4 V=@(LIST3) ID=@(IDVARS) OPT RENU=7001 FILT=4

    &

36)    INFI=W4 V=@(LIST3) ID=@(IDVARS) OPT RENU=8001 FILT=1

37)    INFI=W4 V=@(LIST2) ID=@(IDVARS) OPT RENU=8001 FILT=2

38)    INFI=W4 V=@(LIST1) ID=@(IDVARS) OPT RENU=8001 FILT=3

39)    INFI=W4 V=@(LIST4) ID=@(IDVARS) OPT RENU=8001 FILT=4

    &

40)    INFI=W4 V=@(LIST4) ID=@(IDVARS) OPT RENU=9001 FILT=1

41)    INFI=W4 V=@(LIST3) ID=@(IDVARS) OPT RENU=9001 FILT=2

42)    INFI=W4 V=@(LIST2) ID=@(IDVARS) OPT RENU=9001 FILT=3

43)    INFI=W5 V=@(LIST1) ID=@(IDVARS) OPT RENU=9001 FILT=4

    &

44)    INFI=W5 V=@(LIST1) ID=@(IDVARS) OPT RENU=10001 FILT=1

45)    INFI=W4 V=@(LIST4) ID=@(IDVARS) OPT RENU=10001 FILT=2

46)    INFI=W4 V=@(LIST3) ID=@(IDVARS) OPT RENU=10001 FILT=3

47)    INFI=W5 V=@(LIST2) ID=@(IDVARS) OPT RENU=10001 FILT=4

    &

48)    INFI=W5 V=@(LIST2) ID=@(IDVARS) OPT RENU=11001 FILT=1

49)    INFI=W5 V=@(LIST1) ID=@(IDVARS) OPT RENU=11001 FILT=2

50)    INFI=W4 V=@(LIST4) ID=@(IDVARS) OPT RENU=11001 FILT=3

51)    INFI=W5 V=@(LIST3) ID=@(IDVARS) OPT RENU=11001 FILT=4

    &

52)    INFI=W5 V=@(LIST3) ID=@(IDVARS) OPT RENU=12001 FILT=1

53)    INFI=W5 V=@(LIST2) ID=@(IDVARS) OPT RENU=12001 FILT=2

54)    INFI=W5 V=@(LIST1) ID=@(IDVARS) OPT RENU=12001 FILT=3

55)    INFI=W5 V=@(LIST4) ID=@(IDVARS) OPT RENU=12001 FILT=4
```

56) &END

The first line invokes the OSIRIS.IV command UPDATE that merges two or more files. We will need to select records from different rotation groups for different treatment. Lines 2 thru 5 establish filters that will be used to select Rotation Groups 1, 2, 3 and 4 respectively. The filters will subsequently be referred to by the numbers 1, 2, 3 or 4. The sixth line is a label. If a person does not have data present for all months, variables for missing months will be padded with zeros, PAD=ZERO, as specified in the seventh line, the parameter statement, and the output dictionary will be printed, PRINT=OUTD.

The next forty-eight lines, twelve groups (one group for each month) of four lines each (one for each rotation group), lines 8 thru 55, are update statements. They provide instructions for constructing the calendar year record. A different set of instructions is required for each month for each rotation group. This seems like a lot of instructions. However the syntax is repetitive and the the values of the keywords follow a pattern laid out in the previous table. INFI=Wn specifies the suffix of the input data matrix from which data is to be retrieved, from Wave 2, 3, 4 or 5 for calendar year 1984. ID=@(IDVARS) specifies the set of variables, SU-ID, PP-ENTRY and PP-PNUM, to be used to match the portions of various records to construct a single output record for a person. V=@(LIST1), V=@(LIST2), V=@(LIST3) and V=@(LIST4) specify which set of variables will be taken from the input record. The four lists, LIST1, LIST2, LIST3 and LIST4, specify variables from reference months 1, 2, 3 and 4, respectively. FILT=1, FILT=2, FILT=3 and FILT=4 specify filters which will select records for people in Rotation Groups 1, 2, 3 and 4, respectively. RENU=1001, RENU=2001, etc. specify renumbering of the variables in the output record beginning with 1001 for January, 2001 for February, and so forth. OPT specifies that an output record is to be produced for a person even if they are do not have data for a particular month.

Putting this all together, for the January 1984 portion of the output record the program is being instructed as follows: for Rotation Group 1 – get input data from the Wave 2 rectangular file, INFI=W2, use variables from Reference Month 4, V=@(LIST4), and number this set of variables beginning with 1001 in the output record. RENU=1001; for Rotation Group 2 — get input data from the Wave 2 rectangular file. INFI=W2, use variables from Reference Month 3, V=@(LIST3), and number this set of variables beginning with 1001 in the output record, RENU=1001; for Rotation Group 3 — get input data from the Wave 2 rectangular file, INFI=W2. use variables from Reference Month 2. V=@(LIST2). and number this set of variables beginning with 1001 in the output record. RENU=1001: for Rotation Group 4 — get input data from the Wave 3 rectangular file. INFI=W3. use variables from Reference Month 1, V=@(LIST1). and number this set of variables beginning with 1001 in the output record. RENU=1001.

The instructions continue in an analogous way for selecting records and variables for the February thru December portions of the output record. Thus in one job step it is possible to select appropriate portions of any of four input files and match them with other selected portions to construct a single output record for the calendar year 1984 for each person.

V. Computing Requirements and Practical Considerations

A. Implementing Procedures Described

There are some practical considerations that should be mentioned at this point. In contrast to the relative simplicity of the procedures described in previous sections, there are complexities that are introduced as a result of the size of the SIPP data files and restrictions of the OSIRIS.IV UPDATE command.

