

Data Simulation: Create A Dummy Dataset using Clinical Administrative Database

Jun Liang

Health Indicators, CIHI

April 1, 2010



Canadian Institute
for Health Information

Institut canadien
d'information sur la santé

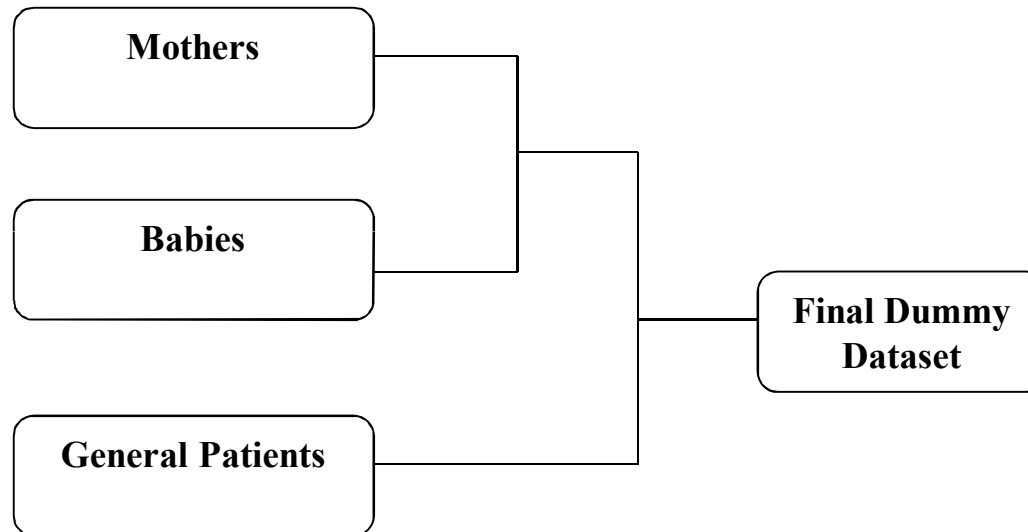
Outline

- Background
- Composition of the Dummy Dataset
- Steps to Create Dataset for “General Patients”
- Steps to Create Dataset for “Mothers and Babies”
- SAS codes to create Dummy Dataset
- Add new data element (data simulation)
- Tests of the Dummy Dataset
- Q & A

Background

- Requested by clients to create a dummy dataset that looks like administrative databases for educational purposes: data quality analysis, data cleaning, and data standardization.
- Required variables Include: record ID, sex, age, admit category, diagnosis code, procedure code, clinical gestation, province code, etc.
- To perform *small for gestational age* (SGA) analysis and to calculate *coronary artery bypass graft surgery* (CABG) rate using this dummy dataset

Composition of the Dummy Dataset



Steps to Create a “General Patients” Dataset

- Group *Discharge Abstract Database* (DAD) ‘general patients’ records by age group, gender, admission category and first 3 diagnosis codes and types, and principal procedure code
- Calculate the frequency of each group (weight)
- Calculate the required number of records for each group based on requested sample size and weight
- Build records until the required sample size is reached
- Simulate province code for each record in the dummy dataset
- Add a unique record ID to each record.

Steps to Create “Mothers and Babies” Datasets

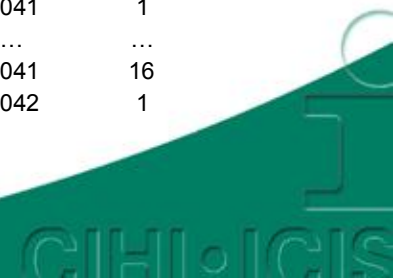
- Group DAD ‘mom’ records following similar steps as for general patients, except
 - one additional variable: gestational_age was used in grouping
 - A chart number was assigned to each mom.
- Group DAD ‘baby’ records following similar steps as for general patients, except
 - No province code
 - A chart number was assigned to each baby

Grouping records (for Demonstration Purpose)

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode
F	U	1	M	1	1	A080	E860	E870	
...					
F	U	1	M	1	1	A080	E860	E870	
F	U	1	M	1	1	A080	E860	E870	
F	U	1	M	1	1	A080	E860	E872	
...					
F	U	1	M	1	1	A080	E860	E872	
F	U	1	M	W	1	A080	E860	E872	

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode	Group
F	U	1	M	1	1	A080	E860	E870		3040
...		
F	U	1	M	1	1	A080	E860	E870		3040
F	U	1	M	1	1	A080	E860	E870		3040
F	U	1	M	1	1	A080	E860	E872		3041
...		
F	U	1	M	1	1	A080	E860	E872		3041
F	U	1	M	W	1	A080	E860	E872		3042

