Using SAS to Analyze CYP-C Data: An introduction

CYP-C Research Champion Webinar
March 12, 2017
Jason D. Pole, PhD

POGO
PEDIATRIC ONCOLOGY GROUP OF ONTARIO

Overview

- SAS overview
- Variable Types
- Data Structure
- Reading in Data
- Creating and Redefining Variables
- Sub-Setting Data
- Working with Dates
- Working with Formats

SAS Overview

- Developed in the 1970s
- Designed to access, manage, analyze and report on data
  - we will discuss mainly analysis
- $3.2 billion company
- You can never own SAS, you can only borrow it

SAS Overview II

- For our purposes only two major things you can do in SAS
  - DATA step
    - Manipulate the data in some way
    - Calculate things, drop variables etc.
  - Procedure step
    - Analyze the data
    - Produce frequency tables
    - Estimate a regression model

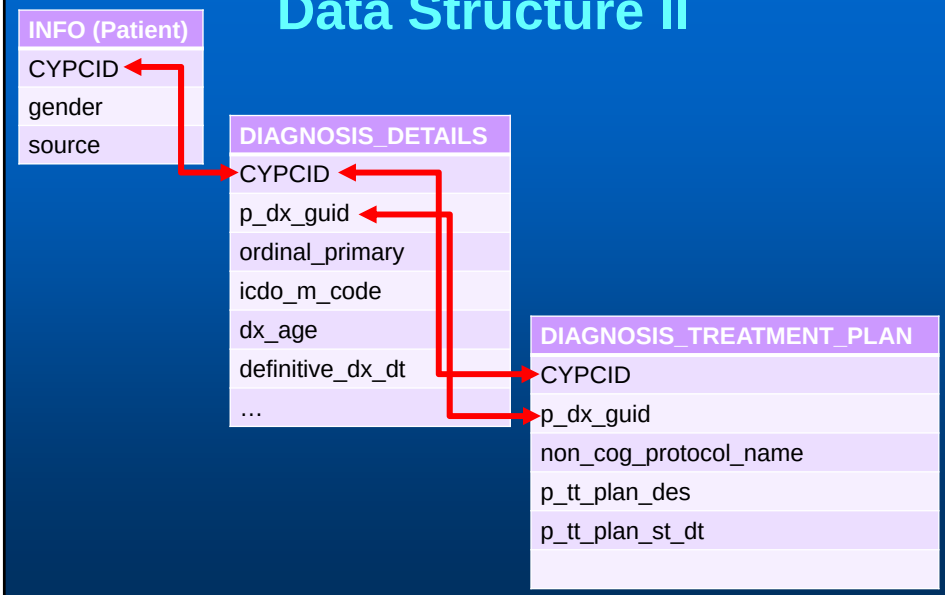
Variable Types in SAS

- There main data types in SAS
  - Character
    - Data stored as text
    - When referencing values in code need quotes around string ‘acute leukemia’
  - Numeric
    - Data stored as number
    - Need to know what value corresponds to what you intend (0 = no, 1 = yes)
  - Date
    - Data stored as number of days from fixed point in time (01JAN1960)
    - Can be positive (after 01JAN1960)
    - Can be negative (before 01JAN1960)

Data Structure

- CYP-C is a relational database
  - Multiple ‘tables’ that contain data
  - Each table may have more than one record per unit observation (person, event)
  - Linkage key that allows all tables to be combined

Data Structure II



Data Structure III

- **Think clinically about the data**
  - **Can you have multiple records of this event?**
- **Know your linking variables**
  - **Links may be different for different parts of the data**

Data Structure of a CYP-C Data Request

- Arrives both as a set of tables and a merged flat file
- Merged flat file
  - Assumes a denominator or population
    - Normally person or but could be diagnosis
    - Repeats the variables to accommodate denominator

Data Structure of a CYP-C Data Request II

- Assume denominator is person
  - Merged dataset has one record per person
  - If, in your population, you have multiple diagnosis per person
    - There will be multiple variables in the merge dataset (icdo_m_code1, icdo_m_code2, icdo_m_code3 etc.)