For instance, consider the creation of the 1984 calendar year record. The SIPP rectangular data files for Waves 2 thru 5 occupy ten 6,250 bpi tapes. One or more output tapes will also be required. At most installations this would require having five tape drives available simultaneously. However, even if that were feasible, it would not be sufficient because the UPDATE command requires that all input files be open simultaneously. For instance, the input file from Wave 3 is referenced sixteen times in the procedure described above. It is not possible for OSIRIS.IV to have a tape file open to more than one location. However, it is possible to have a disk file with pointers pointing to more than one location. Thus all four input files need to be located on disk files. Putting approximately 267,600,000 bytes of data on disk for each of the four input files is also likely to be prohibitive. Thus it is suggested creating a previous job step to subset variables from each of the four waves, writing the selected data to four tape data files and just prior to the matching job step copying these four tape files to scratch disk files. Similar considerations apply for the procedures allowing for skipped waves.

B. Constraints of a Particular Computing Environment

In addition to a reasonable understanding of the SIPP data files and basic data management commands, the analyst wishing to embark on constructing a longitudinal file must also be aware of a number of possible constraints of his or her particular computing environment. In most environments the nine waves of SIPP represent a large amount of data. Limitations at other installations will vary from those mentioned below, but given the magnitude of data of all nine waves of SIPP, limitations such as these are likely to need to be taken into consideration.

Limitations of Hardware. An analyst may be limited by the number of tape drives available at his or her installation. For instance the Food and Drug Administration's Parklawn Computer Center, a site where SIPP is processed, has only four 6,250 bpi tape drives. This means that no more than three tape input files can be used in a single job step. While the University of Michigan has twenty tape drives, it does not recognize multi-volume tapes, and thus all tapes that will be used in a single job step must be mounted simultaneously. In this environment there is a premium on keeping the number of tapes required for a single data file to a minimum.

It is also useful to have access to large amounts of disk storage space on either a temporary or permanent basis. The maximum size of an individual disk file and the maximum total disk storage space available may suggest particular strategies. For instance at Michigan the largest disk file that can be created is about 131,000,000 bytes — about half of a single wave of SIPP data.

Limitations at other installations will vary, but limitations such as these are likely to need to be taken into consideration. The bare minimum that would seem to be required in order to attempt to create a longitudinal file from the nine (or fewer) rectangular files would seem to be at least three tape drives, two for input tapes and one for an output tape. More available tape drives or access to a significant amount of disk storage, either on a permanent or temporary basis, can significantly facilitate this work.

Logical Record Length Limitations. If all nine waves of data were combined the resulting logical record length would be 48,169. This is far greater than is allowed at many, if not most, installations. Thus the analyst wishing to construct a merged longitudinal data file will likely need to select a subset of variables. Variables may be subset by limiting the number of variables selected from each of nine waves of data or by selecting fewer than nine waves of data, for instance selecting all variables from three waves of data.

Cost Considerations. The analyst contemplating working with the entire nine waves of the SIPP panel should be aware that massive amounts of data will be passed and significant computing resources are apt to be required. There may be a limit to the amount of data that it is reasonable to have in the longitudinal file. If the logical record length of 48,168 were possible, records for all 64,928 persons would occupy an estimated eighteen tapes. Passing that amount of data for a single analysis step is apt to be undesirably expensive in most environments. For most analysis requirements the analyst will want to significantly subset variables and possibly cases.

C. Data Management Requirements and Strategies

While the data management tools and concepts described are elementary, the magnitude of the SIPP data warrants careful application of some basic data management guidelines.

Data Management Software. The data management tools that are required to create a longitudinal file from the cross-sectional files are basic. They include the ability to subset variables, to subset cases, to sort records, to match and merge records and to reformat a record by renaming or relocating variables.

Test Data Sets. While gaining an understanding of the structure and special features of the SIPP data and devising data management procedures, it may be useful to test the procedures on a limited set of cases to debug program setups. This allows the analyst to ensure that the approach being used will produce the desired results without the expense implicit in processing the data of the full panel.

