

Many to One Using a SAS® DATA Step and PROC MEANS

Jennifer L. Waller, Ph.D.
Department of Biostatistics
Medical College of Georgia, Augusta GA



Introduction

- ◆ Different types of data sets often contain multiple observations per sampling unit
 - ◆ Insurance claims data
 - ◆ Financial transactions
 - ◆ 911 data from a single address

Introduction

- ◆ Financial Example
 - ◆ Credit card statements sort transactions by type of business / service – travel, entertainment, grocery, clothing, warehouse store, etc.
 - ◆ What if we were interested in determining the total amount spent in each category?

Introduction

- ◆ Administrative insurance claims data often contain multiple observations per individuals.
- ◆ Each observation may correspond to different types of claims that have different information.
 - ◆ Facility
 - ◆ Provider
 - ◆ Prescription

Introduction

- ◆ Within a facility or provider claim there is the potential for 7 ICD-9 diagnosis variables and 3 CPT medical procedure variables.
- ◆ For each individual facility or provider claim
 - ◆ Information for a particular diagnosis, say diabetes, can occur in any of these 7 diagnoses variables.
 - ◆ Information for a procedure can occur in any of the 3 procedure variables.

Introduction

Obs	studyid	dx1	dx2	dx3	dx4	dx5	proc1	proc2	proc3
1432	M0006	585	25040	2859	2753	2520			
1433	M0006	585	25000	2859	2520				
1434	M0006	78900	7295						
1435	M0006	585	25042	2859	2520				
1436	M0006	585	25042	2859	V048	5888			
1437	M0006	4439							
1438	M0006	5184	78609						
1439	M0006	585	25042	2859	2520				
2141	M0010	3310	2941	4019	53081	2859			
2179	M0010	29020					99232		
2184	M0010	29020					99238		
2185	M0010	4439					93926		
2186	M0010	56030					74000		
2203	M0010	3310					99311		

Introduction

- ◆ Not interested in when a diagnosis or procedure occurs, just that an individual has a diagnosis or a procedure.
- ◆ Interested in the number of times a diagnosis or procedure occurs.

Introduction

- ◆ Goal
 - ◆ Collapse data across several variables within and observation
 - ◆ Then collapse several observations to obtain a single observation per individual.

Approach to the Problem

- ◆ Create indicator variables for a particular diagnosis or procedure occurrence within a claim
 - ◆ Use an ARRAY and DO loop in a DATA step
- ◆ Create sums across claims for each individual
 - ◆ Use PROC MEANS on the indicator variables to create the sum
 - ◆ Output the sum to a SAS data set
- ◆ Create indicators from the PROC MEANS output data set to determine whether an individual has a diagnosis or procedure
 - ◆ Use an ARRAY and DO loop in a DATA step

Example Data

- ◆ Charleson Comorbidity Index (CCI)
 - ◆ Created by Charleson et al. (1987).
 - ◆ The CCI is a weighted sum based on the number and severity of co-existing conditions.
 - ◆ Deyo et al. (1992) adapted the CCI for use with the International Classification of Diseases (ICD-9-CM) in administrative data bases.
 - ◆ The CCI uses 17 different diagnoses (corresponding to 158 ICD-9-CM codes) and has a range from 0 to 27.

Steps to Create the CCI

1. Read in the raw data.
2. Create indicator variables for claims having a particular diagnosis or procedure.
3. Sort the data by individual and use PROC MEANS to obtain the sum of each diagnosis or procedure for each individual and output to a SAS data set.
4. Create the CCI from the PROC MEANS outputted SAS data set.

1. Creation of SAS Data Set from Raw Data

The data come as a text file, 279 columns wide.

Some data sets have up to a million claims.

The dataset here has 5 diagnosis variables and 3 procedure variables.

Obs	studyid	dx1	dx2	dx3	dx4	dx5	proc1	proc2	proc3
1432	M0006	585	25040	2859	2753	2520			
1433	M0006	585	25000	2859	2520				
1434	M0006	78900	7295						
1435	M0006	585	25042	2859	2520				
1436	M0006	585	25042	2859	V048	5888			
1437	M0006	4439							
1438	M0006	5184	78609						
1439	M0006	585	25042	2859	2520				
2141	M0010	3310	2941	4019	53081	2859			
2179	M0010	29020					99232		
2184	M0010	29020					99238		
2185	M0010	4439					93926		
2186	M0010	56030					74000		
2203	M0010	3310					99311		