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode	Group	group_count
F	U	1	M	1	1	A080	E860	E870		3040	1
...		
F	U	1	M	1	1	A080	E860	E870		3040	29
F	U	1	M	1	1	A080	E860	E870		3040	30
F	U	1	M	1	1	A080	E860	E872		3041	1
...		
F	U	1	M	1	1	A080	E860	E872		3041	16
F	U	1	M	W	1	A080	E860	E872		3042	1



SAS code to Create Dummy Dataset: Grouping records

```
Proc sort data=&original_data(keep=...) out=interim01; by ...; run;
```

```
proc sort data=interim01 out=temp001 nodupkey; by ...; run;
```

```
data temp001; set temp001; group=_N_; run;
```

```
data interim01; merge interim01 temp001;  
  by agegp sex diagcode1 diagcode2 diagcode3 diagtyp1 diagtyp2 diagtyp3 admcat proccode;  
run;
```

```
data interim01; set interim01; by group;  
  if first.group then group_count=0;  
  retain accumulated_group accumulated_total 0;  
  group_count=group_count+1;  
  accumulated_total=accumulated_total+1;  
run;
```

Grouping records, Cont'd (for Demonstration Purpose)

- Note: Suppose that there are 4,000,000 records from original dataset and the required dummy dataset sample size is 2,000,000.

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode	Group	group_count
F	U	1	M	1	1	A080	E860	E870		3040	30
F	U	1	M	1	1	A080	E860	E872		3041	16
F	U	1	M	W	1	A080	E860	E872		3042	1

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode	Group	group_count	weight	required_sample_size
F	U	1	M	1	1	A080	E860	E870		3040	30	30/4000000	15
F	U	1	M	1	1	A080	E860	E872		3041	16	16/4000000	8
F	U	1	M	W	1	A080	E860	E872		3042	1	1/4000000	0

Sex	Admcat	Agegp	Diagtyp1	Diagtyp2	Diagtyp3	Diagcode1	Diagcode2	Diagcode3	Proccode	Group	group_count	weight	required_sample_size
F	U	1	M	1	1	A080	E860	E870		3040	30	30/4000000	15
F	U	1	M	1	1	A080	E860	E872		3041	16	16/4000000	8

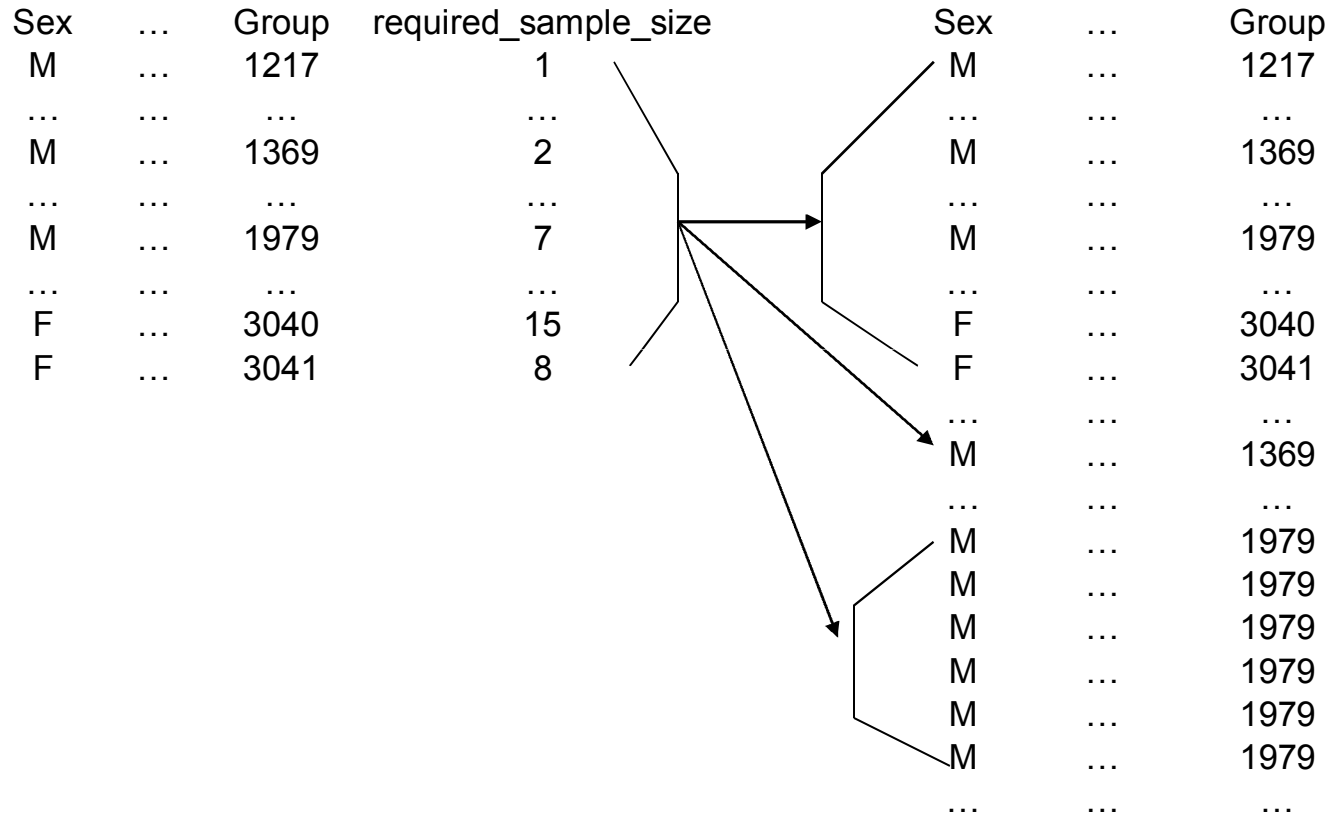
SAS code to Create Dummy Dataset: Grouping records, Cont'd

```
data interim01; set interim01; by group; if last.group; run;
```