Checking Results against Census Produced Counts. Whenever possible, the number of records processed, merged, padded or whatever should be matched against expected counts. For instance, the count of records for each wave should match the number of records that are supposed to be in that wave. If 36,830 records were found on data tapes for Wave 4 where there are supposed to be 55,993 records, one would need to check to see if the proper tapes had been received or were being accessed. Or if producing new aggregate counts for families, a check could be made to confirm that the number of records processed for each family matched the Census produced counts of number of persons in the family.

Breaking Up Job into Separate Steps. When working with the complete data set it is advisable to break up a complex series of operations into separate parts. Thus if a later job step fails, the procedure can be restarted at the point of failure and earlier successful job steps will not need to be re-executed. This may mean writing intermediate files to permanent disk files or to tape.

With reasonable care, consideration, an understanding of the format of the data and basic data management functions available in standard software packages. the analyst can construct a data file suitable for longitudinal analysis from the SIPP rectangular cross-sectional person-level files.

Footnotes

[1] The author wishes to thank Steven G. Heeringa of the University of Michigan for reviewing this paper and providing helpful comments and David B. McMillen of the U. S. Bureau of the Census for providing invaluable information about the intricacies of the SIPP dataset and for reviewing this paper to ensure that they were described accurately therein.

[2] There are minor exceptions. Sixteen people moved from one sample household and joined another unrelated sample household during the course of the panel. These people are identified by their original sample SU-ID while they resided in the original sample household and by a second sample SU-ID while they resided in the second sample household. They were assigned new person numbers in in the range 180–199 as members of the second household. Another person moved from one sample household to another sample household derived from the same original household and for administrative purposes was assigned a new person number in the second household. Since these seventeen people have two, non-matching, sets of values for their identification variables, the matching procedures described subsequently will not match the portions of these persons's records associated with the original household with the portions of the person's records associated with the second household.

[3] When working with the actual data PP-ENTRY would also be used as an identifier and SU-ID would be a nine-digit number.

[4] OSIRIS.IV has been written and is maintained by the Institute for Social Research of the University of Michigan. Information about leasing OSIRIS.IV can be obtained by contacting OSIRIS Distribution, Institute for Social Research, P. O. Box 1248, Ann Arbor, MI 48106, by telephone (313) 764–4417, or via electronic mail –
OSIRIS_DISTRIBUTION@UM.CC.UMICH.EDU

[5] The parenthetical numbers are not part of the actual setup.

[6] It is the assumption in this and following examples that input data files. Wave 1 and Wave 5, and Wave 9 in this case, have their records in sort order by the person identification variables, SU-ID, PP-ENTRY and PP-PNUM. If this is not the case, the program can be requested to sort the portions of the input records that will be needed for the job step by adding the keyword SORT=50000 to each of the update statements where 50,000 is the approximate number of records that are present in the input file.

[7] The values of IDVARS and LIST are specified prior to this job step. The values of IDVARS would be the SU-ID, PP-ENTRY, and PP-PNUM variables. The values of LIST would be the variables to be selected from the input data files. The use of this syntax is optional and has been used in this paper for its mnemonic utility. Alternatively, it is possible to specify the variables directly, e.g., ID=V149,V835,V836.

[8] OSIRIS.IV uses numbers to refer to variables. Variable numbers must be unique within a data record. The output data record is constructed so that the variables are located in the data record in ascending variable number order – a variable numbered 1001 will be located nearer the beginning of the data record than a variable numbered 5001.

[9] The more complete job stream, including I/O assignments (with the JCL in an abbreviated form), would look something like this:

```
//EXEC OSIRISIV
```

```
//DICTW1 DD parameters describing the Wave 1 dictionary file

//DATAW1 DD parameters describing the Wave 1 data file

//DICTW5 DD parameters describing the Wave 5 dictionary file

//DATAW5 DD parameters describing the Wave 5 data file

//DICTW9 DD parameters describing the Wave 9 dictionary file

  DATAW9 DD parameters describing the Wave 9 data file

  DICTOUT DD parameters describing the merged output dictionary

  DATAOUT DD parameters describing the merged output data file

//SCARDS DD *

&MACRO IDVARS=V149,V835,V836

&MACRO LIST=V164-V166,V648-V655,V740-V747

&UPDATE

    COMBINING THREE WAVES OF DATA

    PRINT=OUTD PAD=ZERO

    INFI=W1 V=@(LIST) ID=@(IDVARS) OPT RENU=1001

    INFI=W5 V=@(LIST) ID=@(IDVARS) OPT RENU=5001

    INFI=W9 V=@(LIST) ID=@(IDVARS) OPT RENU=9001

&END

/*
```