2. Creation of Indicator Variables

- ◆ This step is used to
 - ◆ Create from many variables, one indicator variable per diagnosis per claim.
 - ◆ Value can occur in any of the diagnosis variables.
 - ◆ Only interested in occurrence, not when the diagnosis occurred.
- ◆ Two different ways to do this
 - ◆ IF-THEN-ELSE-IF statements for each diagnosis variable
 - ◆ A lot of code
 - ◆ A lot of copying of code and potential for more typos
 - ◆ Less efficient
 - ◆ Using an ARRAY and DO loop with nested IF-THEN-ELSE-IF statements
 - ◆ Less code
 - ◆ No copying of code and less potential for typos
 - ◆ More efficient

2. Creation of Indicator Variables

Example 1 – Lengthy IF-THEN-ELSE-IF statements

```
data claims;
  set in.claims;
  *****
  ** Identification of claims with a MI diagnosis. **
  *****;
  if substr(dx1,1,3) in ('410','412') or substr(dx2,1,3) in ('410','412')
    or substr(dx3,1,3) in ('410','412') or substr(dx4,1,3) in ('410','412')
    or substr(dx5,1,3) in ('410','412') then mi=1;
  *****
  ** Identification of claims with a PVD diagnosis. **
  *****;
  if substr(dx1,1,3)='441' or substr(dx1,1,4) in ('4439','7854','V434') or
    substr(dx2,1,3)='441' or substr(dx2,1,4) in ('4439','7854','V434') or
    substr(dx2,1,3)='441' or substr(dx3,1,4) in ('4439','7854','V434') or
    substr(dx4,1,3)='441' or substr(dx4,1,4) in ('4439','7854','V434') or
    substr(dx5,1,3)='441' or substr(dx5,1,4) in ('4439','7854','V434')
    or proc1='3848' or proc2='3848' or proc3='3848' then pvd=1;
run;
```

2. Creation of Indicator Variables

Example 2 – Using an ARRAY and DO loop

```
data claims;
  set in.claims;
  array dx $ dx1-dx5;
  do over dx;
    *****
    ** Identification of claims with a MI diagnosis. **
    *****;
    if substr(dx,1,3) in ('410','412') then mi=1;
    *****
    ** Identification of claims with a PVD diagnosis. **
    *****;
    if substr(dx,1,3)='441'
       or substr(dx,1,4) in ('4439','7854','V434')
       or proc1='3848' or proc2='3848' or proc3='3848'
       then pvd=1;
  end;
run;
```

Data Set - Indicators Created

Obs	studyid	dx1	dx2	dx3	dx4	dx5	proc1	proc2	proc3	mi	pvd
1432	M0006	585	25040	2859	2753	2520				.	.
1433	M0006	585	25000	2859	2520					.	.
1434	M0006	78900	7295							.	.
1435	M0006	585	25042	2859	2520					.	.
1436	M0006	585	25042	2859	V048	5888				.	.
1437	M0006	4439								.	1
1438	M0006	5184	78609							.	.
1439	M0006	585	25042	2859	2520					.	.
2141	M0010	3310	2941	4019	53081	2859				.	.
2179	M0010	29020					99232			.	.
2184	M0010	29020					99238			.	.
2185	M0010	4439					93926			.	1
2186	M0010	56030					74000			.	.
2203	M0010	3310					99311			.	.

3. Create Sums Using PROC MEANS for Each Individual

- ◆ This step is used to create one observation per subject.
- ◆ To collapse the data across claims (observations) per individual
 - ◆ Sort the data by individual
 - ◆ Use PROC MEANS to produce the summary statistic of interest.
 - ◆ Output the summary statistic to a SAS data set.

```
proc sort data=clms; by studyid;  
run;
```

```
proc means data=clms sum noprint;  
  var mi chf pvd cvd dem cpd rhemz pud mliverd diabnc diabc hppegia  
  renald cancer msliver mcancer aids;  
  by studyid;  
  output out=sumcci sum=smi schf spvd scvd sdem scpd srhemz  
  spud smliverd sdiabnc sdiabc shppega  
  srenald scancer smsliver smcancer saids;  
run;
```

Where We Are

- ◆ At this point
 - ◆ The SAS® data set SUMCCI has 18 variables that includes the unique identifier for each individual and the number of times each diagnosis or procedure occurred for each individual.
- ◆ We have now gone from diagnoses that could occur in **many** fields (variables) across **many** claims (observations) to **one** observation per subject that contains the number of each type of diagnosis.

SAS Data Set of Summary Statistics from PROC MEANS

Obs	studyid	smi	schf	spvd	scvd	sdem	scpd	srhemz	spud	smliverd	sdiabnc	sdiabc	shpplega
1	M0001	.	1	.	21	1	1
2	M0002	.	7	.	.	1
3	M0003	3	3	.	6
4	M0004	1
5	M0006	.	2	1	14	23	33	8
6	M0007	16	.	.