Reading in the Data

```
OPTIONS LS = 120 PS = 67 NODATE PAGENO = 1 NOFMTERR;
TITLE 'CONFIDENTIAL - CYP-C CLINICAL TRIAL DATA - CONFIDENTIAL';
FOOTNOTE "FILENAME: CYPC TRIAL V7.SAS - DATE: &SYSDATE";

/* THIS PROGRAM PERARES THE ANALYSSI FILE FOR THE TRIAL DATA */

LIBNAME I 'H:\CYP-C Projects\Trial\Data\';
LIBNAME LIBRARY 'H:\CYP-C Projects\Trial\Data\';

/* GOING TO BUILD THE ANALYSIS DATASET FROM ALL THE PIECES */
DATA D1; SET I.DIAGNOSIS_DETAILS;
IF DEFINITIVE_DX_DT NE ' ' THEN DO;
DX_DATE = INPUT(STRIP(DEFINITIVE_DX_DT),YYMMDD10.);
END;
DROP DEFINITIVE_DX_DT;
FORMAT DX_DATE DATE9.;
LABEL DX_DATE = 'DIAGNOSIS DATE';
RUN;
```

Reading in the Data II

Data Set Name	INFO (.sas7bdat or .txt)	Observations	11888
Created	Thu, Nov 24, 2016 09:44:52 AM	Observation Length	12

Variables in Creation Order			
Start	Variable	Type	Len
1	CYPCID	Char	7
8	Gender	Char	1
9	source	Char	4

```
DATA INFO; INFILE 'H:\CYP-C Projects\Trial\Data\INFO .TXT';
INPUT CYPCID $7. GENDER $1. SOURCE $4.;
RUN;
```

Exporting Data

```
PROC IMPORT OUT = C1 DATAFILE = 'H:\NAME OF DATASET.XLSX‘
DBMS = XLSX REPLACE;
SHEET = "COUNTS";
GETNAMES = YES;
RUN;


PROC EXPORT DATA = C1 OUTFILE = 'H:\NAME OF DATASET.XLSX‘
DBMS = XLSX REPLACE;
RUN;
```

Creating and Redefining Variables

```
DATA D1; SET I.DIAGNOSIS_DETAILS;

/* CREATES A FLAG FOR EARLY DIAGNOSIS */
IF '01JAN2001'D <= DX_DATE <= '31DEC2003'D THEN EARLY = 1;
ELSE EARLY = 0;

/* CREATES A FLAG FOR EARLY DIAGNOSIS */
IF '01JAN2004'D <= DX_DATE <= '31DEC2006'D THEN EARLY = 0;
ELSE EARLY = 1;

/* CREATES A FLAG FOR EARLY DIAGNOSIS */
IF '01JAN2001'D <= DX_DATE <= '31DEC2003'D THEN EARLY = 1;
IF '01JAN2004'D <= DX_DATE <= '31DEC2006'D THEN EARLY = 0;

RUN;
```

Creating and Redefining Variables

```
DATA D1; SET I.DIAGNOSIS_DETAILS;

/* CREATES DIAGNOSTIC CATEGORIES */
IF ICD_M_CODE IN (9820, 9823, 9826, 9827, 9831:9837, 9940,
9948) AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1; /* ALL */

IF ICD_M_CODE IN (9840, 9861, 9866, 9867, 9870:9874, 9891,
9895:9897, 9910, 9920, 9931) AND ICD_T_CODE IN (000:809) THEN
DX_GRP = 2; /* AML */

RUN;
```

Equivalent Code

```
DATA D1; SET I.DIAGNOSIS_DETAILS;

/* CREATES DIAGNOSTIC CATEGORIES */
IF ICD_M_CODE  = 9820 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9823 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9826 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9827 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9831 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9832 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9833 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9834 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9835 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9836 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9837 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9940 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;
IF ICD_M_CODE  = 9948 AND ICD_T_CODE IN (000:809) THEN DX_GRP = 1;

RUN;
```

Sub-Setting Data

```
DATA LATE; SET D1;
IF EARLY = 0;
RUN;
```
Data with only late cases

```
DATA EARLY; SET D1;
IF EARLY = 1;
RUN;
```
Data with only early cases

```
DATA LATE; SET D1;
IF EARLY = 0;
IF DX_AGE IN (0,1,2,3,4,5);
IF GENDER = 'M';
RUN;
```
Data with only late cases that are aged 0-5 years at time of diagnosis and are male

```
DATA EARLY; SET D1;
IF EARLY = 1;
IF 0 <= DX_AGE < 6 AND GENDER = 'M';
RUN;
```
Data with only early cases that are aged 0-5 years at time of diagnosis and are male

Working with Dates

- **Dates in SAS are stored as number of days from fixed point in time (01JAN1960)**
- **Allows you to simply add and subtract dates**
  - **Results in the number of days between two dates**
- **Allows you to use SAS functions**
  - **YRDIF**
  - **INTCK**

