Parmvir Parmar Quiz2

Questions:

4. The most frequent diagnoses codes are: Z38, Z37, I10, Y83 and U98.

5. Z38: (category) liveborn infant according to place and type of delivery

Z37: (category) delivery outcome

I10: primary (essential) hypertension

Y83: Surgical/ procedural complication, developing after time of surgery/ procedure (without mention of complication at time of procedure).

U98: patient refusal to disclose clinical information

8. There are 2419 unique patients in the spine dataset.

SAS code:

**data** largedb.quiz2;

set Largedb.Quiz2\_visits;

**run**;

**proc** **contents** data=largedb.quiz2 varnum;

**run**;

\*1. creation of spine dataset from visits dataset;

**data** largedb.spinevisit;

set largedb.quiz2;

admit\_date=datepart(encstartdtm);

if **'01JAN2003'd**<=admit\_date<= **'31JAN2004'd**;

keep encwid encpatwid encstartdtm encpatgendercd;

**run**;

\*ensuring no duplicates by encwid in spine dataset;

**proc** **sort** data=largedb.spinevisit out=unique nodupkey;

by encwid;

**run**;

\*sorting spine dataset by encwid in preparation for merge. 2728 observations in spine dataset from log;

**proc** **sort** data=largedb.spinevisit out=spine;

by encwid;

**run**;

\*making rib dataset only keep 1 row for hdghraencwid variable;

**data** diag;

set largedb.nhrdiagnosis;

by hdghraencwid;

if last.hdghraencwid=**1** then output;

**run**;

\*checking rib dataset will only have 1 row for hdghraencwid variable;

**proc** **sort** data=diag out=check nodupkey;

by hdghraencwid;

**run**;

\*sorting rib dataset by encwid (after renaming hraencwid variable) in preparation for merge;

**proc** **sort** data=diag(rename=(hdghraencwid=encwid)) out=diagnosis;

by encwid;

**run**;

**proc** **sort** data=diagnosis out=check1 nodupkey;

by encwid;

**run**;

\*2. merging diagnosis and spine datasets;

\*3 creating diagcat variable with 3 character strings in datastep;

**data** spine\_diagnosis;

merge spine(in=a) diagnosis;

by encwid;

if a=**1**;

diagcat=substr(hdgcd,**1**,**3**);

**run**;

\*4. frequency procedure for diagcat variable with diagnoses displayed by descending frequencies;

**proc** **freq** data=spine\_diagnosis order=freq;

tables diagcat;

**run**;

\*ensuring merged dataset has same number of observations as spine dataset. 2728;

\*proc sort data=spine\_diagnosis out=check nodupkey;

\*by encwid;

\*run;

\*6. creating a flag for z38 diagnosis and a count of total number of diagnoses recorded per encounter;

\*upper limit of diagnosis checked by first setting diagnoses to 50, and was determined 1 encounter had 24 diagnoses, so limits were scaled back;

**proc** **sort** data=largedb.nhrdiagnosis out=flag;

by hdghraencwid;

**run**;

**data** diagflag;

set flag;

by hdghraencwid;

array ddiag{**25**} $diagnosis1-diagnosis25;

retain diagnosis1-diagnosis25 counter;

if first.hdghraencwid=**1** then do

x=**1** to **25**;

ddiag{x}=' ';

counter=**1**;

liveborn=**0**; count=**0**;

end;

if counter<=**25** then do;

ddiag{counter}=hdgcd;

counter=counter+**1**;

end;

if hdgcd in :('Z38') then do;

liveborn=**1**; count=count+**1**;

end;

if last.hdghraencwid=**1** then do;

retain liveborn count;

keep hdghraencwid diagnosis1-diagnosis25 liveborn count counter ;

output;

end;

**run**;

\*checking new dataset and newly created flag variables;

**proc** **sort** data=diagflag out=check2 nodupkey;

by hdghraencwid;

**run**;

**proc** **freq** data= diagflag order=freq;

tables liveborn count diagnosis1-diagnosis25 counter;