Obs	srenald	scancer	smsliver	smcancer	said
1
2	13	2	.	.	.
3
4
5	203
6	.	4	.	.	.

4. Using the Summary Indicator Variables

- ◆ The summary data per individual can now be used in further data manipulation or analysis.
- ◆ Example - Creation of the CCI
 - ◆ Use the summed diagnosis variables to create indicator variables for whether the individual has the diagnosis or not
 - ◆ Two ARRAY statements and a single DO loop.
 - ◆ Create a weighted sum of the diagnoses using ARRAYs and DO loops to calculate the overall CCI score.

4. Using the Summary Variables from PROC MEANS

- ◆ Create the indicator variables for each diagnosis from the summary variables from PROC MEANS.

```
data cci;
  set sumcci; by studyid;
  *****
  ** Array for the summary variables from PROC MEANS. **
  *****;
  array asum smi schf spvd scvd sdem scpd srhemz spud smliverd sdiabnc
           sdiabc shpplega srenald scancer smsliver smcancer saids;
  *****
  ** Array for new indicator variables for each diagnosis. **
  *****;
  array ai imi ichf ipvd icvd idem icpd irhemz ipud impliverd idiabnc
           idiabc ihpplega irenald icancer imsliver imcancer iaids;
  do over asum;
    if asum>=1 then ai=1;
    else ai=0;
  end;
```

Create the Weighted CCI Score

```
...  
cci1=0; cci2=0; cci3=0; cci6=0;  
array one imi ichf ipvd icvd idem icpd irhemz ipud imliverd idiabnc;  
array two idiabc ihpplega irenald icancer;  
array six imcancer iaids;  
do over one;  
  if one=1 then cci1=cci1+1;  
end;  
do over two;  
  if two=1 then cci2=cci2+2;  
end;  
do over six;  
  if six=1 then cci6=cci6+6;  
end;  
if imsliver=1 then cci3=cci3+3;  
  
cci=cci1+cci2+cci3+cci6;  
run;
```

Summary Data from PROC MEANS

Obs	studyid	smi	schf	spvd	scvd	sdem	scpd	srhemz	spud	smliverd	sdiabnc	sdiabc	shpplega
1	M0001	.	1	.	21	1	1
2	M0002	.	7	.	.	1
3	M0003	3	3	.	6
4	M0004	1
5	M0006	.	2	1	14	23	33	8
6	M0007	16	.	.

Obs	srenald	scancer	smsliver	smcancer	said
1
2	13	2	.	.	.
3
4
5	203
6	.	4	.	.	.

Final Data Set for Analysis

Obs	studyid	imi	ichf	ipvd	icvd	idem	icpd	irhemz	ipud	impliverd	idiabnc	idiabc
1	M0001	0	1	0	1	1	1	0	0	0	0	0
2	M0002	0	1	0	0	1	0	0	0	0	0	0
3	M0003	1	1	0	1	0	0	0	0	0	0	0
4	M0004	0	0	0	0	1	0	0	0	0	0	0
5	M0006	0	1	1	1	0	0	0	0	0	1	1
6	M0007	0	0	0	0	0	0	0	0	0	1	0

Obs	ihpplega	irenald	icancer	imcancer	iaids	cci
1	0	0	0	0	0	4
2	0	1	1	0	0	6
3	0	0	0	0	0	3
4	0	0	0	0	0	1
5	1	1	0	0	0	10
6	0	0	1	0	0	3

Other Many-to-One Applications

- ◆ Other applications using insurance claims data
 - ◆ Cost data – allowed, billed, paid, co-pays
 - ◆ Average length of stay
 - ◆ Utilization data
 - ◆ Inpatient visits
 - ◆ Outpatient visits
 - ◆ Prescription use
 - ◆ Short nursing home stays
- ◆ Other types of administrative data bases
 - ◆ Credit Card transactions
 - ◆ Number
 - ◆ Total purchase cost
 - ◆ 911 data
 - ◆ Determine multiple calls from single address
 - ◆ Reasons for calls

Conclusions

- ◆ The many-to-one problem is prevalent in many areas.
- ◆ Relatively simple programming methods
 - ◆ ARRAYs and DO loops and IF-THEN-ELSE-IF statements
 - ◆ Create a single indicator variable from many variables.
 - ◆ Use of SAS/Base ® procedures (PROC MEANS, PROC FREQ)
 - ◆ Create a single observation from multiple observations.
- ◆ Other methods
 - ◆ PROC TRANSPOSE

Contact Information

Jennifer Waller
Medical College of Georgia
Department of Biostatistics, AE -1012
Augusta, GA 30912-4900
jwaller@mcg.edu