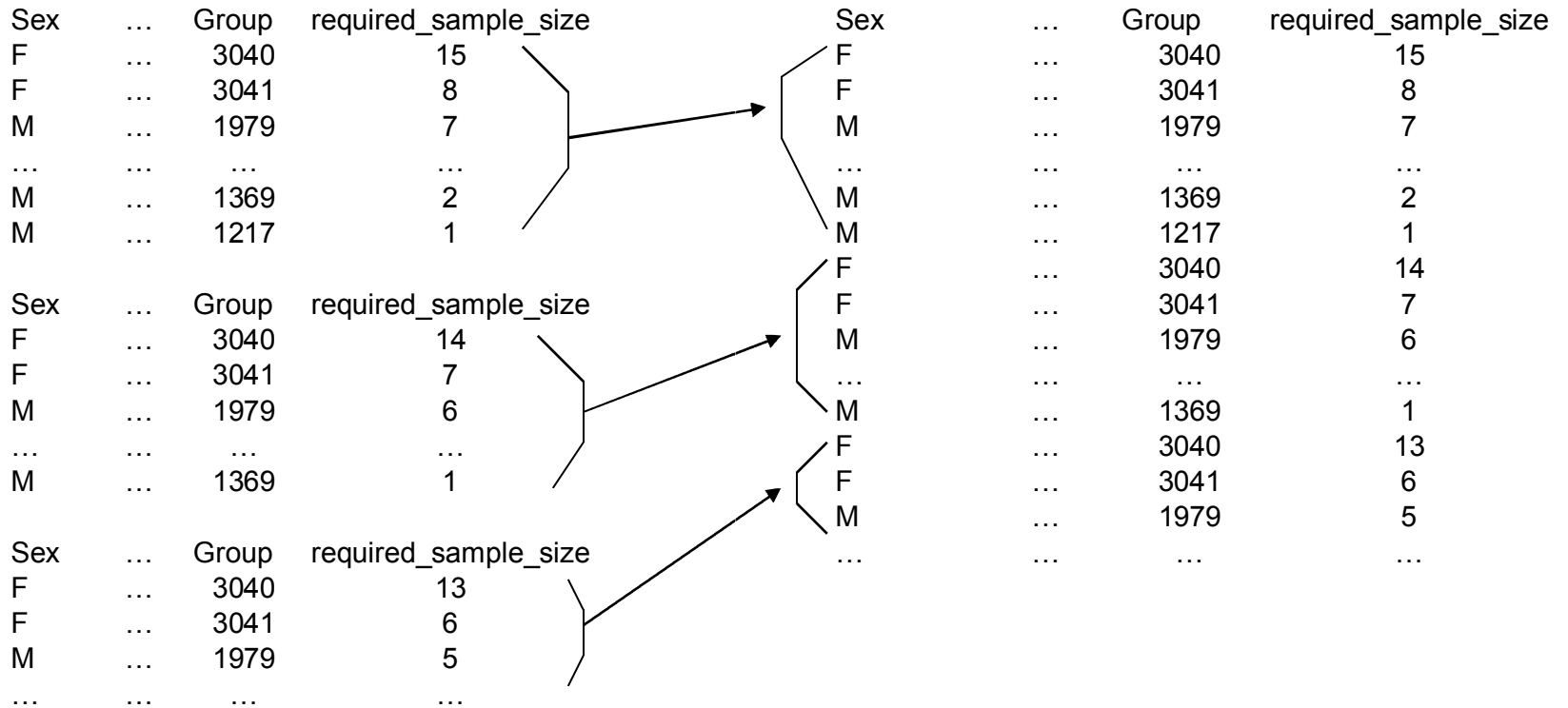
```
data _null_; set interim01 end=last;  
  if last then call symput('total_count',accumulated_total);  
run;
```

```
data interim02; set interim01;  
  weight=group_count/&total_count;  
  required_sample_size=round(weight*&sample_size,1);  
  if required_sample_size=0 then delete;  
run;
```