**run**;

\*merging flag dataset with original spine dataset;

\*renaming hdghraencwid to encwid and sorting by encwid in diagflag dataset;

**proc** **sort** data= diagflag(rename=(hdghraencwid=encwid)) out=ribflag;

by encwid;

**run**;

\*sorting spine dataset by encwid in preparation for merge. 2728 observations in spine dataset from log;

**proc** **sort** data=largedb.spinevisit out=spine;

by encwid;

**run**;

\*merging ribdataset with flag variables and spine dataset;

**data** spinewithflag;

merge spine (in=a) ribflag;

by encwid;

if liveborn=**.** then liveborn=**0**;

if count=**.** then count=**0**;

if a=**1** then output;

**run**;

\*checking new merged dataset to ensure same number of observations as spine dataset;

\*verified 2728 observations in log unique by encwid;

**proc** **sort** data=spinewithflag nodupkey;

by encwid;

**run**;

\*7. frequency table of diagnosis flag (0 or 1), flag diagnosis count (count), and count of # of diagnoses per visit (counter) ;

**proc** **freq** data=spinewithflag;

tables liveborn count counter;

**run**;

\*8.# of unique patients in spine dataset;

\*2419 unique patients were identified by encpatwid in the spine dataset;

**proc** **sql**;

select count(distinct encpatwid) as encpatwid

from spine;

**quit**;

\*\*\*\*\*\*\*9. creating new dataset with patient as unit of analysis still using visit dataset as spine dataset;

**data** patient2;

merge spine (in=a) ribflag;

by encwid;

if liveborn=**.** then liveborn=**0**;

if count=**.** then count=**0**;

if a=**1** then output;

**run**;

**proc** **sort** data=patient2;

by encpatwid;

**run**;

**data** vet;

set patient2;

by encpatwid;

array dencounter{**50**} $encounter1-encounter50;

retain encounter1-encounter50 visitcount;

if first.encpatwid=**1** then do x=**1** to **50**;

dencounter{x}=' ';

visitcount=**1**;

end;

if visitcount<=**50** then do;

dencounter{visitcount}=encwid;

visitcount=visitcount+**1**;

visits=(visitcount-**1**);

end;

if last.encpatwid=**1** then do;

keep encounter1-encounter50 visitcount encpatwid encpatgendercd visits;

output;

end;

**run**;

\*checking no duplicates by encpatwid in new analysis dataset;

**proc** **sort** data=vet out=vine nodupkey;

by encpatwid;

**run**;

\*ensuring same number of unique patients as spine dataset;

**proc** **sql**;

select count(distinct encpatwid) as encpatwid

from vet;

**quit**;

\*10. frequency tables of new visit counter variable and visits by gender;

**proc** **freq** data=vet;

tables visitcount visits visits\*encpatgendercd;

**run**;

\*number of total visits by gender;

\*total of 2728 visits for females and 2726 visits for males;

**data** sum;

set vet;

cum\_sum+visits;

**run**;

**data** females;

set sum;

where encpatgendercd='F';

**run**;

**proc** **freq** data=females;

tables cum\_sum;

**run**;

**data** males;

set sum;

where encpatgendercd='M';

**run**;

**proc** **freq** data=males;

tables cum\_sum;

**run**;

SAS output:

4.

Graphical user interface

Description automatically generated

7.

Graphical user interface, application

Description automatically generated

7. Diagnosis counter

Graphical user interface, application

Description automatically generated

8. Unique patients in spine dataset

Graphical user interface, text, application

Description automatically generated

10. Visitcount is new variable created by flag.

Visits is mathematical operator of (visitcount-1)

Graphical user interface

Description automatically generated

10. Frequency of patients by gender and by number of visits

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

10b? Total number of visits by gender

Graphical user interface

Description automatically generated with medium confidence

Females

Graphical user interface, application

Description automatically generated

Males

Graphical user interface, application

Description automatically generated