Adding and Subtracting Dates

```
/* CALCULATES THE AGE AT THE TIME OF DIAGNOSIS IN DAYS*/
AGE = DX_DATE - DOB;


/* CALCULATES AGE AT END OF FOLLOW-UP (1 JUNE, 2016) IN DAYS */
AGE_END = '01JUN2016'D - DOB;
```

SAS Function: YRDIFF

```
/* CALCULATES THE AGE AT THE TIME OF DIAGNOSIS IN DAYS*/
AGE = DX_DATE - DOB;

/* CALCULATES THE AGE AT THE TIME OF DIAGNOSIS IN YEARS*/
AGE = (DX_DATE - DOB) / 365.25;

/* CALCULATES AGE IN YEARS TRULY BASED ON ACUTAL CALENDAR TIME */
AGE = YRDIF(DOB,DX_DATE,'ACTUAL');

/* PERSON-TIME - YEARS */
PT = YRDIF(DX_DATE,MIN(DOD,'31DEC2014'D),'ACTUAL');
```

SAS Function: INTCK

```
/* PERSON-TIME - YEARS - USING THE INTCK FUNCTION */
PT = INTCK('YEAR',DX_DATE,MIN(DOD,'31DEC2014'D));

/* PERSON-TIME - MONTHS - USING THE INTCK FUNCTION */
PT = INTCK('MONTH',DX_DATE,MIN(DOD,'31DEC2014'D));

/* PERSON-TIME - WEEKS - USING THE INTCK FUNCTION */
PT = INTCK('WEEK',DX_DATE,MIN(DOD,'31DEC2014'D));

/* Count the difference between two dates: 01JAN2009 and 01JAN2010
   The result values will be:
    'year' = 1
    'semiyear' = 2
    'quarter' = 4
    'month' = 12
    'week' = 52
    'day' = 365
*/
```

SAS Formats

- Helps improve readability of output
- Takes less space to store
  - dataset size is reduced
- Storing values as numbers eases analysis
- User-Defined and Built-In

SAS Formats – User Defined

```
PROC FORMAT LIBRARY=LIBRARY;
VALUE YESNO
. = '. MISSING'
0 = '0. NO'
1 = '1. YES';
VALUE PRIM
. = 'UNKNOWN'
1 = '1. FIRST PRIMARY'
2 = '2. SECOND PRIMARY'
3 = '3. THIRD PRIMARY'
4 = '4. FOURTH PRIMARY';
VALUE AGREE
.A = '.A. MISSING'
.B = '.B. NOT APPLICABLE'
1 = '1. AGREE'
2 = '2. DISAGEE';
RUN;
```

SAS Formats II

```
DATA D1; SET I.DIAGNOSIS_DETAILS;
IF DEFINITIVE_DX_DT NE ' ' THEN DO;
DX_DATE = INPUT(STRIP(DEFINITIVE_DX_DT),YYMMDD10.);
END;
DROP DEFINITIVE_DX_DT;
FORMAT DX_DATE DATE9.;
LABEL DX_DATE = 'DIAGNOSIS DATE';
FORMAT ORDINAL_PRIMARY PRIM.;
RUN;
```

SAS Formats III

ordinal_primary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
UNKNOWN	1	0.01	1	0.01
1. FIRST PRIMARY	11984	98.83	11985	98.84
2. SECOND PRIMARY	132	1.09	12117	99.93
3. THIRD PRIMARY	7	0.06	12124	99.98
4. FOURTH PRIMARY	2	0.02	12126	100.00

ordinal_primary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	1	0.01	1	0.01
1	11984	98.83	11985	98.84
2	132	1.09	12117	99.93
3	7	0.06	12124	99.98
4	2	0.02	12126	100.00

SAS Formats IV

DX_DATE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01JAN2001	1	0.01	1	0.01
02JAN2001	1	0.01	2	0.02
03JAN2001	2	0.02	4	0.03
04JAN2001	6	0.05	10	0.08
05JAN2001	3	0.02	13	0.11
06JAN2001	2	0.02	15	0.12

DX_DATE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
14976	1	0.01	1	0.01
14977	1	0.01	2	0.02
14978	2	0.02	4	0.03
14979	6	0.05	10	0.08
14980	3	0.02	13	0.11
14981	2	0.02	15	0.12

Topics Covered

- SAS overview
- Variable Types
- Data Structure
- Reading in Data
- Creating and Redefining Variables
- Sub-Setting Data
- Working with Dates
- Working with Formats