Build Records Until the Required Sample Size reached (For Demonstration Purpose) method 1



Build Records Until the Required Sample Size reached (For Demonstration Purpose) method 2



SAS code to Create Dummy Dataset: Build Records Until the Required Sample Size reached

```
data interim02; set interim02; indx=_N_; run;

data _null_; set interim02 end=last;
if last then call symput('counter',indx);
run;

%do i=1 %to &counter;
  proc sql noprint;
    select required_sample_size, group
      into : required_sample_size, :group
    from interim02 where indx=&i;
  quit;
  %let t = %eval(&required_sample_size - 1);
  %if &t > 0 %then %do;
    data temp002; set interim02;
      %do k=1 %to &t; if group=&group; output; %end;
    run;
    proc append base=interim02 data=temp002; run;
  %end;
%end;
```

```
proc sort data=interim02;
  by required_sample_size;
run;

data _null_; set interim02 end=last;
if last then call symput('time', required_sample_size);
run;

%do i=1 %to &time;
  data tempdata&i; set interim02;
    required_sample_size=required_sample_size - &i;
    if required_sample_size < 1 then delete;
  run;

  data &output.;
  set %if &i=1 %then interim02;
    %else &output.; tempdata&i;
  run;
%end;
```

SAS code to Simulate province code

```
cx=round(99*RANUNI(1),1);  
cy=round(99*RANUNI(1),1);  
if cx=0 and cy<67 then province='NT';  
else if cx=0 and cy>66 then province='YT';  
else if cx=1 and cy>90 then province='NU';  
else if cx=1 and cy<91 then province='PE';  
else if 1<cx<37 then province='ON';  
else if 36<cx<45 then province='SK';  
else if 44<cx<56 then province='AB';  
else if 55<cx<78 then province='BC';  
else if 77<cx<85 then province='MB';  
else if 84<cx<90 then province='NB';  
else if 89<cx<94 then province='NL';  
else province='NS';
```

Frequency Table: Record distribution among provinces

Province	Percentage (DAD)	Percentage (Dummy Dataset)
AB	11.14	11.1
BC	22.76	22.23
MB	7.06	7.08
NB	4.6	5.03
NL	3.97	4.02
NS	5.97	5.56
NT	0.29	0.34
NU	0.08	0.09
ON	35.64	35.36
PE	0.87	0.93
SK	7.47	8.1
YT	0.15	0.16

SAS code to Create Dummy Dataset: Create a Unique Record ID and Age for Each Record

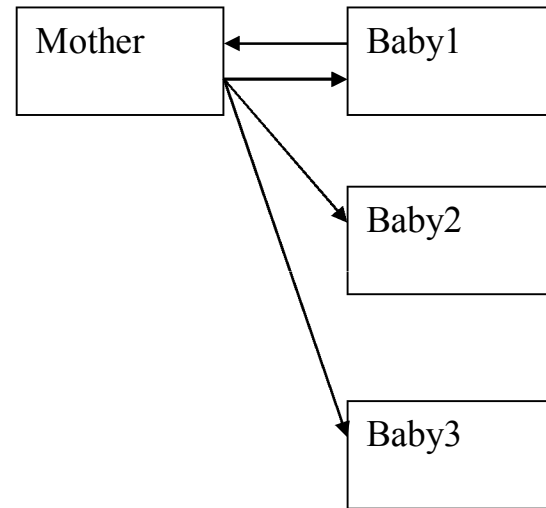
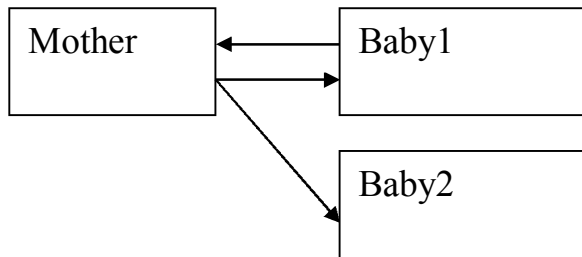
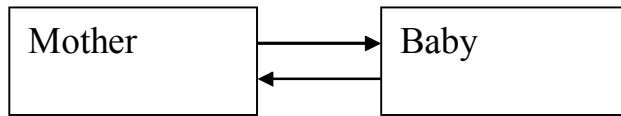
```
data &output; set &input;  randnum=RANUNI(1); run;

proc sort data=&output; by randnum; run;

data &output; set &output;
  length id $10;
  newnum=_N_;
  tri=put(newnum, z6.);
  id=compress(province||tri||sex);

  temp=input(agegp, 2.0);
  if temp=1 then age=0;
  else if temp=2 then age=round(3*RANUNI(1)+1,1);
  else if 2<temp<20 then age=round(4*RANUNI(1)+1,1)+5*temp-11;
  else age=90+round(9*RANUNI(1),1);
run;
```

Mom-Baby Linkage: 3 Scenarios



SAS code to Create Dummy Dataset: Mom-Baby Linkage

```
/*in1~dataset from mom side; in2~dataset from baby side;
type=1~singleton; type=2~twin; type=3~other multiple delivery; */

%macro mom_baby_link(in1=, in2=, type=);

data &in1; set &in1 %if &type=2 %then &in1; %if &type=3 %then &in1 &in1;; run;
%if &type ne 1 %then %do; proc sort data=&in1; by mom_chart_number; run; %end;
data &in1; set &in1; indx=_N_; run;
data mom_for_merge; set &in1; keep indx mom_chart_number province clinical_gestation; run;
proc sort data=&in2; by baby_chart_number; run;
data baby_for_merge; set &in2; indx=_N_; keep baby_chart_number indx; run;
data mom_baby_merge; merge mom_for_merge baby_for_merge; by indx; run;

proc sort data=mom_baby_merge out=mom_for_merge2 nodupkey; by mom_chart_number; run;
proc sort data=mom_baby_merge out=baby_for_merge2; by baby_chart_number; run;
proc sort data=&in1. nodupkey; by mom_chart_number; run;

data &in1.; merge &in1.(in=a) mom_for_merge2(in=b); by mom_chart_number; if a; run;
data &in2.; merge &in2.(in=a) baby_for_merge2(in=b); by baby_chart_number; if a; run;

%mend mom_baby_link;
```

Add a new data element into dummy dataset

- To get a SAS dataset from DAD including the following information:
 - count in every group with certain existing data elements (1)
 - count in every subgroup by the required new data element
 - percent in every subgroup within a group
 - percent range for each subgroup, associated with the value of the new data element
- To assign a random number for each record in the dummy dataset
- To merge the two datasets by use of data elements (1) (many-to-many merge)
- To keep records whose random numbers fall into the percent range, therefore, the corresponding values of the new data element were added to the dummy dataset

Add a new data element – an example

- Current data elements: sex, ..., diagcde1,diagcde2,diagcde3,proccde1;
- New data element: discharge_disposition;

sex	...	diagcde1	diagcde2	diagcde3	proccde1	count
M	...	I500	M8687	J449	1I53GRLF	20

sex	...	diagcde1	diagcde2	diagcde3	proccde1	count	discharge_disposition	count2	percent	lower_bound	upper_bound
M	...	I500	M8687	J449	F	20	1	2	10%	1	10
							4	3	15%	11	25
							5	4	20%	26	45
							6	11	55%	46	100

SAS code to Create Dummy Dataset: Add a new data element into dummy dataset

- `proc sql;`
- `create table new_dummy_dataset as`
- `select a.*, b.discharge_disposition,b.lower_bound,b.upper_bound`
- `from current_dummy_dataset as a , SAS_dataset_from_DAD as b`
- `Where a.sex=b.sex and ... and a.proccde1=b.proccde1`
- `%do i=1 %to 3; and a.diagcde&i=b.diagcde&i %end;`
- `and a.rand>=b.lower_bound and a.rand<=b.upper_bound;`
- `quit;`

Test of the dummy dataset

- ACSC (Ambulatory Care Sensitive Conditions)

Defined as Conditions where appropriate ambulatory care may prevent or reduce the need for hospitalization.

121,834 ACSC cases were identified in the dummy dataset, 125,373 were identified in DAD (97%)

- CABG (Coronary Artery Bypass Graft Surgery)

16,198 procedures identified in the dummy dataset, which account for 70% of all CABGs in DAD (23,122).

- Injury

114,810 Injuries in the dummy dataset, which covered 59% of injuries in DAD (195,029)

Question?

